



販売者ガイド

AWS Marketplace



AWS Marketplace: 販売者ガイド

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

とは AWS Marketplace	1
販売 AWS Marketplace 者としての の使用	1
の契約構造 AWS Marketplace	3
での製品の料金 AWS Marketplace	4
入門	6
無料のソフトウェア製品を公開するための販売者の要件	6
有料製品の販売者の追加要件	7
有料製品の対象となる管轄区域	8
AWS Marketplace 管理ポータル	14
AWS Marketplace 販売者としての登録	14
ステップ 1: 公開プロフィールを作成する	15
ステップ 2: 税金情報を入力する	16
ステップ 3: 銀行口座情報を入力する	18
ステップ 4: 支払い設定を設定する	19
ステップ 5: 顧客確認 (KYC) プロセスを完了する	19
ステップ 6: 銀行口座検証プロセスを完了する	23
すでに AWS Marketplace 販売者ですか？	24
支払い設定	25
支払いの詳細の設定	25
AWS Marketplace サブスクリプションの請求	26
販売者が支払いを受ける方法	26
利用可能な通貨	27
出品料金	28
での標準オファーストの料金について AWS Marketplace	28
でのリージョン別オファーストの料金について AWS Marketplace	29
Amazon Payments Europe の苦情	29
苦情の申し立て	30
Amazon Payments Europe の苦情解決期間	30
苦情のエスカレーション	30
AWS Marketplace 販売者向けのその他のツール	31
AWS Marketplace Commerce Analytics サービス	32
AWS Marketplace フィールドデモンストレーションプログラム	51
AWS パートナーアシスタント	52
製品の準備	58

製品デリバリー	58
で実行されている製品の指定 AWS	63
製品の料金	63
料金モデル	64
料金モデルの変更	68
料金の変更	69
プライベートオファー	69
製品の返金	70
リージョンと国	75
AWS リージョン	75
Countries (国)	75
翻訳と言語	76
入門	76
翻訳のオプトアウト	77
標準契約書	78
の標準契約 AWS Marketplace	78
のリセラー契約 AWS Marketplace	80
カテゴリとメタデータ	81
製品の名前付けと説明	81
カテゴリとキーワードの選択	83
AMI とコンテナ製品の使用手順	84
要件	84
リリースノートの記事	85
使用手順の記事	85
アップグレード手順の記事	86
CloudFormation 配信手順の記事	86
検索エンジンの最適化	86
検索エンジンの最適化	87
AWS Marketplace 検索	88
プロモーションメディア	90
プロモーションメディアのベストプラクティス	90
プロモーションメディアを追加および管理するためのヒント	92
プライベートオファーの準備	94
プライベートオファーの仕組み	95
プライベートオファーに関する考慮事項	95
プライベートオファーの購入者エクスペリエンス	96

プライベートオファーのレポート	97
サポートされる製品タイプ	98
AMI 製品のプライベートオファー	98
コンテナ製品のプライベートオファー	99
プロフェッショナルサービス製品のプライベートオファー	100
SaaS 製品のプライベートオファー	100
ML 製品のプライベートオファー	100
プライベートオファーの作成と管理	101
新しいプライベートオファーを開始する	102
オファーのステータスについて	102
プライベートオファーの草案作成と公開	103
プライベートオファーとデモリクエストボタンの追加	105
購入者へのプライベートオファーの送信	108
プライベートオファーのクローン作成	108
オファーの詳細のダウンロード	109
プライベートオファーの進行状況を保存する	109
プライベートオファーの有効期限の更新	110
プライベートオファーをキャンセルする	110
チャンネルパートナーのプライベートオファー	110
追加情報	112
ISV として販売認証を作成する	112
分割プランの作成	117
プライベートオファーの分割プランの作成	117
分割プランレポート	118
将来の日付の契約	119
将来の日付の契約に関する考慮事項	120
将来の日付の契約の作成	120
将来の日付の契約での分割プランの使用	121
将来の日付の契約の通知を受け取る	121
チャンネルパートナーのプライベートオファーの再販に関する将来の日付の契約を使用する	122
プライベートオファーに関するよくある質問	123
Private Offer Success Team (POST) とは何ですか？ 販売者はどのように連絡すればよいですか？	123
販売者はオファータブにアクセスするときにどのようにエラーを解決できますか？	123
分割払いプランまたは柔軟な支払いスケジュールとは	124

プライベートオファーにアクセスするときに 404 エラーを受け取った購入者を支援するにはどうすればよいですか？	126
プライベートオファーを承諾しようとする、「既にアクティブな契約がある」というエラーが表示されるのはなぜですか？	127
販売者または購入者はプライベートオファーをキャンセルできますか？	129
返金または契約のキャンセルをリクエストするにはどうすればよいですか？	129
購入者に請求されるのはいつですか？	130
オファーが承諾されたら、販売者はどのような手順を実行する必要がありますか？	130
販売者やパートナーへの AWS 支払い方法	131
は税金をどのように AWS 評価しますか？	131
リソースとサポート	131
契約の使用	133
契約タイプ	133
契約の使用	133
契約の検索	134
契約のソート	134
契約の詳細の表示	134
契約の詳細のダウンロード	134
契約の修正	135
パブリックオファーとプライベートオファーの修正でサポートされている製品タイプ	135
パブリックおよびプライベートオファーのアップグレード、更新、修正の作成	136
アップグレード、更新、および修正のレポート	138
AMI ベースの製品	139
AMI ベースの製品配信方法	139
追加リソース	139
AMI ベースの製品について	140
製品のライフサイクル	140
AMI 製品コード	143
変更リクエスト	143
製品ロードフォーム	145
年次契約の変更	145
AMI ベースの製品の作成	146
前提条件	146
セルフサービスエクスペリエンスを理解する	147
リストを作成する	148
追加リソース	150

CloudFormation テンプレートを製品に追加する	151
CloudFormation テンプレートの準備	151
アーキテクチャ図	159
既存の製品の CloudFormation テンプレートを変換する	160
サーバーレスアプリケーションコンポーネントの追加	165
AMI ベースの製品の管理	175
変更リクエストの作成	176
製品の可視化の更新	179
インスタンスの追加と制限	181
バージョンの管理	183
製品情報の更新	190
可用性の管理	191
EULA を更新する	195
返金ポリシーの更新	195
AMI AWS Marketplace へのアクセスの許可	196
製品を削除する	197
トラブルシューティング	198
AMI 構築のベストプラクティス	200
再販権の確保	200
AMI の構築	201
の AMI の準備と保護 AWS Marketplace	202
公開要件を確認するための AMI のスキャン	203
AWS Marketplace AMI でソフトウェアが実行されていることを確認する	204
AMI 製品の料金	206
AMI 料金モデル	207
AWS 料金とソフトウェア料金	211
AWS Marketplace Metering Service によるカスタム計測	212
AMI 製品の契約料金	228
の使用 AWS License Manager	232
Amazon SNS 通知を受信する	246
Amazon SNS トピック: aws-mp-subscription-notification	246
Amazon SNS トピックへ Amazon SQS キューをサブスクライブする	247
AMI 製品チェックリスト	248
製品の使用	248
AMI の準備	248
Windows AMI	249

Linux AMI	249
製品ロードフォームまたは製品タブ	249
AMI ベースの製品要件	250
AMI 製品販売者ポリシー	251
セキュリティポリシー	251
アーキテクチャポリシー	253
AMI 製品使用説明書	254
AMI 製品バージョンポリシー	254
カスタマー情報ポリシー	254
製品使用ポリシー	254
Image Builder コンポーネントベースの製品	257
変更リクエストの作成	257
セルフサービスを使用して変更リクエストを作成します。	258
変更リクエストを作成する	259
変更リクエストのステータスを取得する	260
追加リソース	260
製品の可視化の更新	260
製品の可視性を更新する	261
許可リスト (プレビューアカウント) を更新する	261
インスタンスの追加と制限	262
インスタンスの追加	262
インスタンスの制限	263
バージョンの管理	264
製品情報の更新	264
可用性の管理	266
を追加する AWS リージョン	266
の制限 AWS リージョン	267
将来の AWS リージョンのサポートの更新	268
国別の可用性の更新	268
EULA を更新する	269
返金ポリシーの更新	270
製品を削除する	270
コンテナベースの製品	273
ヘルプの利用	274
コンテナ製品の開始方法	274
製品のライフサイクル	274

前提条件	275
概要: コンテナ製品を作成する	276
ステップ 1: コンテナ製品の製品 ID と製品コードを作成する	276
ステップ 2: 初期リストを作成する	277
ステップ 3: 製品の初期バージョンを追加する	277
ステップ 4: (有料製品のみ) 計測または契約料金を統合する	278
ステップ 5: 製品の可視性を更新する	278
次のステップ	279
コンテナ製品はセキュリティ上の問題をスキャンします。	280
製品情報の更新	280
製品バージョンを追加する	281
製品価格を管理する	291
国別の可用性の更新	294
EULA を更新する	295
製品のテストとリリース	295
コンテナベースの製品要件	297
セキュリティポリシー	298
顧客情報の要件	298
製品の使用要件	299
アーキテクチャの要件	300
コンテナ製品の使用手順	301
Amazon EKS アドオン製品の要件	301
コンテナ製品の料金	319
コンテナ料金モデル	320
コンテナ製品の契約料金	324
コンテナ製品の請求、計測、ライセンスの統合	327
AWS Marketplace Metering Serviceによる時間単位計測およびカスタム計測	328
との契約料金 AWS License Manager	331
AWS Marketplace Metering Service による時間単位の計測を設定する	332
AWS Marketplace Metering Service を使用したカスタム計測の設定	344
との契約料金 AWS License Manager	357
コンテナ製品の Amazon SNS 通知	391
Amazon SNS トピック: aws-mp-subscription-notification	391
Amazon SNS トピックへ Amazon SQS キューをサブスクライブする	392
機械学習製品	394
機械学習製品について	394

SageMaker AI モデルパッケージ	394
SageMaker AI アルゴリズム	395
推論モデルのデプロイ	395
製品のライフサイクル	396
機械学習製品の料金	397
サービスの制限とクォータ	400
セキュリティと知的財産	403
の機械学習レポート AWS Marketplace	405
SageMaker AI で製品を準備する	406
コードをイメージにパッケージ化する	406
イメージのアップロード	431
Amazon SageMaker AI リソースの作成	434
での製品の一覧表示 AWS Marketplace	440
前提条件	441
ステップ 1: 新しいリストを作成する	444
ステップ 2: 製品情報を提供する	444
ステップ 3: 初期製品バージョンを追加する	445
ステップ 4: 料金モデルを設定する	446
ステップ 5: 返金ポリシーを設定する	447
ステップ 6: EULA を設定する	447
ステップ 7: 許可リストを設定する	447
製品の管理	448
.....	448
.....	449
.....	449
.....	450
.....	452
.....	452
.....	452
製品を削除する	453
プライベートオファターの作成	453
要件とベストプラクティス	455
ML 製品の一般的なベストプラクティス	456
使用状況情報の要件	457
入力と出力の要件	457
Jupyter Notebook の要件	458

ML 製品リストの要件と推奨事項の要約	458
トラブルシューティング	462
製品 ARN を追加するときの 400 エラー	463
製品 ARN を追加するときの 404 エラー	463
製品作成時の SageMaker AI アクセス拒否エラー	464
製品作成時の SageMaker AI の失敗	465
SaaS 製品	466
SaaS 製品を開始する	466
前提条件	467
次のステップ	467
SaaS 製品のライフサイクル	467
SaaS 製品の作成	468
初期 SaaS 製品ページの作成	471
SaaS 製品設定の設定	473
SaaS サブスクリプション製品の統合	484
SaaS 契約製品の統合	488
従量制料金製品を含む SaaS 契約との統合	491
サーバーレス SaaS 統合のデプロイ	496
SaaS 製品の計画	496
料金の計画	496
請求の統合の計画	497
Amazon SNS インテグレーションを計画する	497
顧客が製品にアクセスする方法の計画	497
SaaS 製品ガイドライン	498
製品設定ガイドライン	499
顧客情報の要件	499
製品使用ガイドライン	500
アーキテクチャガイドライン	500
SaaS 製品の料金	506
SaaS サブスクリプションの料金	507
SaaS 契約の料金	509
SaaS 無料トライアルの作成	513
SaaS 無料トライアルオファーの作成	514
SaaS 無料トライアルオファーのキャンセル	515
顧客オンボーディング	515
新しい購入者を受け入れるための SaaS 製品の設定	516

セキュリティと注文	518
SaaS 製品に関する Amazon SNS 通知	518
Amazon SNS トピック: aws-mp-entitlement-notification	519
Amazon SNS トピック: aws-mp-subscription-notification	520
SNS トピックへの SQS キューのサブスクライブ	522
AWS Marketplace Metering Service API および使用権限管理サービス API へのアクセス	522
使用量の計測	523
権限を確認する	530
SaaS 製品統合チェックリスト	531
レポート作成	535
SaaS のコード例	536
ResolveCustomer コード例	536
GetEntitlement コード例	537
BatchMeterUsage コード例	539
使用量割り当てタグ付きの BatchMeterUsage のコード例 (オプション)	541
の使用 AWS PrivateLink	543
序章	544
製品の設定	545
製品を に送信する AWS Marketplace	546
VPC エンドポイントへの購入者のアクセス	547
付録: チェックリスト	548
のプロフェッショナルサービス製品 AWS Marketplace	550
ヘルプの利用	551
プロフェッショナルサービス製品の開始方法	551
前提条件	551
プロフェッショナルサービス製品の作成	552
プライベートオファーを作成する	553
製品情報の編集	555
製品の料金の編集	556
製品の可視性の編集	557
プロフェッショナルサービス製品の削除	557
製品の詳細の提供	558
製品の説明	559
追加リソース	561
サポート情報	561
料金ディメンション	561

製品の可視性	561
製品要件	562
製品設定ガイドライン	562
顧客情報の要件	563
製品使用ガイドライン	563
アーキテクチャガイドライン	563
プロフェッショナルサービス製品の料金	564
プライベートオファターの仕組み	564
データ製品	566
製品の送信	567
[製品] タブの使用	569
会社と製品のロゴ要件	570
有料の再パッケージ版ソフトウェアを送信するための要件	570
ハードウェアコンポーネントを持つ製品の要件	571
AWS CloudFormationが発表した製品 (無料または有料) または使用量ベースの有料 AMI 製品	572
製品の更新	576
製品の変更と更新	577
タイミングと心構え	578
への AMIs の送信 AWS Marketplace	578
AMI セルフサービススキャン	579
AMI クローン作成と製品コード割り当て	579
最終チェックリスト	580
製品のマーケティング	582
180 日間の GTM アカデミー	582
製品が入手可能になったことの発表	583
AWS Marketplace メッセージング	583
のレビュー AWS Marketplace	584
へのリンク AWS Marketplace	585
AWS Marketplace ブランディングの使用	585
販売 AWS 者として での購入について	586
AWS 前提条件付きで購入する	587
.....	588
ブランドの変更を確認する	588
AWS レポートで購入する	588
で製品に直接リンクする AWS Marketplace	589

プレスリリース	589
AWS Marketplace 商標使用ガイドライン	590
通知	592
E メール通知	593
イベントタイプ	593
フィールドの説明	606
通知の管理	611
Amazon EventBridge 通知	612
新しいオファターのイベント	613
変更セットのイベント	617
セキュリティ概要レポートのイベント	619
支払いのイベント	620
出品者レポート、データフィード、ダッシュボード	622
販売者配信データフィード	623
データフィードのストレージと構造	623
データフィードへのアクセス	626
データフィードの使用	631
データフィードテーブルの概要	632
データフィードクエリの例	640
データフィード	763
販売者レポート	805
レポートへのアクセス	806
利用可能な AWS Marketplace 販売者レポート	807
日別ビジネスレポート	807
日次顧客サブスクライバーレポート	818
支払いレポート	821
月別請求済み収益レポート	829
販売補償レポート	839
補足レポート	842
契約詳細レポート	842
販売者ダッシュボード	843
ダッシュボードへのアクセス	806
財務業務用ダッシュボード	846
販売業務用ダッシュボード	878
マーケティング用ダッシュボード	895
AWS Marketplace Vendor Insights	914

AWS Marketplace Vendor Insights について	915
販売者として設定する	916
セキュリティプロファイルを作成する	916
証明書のアップロード	917
自己評価をアップロードする	919
AWS Audit Manager 自動評価を有効にする	920
プロファイルの表示	926
販売者としてのセキュリティプロファイルを確認する	927
スナップショットの管理	928
スナップショットを作成する	929
スナップショットを表示する	929
スナップショットをエクスポートする	930
最新リリースのスナップショットを表示する	930
スナップショットリリースを延期する	931
スナップショットリストの設定を変更する	931
アクセスコントロール	932
AWS Marketplace Vendor Insights 販売者のアクセス許可	933
CreateDataSource	933
DeleteDataSource	933
GetDataSource	934
UpdateDataSource	934
ListDataSources	934
CreateSecurityProfile	934
ListSecurityProfiles	935
GetSecurityProfile	935
AssociateDataSource	935
DisassociateDataSource	935
UpdateSecurityProfile	935
ActivateSecurityProfile	936
DeactivateSecurityProfile	936
UpdateSecurityProfileSnapshotCreationConfiguration	936
UpdateSecurityProfileSnapshotReleaseConfiguration	937
ListSecurityProfileSnapshots	937
GetSecurityProfileSnapshot	937
TagResource	937
UntagResource	938

ListTagsForResource	938
追加リソース	938
セキュリティ	251
の IAM AWS Marketplace	940
ユーザーの作成	942
グループの作成または使用	943
ユーザーとしてのサインイン	945
AWS Marketplace 販売者のポリシーとアクセス許可	945
AWS Marketplace 販売者のポリシー	946
AWS Marketplace 販売者のアクセス許可	947
例 1: KYC ステータスを表示するアクセス許可	950
例 2: プライベートオファターのアップグレードと更新を作成するためのアクセス許可	951
例 3: オファラーページにアクセスして新しいプライベートオファラーを作成する権限	952
例 4: 設定ページにアクセスする権限	952
例 5: ファイルのアップロードページにアクセスする権限	953
IAM グループの使用	953
AWS マネージドポリシー	954
AWSMarketplaceAmiIngestion	955
AWSMarketplaceFullAccess	955
AWSMarketplaceGetEntitlements	956
AWSMarketplaceMeteringFullAccess	956
AWSMarketplaceMeteringRegisterUsage	956
AWSMarketplaceSellerFullAccess	956
AWSMarketplaceSellerProductsFullAccess	960
AWSMarketplaceSellerProductsReadOnly	962
AWSMarketplaceSellerOfferManagement	963
AWSVendorInsightsVendorFullAccess	964
AWSVendorInsightsVendorReadOnly	964
ポリシーの更新	964
AWS Marketplace Commerce Analytics Service アカウントのアクセス許可	970
Amazon SQS のアクセス許可	972
AWS Marketplace 計測と使用権限 API アクセス許可	973
SaaS 製品の IAM ポリシー	973
AMI 製品の IAM ポリシー	974
コンテナ製品の IAM ポリシー	974
再販認証用のサービスリンクロール	975

のサービスにリンクされたロールのアクセス許可 AWS Marketplace	976
のサービスにリンクされたロールの作成 AWS Marketplace	979
のサービスにリンクされたロールの編集 AWS Marketplace	979
のサービスにリンクされたロールの削除 AWS Marketplace	979
AWS Marketplace サービスにリンクされたロールでサポートされているリージョン	980
を使用した AWS Marketplace API コールのログ記録 AWS CloudTrail	980
AWS Marketplace 計測 API ログファイルエントリの例	981
トラブルシューティング	987
一般的な販売者登録エラーメッセージ	987
ドキュメント履歴	990
AWS 用語集	1014
.....	mxv

とは AWS Marketplace

AWS Marketplace は、お客様がサードパーティーのソフトウェア、データ、サービスを検索、購入、デプロイ、管理してソリューションを構築し、ビジネスを運営するために使用できる厳選されたデジタルカタログです。AWS Marketplace には、セキュリティ、ビジネスアプリケーション、機械学習、医療、金融サービス、通信などの特定の業界にわたるデータ製品などの一般的なカテゴリからの数千のソフトウェアリストが含まれています。お客様は、数回クリックするだけで事前設定されたソフトウェアをすばやく起動し、ソフトウェアソリューションを Amazon マシンイメージ (AMI) 形式、Software as a Service (SaaS)、およびその他の形式で選択できます。また、サードパーティー製ソフトウェアの設定、デプロイ、管理を支援するプロフェッショナルサービスも利用できます。配送方法の詳細なリストについては、「[製品の配送](#)」を参照してください。

購入者 (サブスクライバー)、販売者 (プロバイダー)、またはその両方 AWS Marketplace として使用できます。を持つユーザーは誰でも購入者 AWS Marketplace として AWS アカウント を使用し、販売者として登録できます。販売者は、独立系ソフトウェアベンダー (ISV)、チャネルパートナー、マネージドサービスプロバイダー (MSP)、または AWS 製品やサービスと連携する何かを提供する個人です。

Note

データ製品プロバイダーは、AWS Data Exchange の資格要件を満たす必要があります。詳細については、「AWS Data Exchange ユーザーガイド」の「[AWS Data Exchange でのデータ製品の提供](#)」を参照してください。

対象となるパートナーは、プログラムでの外部に AWS Marketplace 製品を一覧表示できます AWS Marketplace。対象となるパートナーになる方法の詳細については、AWS Marketplace 事業開発パートナーにお問い合わせください。

次の動画では、販売について詳しく説明します AWS Marketplace。

[の概要 AWS Marketplace](#)

販売 AWS Marketplace 者としての の使用

でソフトウェア製品を販売するプロセス AWS Marketplace には、次の 7 つのステップが含まれます。

販売者のプロセス

[ステップ]	[アクション]	説明
1	登録	販売者は、まずに登録します AWS Marketplace 管理ポータル。既存の AWS 組織と簡単にリンク AWS アカウント できる新しい専用 を実装することをお勧めします。AWS パートナーの税情報が管轄区域の資格基準を満たしていることを確認します。支払いを受けるには、銀行口座を指定する必要があります。
2	製品タイプを決定する	販売する製品の種類を決定します。で製品タイプを作成する方法の詳細については AWS Marketplace、以下を参照してください。 • の AMI ベースの製品 AWS Marketplace • のコンテナベースの製品 AWS Marketplace • の機械学習製品 AWS Marketplace • の SaaS ベースの製品 AWS Marketplace • のプロフェッショナルサービス製品 AWS Marketplace • データ製品 (データ製品の詳細については、「AWS Data Exchange ユーザーガイド」の「AWS Data Exchange とは」を参照してください)。
3	製品を準備する	パッケージと料金スキームを設定し、製品が表示される関連するカテゴリを決定して、製品が関連のある検索に表示されるようにキーワードを追加します。調達プロセスを簡素化するには、公開出品とプライベートオファーの両方で 標準化されたライセンス条項 を使用できます。
4	製品を送信する	製品を AWS Marketplaceで入手可能にするには、製品送信プロセスを使用します。製品はシンプルにすることができます。料金体系が1つの、単一の Amazon マシンイメージ (AMI) がその例です。または、複数の AMI、AWS CloudFormation テンプレート、複雑な価格オプションと支払いスケジュールを含めて、複雑にすることもできます。

[ステップ]	[アクション]	説明
5	製品を売る	の認識を高め、製品ページに直接トラフィックを誘導することで AWS Marketplace、製品の成功に貢献します AWS Marketplace。
6	レポートとデータフィードを表示する	販売者として登録したら、AWS Marketplace 管理ポータル を使用して製品の使用状況レポートにアクセスします。は、製品の販売に関する情報を収集および分析するためのツール AWS Marketplace を提供します。
7	製品を管理する	AWS Marketplace 管理ポータル を使用してアカウントと製品ページを管理します。

販売者は、[AWS Marketplace 管理ポータル](#) に移動して登録します。製品の使用に対して請求する場合は、登録の一部として税金と銀行の情報も指定する必要があります。登録時に AWS Marketplace で検索可能な会社のプロフィールまたは自分自身のプロフィールを作成します。[AWS Marketplace 管理ポータル](#) を使用して製品用の製品ページを作成して管理することもできます。

の契約構造 AWS Marketplace

で販売されるソフトウェア、サービス、およびデータ製品の使用は、購入者と販売者の間の契約によって管理 AWS Marketplace されます。AWS は、これらの契約の当事者ではありません。

販売者に適用される契約は以下のとおりです。

- 購入者とのエンドユーザーライセンス契約 (EULA)。この契約は、AWS Marketplaceに出品されている公開ソフトウェアの製品リストページに記載されています。多くの販売者は、デフォルトの EULA として [AWS Marketplace \(SCMP\) の標準契約](#) を使用します。SCMP をプライベートオファターの交渉のベースとして使用したり、修正テンプレートを使用して SCMP を変更したりすることもできます。プライベートオファターには、当事者間で交渉したカスタム契約条件を含めることもできます。

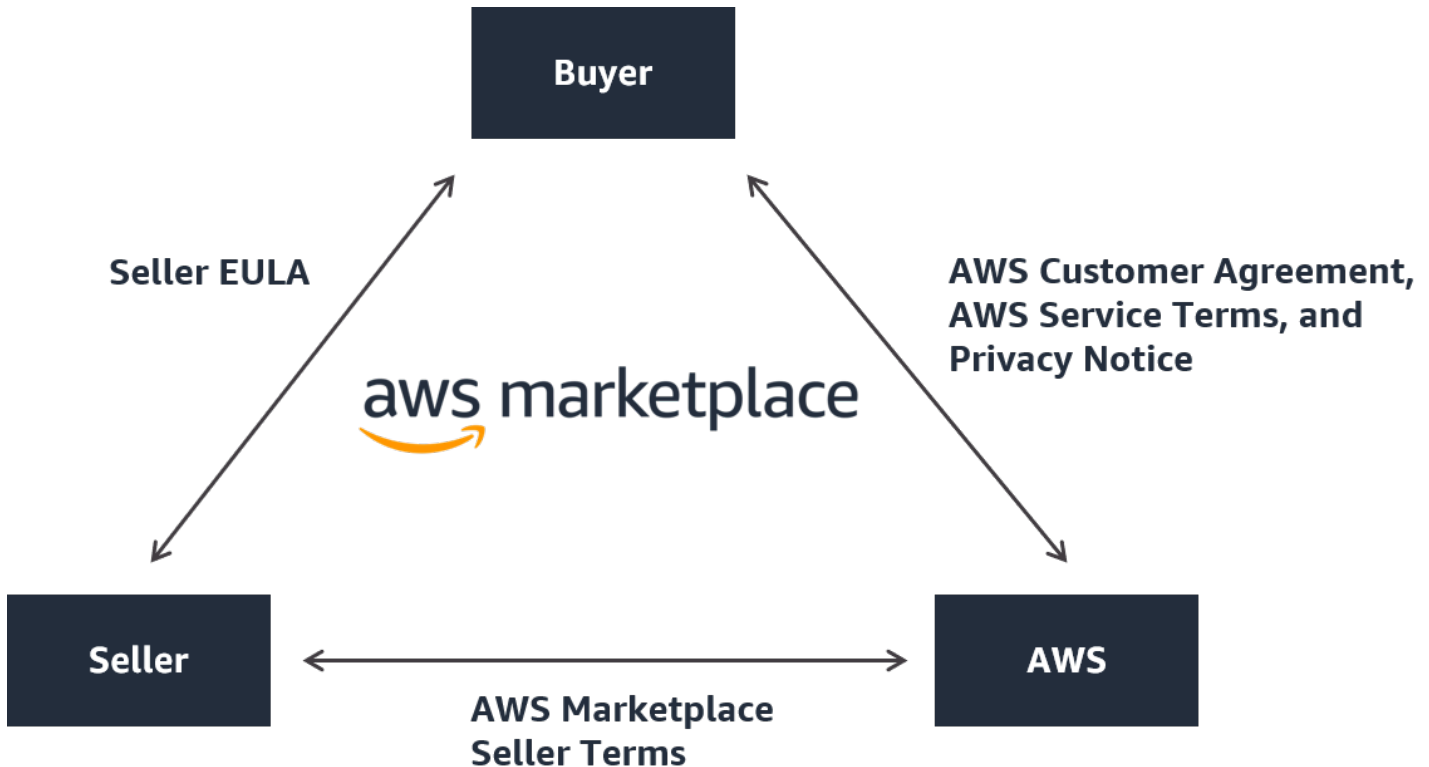
Note

オファertypeと料金モデルに基づく EULA 更新の発生タイミングについては、「AWS Marketplace 購入者ガイド」の「[EULA updates](#)」を参照してください。

- AWS Marketplaceでのアクティビティを規定する「[AWS Marketplace 販売者規約](#)」。

購入者による の使用 AWS Marketplace には、[AWS サービス条件](#)、[AWS カスタマーアグリーメント](#)、および[プライバシー通知](#)が適用されます。

次の図は、 の契約構造を示しています AWS Marketplace。



での製品の料金 AWS Marketplace

では AWS Marketplace、製品は無料で使用でき、関連する料金が発生する可能性があります。料金は購入者の AWS 請求書の一部となり、購入者が支払うと、 が販売者に AWS 支払います。製品はたくさんの形式を持つことができます。例えば、製品を購入者の AWS アカウントを使用してインスタンス化された Amazon マシンイメージ (AMI) として提供することができます。購入者への配送に CloudFormation テンプレートを使用するように製品を設定することもできます。製品は、ISV からの SaaS サービス、ウェブアクセスコントロールリスト (ウェブ ACL)、ルールセット、または AWS WAF の条件の場合もあります。ISV、チャネルパートナー、MSP が提供するプロフェッショナル サービスもあります。

柔軟な料金オプションには、無料トライアル、時間単位、月単位、年単位、複数年単位、Bring Your Own License モデル (BYOL) が含まれ、1 つのソースから請求されます。は請求と支払い AWS を処理し、料金はお客様の AWS 請求に表示されます。

ソフトウェア製品は、ISV の標準のエンドユーザーライセンス契約 (EULA) を利用して出品された価格で購入できます。さらに、ソフトウェア製品は、独自の料金や EULA を適用してプライベートオファーで提供することもできます。製品は、時間や使用範囲を指定した契約の下で購入することもできます。製品をサブスクライブした後、購入者は AWS Service Catalog を使用して製品をコピーし、購入者の組織で製品がどのようにアクセスおよび使用されるかを管理できます。購入者のエクスペリエンスについて詳しくは、「<https://docs.aws.amazon.com/marketplace/latest/buyerguide/service-catalog.html>」を参照してください。料金の詳細については、「[the section called “製品の料金”](#)」を参照してください。

AWS Marketplace 販売者としての開始方法

独立系ソフトウェアベンダー (ISV)、チャネルパートナー、マネージドサービスプロバイダー (MSP)、または AWS 製品やサービスと連携する何かを提供している個人は、販売者として登録できます AWS Marketplace。販売者として登録することは、製品を販売用に公開するための前提条件です AWS Marketplace。お住まいの場所と販売する製品の種類に応じて、の登録要件 AWS Marketplace は異なります。以下のセクションでは、販売者登録プロセス、要件、および関連ツールの概要について説明します。

でソフトウェアを販売するには AWS Marketplace、次の手順に従います。

- [無料製品の販売者要件](#)および[有料製品の要件](#)を確認します。
- [販売者登録プロセス](#)を完了します。
- [次のステップ](#)と[販売者ツール](#)の詳細を確認します。

メモ

- AWS Marketplace 販売者として登録することは、AWS Data Exchange でデータ製品を一覧表示し、利用可能にするための前提条件です AWS Marketplace。これらの要件の詳細については、AWS [Data Exchange ユーザーガイドの AWS 「Data Exchange でのデータ製品の提供」](#)を参照してください。
- AWS Marketplace 販売者が必要とするアクセス許可については、「」を参照してください [AWS Marketplace 販売者のポリシーとアクセス許可](#)。
- 商品出品手数料の詳細について、登録済み販売者は AWS Marketplace 管理ポータル「[AWS Marketplace 販売者規約](#)」を参照することができます。
- よくある質問への回答については、[AWS Marketplace 「Sellers FAQ」](#)を参照してください。

無料のソフトウェア製品を公開するための販売者の要件

製品を提供するときに料金を請求するかどうかにかかわらず AWS Marketplace、その製品を販売していることになります。顧客への費用は 0.00 USD ですが、販売者と顧客は製品の使用に関する相互契約に同意することになります。無料の製品のみを提供する場合は、AWS Marketplaceに銀行情報を提供する必要はありません。

で無料の製品を作成して提供するには AWS Marketplace、以下を実行する必要があります。

- 全機能を使用できる一般公開版の本番環境用ソフトウェアを販売する。
- 明確なカスタマーサポートプロセスおよびサポート組織があること。
- ソフトウェアを定期的に更新し、脆弱性の影響を受けないようにする手段を提供している。
- AWS Marketplaceで製品を販売するときのベストプラクティスとガイドラインに従う。
- 優良な AWS 顧客であり、AWS Marketplace 販売者の利用規約の要件を満たしている。

有料製品の販売者の追加要件

製品の代金を請求する場合、または自分のライセンス使用モデル (BYOL) 製品を販売する場合は、以下の要件も満たし、以下の追加情報を提供する必要があります。

- [対象となる管轄区域](#)の永住者または市民であるか、これらの区域のいずれかで組織化または法人化された事業体である必要があります。
- 税金および銀行口座の情報を提供する必要があります。米国を拠点とする企業の場合は、W-9 フォームと、米国を拠点とする銀行からの銀行口座が必要です。パブリックオファーまたはプライベートオファーを作成するには、銀行口座情報を[支払い設定](#)に関連付ける必要があります。
- 米国以外の販売者は、(i) W-8 フォーム、付加価値税 (VAT)、商品・サービス税 (GST) 登録番号、および (ii) 対象の管轄区域の銀行口座および SWIFT コードを提供する必要があります。必要に応じて、[Hyperwallet](#) からオンラインの米国銀行口座を登録できます。
- データ製品を提供するには、[ケース作成](#)ウィザードを使用してオンボーディングをリクエストする必要があります サポート。
- Amazon Web Services EMEA SARL を通じて、欧州、中東、アフリカ (EMEA) (トルコと南アフリカを除く) の国と地域に拠点 AWS アカウント を置く顧客に製品を販売するには、[顧客確認プロセスを完了](#)する必要があります。加えて:
 - 最大 2 つの支払いを受け取ります (AWS Inc. および Amazon Web Services EMEA SARL を介したトランザクションの場合)。
 - 場所によっては、特定の取引について出品手数料に課税される場合があります。税金の詳細については、[AWS Marketplace 「Sellers Tax」 help](#) ページを参照してください。出品料に付加価値税 (VAT) が課せられる場合、AWS Marketplace は税制に準拠した請求書を発行します。
 - Amazon Web Services EMEA SARL の詳細については、「[Amazon Web Services ヨーロッパ FAQ](#) ウェブサイト」の「AWS EMEA Marketplace - 出品者」を参照してください。

に販売するには AWS GovCloud (US) Region、販売者が [AWS GovCloud \(US\) アカウント](#) を持っている必要があります。ITAR 要件の詳細については、「[AWS GovCloud \(US\) ユーザーガイド](#)」を参照してください。

AWS Marketplace 販売者の要件または登録プロセスに関するご質問は、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。

有料製品の対象となる管轄区域

有料ソフトウェアを販売するには AWS Marketplace、次のいずれかの国または SARs の永住者または市民であるか、その中に組織または法人化された事業体である必要があります。

- オーストラリア¹
- バーレーン^{1 2}
- コロンビア^{1 2}
- 欧州連合 (EU) 加盟国¹
- 香港特別行政区
- イスラエル^{1 2}
- 日本³
- ニュージーランド¹
- ノルウェー^{1 2}
- カタール
- 韓国
- スイス^{1 2}
- アラブ首長国連邦 (UAE)^{1 2}
- 英国 (UK)¹
- [米国 (US)]

¹ これらの国の有料製品の販売者は、設立国の VAT 登録情報を提供する必要があります。

² 販売者が購入者と同じ国に居住している場合、販売者は税金の請求、回収、送金の責任を担う場合があります。税務顧問にお問い合わせください。

³ 2025 年 4 月 1 日以降、10% の日本の消費税 (JCT) の徴収と送金、および で販売された製品に対する AWS 日本の税対象請求書 (T") の発行には、以下の手順が適用されます AWS Marketplace。

販売者と顧客のシナリオ	税金の徴収と請求の手順
販売者 – 住所が日本以外の独立系ソフトウェアベンダー (ISV)。 顧客 – AWS 日本の住所を持つアカウント。	指定されたプラットフォーム税ルールに従って、AWS 日本は以下を実行します。 • 10% の日本の消費税 (JCT) を徴収します。 • AWS 日本の JCT 番号 T6011001106696 に基づいて、税対象請求書 (T") を顧客に発行します。 • 徴収した日本の消費税 (JCT) を日本の税務局 (JTA) に送金します。 重複を避けるために、ISV は日本の消費税 (JCT) の徴収または送金と、税対象請求書 (TQIs) の発行を停止する必要があります。
販売者 – 日本の住所を持つ独立系ソフトウェアベンダー (ISV)。 顧客 – AWS 日本の住所を持つアカウント。	Special Agency ルールに従って、AWS 日本は以下を実行します。 • 10% の日本の消費税 (JCT) を徴収します。 • AWS 日本の JCT 番号 T6011001106696 に基づいて、税対象請求書 (T") を顧客に発行します。 • 収集した JCT を ISV に支払います。 ISV は以下を実行する必要があります。 • Amazon Marketplace 管理ポータルに日本の企業番号と日本の消費税 (JCT) 番号を入力します。 • Amazon Marketplace 管理ポータルで利用可能な税対象請求書 (T") 情報を使用して、収集された JCT を日本の税務局 (JTA) に送金します。

販売者と顧客のシナリオ	税金の徴収と請求の手順
日本以外のチャネルパートナープライベートオフアー (CPPO)、ISV 1. 最初のトランザクション - 販売者 – 住所が日本以外の独立系ソフトウェアベンダー (ISV)。 - 顧客 – AWS Marketplace 日本の住所を持つチャネルパートナー。 2. 2 番目のトランザクション - 販売者 – 同じ AWS Marketplace チャネルパートナー。 - 顧客 – AWS 日本の住所を持つアカウント。	最初の取引では、チャネルパートナーは、リバースチャージ税メカニズムに従って日本の消費税 (JCT) を報告する必要があります。 2 番目のトランザクションの場合： - Special Agency ルールに従って、AWS 日本は以下を実行します。 10% の日本の消費税 (JCT) を徴収します。 - AWS 日本の JCT 番号 T6011001106696 に基づいて、税対象請求書 (T") を顧客に発行します。 - 収集した日本の消費税 (JCT) をチャネルパートナーに支払います。 - チャネルパートナーは以下を実行する必要があります。 - Amazon Marketplace 管理ポータルに日本の企業番号と日本の消費税 (JCT) 番号を入力します。 - Amazon Marketplace 管理ポータルで利用可能な税対象請求書 (T") 情報を使用して、収集された JCT を日本の税務局 (JTA) に送金します。

販売者と顧客のシナリオ	税金の徴収と請求の手順
チャンネルパートナープライベートオファー (CPPO)、日本のすべての販売者および顧客 1. 最初のトランザクション - 販売者 – 日本の住所を持つ独立系ソフトウェアベンダー (ISV)。 - 顧客 – AWS Marketplace 日本の住所を持つチャンネルパートナー。 2. 2 番目のトランザクション - 販売者 – 同じ AWS Marketplace チャンネルパートナー。 - 顧客 – AWS 日本の住所を持つアカウント。	最初のトランザクションの場合： - チャンネルパートナーは、ISV から税対象請求書 (T") を受け取ります。 - AWS 日本は、請求書を発行したり、日本の消費税 (JCT) を徴収したりしません。 2 番目のトランザクションの場合： - Special Agency ルールに従って、AWS 日本は以下を実行します。 10% の日本の消費税 (JCT) を徴収します。 - AWS 日本の JCT 番号 T6011001106696 を使用して、税対象請求書 (T") を顧客に発行します。 - 収集した日本の消費税 (JCT) をチャンネルパートナーに支払います。 - チャンネルパートナーは以下を実行する必要があります。 - Amazon Marketplace 管理ポータルに日本の企業番号と日本の消費税 (JCT) 番号を入力します。 - Amazon Marketplace 管理ポータルで利用可能な税対象請求書 (T") 情報を使用して、収集された JCT を日本の税務局 (JTA) に送金します。

販売者と顧客のシナリオ	税金の徴収と請求の手順
チャンネルパートナープライベートオファー (CPPO)、ISV、および日本以外のチャンネルパートナー 1. 最初のトランザクション - 販売者 – 住所が日本以外の独立系ソフトウェアベンダー (ISV)。 - 顧客 – AWS Marketplace 住所が日本以外のチャンネルパートナー。 2. 2 番目のトランザクション - 販売者 – 同じ AWS Marketplace チャンネルパートナー。 - 顧客 – AWS 日本に住所があるアカウント。	最初のトランザクションでは、AWS 日本は請求書を発行したり、日本の消費税 (JCT) を徴収したりしません。これは JCT の範囲外であるためです。 2 番目のトランザクションの場合： - 指定されたプラットフォームビジネスルールに従って、AWS 日本は以下を実行します。 10% の日本の消費税 (JCT) を徴収します。 - AWS 日本の JCT 番号 T6011001106696 に基づいて、税対象請求書 (T") を顧客に発行します。 - 回収した日本の消費税 (JCT) を日本の税務局 (JTA) に送金します。 重複を避けるために、チャンネルパートナーは日本の消費税 (JCT) の徴収または納付を停止し、税対象請求書 (TQIs) の発行を停止する必要があります。

販売者と顧客のシナリオ	税金の徴収と請求の手順
チャンネルパートナープライベートオファァー (CPPO)、日本以外のチャンネルパートナー 1. 最初のトランザクション - 販売者 – 日本の住所を持つ独立系ソフトウェアベンダー (ISV)。 - 顧客 – AWS Marketplace 日本以外の住所を持つチャンネルパートナー。 2. 2 番目のトランザクション - 販売者 – 同じ AWS Marketplace チャンネルパートナー。 - 顧客 – AWS 日本 of の住所を持つアカウント。	最初のトランザクションでは、AWS 日本は請求書を発行したり、日本の消費税 (JCT) を徴収したりしません。これは JCT の範囲外であるためです。 2 番目のトランザクションの場合： - 指定されたプラットフォームビジネスルールに従って、AWS 日本は以下を実行します。 10% の日本の消費税 (JCT) を徴収します。 - AWS 日本 of の JCT 番号 T6011001106696 に基づいて、税対象請求書 (T") を顧客に発行します。 - 回収した日本の消費税 (JCT) を日本の税務局 (JTA) に送金します。 重複を避けるために、チャンネルパートナーは日本の消費税 (JCT) の徴収または納付を停止し、税対象請求書 (TQIs) の発行を停止する必要があります。

販売者としての VAT、請求、納税義務の詳細については、[Amazon Web Service Tax Help](#) の [AWS Marketplace 「販売者」](#) を参照してください。

上記の国または SARs [AWS Marketplace 「管轄区域外の企業のリソース」](#) を参照してください。

Note

AWS India Private Limited (AWS インド) アカウントは、以前は Amazon Internet Services Private Limited と呼ばれていましたが、登録には使用できません AWS Marketplace。

AWS Marketplace 管理ポータル

[AWS Marketplace 管理ポータル](#) を使用して、AWS Marketplaceで販売する商品を管理できます。登録されたすべての販売者は、製品の作成に使用したアカウントの AWS 認証情報 [AWS Marketplace 管理ポータル](#) を使用して にアクセスできます。使用するアカウントは、顧客が製品をサブスクライブするときの登録販売者として定義されます。製品の登録販売者である特定アカウントの決定に関してサポートが必要な場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

ポータルでは、以下のタスクを完了できます。

- AWS Marketplace 販売者として登録します。
- [製品] ページを使用して、新しいソフトウェア製品を送信したり、既存のソフトウェア製品を更新したりする。
- リクエストのステータスをモニタリングする。
- 新しいソフトウェア製品を作成して管理するために必要なファイルをアップロードする。
- 販売促進活動を活用してソフトウェア製品を管理し、チャネル収益増につなげる。
- ローンチから数時間以内に、使用状況およびキャンペーンを使用した収益を含め、マーケティング活動の結果を測定する。
- カスタマーサービス担当者がリアルタイムで顧客データを取得できるようにします。
- 自動 Amazon マシンイメージ (AMI) スキャンを開始して脆弱性を検出する。

Note

データ製品は AWS Data Exchange コンソールから公開および管理します。AWS Data Exchange プロバイダーは、AWS Marketplace 管理ポータル を使用して、販売者としての登録、AWS Data Exchange のオンボーディングのリクエスト、販売者レポートへのアクセス、返金リクエストの送信を行うことができます。

AWS Marketplace 販売者としての登録

独立系ソフトウェアベンダー (ISV)、チャネルパートナー、マネージドサービスプロバイダー (MSP)、または AWS 製品やサービスと連携する何かを提供している個人は、販売者として登録できます AWS Marketplace。販売者として登録することは、製品を販売用に公開するための前提条件で

す AWS Marketplace。以下のセクションでは、販売者として正常に登録する手順について説明します AWS Marketplace。

で販売者として登録するには AWS Marketplace、既存の を使用する AWS アカウント か、新しいアカウントを作成します。すべての AWS Marketplace インタラクションは、選択したアカウントに関連付けられます。では、ルートアカウントの認証情報を使用するの AWS Marketplace 管理ポータルではなく、AWS Identity and Access Management (IAM) ロールを使用して にサインインすることを AWS Marketplace 強くお勧めします。詳細については、「[AWS Marketplace セキュリティ](#)」を参照してください。IAM を使用して、さまざまなアクセス許可を持つ複数のユーザーに へのアクセスを許可する AWS アカウント ようにプライマリを設定することもできます AWS Marketplace 管理ポータル。詳細については、「[the section called “の IAM AWS Marketplace”](#)」を参照してください。

Note

重要な通知とアカウントの更新を受け取るには、組織内の適切な連絡先がアクセスできる E メールアドレスを指定する必要があります。一部の AWS チームは登録された E メールアドレスからのみ E メールを受信できるため、エイリアスを E メールアドレスに置き換えることはできません。

販売者として登録するには、次の手順を実行します。

トピック

- [ステップ 1: 公開プロフィールを作成する](#)
- [ステップ 2: 税金情報を入力する](#)
- [ステップ 3: 銀行口座情報を入力する](#)
- [ステップ 4: 支払い設定を設定する](#)
- [ステップ 5: 顧客確認 \(KYC\) プロセスを完了する](#)
- [ステップ 6: 銀行口座検証プロセスを完了する](#)
- [すでに AWS Marketplace 販売者ですか？](#)

ステップ 1: 公開プロフィールを作成する

登録する最初のステップは、プライマリ AWS Marketplace アカウント AWS アカウント として使用する を選択し、コンソールで潜在的な購入者に表示される情報を提供することです AWS

Marketplace。このアカウントは、で製品の登録販売者 AWS Marketplace となり、からお客様 AWS Marketplace への報告、支払い、通信に使用されます。

を使用して販売者として AWS アカウント 登録し、製品を一覧表示すると AWS Marketplace、製品に関連付けられたアカウントを変更することはできません。AWS Marketplace 販売者として登録するには、新しいアカウントを使用することをお勧めします。ただし、2017 年 9 月 27 日以降に作成されたアカウントであれば、既存のアカウントを使用できます。

公開プロフィールを作成するには

1. [AWS Marketplace 管理ポータル](#) (AMMP) から今すぐ登録を選択し、販売者にサインインします AWS アカウント。
2. [公開プロフィールを追加] を選択して販売者情報を入力します。

公開プロフィールを作成すると、無料の製品を公開して販売できます。有料製品を販売するには、税および銀行に関する情報を提供する必要があります。

ステップ 2: 税金情報を入力する

が製品の売上に対する税金 AWS Marketplace を正確に報告および徴収できるように、税金情報と該当する場合は付加価値税 (VAT) 情報を提供する必要があります。

税に関する情報を入力するには

1. [AWS Marketplace 管理ポータル](#) にサインインして [設定] を選択します。
2. [支払い情報] セクションの [税金ダッシュボードに移動] を選択します。
3. 米国税に関する質問に回答してください。
 - プロフェッショナルサービスを販売するには AWS Marketplace、DAC7 の税務調査を完了する必要があります。
 - 「税務面接の場所がビジネスの場所と一致しない場合」というエラーメッセージが表示された場合は、[請求情報とコスト管理で提供される銀行情報](#)と税金情報が、に入力されている情報と一致していることを確認してください AWS Marketplace 管理ポータル。Tax Interview の場所は、ビジネスの場所と一致する必要があります。
4. 税金情報を完了したら、設定ページに戻り、利用可能な場合は VAT 情報の入力を選択します。この選択により、AWS Billing コンソールの税設定ページにリダイレクトされます。

 Note

VAT 情報セクションは、VAT AWS リージョン をサポートする にいる場合にのみ使用できます。

補助税登録

特定の地域の売上にローカルの税登録番号 (TRN) を使用する場合は、アカウントに補足の TRN として追加できます AWS Marketplace 。たとえば、補足的なスイス TRN は、スイスの購入者の税金の徴収と請求を管理するために使用されます。補足 TRN に関連する会社住所は、独立したソフトウェアベンダー、チャネルパートナー、および購入者と共有される場合があります。

補足的な TRN がない場合、はプライマリ TRN を使用して税管轄区域 AWS Marketplace を決定します。詳細については、[「がアカウントの場所 AWS を決定する方法」を参照してください。](#)

補助税登録を追加するには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [設定] を選択します。
3. 設定ページで、税金設定タブを選択します。
4. 「補足税情報の追加」を選択します。
5. 補足税登録の詳細を入力します。TRN には、付加価値税 (VAT) 番号、VAT ID、VAT 登録番号、または事業者登録番号を指定できます。
6. [Submit] を選択してください。

税務書類にアクセスする

1099 フォームなどの税務書類には、 からアクセスできます AWS Marketplace 管理ポータル。

税務書類にアクセスするには

1. [AWS Marketplace 管理ポータル](#) にサインインして [設定] を選択します。
2. [支払い情報] セクションに移動します。
3. 関連する [税申告書] (1099K または DAC7) を選択します。
4. 納税申告書が利用可能な場合は、税務ダッシュボードページからダウンロードできます。

ステップ 3: 銀行口座情報を入力する

AWS Marketplaceで有料製品を販売するすべての販売者には、対象となる管轄区域の銀行口座が必要です。銀行口座は、米ドルの支払いを受け入れることができる必要があります。

Note

有料製品を提供できる国のリストについては AWS Marketplace、「」を参照してください [有料製品の対象となる管轄区域](#)。

銀行情報を入力するには

1. [AWS Marketplace 管理ポータル](#) にサインインして [設定] を選択します。
2. [支払い情報] セクションで [銀行情報の入力] を選択します。
3. 銀行口座に関する必要な情報を入力します。複数の銀行口座を指定することもできます。銀行口座には、米国の ACH アカウント、対象となる管轄区域の SWIFT 銀行口座、または Hyperwallet アカウントを使用することができます。

Note

税に関する情報 (および該当する場合は付加価値税情報) をまだ入力していない場合、銀行情報を入力することはできません。

米国ベースの銀行口座を使用して米ドルの支払いを受け入れる場合、Hyperwallet は米国アカウントを提供できます。

Hyperwallet は、資金を他の銀行口座にサポートされている通貨で送金することを可能にする、独立したサービスプロバイダーです。期間限定で、AWS Marketplace の支払いに対する特定の Hyperwallet サービス料金を支払うことは要求されません。

- Hyperwallet アカウントの詳細を AWS Marketplace 販売者アカウントに追加することで、AWS Marketplace が販売者としてのステータス AWS Marketplace を確認するためにお客様の名前、E メールアドレス、およびアカウント番号を と共有Hyperwalletすることに合意し、承認します。
- Hyperwallet サービス (資金を現地の通貨に送金するために必要な送金手数料および為替手数料を含みます) の使用に加え、為替レートに追加料金が適用される場合があります。Hyperwallet サービス料金は、有料製品からHyperwalletお客様のアカウントへの収益 AWS Marketplace の支払い

に関してのみ、期間限定で免除されます。詳細については、Hyperwallet サイトの料金セクションを参照してください。または、Hyperwallet まで詳細と適用される料金をお問い合わせください。サービスの詳細については、[Hyperwallet サポートサイト](#)を参照してください。

Hyperwallet への登録を開始し、米国の銀行口座情報を取得するには

1. [AWS Marketplace 管理ポータル](#) にログインして [設定] を選択します。次に [支払い情報] セクションの [銀行情報の入力] を選択します。
2. Hyperwallet 口座を持っておらず、AWS Marketplaceで使用する必要がある場合は、[米国の銀行口座をお持ちですか?] と [Hyperwallet に登録していますか?] に対して [いいえ] を選択してください。Hyperwallet に登録するための個人識別番号 (PIN) とリンクが提供されます。
3. Hyperwallet アカウントを有効にしたら、Hyperwallet 登録ポータルに記載されているステップに従って登録を完了し、預金口座情報を受け取ります。
4. からアカウントを取得したらHyperwallet、 にサインイン AWS アカウント して にHyperwallet アカウント情報を追加します[AWS Marketplace 管理ポータル](#)。次に [設定] を選択し、[支払い情報] セクションで [銀行情報の入力] を選択します。

Note

Hyperwallet アカウントの情報を更新するには、[Hyperwallet ポータル](#)の Hyperwallet サポートチームにお問い合わせください。時間および連絡先情報については、サポートタブを参照してください。

ステップ 4: 支払い設定を設定する

日別または月別の支払いオプションを選択し、支払いを受け取る日を選択し、銀行口座に通貨を割り当てます。詳細については、「[AWS Marketplace 販売者の支払い設定](#)」を参照してください。

ステップ 5: 顧客確認 (KYC) プロセスを完了する

販売者は、顧客確認 (KYC) プロセスを完了する必要があります。

- 韓国の取引に対して支払いを受ける
- 英国ベースの銀行口座を使用した取引
- EMEA のお客様への販売を簡素化する

このステップはオプションですが、他のリージョンでは推奨されます。KYC には、会社、主要連絡先、受益所有権、および関連ドキュメントに関する追加情報の提供が含まれます。

 Note

返金の処理、KYC 情報の修正、銀行口座の詳細などの財務情報の変更は、検証済みのユーザーのみが行うことができます。検証済みユーザーは、KYC 検証済みユーザーと検証を完了したセカンダリユーザーです。

KYC は、金融機関やオンライン企業が顧客の身元を確認するために使用するコンプライアンス要件です。この要件は、以下が原因です。

- 韓国における指定金融取引情報の報告および使用に関する法律第 5 条 2 項
- Payment Services Directive (PSD 2)
- 金融機関を管理する欧州連合のマネーローンダリング防止指令

AWS Marketplace Amazon Web Services EMEA SARL 経由のトランザクションは、Amazon Payments Europe, S.C.A. (APE) を通じて処理されます。APE は、ルクセンブルクのライセンスされた電子マネー機関であり、支払いサービスを使用するにはお客様の ID の検証が必要です。

 Note

英国の銀行口座では、EUR と GBP でのみ支払いを受けることができます。他の通貨での支払いには、別の管轄区域のアカウントが必要です。また、Amazon Web Services EMEA SARL による販売に限定されます。

KYC プロセスを完了させるには

1. AWS Marketplace 管理ポータルで、設定 を選択します。
2. [アカウントサマリ] セクションで、表示されている [国] が正しいことを確認します。

 Note

[情報] リンクをクリックすると、国を変更する方法が表示されます。

3. [KYC 情報に移動] または [本人確認 (KYC)] タブを選択し、[KYC コンプライアンスを開始] を選択すると、KYC 登録ポータルにリダイレクトされます。

による情報の使用と共有方法の詳細については AWS Marketplace、[Amazon Payments Europe プライバシー通知](#)」を参照してください。

4. [KYC 概要に移動] を選択します。
5. [本人確認 (KYC) の概要] で、必要な情報と書類のリストに目を通し、必要な書類を集めてください (まだ行っていない場合)。次に、[KYC コンプライアンスを続行] を選択します。
6. 指示に従って [基本情報] を入力します。Amazon Payments Europe 利用規約を確認したら、[同意して続行] を選択します。

KYC プロセスの次のページまたは次のステップに進むと、Amazon Payments Europe 利用規約に同意したものとみなされます。

質問がある場合は、コンソールの右側にある [よくある質問 (FAQ)] を参照してください。

7. 指示に従って必要な [ビジネス情報] を入力し、[次へ] を選択します。

 Note

Next を選択すると、次のステップに進むたびに情報が保存されます。

8. 指示に従って必要な [連絡先情報] を入力し、[次へ] を選択します。
9. [実質的所有者] が連絡先と同じかどうかを選択し、必要に応じて実質的所有者 (最大 4 人) を追加します。追加内容を確認して [次へ] を選択します。
10. [法定代理人] が連絡先または実質的所有者と同じかどうかを選択します。法定代理人が別のエンティティである場合は、必要な情報を入力し、エントリを保存してから、[次へ] を選択します。
11. [その他の書類] については、事業許可証、身分証明書、委任状 (該当する場合) をアップロードしてください。

 Note

認可書が必要な場合は、次のサンプルを使用できます。

Letterhead of the company

POWER TO ACT ON BEHALF OF THE COMPANY

The undersigned ****name of Company here**** (herein after, the "Company"), duly represented by

(name, date of birth, and function) **add full name, date of birth, and function of the signatory here**, confirms that **add full name of the Person of Contact here** is authorized to open an Amazon Web Services account with Amazon Payments, accept the User Agreement and other Policies, have access to the Amazon Web Services account, and initiate transactions in the name and on behalf of the Company.

Dated this:

Signed by:

12. [確認と送信] で、入力したすべての情報を確認して確認します。

必要に応じて [編集] を選択して前のセクションに戻ることができます。

13. [検証のため送信] をクリックします。

KYC コンプライアンスの状況がレビューされます (通常は 24 時間以内)。レビューが完了すると、E メールで通知されます。通常、KYC プロセス全体で約 2 週間かかります。

[設定] タブに戻ると、[アカウントサマリ] カードで KYC コンプライアンスの状況を確認できます。KYC ステータスの詳細については、[アカウントサマリ] カードで [本人確認 (KYC)] タブをクリックしてください。レビューが完了するまで [レビュー中] と表示されます。

KYC が確認された後で、APE を通じて支払いを受け取るには、[支払い情報] タブに銀行取引明細を入力する必要があります。

(オプション) 本人確認手順にセカンダリユーザーを追加する

Note

支払い情報を更新するには、ユーザーは多要素認証 (MFA) を有効にする必要があります。MFA の詳細については、[「IAM の多要素認証 \(MFA\)」](#) を参照してください。

セカンダリユーザーとは、KYC 情報の修正、資金や返金の流れの管理、銀行口座情報などの財務情報の変更ができる個人のことです。

KYC 認証を受けたセカンダリユーザーのみが、前述の更新を行うことができます。こうしたセカンダリユーザーは、ルートアカウントの所有者と同じ継続的なスクリーニング管理の対象となります。

セカンダリユーザーが KYC 認証を受けるには、[ステップ 5: 顧客確認 \(KYC\) プロセスを完了する](#) に記載の手順を完了させる必要があります。

本人確認手順にセカンダリユーザーを追加するには

1. ユーザーに AWS Marketplace 管理ポータルにサインインするよう依頼します。
2. [設定] タブに移動します。
3. [本人確認 (KYC)] タブを選択し、[セカンダリユーザー情報] のセクションを確認します。
4. [セカンダリユーザー情報の入力] を選択します。

[セカンダリユーザー] 登録ポータルにリダイレクトされます。

5. [セカンダリユーザー] 登録ポータルで、必須フィールドを入力し、[次へ] を選択します。
6. [確認と送信] ページで、本人確認書類 ([パスポートのアップロード]) と住所証明 ([ドキュメントのアップロード]) のコピーをアップロードします。
7. [検証のため送信] をクリックします。

KYC コンプライアンスの状況がレビューされます (通常は 24 時間以内)。レビューが完了すると、Eメールで通知されます。通常、KYC プロセス全体で約 2 週間かかります。

ステップ 6: 銀行口座検証プロセスを完了する

Amazon Payments Europe (APE) からの支払いを受け取るには、追加情報を入力して、AWS Marketplace 管理ポータルの [支払い情報] タブに記載されている支払い銀行口座を検証する必要があります。

追加の銀行情報の入力

追加の銀行情報を入力するには

1. にサインインし AWS Marketplace 管理ポータル、設定を選択します。
2. [支払い情報] セクションで [銀行情報の更新] を選択します。
3. 適切な支払い口座を選択します。

[検証ステータス] には [未検証] と表示されます。

4. 確認を選択します。

- 銀行口座検証 登録ポータルにリダイレクトされるので、そこで銀行取引明細書をアップロードして送信できます。

Hyperwallet 仮想銀行口座ソリューションを使用している場合は、「[the section called “Hyperwallet から銀行口座確認書をダウンロードします。”](#)」を参照してください。

- ポータルで [銀行書類のアップロード] を選択し、[送信] を選択します。

Hyperwallet から銀行口座確認書をダウンロードします。

[Hyperwallet仮想銀行口座ソリューション](#)を使用する販売者は、以下の手順で Hyperwallet 銀行取引明細書をダウンロードできます。その後、[ステップ 6: 銀行口座検証プロセスを完了する](#) に記載の指示に従って銀行書類をアップロードできます。

Hyperwallet から銀行取引明細書をダウンロードするには

- [Hyperwallet アカウント](#)にサインインします。
- [\[預金口座情報\]](#) ページに移動します。
- [銀行口座確認書] をダウンロードします。

すでに AWS Marketplace 販売者ですか？

販売者として登録したら、以下のトピックでの詳細 AWS Marketplace と次のステップを確認してください。

- 支払い – AWS Marketplace 販売者は、選択した通貨で残高を受け取るように支払い設定を行うことができます。詳細については、「[AWS Marketplace 販売者の支払い設定](#)」を参照してください。
- 出品料金 – AWS Marketplace 販売者の出品料金の詳細については、「」を参照してください [AWS Marketplace 販売者の出品料金について](#)。
- Amazon Payments Europe (APE) AWS Marketplace を使用した の苦情処理ポリシー – Amazon Payments Europe (APE) が提供するサービスに問題がある場合は AWS Marketplace、お知らせください。お客様からのフィードバックは、お客様だけでなく、購入者と販売者全員にとってより良い体験を提供するのに役立ちます。詳細については、「[Amazon Payments Europe \(APE\) に関する AWS Marketplace 販売者の苦情を送信する](#)」を参照してください。
- 追加の販売者ツール – は、顧客ベースをより深く理解し、販売をよりよく理解するために使用できる追加の販売者ツール AWS Marketplace を提供します。詳細については、「[AWS Marketplace 販売者向けのその他のツール](#)」を参照してください。

- 製品の準備 – アカウントを販売者として登録したら、購入者に販売する製品を作成できます AWS Marketplace。詳細については、「[用の製品の準備 AWS Marketplace](#)」を参照してください。

AWS Marketplace 販売者の支払い設定

AWS Marketplace 独立系ソフトウェアベンダー (ISVs) やチャネルパートナーなどの販売者は、未払い残高を受け取るための支払い設定を行うことができます。販売者は、日次または月次の支払いオプションの選択、支払日の選択、銀行口座への通貨の割り当てを行えます。銀行口座には複数の通貨を割り当てることができます。これらのセクションでは、支払い設定を設定する方法を示します。また、AWS Marketplace サブスクリプションの請求と AWS 支払い方法に関する情報も提供します。

Note

米国ベースの ACH アカウントと Hyperwallet アカウントは、米ドルでのみ支払いを受け取ることができます。米ドル以外の支払いについては、SWIFT 銀行口座を指定する必要があります。

トピック

- [支払いの詳細の設定](#)
- [AWS Marketplace サブスクリプションの請求](#)
- [販売者が支払いを受ける方法](#)
- [利用可能な通貨](#)

支払いの詳細の設定

支払いの詳細を設定するには、以下の手順に従います。最近登録した場合は、パブリックプロフィールの作成から支払い方法の追加まで 2 営業日かかる場合があります。

1. にサインインし AWS Marketplace 管理ポータル、設定を選択します。
2. [支払い情報] タブを選択します。
3. [支払い方法] セクションで、[支払い方法の追加] を選択します。または、支払い方法を選択し、[編集] を選択して設定を更新することもできます。
4. [通貨] で、希望する支払い通貨を選択します。次に、[銀行口座] で、選択した通貨で支払いを受け取ることができる口座を選択します。

5. 支払いを日次から月次に変更するには、[毎月] を選択し、支払いを処理する日を 1～28 の範囲から選択します。
6. [支払い方法の追加] を選択します。

Note

販売者は、支払い設定オプションにアクセスするため、きめ細かな IAM アクセス許可を受ける必要があります。きめ細かい IAM アクセス許可を受けるには、「[the section called “AWS Marketplace 販売者のポリシーとアクセス許可”](#)」を参照してください。

AWS Marketplace サブスクリプションの請求

AWS は、ユーザーに代わって請求メカニズムとして機能します。購入者が利用できる最も一般的な支払い方法は、クレジットカードによるものと請求書によるものの 2 つです。

詳細については AWS Marketplace、次の点に注意してください。

- 前払いによる購入は、サブスクリプション購入後すぐに請求されます。
- プライベートオファターの請求スケジュールは、購入者と販売者の間で合意されます。
- 請求書の支払い条件 (請求書の期日を含む) は、購入者と AWS の間で合意されます。条件はベンダーには開示されません。
- フレキシブルな支払いスケジュールを使用したプライベートオファターは、支払いオプションとして請求書に記載する必要があります。
- [請求収益ダッシュボード](#) を使用して請求を検証できます。このダッシュボードは、AWS ユーザーに代わって による請求を要約し、請求書作成日と請求書期日を可視化します。

販売者が支払いを受ける方法

- 支払いには、有効な [支払い方法](#)、[登録された銀行口座](#)、および提出された W9 フォームが必要です。
- 有料商品の出品者は、W-8、付加価値税 (VAT) または物品およびサービス税 (GST) の登録番号、および [適切な対象管轄区域](#) の銀行口座情報を提供する必要があります。
- AWS は、次の方法で支払いを行います。

- 毎日 - 支払いが可能になったときに日単位で支払いが行われます。支払いを受けるには、販売者の残高がプラスである必要があります。
- 毎月 - 販売者は支払いを受ける日 (1 ~ 28 日) を選択します。[回収と支払いダッシュボード](#)には、支払い日が表示されます。
- AWS は、購入者が請求書の支払いを行った後、自動決済機関 (ACH) 移管または SWIFT 移管を使用して支払いを行います。
- 資金は、顧客から回収された後にのみ支払われます。
- 支払いが販売者の銀行に届くまでに支払い日から約 1 ~ 2 営業日かかります。正確なタイミングは、銀行とタイムゾーンによって異なります。
- 支払いダッシュボードは、支払いから AWS Marketplace 管理ポータル 3 ~ 5 日後に更新されます。
- 支払われた資金と未回収の資金に関する詳細は、未払いの売掛金を含め、支払いレポートに記載されています。

Note

購入者がプライベートオファーで合意された通貨をサポートしていない管轄区域に移転する場合は、サポートされている通貨で新しいプライベートオファーを交渉する必要があります。詳細については、「[the section called “プライベートオファー”](#)」を参照してください。

利用可能な通貨

販売者は、契約料金でプライベートオファーを作成し、以下の通貨で支払いを受け取ることができます。

- 米ドル (USD)
- ユーロ (EUR)
- 英国ポンド (GBP)
- オーストラリアドル (AUD)
- 日本円 (JPY)

Note

すべてのパブリックオファーと消費料金によるプライベートオファーは、USD のみ作成できます。

AWS Marketplace 販売者の出品料金について

AWS Marketplace では、AWS Marketplace 販売者に標準およびリージョン別出品料金を提供しています。出品料金は、によって発行された出品料金請求書に 1 行項目として表示されます AWS。

Note

これらの出品料金は、2024 年 1 月 5 日午前 0 時 (UTC) から適用されます。

での標準オファーリストの料金について AWS Marketplace

標準のパブリックオファー、プライベートオファー、チャネルパートナープライベートオファー (CPPOs)、プロフェッショナルサービスの出品料金について説明します。AWS Marketplace では、製品の次の出品料金が提供されています。

公開オファー出品料金

ソフトウェアとデータのパブリックオファーの出品料金は、デプロイ方法によって決まります。

- Software as a service (SaaS) – 3%
- サーバー (Amazon マシンイメージ (AMI)、コンテナ、機械学習) — 20%
- AWS Data Exchange – 3%

プライベートオファー出品料金

プライベートオファーの出品料金は、契約総額と、プライベートオファーが以前のプライベートオファーからの更新、または AWS Marketplace 以外の契約からの更新であるかによって決まります。

- 100 万 USD 未満 — 3%
- 100 万 USD 以上 1,000 万 USD 未満 — 2%

- 1,000 万 USD 以上 — 1.5%
- すべての更新 — 1.5%

チャネルパートナープライベートオファー (CPPO) 出品料金

CPPO 製品は、オファータイプやデプロイ方法にかかわらず、出品料金が 0.5% 上乗せされます。例えば、その製品が SaaS プライベートオファーで、契約総額が 100 万 USD 未満の場合、出品料金は 3.5% になります。

プロフェッショナルサービス出品料金

すべてのプロフェッショナルサービスのプライベートオファーには 2.5% の出品料金がかかります。

でのリージョン別オファーリストの料金について AWS Marketplace

次の管轄区域 (複数可) の購入者との取引の標準出品料金には、追加のリージョン出品料金が適用されます。

Note

リージョン別出品料金は、標準出品料金に追加されます。例えば、製品が SaaS プライベートオファーで、合計契約価格が 100 万米ドル未満の韓国の購入者に販売された場合、出品料金は 4% (3% の標準プライベートオファー出品料金と 1% のリージョン出品料金) になります。

リージョン	追加のリージョン出品料金	有効日
韓国	1%	04/01/2025

Amazon Payments Europe (APE) に関する AWS Marketplace 販売者の苦情を送信する

AWS Marketplace 販売者として、Amazon Payments Europe S.C.A (APE) が提供するサービスに問題を抱えている場合は、苦情を送信できます。Amazon Payments Europe (APE) が提供するサービスに問題がある場合は、お知らせください。お客様からのフィードバックは、お客様だけでなく、

購入者と販売者全員にとってより良い体験を提供するのに役立ちます。次のセクションでは、苦情ポリシーに基づいた Amazon Payments Europe (APE) 関連の苦情の送信に必要な特定の手順について詳述します。このトピックでは、解決時間枠や苦情のエスカレーションなど Amazon Payments Europe (APE) に対する苦情ポリシーについても説明します。

Note

AWS Marketplace にのみ当てはまる苦情に限り、以下の手順で対処します。Amazon Payments Europe S.C.A. が提供するサービスには、支払い取引の処理、手数料に発生する可能性のあるエラーの検証、資金の支払いなどが含まれます。

苦情の申し立て

Amazon Payments Europe S.C.A. と紐づいた AWS Marketplace アカウントをお持ちの場合、苦情は Amazon Payments Europe S.C.A. が処理します。

苦情を申し立てるには

1. [AWS Marketplace](#) 販売者アカウントにサインインします。
2. [お問い合わせ] にアクセスします。
3. [コマーシャルマーケットプレイス]、[販売者アカウント]、[登録] をクリックします。
4. 苦情の詳細を入力し、[送信] をクリックします。

Amazon Payments Europe の苦情解決期間

Amazon Payments Europe S.C.A. (APE) は、苦情を受領した日から 15 営業日以内に回答します。APE の管理が及ばない例外的な状況の場合、苦情の解決に、APE が苦情を最初に受領した日から最大 35 営業日かかる場合があります。

苦情のエスカレーション

対応にご満足いただけない場合は、以下の連絡先に苦情を申し立てることができます。

- Amazon Payments Europe 上級管理職

苦情は、上級管理職 <ape-management@amazon.lu> 宛てにメールを送信してください。お客様のコメントを慎重に検討し、上級管理職が苦情を受領した日から 15 営業日以内に回答しま

す。Amazon Payment Europe の管理が及ばない例外的な状況では、苦情の解決に、上級管理職が最初に苦情を受けた日から最大 35 営業日かかる場合があります。

- Commission de Surveillance du Secteur Financier (CSSF)

ルクセンブルク所在の CSSF は、金融部門の企業を慎重に監督する機関です。110 Route d'Arlon L-2991 ルクセンブルクにある CSSF に連絡するか、<https://www.cssf.lu/contacts/> の [連絡先ページ] をご利用ください。CSSF に関する詳細情報と連絡方法については、CSSF ウェブサイトの「[Customer complaints](#)」を参照してください。

- オンライン紛争解決

EU においてオンラインでアカウントを開設した場合、オンライン紛争解決プラットフォームを使用して CSSF に苦情を申し立てることもできます。このオプションが利用できるのは、Amazon Payments Europe S.C.A. が金融サービスを提供しており、CSSF がそのライセンスを担当する機関だからです。詳細については、欧州委員会ウェブサイトの [Online Dispute Resolution](#) プラットフォームを参照してください。

AWS Marketplace 販売者向けのその他のツール

AWS Marketplace には、製品の販売と管理に使用できる販売ツールが用意されています。これらのツールを使用すると、顧客ベースのインサイトを得られ、自身の販売についてより良く理解できます。このトピックでは、これらのツールと追加のリソースへのリンクについて説明します。

[AWS Marketplace 管理ポータル](#) は、で製品を販売するための主要なツールです AWS Marketplace。 [AWS Marketplace 管理ポータル](#) で利用できる販売促進活動を有効活用することで、製品を管理し、チャネル収益を増加させることができます。詳細については、「[AWS Marketplace 管理ポータル](#)」を参照してください。

AWS Marketplace には、次の追加の販売者ツールが用意されています。

- のその他のリソース AWS Marketplace 管理ポータル – を開いて [AWS Marketplace 管理ポータル](#) サインインすると、ホームページの Marketplace リソースセクションに追加のリソースへのリンクが表示されます。例えば、発売前 90 日間および発売後 90 日間に製品のマーケティングのサポートを受けるには、Marketplace リソースの AWS Marketplace 管理ポータル ホームページから [リンク180 日間の GTM アカデミー](#)されているを確認できます。
- AWS Marketplace Commerce Analytics Service – AWS Marketplace Commerce Analytics Service を使用すると、を通じて製品と顧客データにプログラムでアクセスできます AWS Marketplace。 サービスに登録すると、AWS SDK から使用状況、サブスクリプション、請求レポートにアクセ

できます。詳細については、「[AWS Marketplace Commerce Analytics Service を使用して、製品および顧客データにアクセスする](#)」を参照してください。

- AWS Marketplace フィールドデモンストレーションプログラム (FDP) – FDP を使用すると、AWS フィールドチーム (内部的に承認された AWS 従業員) は、を通じて一部の製品やソリューションを無償 AWS Marketplace で使用できます。詳細については、「[AWS Marketplace フィールドデモンストレーションプログラム](#)」を参照してください。
- 販売者レポート、データフィード、ダッシュボード – AWS Marketplace は、製品販売に関する情報を収集および分析するためのツールを提供します。詳細については、「[の販売者レポート、データフィード、ダッシュボード AWS Marketplace](#)」を参照してください。

詳細については、以下のトピックを参照してください。

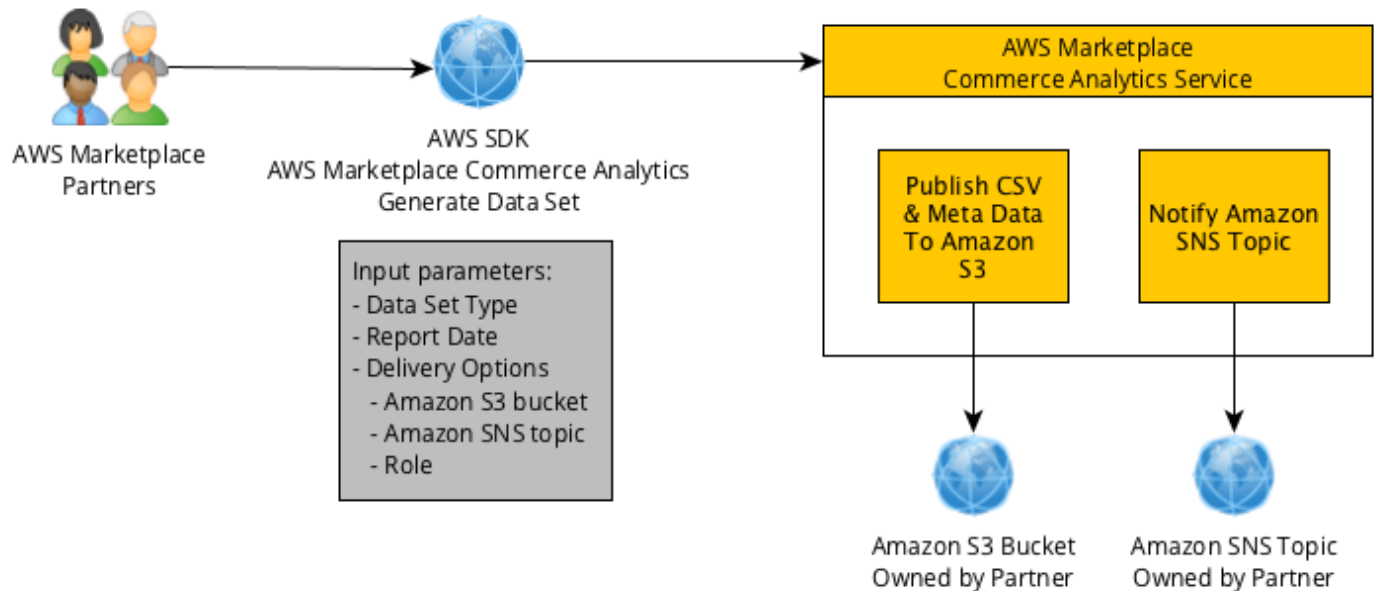
トピック

- [AWS Marketplace Commerce Analytics Service を使用して、製品および顧客データにアクセスする](#)
- [AWS Marketplace フィールドデモンストレーションプログラム](#)
- [AWS パートナーアシスタント](#)

AWS Marketplace Commerce Analytics Service を使用して、製品および顧客データにアクセスする

AWS Marketplace Commerce Analytics Service を使用すると、を通じて製品と顧客データにプログラムでアクセスできます AWS Marketplace。サービスに登録すると、AWS SDKs を通じて使用状況、サブスクリプション、請求レポートにアクセスできます。SDK ツールを使用してリクエストしたデータは、データセットとして AWS アカウント に配信されます。ほとんどのデータセットは、[AWS Marketplace 管理ポータル](#) で利用可能なテキストベースのレポートと同じデータに対応します。特定の日付のデータセットをリクエストできます。データは指定先の Amazon S3 バケットに配信されます。データ配信の通知は、Amazon Simple Notification Service (Amazon SNS) によって提供されます。このトピックでは、Commerce Analytics Service AWS Marketplace を使用するための利用規約について説明します。

次の図は、Commerce Analytics Service が AWS Marketplace の製品および顧客にアクセスして、データセットとして Amazon S3 バケットに配信し、Amazon SNS 経由で通知を送信している状況を図解しています。



利用規約

これらの AWS Marketplace Commerce Analytics Service 利用規約 (これらの「CAS 条件」) AWS Marketplace には、Commerce Analytics Service (「CA Service」) の使用とアクセスに固有の条件が含まれており、これらの CAS 条件に表示された「I Accept」ボタンまたはチェックボックスをクリックした日付、またはそれ以前であれば CA サービスサービスの使用時に有効になります。これらの CAS 条件は、お客様と Amazon Web Services, Inc. の間の AWS Marketplace 販売者の利用規約 (AWS Marketplace 「販売者条件」) の補足です。 (「AWS」、「we」、「us」、または「our」)。これらの条件はここに組み込まれます。これらの CAS 規約と AWS Marketplace 販売者規約の間に競合が発生した場合は、当該の競合の範囲内で、およびお客様の CA サービスの使用に関してのみ、これらの CAS 規約が適用されます。ここで使用されているが、ここで定義されていない大文字の用語は、AWS Marketplace 販売者規約に規定された意味を持つものとします。

1. CA サービスおよび CAS データ。CA サービスへのアクセス資格を得るには、既存の AWS Marketplace 販売者規約に拘束されている AWS Marketplace 販売者である必要があります。CA サービス (「CAS データ」) に関連して受信またはアクセスできる情報およびデータは、サブスクリャー情報であり、AWS Marketplace 販売者規約に規定された制限および義務の対象となります。お客様は、(a) CAS データを第三者に開示しないことを条件として、お客様の AWS Marketplace コンテンツに関連するマーケティングやその他のプロモーション活動を改善およびターゲットにするために、CAS データを機密ベースで 사용할 수 있습니다。 (b) 適用されるプライバシーポリシーまたは法律と何らかの点で矛盾して CAS データを使用する。 (c) サブスクリャーに連絡して、の外部で代替購入を行うよう影響を与える AWS Marketplace。 (d) 誹謗中傷する 当社の関連会社 またはそのいずれかの製品、または当社のそれぞれの製品。 または (e) 目

的の受信者が AWS Marketplace サブスクライバーであることに基づいて、あらゆる種類の通信をターゲットにします。

2. CA サービスの制限とセキュリティ。CA サービスドキュメントに記載されている方法によってのみ、CA サービスにアクセスします (またはアクセスを試みます)。CA サービスを使用中に自身のアイデンティティまたは顧客のアイデンティティを偽ったり、隠したりしてはなりません。いずれかの期間において CA サービスへのアクセスを許可された接続、通話、またはサーバーの数に関して、これを含みますが限定されず、CA サービスに使用を設定または制限する権利を単独裁量にて保有します。当該の制限に同意し、迂回することを回避しようとしません。これらの CAS 規約に違反している、または CA サービスを悪用していると確信した場合、CA サービスにアクセスする権利を制限、停止または終了する権利を保有します。
3. CA サービス認証情報の機密性とセキュリティ。CA サービス認証情報 (パスワード、キー、およびクライアント ID など) は、お客様が API クライアントを識別するために使用されることが目的です。認証情報を秘匿する全責任はお客様が負い、少なくとも同様の性質の機密情報を保護するために取る措置を含め、当該認証情報の開示、流布、または不正使用を避けるための合理的な措置を取ります。CA サービス認証情報をオープンソースプロジェクトに埋め込むことはできません。お客様の認証情報での CA サービスへのあらゆるアクセスの全責任はお客様が負います。
4. 変更。当社は、AWS サイトに改訂されたバージョンを投稿するか、AWS Marketplace 販売者規約に従ってお客様に通知することで、これらの CAS 規約をいつでも変更することができます。変更された規約は、掲載時に、または、E メールによって通知した場合、E メールメッセージに記載されているとおりに、発行します。これらの CAS 規約への変更の発行日後に CA サービスの使用またはアクセスを継続することで、変更された規約に縛られることに同意したものと見なされます。
5. ターミネーション。これらの CAS 規約およびここに付与される CAS データを使用する権限は、あらゆる理由による AWS Marketplace 販売者規約の終了により、通知の有無に関わらず、終了します。さらに、あらゆるときにあらゆる理由で CA サービスの提供を停止するか、CA サービスへのアクセスを終了する場合があります。

入門

オンボーディング、技術実装、トラブルシューティング情報など、AWS Marketplace Commerce Analytics Service の詳細については、以下のトピックを参照してください。

トピック

- [Commerce Analytics Service AWS Marketplace へのオンボーディング](#)
- [AWS CLI および での AWS Marketplace Commerce Analytics Service の使用 AWS SDK for Java](#)

- [Commerce Analytics Service AWS Marketplace を使用したデータセットの生成](#)
- [Commerce Analytics Service AWS Marketplace のトラブルシューティング](#)

Commerce Analytics Service AWS Marketplace へのオンボーディング

AWS Marketplace Commerce Analytics Service を使用すると、を通じて製品と顧客データにプログラムでアクセスできます AWS Marketplace。Commerce Analytics Service AWS Marketplace の使用を開始するには、Commerce Analytics Service を使用する AWS のサービス ように AWS アカウントと AWS Marketplace を設定する必要があります。これらのセクションでは、Commerce Analytics Service を使用する AWS のサービス ように AWS アカウントと AWS Marketplace を設定する方法について説明します。

Commerce Analytics Service AWS Marketplace を使用するには

- [ステップ 1: アクセス許可 AWS アカウント を使用して をセットアップする](#)
- [ステップ 2: 送信先の Amazon S3 バケットを作成する](#)
- [ステップ 3: レスポンス通知用の Amazon SNS トピックを設定する](#)
- [ステップ 4: Commerce Analytics Service プログラムに登録する](#)
- [ステップ 5: 設定を確認する](#)

ステップ 1: アクセス許可 AWS アカウント を使用して をセットアップする

では、ルートアカウントの認証情報を使用するの AWS Marketplace 管理ポータル ではなく、AWS Identity and Access Management (IAM) ロールを使用して にサインインすることをAWS Marketplace 強くお勧めします。AWS Marketplace Commerce Analytics Service のアクセス許可の、特定の IAM のアクセス許可については、「[the section called “AWS Marketplace 販売者のポリシーとアクセス許可”](#)」を参照してください。お客様のアカウントにアクセスする人に対して個別ユーザーを作成することにより、ユーザーそれぞれに一意の認証情報を設定することができます。さらに、各ユーザーにそれぞれ異なるアクセス権限を付与することもできます。必要な場合には、いつでもユーザーのアクセス許可を変更、または無効にすることができます。

ステップ 2: 送信先の Amazon S3 バケットを作成する

Commerce Analytics Service により、リクエストしたデータが、指定した Amazon S3 バケットに配信されます。既に Amazon S3 バケットをお持ちで利用できる場合は、次のステップに進みます。

Amazon S3 バケットがない場合、またはこのデータ専用の Amazon S3 バケットを作成する場合は、[Amazon S3バケットの作成方法](#)」を参照してください。

ステップ 3: レスポンス通知用の Amazon SNS トピックを設定する

Commerce Analytics Service は、Amazon SNS を使用してレスポンス通知を配信します。このサービスでは、データセットが利用可能になったり、エラーが発生したりしたときに、このトピックにメッセージを発行して通知します。既にこの目的で Amazon SNS トピックをお持ちの場合は、次のステップに進みます。

このサービスに対して Amazon SNS トピックを設定していない場合は、この時点で設定します。手順については、「[トピックの作成](#)」を参照してください。

作成したトピックの Amazon リソースネーム (ARN) をメモしておきます。サービスを呼び出すときに ARN が必要になります。

ステップ 4: Commerce Analytics Service プログラムに登録する

トピックの ARN およびバケットの名前を使用してサービスを設定すると、Commerce Analytics Service は Amazon S3 バケットと Amazon SNS トピックにアクセスします。

そのアクセスを有効にするには

1. AWS Marketplace 製品の管理 AWS アカウント に使用する [AWS Marketplace 管理ポータル](#) を使用して にログインします。
2. AWS Marketplace Commerce Analytics Service に登録するために [必要な IAM 権限](#) があることを確認してください。
3. [Commerce Analytics Service 登録ページ](#) に移動します。
4. Amazon S3 バケット名および Amazon SNS トピック ARN を入力し、[登録] を選択します。
5. [permissions (権限)] ページの [Allow (許可)] を選択します。
6. で AWS Marketplace 管理ポータル、成功メッセージにロール名 ARN を記録します。サービスを呼び出すには ARN が必要です。

Note

Commerce Analytics Service へのオンボーディングにより、に IAM ロールが作成されます AWS アカウント。IAM ロールにより、AWS Marketplace は Amazon S3 バケットに書き込み、Amazon SNS トピックに通知を発行できます。AWS Marketplace はアカウント 452565589796 を使用して、この IAM ロールでこれらの関連アクションを実行します。

ステップ 5: 設定を確認する

最後のステップは、設定が正常に動作しているかどうかを確認することです。

設定をテストするには

1. [AWS コマンドラインインターフェイス](#) () をダウンロード、インストール、設定します AWS CLI。
2. を使用して AWS CLI、このコマンドを実行します。

```
aws marketplacecommerceanalytics generate-data-set \
--data-set-type "customer_subscriber_hourly_monthly_subscriptions" \
--data-set-publication-date "{TODAY'S-DATE}" \
--role-name-arn "{YOUR-ROLE-NAME-ARN}" \
--destination-s3-bucket-name "{amzn-s3-demo-bucket}" \
--destination-s3-prefix "TEST_PREFIX" \
--sns-topic-arn "{YOUR-SNS-TOPIC-ARN}"
```

- --data-set-publication-date で、{TODAY'S DATE} を ISO-8601 形式を使用して現在の日付 YYYY-MM-DDT00:00:00Z に置き換えます。YYYY は 4 桁の年、MM は 2 桁の月、DD は 2 桁の日を表します。
- --role-name-arn で、{YOUR-ROLE-NAME-ARN} を [ステップ 4: Commerce Analytics Service プログラムに登録する](#) の登録プロセスで取得したロールの ARN に置き換えます。
- --destination-s3-bucket-name では、{amzn-s3-demo-bucket} を [ステップ 2: 送信先の Amazon S3 バケットを作成する](#) で作成した Amazon S3 バケットの名前で置き換えます。
- --sns-topic-arn では、{YOUR-SNS-TOPIC-ARN} を、[ステップ 3: レスポンス通知用の Amazon SNS トピックを設定する](#) で作成した Amazon SNS トピックに置き換えます。

dataSetRequestId レスポンスが含まれているレスポンスをサービスから受信した場合は、オンボーディングプロセスが正常に完了しています。正常な応答は次のようになります。

```
{
  "dataSetRequestId": "646dd4ed-6806-11e5-a6d8-fd5dbcaa74ab"
}
```


AWS CLI および での AWS Marketplace Commerce Analytics Service の使用 AWS SDK for Java

AWS Marketplace Commerce Analytics Service を使用すると、 を通じて製品と顧客データにプログラムでアクセスできます AWS Marketplace。Commerce Analytics Service AWS Marketplace は [AWS SDK](#) を通じて提供されます。[AWS CLI](#) および [AWS SDK for Java](#) を使用して Commerce Analytics Service を操作します。これらのセクションでは、AWS CLI および SDK for Java を使用して Commerce Analytics Service を実装する方法について説明します。

トピック

- [Commerce Analytics Service の IAM ポリシー](#)
- [を使用したリクエストの実行 AWS CLI](#)
- [AWS SDK for Javaによるリクエストの作成](#)

Commerce Analytics Service の IAM ポリシー

ユーザーが Commerce Analytics Service を使用できるようにするには、以下の権限が必要です。

Commerce Analytics Service に登録するには、次の AWS Marketplace IAM アクセス許可ポリシーを使用します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles",
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "aws-marketplace-management:viewReports"
      ],
      "Resource": "*"
    }
  ]
}
```

以下の IAM アクセス許可ポリシーを使用して、ユーザーが AWS Marketplace Commerce Analytics Service にリクエストを送信できるようにします。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "marketplacecommerceanalytics:GenerateDataSet",
      "Resource": "*"
    }
  ]
}
```

詳細については、「IAM ユーザーガイド」の「[IAM コンソールでのポリシーの作成](#)」を参照してください。

を使用したリクエストの実行 AWS CLI

開始するには、[AWS CLI](#)をダウンロードします。次の AWS CLI 例では、2017 年 10 月 1 日の時間単位/月単位のサブスクリプションデータセットをリクエストします。このデータセットはプレフィックス demo-prefix を使用して Amazon S3 バケット demo-bucket に発行され、通知メッセージは Amazon SNS トピック demo-topic に配信されます。

```
aws marketplacecommerceanalytics generate-data-set \
--data-set-type "customer_subscriber_hourly_monthly_subscriptions" \
--data-set-publication-date "2017-10-01T00:00:00Z" \
--role-name-arn "arn:aws:iam::123412341234:role/MarketplaceCommerceAnalyticsRole" \
--destination-s3-bucket-name "demo-bucket" \
--destination-s3-prefix "demo-prefix" \
--sns-topic-arn "arn:aws:sns:us-west-2:123412341234:demo-topic"
```

このリクエストは、リクエストごとに一意の識別子を返します。この識別子を使用して、リクエストを Amazon SNS トピックに発行された通知に関連付けることができます。以下は、この識別子の例です。


```
{
  "dataSetRequestId": "646dd4ed-6806-11e5-a6d8-fd5dbcaa74ab"
}
```

AWS SDK for Javaによるリクエストの作成

開始するには、[AWS Java SDK](#) をダウンロードします。次の AWS SDK for Java 例では、2015 年 10 月 1 日の時間単位/月単位のサブスクリプションデータセットをリクエストします。このデータセットはプレフィックス demo-prefix を使用して Amazon S3 バケット demo-bucket に発行され、通知メッセージは Amazon SNS トピック demo-topic に配信されます。

```
/*
 * Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 * http://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import com.amazonaws.AmazonClientException;
import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import
    com.amazonaws.services.marketplacecommerceanalytics.AWSMarketplaceCommerceAnalyticsClient;
```

```

import
    com.amazonaws.services.marketplacecommerceanalytics.model.GenerateDataSetRequest;
import com.amazonaws.services.marketplacecommerceanalytics.model.GenerateDataSetResult;
/**
 * This sample demonstrates how to make basic requests to the AWS Marketplace Commerce
 * Analytics service using the AWS SDK for Java.
 * <p>
 * <b>Prerequisites:</b> Follow the on-boarding guide: {URL OR SOMETHING}
 * <p>
 * Fill in your AWS access credentials in the provided credentials file
 * template, and be sure to move the file to the default location
 * (~/.aws/credentials) where the sample code will load the credentials from.
 * <p>
 * <b>WARNING:</b> To avoid accidental leakage of your credentials, DO NOT keep
 * the credentials file in your source directory.
 * <p>
 * http://aws.amazon.com/security-credentials
 */
public class MarketplaceCommerceAnalyticsSample {
    public static void main(String[] args) throws ParseException {
        /*
         * The ProfileCredentialsProvider will return your [default]
         * credential profile by reading from the credentials file located at
         * (~/.aws/credentials).
         */
        AWSCredentials credentials = null;
        try {
            credentials = new ProfileCredentialsProvider().getCredentials();
        } catch (Exception e) {
            throw new AmazonClientException("Cannot load the credentials from the credential
                profiles "
                + "file. Make sure that your credentials file is at the correct "
                + "location (~/.aws/credentials), and is in valid
                format.", e);
        }
        AWSMarketplaceCommerceAnalyticsClient client = new
            AWSMarketplaceCommerceAnalyticsClient(credentials);
        Region usEast1 = Region.getRegion(Regions.US_EAST_1);
        client.setRegion(usEast1);
        System.out.println("=====");
        System.out.println("Getting Started with AWS Marketplace Commerce Analytics Service");
        System.out.println("=====
        \n");
        // Create a data set request with the desired parameters

```

```
GenerateDataSetRequest request = new GenerateDataSetRequest();
request.setDataSetType("customer_subscriber_hourly_monthly_subscriptions");
request.setDataSetPublicationDate(convertIso8601StringToDateUtc("2014-06-09T00:00:00Z"));
request.setRoleNameArn("arn:aws:iam::864545609859:role/
MarketplaceCommerceAnalyticsRole");
request.setDestinationS3BucketName("awsmp-goldmine-seller");
request.setDestinationS3Prefix("java-sdk-test");
request.setSnsTopicArn("arn:aws:sns:us-west-2:864545609859:awsmp-goldmine-seller-
topic");
System.out.println(
String.format("Creating a request for data set %s for publication date %s.",
request.getDataSetType(), request.getDataSetPublicationDate()));
try {
// Make the request to the service
GenerateDataSetResult result = client.generateDataSet(request);
// The Data Set Request ID is a unique identifier that you can use to correlate the
// request with responses on your Amazon SNS topic
System.out.println("Request successful, unique ID: " + result.getDataSetRequestId());
} catch (AmazonServiceException ase) {
System.out.println("Caught an AmazonServiceException, which means your request made it
"
+ "to the AWS Marketplace Commerce Analytics service, but was rejected with an "
+ "error response for some reason.");
System.out.println("Error Message: " + ase.getMessage());
System.out.println("HTTP Status Code: " + ase.getStatusCode());
System.out.println("AWS Error Code: " + ase.getErrorCode());
System.out.println("Error Type: " + ase.getErrorType());
System.out.println("Request ID: " + ase.getRequestId());
} catch (AmazonClientException ace) {
System.out.println("Caught an AmazonClientException, which means the client encountered
"
+ "a serious internal problem while trying to communicate with the AWS Marketplace"
+ "Commerce Analytics service, such as not being able to access the "
+ "network.");
System.out.println("Error Message: " + ace.getMessage());
}
}

private static Date convertIso8601StringToDateUtc(String dateIso8601) throws
ParseException {
    TimeZone utcTimeZone = TimeZone.getTimeZone("UTC");
    DateFormat utcDateFormat = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ssX");
    utcDateFormat.setTimeZone(utcTimeZone);
    return utcDateFormat.parse(dateIso8601);
}
```

```
}
```

正常な結果はこの例に似たものとなります。

```
=====
Getting Started with AWS Marketplace Commerce Analytics Service
=====
Creating a request for data set customer_subscriber_hourly_monthly_subscriptions for
publication
date Sun Jun 08 17:00:00 PDT 2014.
Request successful, unique ID: c59aff81-6875-11e5-a6d8-fd5dbcaa74ab
```

Commerce Analytics Service AWS Marketplace を使用したデータセットの生成

Commerce Analytics Service AWS Marketplace に登録すると、AWS CLI および を通じて使用状況、サブスクリプション、請求レポートにアクセスできます AWS SDK for Java。SDK ツールを使用してリクエストしたデータは、データセット AWS アカウント として に配信されます。ほとんどのデータセットは、[AWS Marketplace 管理ポータル](#) で利用可能なテキストベースのレポートと同じデータに対応します。詳細については、「[AWS Marketplace Commerce Analytics Service を使用して、製品および顧客データにアクセスする](#)」を参照してください。これらのセクションでは、Commerce Analytics Service によって生成されたデータセットに関連するパラメータ、応答、出力について説明します。

トピック

- [GenerateDataSet 個のパラメータ](#)
- [レスポンス](#)
- [出力](#)

GenerateDataSet 個のパラメータ

AWS Marketplace Commerce Analytics Service は、Amazon S3 バケットへのデータセットの発行をリクエストするためにGenerateDataSet使用できる 1 つのメソッド を公開します。次の表にGenerateDataSet のパラメータの一覧を示します。

データセットのパラメータ

フィールド	説明
Data Set Type	このデータセットはリクエストの結果として返されます。
Data Set Publication Date	データセットの発行日。 日別データセットでは、該当日の日付が日単位の粒度で提供されます。 月別データセットでは、該当月の日付が月単位の粒度で提供されます。日付の値は無視されます。
Role Name ARN	ロールの ARN の後に、リソースへのアクセス権を持つサービスを提供するアクセス権限ポリシーが続いています。
Destination Amazon S3 Bucket Name	送信先 Amazon S3 バケットの名前 (ARN ではなくフレンドリ名)。データセットはこの場所に発行されます。
Destination Amazon S3 Prefix	(オプション) データセットの発行先である Amazon S3 のプレフィックス。標準ファイルシステムのディレクトリパスに似ています。 たとえば、バケットの名前が mybucket であり、プレフィックスが myprefix/mydatasets の場合、出力ファイルは s3://amzn-s3-demo-bucket/myprefix/mydatasets/outputfile に発行されます。 プレフィックスのディレクトリ構造が存在しない場合は作成されます。

フィールド	説明
	プレフィックスを指定しないと、データセットの発行先は Amazon S3 バケットのルートになります。
SNS トピックの ARN	データセットの発行時またはエラーの発生時に通知先となる Amazon SNS トピックの ARN。

レスポンス

Commerce Analytics Service は 2 AWS Marketplace でのレスポンスを返します。1 つ目は同期で即座に返され、2 つ目は非同期でこれは Amazon SNS を使用して返されます。同期のレスポンスは次の例のようになります。

データセットパラメータ

フィールド	説明
Data Set Request ID	サービスに対する特定のリクエストを表す一意の識別子。この識別子を使用して、リクエストを Amazon SNS トピックへの通知と関連付けることができます。

非同期レスポンスが JSON 形式のドキュメントとして Amazon SNS トピックに書き込まれ、次の例のようになります。

データセットのパラメータ

フィールド	説明
Data Set S3 Location	配信されたデータセットのバケット名とキー。
Data Set Meta Data S3 Location	配信されたデータセットのメタデータファイルのバケット名とキー。
Data Set Request ID	サービスに対する特定のリクエストを表す一意の識別子。この識別子を使用して、リクエスト

フィールド	説明
	を Amazon SNS トピックへの通知と関連付けることができます。
Success (成功)	オペレーションが成功した場合は「true」、それ以外の場合は「false」。
メッセージ	(オプション) エラーが発生した場合 (「Success」が「false」の場合など)、このメッセージは失敗に関する情報を示します。

JSON 形式の非同期レスポンスの例

```
{
  "dataSetS3Location":{
    "bucketName":"demo-bucket",
    "key":"demo-prefix/
customer_subscriber_hourly_monthly_subscriptions_2014-06-09.csv"
  },
  "dataSetMetaDataS3Location":{
    "bucketName":"demo-bucket",
    "key":"demo-prefix/
customer_subscriber_hourly_monthly_subscriptions_2014-06-09.meta.json"
  },
  "dataSetRequestId":"f65b7244-6862-11e5-80e2-c5127e17c023",
  "success":true
}
```

出力

リクエストが成功すると、リクエストしたデータセットは .csv ファイルとして Amazon S3 バケットに配信されます。JSON 形式のメタデータファイルは、データセットファイルと同じ場所に発行されます。メタデータファイルは、データセットや元のリクエストのパラメータに関する有用な情報を提供します。メタデータファイル名は、データセットファイル名と同じですが、拡張子が .meta.json となります。次の表は、.csv ファイルのメタデータフィールドの一覧です。

メタデータフィールド

フィールド	説明
Data Set Request ID	サービスに対する特定のリクエストを表す一意の識別子。この識別子を使用して、リクエストを Amazon SNS トピックへの通知と関連付けることができます。
Data Set Coverage Range	データ範囲の開始日時と終了日時を定義します。これらの日時は ISO 8601 形式です。
Data Set Request Parameters	GenerateDataSet メソッドへの元のリクエストパラメータ。
Data Set S3 Location	配信されたデータセットのバケット名とキー。
Data Set Meta Data S3 Location	配信されたデータセットのメタデータファイルのバケット名とキー。

JSON 形式のメタデータコンテンツの例を次に示します。

```
{
  "dataSetRequestId": "43d7137b-8a94-4042-a09d-c41e87f371c1",
  "dataSetCoverageRange": {
    "startDateTime": "2014-06-08T00:00:00.000Z",
    "endDateTime": "2014-06-08T23:59:59.000Z"
  },
  "dataSetRequestParameters": {
    "sellerAccountId": "123412341234",
    "dataSetType": "customer_subscriber_hourly_monthly_subscriptions",
    "dataSetPublicationDate": "2014-06-09T00:00:00.000Z",
    "roleNameArn": "arn:aws:iam::123412341234:role/MarketplaceCommerceAnalyticsRole",
    "destinationS3BucketName": "demo-bucket",
    "destinationS3Prefix": "demo_prefix/customer_subscriber_hourly_monthly_subscriptions",
    "snsTopicArn": "arn:aws:sns:us-west-2:123412341234:demo-topic"
  },
  "dataSetS3Location": {
    "bucketName": "demo-bucket",
    "key": "demo_prefix/customer_subscriber_hourly_monthly_subscriptions_2014-06-09.csv"
  }
}
```



```
},  
"dataSetMetaDataS3Location": {  
  "bucketName": "demo-bucket",  
  "key": "demo_prefix/  
customer_subscriber_hourly_monthly_subscriptions_2014-06-09.meta.json"  
}  
}
```

利用可能なデータセットの完全なリストについては、[AWS SDK ドキュメント](#)を参照してください。

Commerce Analytics Service AWS Marketplace のトラブルシューティング

Commerce Analytics Service AWS Marketplace の問題をトラブルシューティングできます。このサービスでは、製品と顧客のデータをプログラムで提供します AWS Marketplace。エラーやその他の設定の問題が発生した場合は、Commerce Analytics Service のトラブルシューティングが必要になる場合があります。以下のセクションでは、Commerce Analytics Service でよくある問題を診断して解決する手順を説明するトラブルシューティングプロセスについて説明します。

許可リストの問題のため、サービスにアクセスできません。

で販売者としてまだ登録していない場合は AWS Marketplace、[AWS Marketplace 管理ポータル](#) にアクセスして登録します。で販売者として登録済みの場合は AWS Marketplace、[AWS Marketplace 販売者オペレーションチーム](#)にお問い合わせください。

過去の日付のデータセットをリクエストできません。ただし、SDK ドキュメントでは、この日付のデータが利用可能となっています。

過去の特定期間の日付に関するデータセットが利用可能であると記載されていても、利用できるのは AWS Marketplaceに参加した時点以降のデータのみです。エラーが発生したと思われる場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

このサービスを呼び出すと、「Could not connect to the endpoint URL: https://marketplacecommerceanalytics.eu-central-1.amazonaws.com/」というエラーメッセージが表示されます。

Commerce Analytics Service AWS Marketplace は、米国東部 (バージニア北部) リージョンでのみ使用できます。Commerce Analytics Service へのすべての呼び出しは us-east-1 エンドポイントに対して行う必要があります。

を使用している場合は AWS CLI、次の例に示すように、各呼び出しに `--region` フラグを追加し `us-east-1`、を AWS リージョン として指定します。

```
aws marketplacecommerceanalytics generate-data-set \  
--data-set-type "customer_subscriber_hourly_monthly_subscriptions" \  
--data-set-publication-date "2016-04-21T00:00:00Z" \  
--role-name-arn "arn:aws:iam::138136086619:role/MarketplaceCommerceAnalyticsRole" \  
--destination-s3-bucket-name "marketplace-analytics-service" \  
--destination-s3-prefix "test-prefix" \  
--sns-topic-arn "arn:aws:sns:eu-  
central-1:138136086619:Marketplace_Analytics_Service_Notice" \  
--region us-east-1
```

オンボーディングプロセスで選択したのとは別の Amazon S3 バケットまたは別の Amazon SNS トピックを使用するにはどうすればいいですか。

Commerce Analytics Service AWS Marketplace に登録するときに、Amazon S3 バケットと Amazon SNS トピックを指定しました。オンボーディングプロセスでは、サービスがこれらの特定のリソースにのみアクセスできるように IAM 権限を設定します。別のリソースを使用するには、IAM ポリシーを変更する必要があります。

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/iam/> で IAM コンソールを開きます。
2. IAM コンソールの左側にある [ロール] を選択します。
3. [MarketplaceCommerceAnalyticsRole] を選択します。
4. [Inline Roles (インラインロール)] セクションが展開されていない場合は展開します。
5. oneClick_MarketplaceCommerceAnalyticsRole で始まる名前のポリシーを探し、[Edit Policy] を選択します。
6. そのポリシードキュメントで、変更するサービスに関連するアクションが指定されているセクションを見つけます。例えば、Amazon S3 バケットを変更するには、s3: で始まるアクションが含まれているセクションを見つけ、各リソースの選択肢を新しい Amazon S3 バケットに変更します。

IAM ポリシーの詳細については、「https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html」を参照してください。

GenerateDataSet アクションを呼び出すと、**AccessDeniedException** エラーが表示される

これは、GenerateDataSet を呼び出すために必要なアクセス許可がユーザーにない場合に発生することがあります。以下の手順は、IAM コンソールを使用してこれらの権限を持つ IAM ポリシーを作成し、ユーザー、グループ、またはロールにアクセス許可を追加するために必要な手順の概要を示しています。

JSON ポリシーエディタでポリシーを作成するには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/iam/> で IAM コンソールを開きます。
2. 左側のナビゲーションペインで、[ポリシー] を選択します。

初めて [ポリシー] を選択する場合には、[管理ポリシーによるこそ] ページが表示されます。[今すぐ始める] を選択します。

3. ページの上部で、[ポリシーを作成] を選択します。
4. [ポリシーエディタ] セクションで、[JSON] オプションを選択します。
5. 次の JSON ポリシードキュメントを入力します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "marketplacecommerceanalytics:GenerateDataSet",
      "Resource": "*"
    }
  ]
}
```

6. [次へ] をクリックします。

Note

いつでも [Visual] と [JSON] エディタオプションを切り替えることができます。ただし、[Visual] エディタで [次へ] に変更または選択した場合、IAM はポリシーを再構成して visual エディタに合わせて最適化することがあります。詳細については、「IAM ユーザーガイド」の「[ポリシーの再構成](#)」を参照してください。

7. [確認と作成] ページで、作成するポリシーの [ポリシー名] と [説明] (オプション) を入力します。[このポリシーで定義されているアクセス許可] を確認して、ポリシーによって付与されたアクセス許可を確認します。
8. [ポリシーの作成] をクリックして、新しいポリシーを保存します。

アクセス権限を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM Identity Center :

アクセス許可セットを作成します。「AWS IAM Identity Center ユーザーガイド」の「[権限設定を作成する](#)」の手順に従ってください。

- IAM 内で、ID プロバイダーによって管理されているユーザー:

ID フェデレーションのロールを作成します。詳細については「IAM ユーザーガイド」の「[サードパーティー ID プロバイダー \(フェデレーション\) 用のロールを作成する](#)」を参照してください。

- IAM ユーザー:

- ユーザーが担当できるロールを作成します。手順については「IAM ユーザーガイド」の「[IAM ユーザーのロールの作成](#)」を参照してください。
- (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。詳細については「IAM ユーザーガイド」の「[ユーザー \(コンソール\) へのアクセス権限の追加](#)」を参照してください。

ここに記載されていない問題。

[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

AWS Marketplace フィールドデモンストレーションプログラム

AWS Marketplace フィールドデモンストレーションプログラム (FDP) を使用すると、AWS フィールドチーム (内部的に承認された AWS 従業員) は、を通じて一部の製品やソリューション AWS Marketplace を無料で使用できます。

承認された AWS 従業員の例には、ソリューションアーキテクト、営業およびマーケティングの専門家などがあります。FDP により、これらの従業員は、教育のための製品機能と、顧客のワークロードに組み込む可能性を実証することができます。

次の製品タイプがサポートされています。

- [アマゾン マシンイメージ \(AMI\)](#)
- [コンテナ](#)
- [機械学習アルゴリズムとモデルパッケージ \(SageMaker AI\)](#)
- [データセット \(AWS Data Exchange\)](#)

Note

AWS Data Exchange 製品の場合、FDP は公開オファーが 0 USD (無料) の製品にのみ適用されます。

サブスクリプション検証が有効になっている AWS Data Exchange 製品の場合、プロバイダーはサブスクリプションリクエストを承認する必要があります。サブスクリプション検証の詳細については、「AWS Data Exchange ユーザーガイド」の「[サブスクライバーのサブスクリプション検証](#)」を参照してください。

AWS Marketplace 販売者としてサインアップすると、FDP プログラムに自動的に登録されます。オプトアウトするには、[AWS Marketplace Seller Operations](#) チームにサポートリクエストを送信します。

このプログラムでの製品使用状況に関する情報は、[月別請求済み収益レポート](#)の「[AWS フィールド デモ使用量](#)」セクションを参照してください。

AWS パートナーアシスタント

AWS Partner Assistant AWS Partnerは、の生成 AI を活用したチャットボットです。Partner Central と AWS Marketplace 管理ポータル (AMMP) の両方からアクセスでき、登録されたパートナー専用のドキュメントに従ってトレーニングされた Amazon Q Business を使用します。

Note

AWS Partner Assistant にアクセスするには、パートナーは AWS Partner Central アカウントを自分の AWS アカウントにリンクする必要があります。詳細については、[AWS 「Partner Central 入門ガイド」の「Linking Partner Central accounts with AWS Marketplace seller accounts」](#)を参照してください。AWS

チャットボットは、AWS サービス、AWS Marketplace 販売者ガイダンス、AWS パートナーネットワークのインセンティブ、利点、プログラムに関連するさまざまなトピックに関するリアルタイム

ムの回答とドキュメントの概要を提供します。チャットボットが応答を提供できない場合は、サポートチームにチケットを提出する方法についてアドバイスします。また、サムズアップボタンとサムズダウンボタンを使用してチャットボット内からフィードバックを提供することもできます。これにより、チャットボットの精度と品質が向上します。

以下は、AWS Partner Assistant との一般的なやり取りです。

1. 質問する: AWS Partner Central または AMMP からチャットボットを開き、日常的な言語を使用して質問します。
2. AWS Partner Assistant はリクエストを処理します。チャットボットは自然言語処理 (NLP) を使用してクエリを解釈します。
3. AWS Partner Assistant はレスポンスを生成します。チャットボットは、追加のリソースを含む関連するレスポンスを返します。
4. フィードバックを提供する: チャットボットからフィードバックを求められます。

利点と特徴

AWS Partner Assistant には、情報へのアクセスを改善するセルフサービスオプションが用意されています。

- 即時回答: パートナーは通常、ヘルプのために AWS 直接 に連絡する必要がありますが AWS、Partner Assistant は常に利用できます。
- 正確な回答: パートナーは、応答の変動を最小限に抑えながら、事実に基づくタイムリーな情報を受け取ります。
- セルフサービスサポート: 関連するリソースを見つけるのにかかる時間を短縮します。
- 高度なレスポンス: 自然で直感的で人間のようなレスポンスを提供します。
- コンテキスト理解: 複雑なサブジェクトに関する追加情報を提供します。
- 自己学習システム: インタラクションから学び、時間の経過とともにレスポンスを改善し、ナレッジベースを拡張することで、継続的に改善されます。
- スケーラブルなサポート: 大量の問い合わせを同時に処理できます。

AWS Partner Assistant の使用方法

1. [AWS Partner Central](#) または [AMMP](#) にログインします。
2. チャットにアクセスします。

- a. 右下隅にある紫色のフローティングアイコンを選択します。
 - b. 推奨されるプロンプトのいずれかを選択するか、テキストフィールドに質問の入力を開始します。
 - c. チャットボットはソースとリンクで応答します。
3. Partner Central からチャットボットにアクセスします。
 - a. サポートメニューを選択します。
 - b. ドロップダウンから AWS Partner Assistant とのインスタントチャットを選択します。
 - c. (オプション) サポートチケットからチャットボットにアクセスします。サポートチケットを送信する前に、AWS パートナーアシスタントに質問を選択してすぐに回答を取得します。
 4. AMMP からチャットボットにアクセスします。
 - a. お問い合わせページから、AWS パートナーアシスタントへのお問い合わせを選択します。
 - b. (オプション) サポートチケットからチャットボットにアクセスします。サポートチケットを送信する前に、お問い合わせページからサポートリクエストを送信するを選択します。Ask AWS Partner Assistant を選択して、すぐに回答を取得します。
 5. (オプション) フィードバックを提供します。
 - a. 各レスポンスの後、フィードバックを提供するオプションがあります。レスポンスがニーズを満たす場合は、サムアップを選択します。
 - b. レスポンスが役に立たなかった場合は、Thumbs Down を選択します。
 - c. Thumbs Down を選択した場合は、テキストボックスに詳細を入力するか、応答が役に立たなかった理由を最もよく表すオプションをピックリストから選択します。
 - d. チャットボットウィンドウの左下隅からフィードバックを選択してエクスペリエンスを評価します。フィードバックを選択して、評価スケールを含むポップアップを開きます。追加のコメントやフィードバックについては、テキストフィールドを使用します。

Partner Assistant に関するよくある質問

以下は、AWS Partner Assistant に関するよくある質問です。

AWS Partner Assistant とは

AWS Partner Assistant は生成 AI を活用したチャットボットで、パートナーが情報を見つけて質問に答えるのに役立ちます。

AWS Partner Assistant にアクセスするにはどうすればよいですか？

AWS Partner Assistant は AWS 、パートナーと Marketplace 販売者の両方が利用できますが、Marketplace 購入者は利用できません。AWS Partner Central または AWS Marketplace Management Portal (AMMP) から任意のページからチャットボットにアクセスできます。

AWS Partner Assistant は常に利用できますか？

はい。

Partner Assistant は AWS どの言語をサポートしていますか？

現在、AWS パートナーアシスタントは英語のみをサポートしていますが、今後のリリースでは他の言語もサポートします。

AWS Partner Assistant は何ができますか？

チャットボットは、AWS サービス、AWS Marketplace 販売者のガイダンス、AWS Partner プログラム、インセンティブ、利点に関するリアルタイムの回答とドキュメントの概要を提供します。チャットボットは、パートナー専用の公開済みコンテンツのライブラリを使用します。

AWS Partner Assistant に何を依頼できますか？

チャットボットは、AWS サービス、AWS マーケットプレイス販売者ガイダンス、AWS パートナーネットワークプログラム、インセンティブ、および利点に関するリアルタイムの回答とドキュメントの概要を提供します。チャットボットは、パートナー専用の公開済みコンテンツのライブラリを使用します。

AWS Partner Assistant にどのような問題を問い合わせるべきではありませんか？

詳細なトラブルシューティング、アカウント検証、または機密データへのアクセスが必要な問題については、AWS カスタマーサポートに直接お問い合わせください。

AWS Partner Assistant はパーソナライズされたレスポンスを提供できますか？

現在、AWS パートナーアシスタントは一般的な情報とガイダンスのみを提供します。クリエイティブコンテンツを生成したり、タスクを実行したりすることはできません。AWS Partner Assistant が質問に適切に回答できない場合は、サポートチケットを提出するか、フィードバックを送信してください。追加の例を次に示します。

実行する	しない
階層の更新日はどこで確認できますか？	階層の更新日はいつですか？
パスに登録するにはどうすればよいですか？	ソフトウェアパスに登録します。
MSP プログラムに申請するにはどうすればよいですか？	MSP プログラムに申請できますか？

Note

アカウント関連の質問の場合、パートナーは既存の[AWS サポートチャネル](#)を使用する必要があります。

パートナーは AWS Partner Assistant を使用してアカウントまたはサービスを変更できますか？

いいえ。ただし、これらの変更を行う方法や、可能なチームをサポートするように指示する方法を説明できます。

Partner Assistant は AWS 機密情報をどのように処理しますか？

AWS Partner Assistant はユーザーのプライバシーとデータを保護するように設計されていますが、機密情報の共有はお勧めしません。

AWS Partner Assistant が私の質問を理解していない場合はどうなりますか？

AWS Partner Assistant が質問に答えられない場合は、言い換えてみてください。または、サポートチームに案内してサポートを依頼することもできます。このシナリオでは、追加のコンテキストとともにサムズダウンフィードバックを送信します。これにより、チャットボットのパフォーマンスが向上します。必要に応じて、一般的なカテゴリのピックアップリストから構造化されたフィードバックを提供できます。

AWS Partner Assistant が応答しない場合はどうすればよいですか？

ページを更新するか、アプリケーションを再起動してみてください。問題が解決しない場合は、サポートチームに問題を報告してください。

AWS Partner Assistant が誤った情報を提供した場合はどうすればよいですか？

AWS Partner Assistant から間違った情報が提供されたと思われる場合は、セッションの最後にサムズダウンオプションを使用してフィードバックを提供します。

使用しない場合は AWS Partner Assistant をオフにできますか？

AWS Partner Assistant はオプション機能です。使用しない場合は、サポートチームにお問い合わせください。

用の製品の準備 AWS Marketplace

以下のトピックでは、を通じてソフトウェア製品を販売用に準備する方法について説明します AWS Marketplace。製品の準備には、パッケージの設定、料金スキームの設定、製品を一覧表示するカテゴリの決定、製品が関連する検索に表示されるようにキーワードの追加が含まれます。独立系ソフトウェアベンダー (ISV)、チャネルパートナー、マネージドサービスプロバイダー (MSP)、または AWS 製品やサービスと連携するものを提供する個人は、このプロセスを使用して AWS Marketplace で製品を公開する準備を行うことができます。

トピック

- [の製品配信 AWS Marketplace](#)
- [で実行されている製品の指定 AWS](#)
- [の製品料金 AWS Marketplace](#)
- [AWS Marketplace 製品のリージョンと国](#)
- [翻訳と言語](#)
- [での標準化された契約の使用 AWS Marketplace](#)
- [AWS Marketplace 製品のメタデータの提供](#)
- [の AMI およびコンテナ製品の使用手順の作成 AWS Marketplace](#)
- [検索用に AWS Marketplace 製品を最適化する](#)
- [プロモーションメディアで AWS Marketplace 製品を強化する](#)

の製品配信 AWS Marketplace

各配信方法には、パッケージング、料金、および配信に関するオプションが複数あります。一部のメソッドは、それをサポートするプログラムに登録 AWS Marketplace するまで、で販売者として使用することはできません。

製品の作成には、標準の表示価格とエンドユーザーライセンス契約 (EULA) を使用できます。またカスタムの料金と EULA を使用して、個別の顧客向けのプライベートオファーを作成できます。契約条件に追加の変更が必要な場合は、AWS Marketplace チームと連携してプライベートオファーをカスタマイズできます。調達プロセスを簡素化するには、公開出品とプライベートオファーの両方で[標準化されたライセンス条項](#)を使用できます。

Note

AWS は、特定の販売者にガイド付きデモンストレーションを提供するオプションを提供します AWS Marketplace。APN カスタマーエンゲージメント (ACE) の対象となる AWS パートナーネットワーク (APN) パートナーで、このオプションを購入者に提供する場合は、[プライベートオファーとデモリクエストボタンの追加](https://aws.amazon.com/partners/programs/ace/)」を参照してください。 <https://aws.amazon.com/partners/programs/ace/>

次の表に、ソフトウェア製品の配信に使用できるメソッドと、AWS Marketplace 購入者がコンソールで AWS Marketplace 各タイプの配信方法を検索するために使用するフィルターを示します。

製品配信方法	コンソールでの配信方法フィルター	説明
単一 AMI	[Amazon マシンイメージ (AMI)]	製品用のカスタム Amazon マシンイメージ (AMI) を配信します。AMI は Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを起動するために必要な情報を提供します。 購入者は単一 AMI を使用して、製品がインストール済みですぐに使用できる Amazon EC2 インスタンスを作成できます。 詳細については、「の AMI ベースの製品 AWS Marketplace」を参照してください。
AWS CloudFormation テンプレートを使用して配信 AMIs	[CloudFormation テンプレート]	CloudFormation テンプレートを使用して、AWS Marketplace 購入者に配信される AMI ベースの製品を一覧表示できます。

製品配信方法	コンソールでの配信方法フィルター	説明
		CloudFormation テンプレートとしての AMIs 配信の詳細については、「を使用した AMI ベースの配信 AWS CloudFormation」を参照してください。 CloudFormation テンプレートの詳細については、「AWS CloudFormation ユーザーガイド」の「AWS CloudFormation の概念」を参照してください。
EC2 Image Builder コンポーネントベースの製品	EC2 Image Builder コンポーネント	EC2 Image Builder コンポーネントを使用して、AWS Marketplace 購入者に配信される製品を一覧表示できます。 EC2 Image Builder コンポーネントの使用の詳細については、AWS Marketplace API リファレンスのEC2 Image Builder コンポーネント製品の操作」を参照してください。

製品配信方法	コンソールでの配信方法フィルター	説明
コンテナベースの製品またはアプリケーション	[コンテナ]	コンテナイメージにパッケージされた製品を配信します。コンテナ製品は、コンテナイメージとデプロイテンプレートのセットであるオプションで構成されます。 詳細については、「のコンテナベースの製品 AWS Marketplace」を参照してください。
データ製品	AWS Data Exchange	既存の PLF を使用するか、AWS Data Exchange を使用してデータ製品を作成します。 AWS Data Exchange を通じてデータ製品などを公開および管理する方法については、「AWS Data Exchange ユーザーガイド」の「AWS Data Exchange でのデータ製品の提供」を参照してください。

製品配信方法	コンソールでの配信方法フィルター	説明
機械学習アルゴリズムとモデルパッケージ	[SageMaker モデル]	Amazon SageMaker AI を使用してアルゴリズムまたはモデルパッケージを作成し、公開します AWS Marketplace。 機械学習アルゴリズムとモデルパッケージの提供の詳細については、「機械学習製品」を参照してください。 SageMaker AI の詳細については、Amazon SageMaker AI デベロッパーガイドの「SageMaker AI とは」を参照してください。 Amazon SageMaker
Software as a service (SaaS)	SaaS	SaaS 製品は、サブスクリプションベース、契約ベース、または消費料金モデルとの契約で提供できます。 詳細については、「の SaaS ベースの製品 AWS Marketplace」を参照してください。
プロフェッショナルサービス	プロフェッショナルサービス	他の AWS Marketplace 製品をサポートまたは操作するプロフェッショナルサービスを提供できます。

で実行されている製品の指定 AWS

Note

次の指定は、2025 年 5 月 1 日に有効になります。

にデプロイされた製品は、検索結果とその製品詳細ページで特別な指定 AWS AWS Marketplace を受けます。次の表に、が指定を受け取ることができる製品配信タイプを示します。

製品配信方法	にデプロイ済みとして指定 AWS
AMI ベース	はい
コンテナベース	はい。ただし、Amazon EKS Anywhere にデプロイされた製品を除きます。この機能を有効にするコンテナ製品の詳細については、このガイドの AWS Marketplace for Containers Anywhere と License Manager の統合 後半の「」を参照してください。
SageMaker AI アルゴリズムとモデルパッケージベース	はい
SaaS ベース	完全に実行される製品のみ AWS。詳細については、このガイドの アーキテクチャガイドライン 後半を参照してください。
データベース	はい
プロフェッショナルサービスベース	いいえ

の製品料金 AWS Marketplace

AWS Marketplace 販売者は、製品の料金モデルを選択できます。有料製品の場合、AWS Marketplace はお客様にソフトウェア料金を請求します。すべての料金は、米ドル (USD) に基づいています。無料のソフトウェア、または無償で提供されるオープンソースのソフトウェアにはサービ

送料はかかりません。このトピックでは、AWS Marketplaceのソフトウェア製品の料金モデルと料金に関する情報を提供します。

返金については、「[での製品の返金 AWS Marketplace](#)」を参照してください。

トピック

- [料金モデル](#)
- [料金モデルの変更](#)
- [料金の変更](#)
- [プライベートオファー](#)
- [での製品の返金 AWS Marketplace](#)

料金モデル

次のトピックでは、AWS Marketplaceで利用可能な料金モデルに関する一般的な情報を示します。

トピック

- [年間料金](#)
- [使用料金](#)
- [契約料金](#)
- [Bring Your Own License の料金](#)

特定の製品配信方法の料金モデルについては、以下を参照してください。

- [の AMI 製品の料金 AWS Marketplace](#)
- [のコンテナ製品の料金 AWS Marketplace](#)
- [の機械学習製品の料金 AWS Marketplace](#)
- [での SaaS 製品の料金 AWS Marketplace](#)
- [でのプロフェッショナルサービス製品の料金 AWS Marketplace](#)

年間料金

年間価格モデルでは、12 か月のサブスクリプションを購入するお客様に製品を提供できます。例えば、サブスクリプション料金は、同じ製品を 1 時間ごとに長期間実行する場合に比べて、最大 40% 節約できます。サブスクリプション契約時に全額請求されます。年間サブスクリプションがお客様に

提示される方法の詳細については、「[AMI サブスクリプション](#)」または「[有料コンテナ製品の料金モデル](#)」を参照してください。

年間サブスクリプションを使用する場合の考慮事項は以下のとおりです。

- 年間料金は、インスタンスタイプごとに定義されます。すべての Amazon Elastic Compute Cloud (Amazon EC2) インスタンスタイプで同じ場合や、インスタンスタイプごとに異なる場合があります。
- すべての年間インスタンスタイプには、時間単位のインスタンスタイプも定義されている必要があります。AWS Marketplace では、同じ製品に対して年間みの料金または年間なしの時間単位は提供されません。年間料金を提供する製品では、時間単位の料金も指定する必要があります。
- 時間料金が 0 USD で、0 USD 以外の年間インスタンスタイプが定義されている場合は、特定のインスタンスタイプで年間料金を 0 USD に設定することができます。
- 年間サブスクリプション期間の終了時、時間単位の請求が開始されます。
- お客様が年間サブスクリプション X を購入したが Y インスタンスで Y ソフトウェアを実行している場合、年間サブスクリプションの対象外である (Y-X) インスタンスの時間単位のソフトウェア料金で請求されます。そのため、年間料金のインスタンスタイプには必ず、時間レートを含める必要があります。
- 販売者のプライベートオファーを使用すると、複数年 (最大 3 年) またはカスタム期間 AMI を前払い料金で提供したり、柔軟な支払いスケジュールを提供したりできます。複数年契約とカスタム期間契約の詳細については、[AWS Marketplace 製品のプライベートオファーの準備](#)および[プライベートオファーの分割プラン](#)を参照してください。

で年間製品を提供する場合 AWS Marketplace、 のファイルアップローダードキュメントセクションにある年間製品の特定の返金ポリシーに同意することになります[AWS Marketplace 管理ポータル](#)。

料金の変更

年間料金 (年額 1,000 USD など、年額 1,200 USD) は 90 日ごとに変更できます。ただし、既存のお客様には 90 日前に年間料金を通知する必要があります。新しい価格は新しいサブスクリプションに適用されますが、既存のサブスクリプションには影響しません。

Note

90 日間の通知期間中は、サポートされているインスタンスタイプを更新することはできません。

価格の変更は、自動更新日より少なくとも 90 日前に変更された場合にのみ有効です。自動更新前に、新しい価格が記載された E メールメッセージがお客様に送信されます。

エンドユーザーライセンス契約

年間サブスクリプションに基づく 12 か月間の AWS お客様のソフトウェアの使用は、製品の詳細ページで指定した EULA の対象となります AWS Marketplace。

使用料金

従量制料金とも呼ばれる使用量料金モデルでは、使用した分のみを支払うお客様に製品を提供できます。

販売者は、以下の使用カテゴリのいずれかを選択できます。

- [ユーザー]
- [ホスト]
- [帯域幅]
- [データ]
- [階層]
- [単位] (カスタムカテゴリ用)

また、製品に対して最大 24 の分析を定義できます。料金は、ソフトウェアが API を呼び出した際に測定され、報告されます。ユースケースにもよりますが、ベストプラクティスとして API が 1 時間に 1 回呼び出されるよう設定することをお勧めします。すべての使用量は毎月計算され、既存の AWS Marketplace ソフトウェアと同じメカニズムを使用して毎月請求されます。

AWS Marketplace Metering Service では、新しい料金シナリオをいくつか使用することができます。

Example ホスト別料金

ソフトウェアがホストをモニタリングする場合は、モニタリングされるホストごとに請求し、ホストサイズに基づき異なる料金を設定することができます。

Example ユーザー別料金

お客様のソフトウェアで、組織における複数のユーザーの使用を許可している場合は、ユーザー単位で請求することができます。1 時間ごとに、プロビジョニングされたユーザーの合計数が請求されます。

Note

製品積載フォーム (PLF) では、関連する列に「FCP」(Flexible Consumption Pricing) という名前が付けられます。例: [FCP カテゴリ (カスタム料金カテゴリ)]。

AWS Marketplace Metering Service 製品の場合は、次の点に注意してください。

- ソフトウェアがすでにオンになっている場合は AWS Marketplace、代替の使用ディメンションを有効にする製品を作成する必要があります。標準製品を AWS Marketplace 計測サービスを使用するように変更することはできません。新製品が公開されたら、古い製品を削除するか、いずれもウェブサイトに維持することができます。
- AWS Marketplace Metering Service では、ソフトウェアが 1 時間ごとに使用状況を報告し、1 時間の顧客使用状況を記録する必要があります。計測サービスレコードの送信または受信に障害が発生した場合、AWS はそのような使用に対して請求できません。計測レコードを正常に受け取ったことを確認する必要があります。
- AWS Marketplace Metering Service を使用する製品では 1-Click はサポートされていません。購入者は、特定のアクセス許可を持ち、インターネットゲートウェイを持つ AWS Identity and Access Management (IAM) ロールを使用してソフトウェアを起動する必要があります。
- 無料トライアルと年間料金は、AWS Marketplace Metering Service と互換性がありません。
- デイメンション (ユーザー、ホスト、帯域幅、およびデータ) またはデイメンション名の変更はサポートされていません。新しい製品を作成する必要はありません。

契約料金

契約料金モデルを使用すると、1 か月、12 か月、24 か月、または 36 か月間のライセンスを購入できるような前払い料金をお客様に提供できます。

契約料金は、以下の製品で利用できます。

- AWS CloudFormation テンプレートベースの製品を含む単一の AMI ベースの製品と AMI。詳細については、[での AMI 製品の契約料金 AWS Marketplace](#)を参照してください。

- コンテナベースの製品 詳細については、「[コンテナ製品の契約料金](#)」を参照してください。
- Software as a service (SaaS) ベースの製品 詳細については、「[SaaS 契約の料金](#)」を参照してください。

Note

AMI およびコンテナベースの製品の契約料金は、新製品のみを対象としています。既存の AMI またはコンテナベースの製品があり、契約料金を使用する場合は、新しい出品を作成し、Product Load Form (PLF) を使用して契約料金モデルを適用してさまざまなディメンションを追加し、AMI またはコンテナベースの製品を と統合してから AWS License Manager、AMI またはコンテナベースの製品を公開します。顧客が契約料金で製品を購入すると、License Manager API を使用してソフトウェア AWS アカウント がチェックできるライセンスが顧客 AWS Marketplace 内で によって作成されます。お客様が AMI またはコンテナベースの製品のインスタンスを起動するには IAM ロールが必要です。

Bring Your Own License の料金

AWS Marketplaceでは、自分のライセンス使用 (BYOL) 製品のサービス料はかかりません。

お客様の選択を保証するために、すべての BYOL 製品に対し有料オプションが必要です。これにより、既存のライセンスがないお客様が製品を購入して使用することもできるようになります。

BYOL 製品の場合、ソフトウェアをオンラインで購入できるようにすることは、事業内容から逸脱する企業もあると当社では認識しています。したがって、当社は、ソフトウェアには AWS Marketplaceでの購入が可能なバージョンがあるべきとする要件を、起動後 90 日間は緩和します。この間、AWS Marketplace アカウント管理チームはお客様と協力して課題に対処します。チームがソフトウェアを AWS Marketplaceで購入可能にするかどうか、またどのように購入できるかを判断するお手伝いをします。

料金モデルの変更

料金モデルの変更は、ポジティブなカスタマーエクスペリエンスを確保し、すべての当事者へのリスクを軽減 AWS Marketplace するために、 によってレビューおよび承認される必要があります。料金モデルの変更を行いたい場合は、[AWS Marketplace Seller Operations](#) チームに連絡して相談してください。

料金モデルの変更に関するリクエストは、処理とレビューに 30〜90 日ほどかかります。

 Note

SaaS 製品では料金モデルの変更はサポートされていないことに注意してください。SaaS 製品の料金の詳細については、「での [SaaS 製品の料金 AWS Marketplace](#)」を参照してください。

料金の変更

AWS Marketplace 管理ポータルで価格とメタデータを更新できます。

価格を変更するには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [製品] タブに、作成した現在の製品のリストが表示されます。ここで製品リストを編集するか、変更をリクエストします。

 Note

新しいサブスクライバーの場合、料金変更はすぐに有効になります。パブリックオファ어의既存のサブスクライバーの場合、料金変更は 90 日間の期間の翌月 1 日に有効になります。例えば、3 月 16 日に料金変更通知を送信した場合、料金変更は、7 月 1 日となる 90 日間の期間の後の月の最初の日に発生します。

プライベートオファー

AWS Marketplace 販売者プライベートオファープログラムでは、AWS Marketplace 販売者は個々の AWS Marketplace 顧客 (購入者) とカスタム料金と EULAs を交渉できます。詳細については、「[AWS Marketplace 製品のプライベートオファーの準備](#)」を参照してください。

での製品の返金 AWS Marketplace

Note

Amazon Web Services EMEA オペレーターを通じて取引する場合、返金の処理、顧客確認 (KYC) 情報の修正、銀行口座の詳細などの財務情報の変更は、検証済みのユーザーのみが行うことができます。検証済みユーザーは、KYC 検証済みユーザーと検証を完了するセカンダリユーザーです。KYC プロセスの詳細については、[「顧客確認プロセスの完了」](#)を参照してください。

のすべての有料製品には AWS Marketplace、料金モデルに関係なく、ソフトウェア料金の返金ポリシーが明記されている必要があります。返金ポリシーには、返金の条件と、返金をリクエストするために販売者に連絡する方法を含める必要があります。

返金ポリシーの詳細を決定するのは販売者です。ただし、購入者に製品の使用に対して何らかの返金を提供することをお勧めします。お客様は、掲載した返金ポリシーを遵守する必要があります。このトピックでは、AWS Marketplace 製品の返金リクエストのタイプ、関連するポリシーと承認プロセス、購入者に返金リクエストを送信する方法について説明します。

トピック

- [AWS Marketplace 製品の返金リクエストタイプ](#)
- [AWS Marketplace 製品の返金ポリシーと承認](#)
- [製品の返金のリクエスト](#)
- [一括返金のリクエスト](#)

AWS Marketplace 製品の返金リクエストタイプ

購入者は AWS Marketplace、製品のさまざまなタイプの返金をリクエストできます。によって販売される AWS Marketplace 製品については AWS、返金ポリシーページを参照し、を使用して問い合わせサポートフォームを送信します AWS Support Center Console。購入者が から直接ソフトウェアの返金をリクエストする場合 AWS、問題の製品の投稿されたサポート連絡先情報を使用して販売者に連絡するように指示します。AWS インフラストラクチャ料金の返金は、の自由裁量であり AWS、ソフトウェアの返金とは無関係に処理されます。

サードパーティーが販売する製品の場合、購入者は製品詳細ページで返金ポリシーを表示する必要があります。AWS Marketplace サブスクリプションのソフトウェア料金は製品の販売者に支払

い、返金は販売者から直接リクエストする必要があります。各 AWS Marketplace 販売者は、AWS Marketplace ページに返金ポリシーを含める必要があります。

AWS Marketplace 製品の返金ポリシーと承認

次のリストは、AWS Marketplace 返金ポリシーと、承認が必要かどうかを示しています。

- 無料トライアル

ソフトウェアを無料トライアル製品としてリストした場合、AWS は無料トライアルから有料サブスクリプションへの変換から 7 日以内に発生するソフトウェア料金の返金をお客様に代わって行うことができます。返金が無料トライアルの変換に関連して行われる場合、お客様によるアクションは不要です。製品で無料トライアルを有効にすると、このポリシーに同意したことになります。

- プライベートオファー

プライベートオファーのすべての返金は、が AWS 処理する前に、お客様が承認する必要があります。

- ソフトウェア計測の返金

を使用してソフトウェアの使用量を AWS Marketplace Metering Service 計測する場合、AWS はソフトウェア計測エラーに起因するソフトウェア料金の返金をユーザーに代わって行うことができます。これらのエラーが複数の購入者に共通している場合、は各購入者の適切な返金を決定し、各購入者に直接適用する権利 AWS を予約します。に関連して発行された返金は、販売者に 1 回確認 AWS Marketplace Metering Service する必要がありますが、販売者が個々の返金を確認する必要はありません。製品 AWS Marketplace Metering Service でを使用すると、このポリシーに同意したことになります。

- 購入後 48 時間以内のサブスクリプションキャンセル

購入者がプライベートオファー以外の購入から 48 時間以内にサブスクリプションをキャンセルした場合、は全額返金 (100% 返金によるキャンセル) AWS を行います。購入から 48 時間以内のキャンセルに対する返金には、お客様側での対応は必要ありません。48 時間経過後にこのような購入者のリクエストがあった場合は、お客様の裁量に委ねられます。に製品を一覧表示することで AWS Marketplace、このポリシーに同意したものとみなされます。

- サブスクリプションのアップグレード

購入者が既存の非プライベートオファーサブスクリプションをより高価なサブスクリプションまたは同等の価値のサブスクリプションに置き換えた場合、はユーザーに代わって低層サブスクリプ

ションの返金を発行 AWS できます。購入者にとっては 2 段階のプロセスになります。新しいサブスクリプションを購入し、次に古いサブスクリプションの解約と返金をリクエストします。

- サブスクリプションのダウングレード

ダウングレードサブスクリプションの返金リクエストはすべて、が AWS 処理する前に、お客様が承認する必要があります。

AWS 承認された返金はすべて自動的に処理され、お客様側のアクションは必要ありません。

製品の返金のリクエスト

外部購入者または内部テストアカウントのソフトウェア返金をリクエストするには、次の手順に従います。

ソフトウェアの返金をリクエストするには

1. 以下の情報を収集します。

- 製品のサブスクリプションに使用される購入者 AWS アカウント ID (12 桁)。購入者が組織の支払者である場合は、サブスクリプションに使用されるリンクされたアカウントの AWS アカウント ID を取得します。
- に関連付けられている購入者の E メールアドレス AWS アカウント。
- 販売者 AWS アカウント ID (12 桁)。
- 製品 ID 製品 ID は、[契約および更新ダッシュボード](#)または[請求済み収益ダッシュボード](#)にあります。
- オファー ID。オファー ID は、の[契約と更新ダッシュボード](#)または[請求済み収益ダッシュボード](#)、オファータブ、または契約タブにあります[AWS Marketplace 管理ポータル](#)。
- 請求期間。
- [請求済み収益ダッシュボード](#)の請求書番号。
- 税引き前返金額。

2. [AWS Marketplace 管理ポータル](#)にサインインします。

3. サポートを選択します。

4. 返金のリクエストを選択します。

5. サブスクライバーの AWS アカウント 番号、製品 ID、請求期間を入力します。

6. 追加コメント (オプション) フィールドに、製品 ID、返金タイプ、請求書番号 (利用可能な場合) など、リクエストの簡単な説明を入力します。例:

Cancel the buyer's subscription to <product-ID> and <offer-ID> and process a full refund for invoice <invoice-number>.

7. 年間サブスクリプションの返金、アップグレード、またはダウングレードのリクエストを送信するには、次の手順を実行します。
 - a. [契約と更新ダッシュボード](#)または[請求済み収益](#)ダッシュボードを使用して、購入者が年間サブスクリプションを購入していることを確認します。
 - b. 追加コメント (オプション) フィールドに、サブスクリプションのキャンセル日を入力し、購入者が返金、アップグレード、またはダウングレードをリクエストするかどうかを指定します。
8. フォームを送信します。通知が届き、返金の処理が開始され、購入者に発行されます。
9. 返金リクエストを管理するために、AWS Marketplace 購入者サポートチームは[サポート センターコンソール](#)で対応するサポートケースを作成します。返金のステータスは、サポートケースの件名に表示されます。ステータスは次のいずれかになります。
 - 完了 - 返金処理が完了したため、これ以上の操作は必要ありません。
 - 保留中 - 現在の請求サイクルが終了すると、返金が処理されます。
 - アクションが必要 - リクエストを処理できませんでした。追加情報が必要です。サポートケースに直接返信できますが、新しい返金リクエストフォームも送信する必要があります。

返金は 24 ~ 48 時間以内に購入者のアカウントに表示されます。ただし、資金が購入者の金融口座に表示されるまでに最大 5 営業日かかる場合があります。

一括返金のリクエスト

次の手順では、20 件以上の請求書または 20 件以上のアカウントを含む返金リクエストを作成する方法について説明します。このプロセスにより、大規模な返金リクエストが合理化され、必要な情報が提供されるようになります。

一括返金をリクエストするには

1. 以下の必須情報を収集します。
 - 販売者の AWS アカウント ID (12 桁)
 - すべての AWS アカウント IDs のリスト (各 12 桁)。複数の購入者を入力することも、複数の請求期間に対して単一の購入者を入力することもできます。

- すべての製品 IDs
 - 該当する場合はオフアー IDs
 - すべての請求書 IDs
 - 各請求書の請求期間
 - AWS Marketplace 返金チケットリファレンス ID
2. 次の列を使用して、必要な情報を CSV スプレッドシートにコンパイルします。

- **Seller Account ID**
- **Subscriber Account ID**
- **Payer Account ID**
- **Billing Period**
- **Invoice ID**
- **Targeted Amount**
- **Product ID**

3. ルートユーザーとして[サポート センターコンソール](#)にサインインします。

 Note

ルートユーザーとしてサインインしない限り、これらのステップを完了することはできません。

4. アカウントと請求のサポートケースを作成し、 を選択しますAWS Marketplace。
5. 件名に と入力します**Request Bulk Refund**。
6. 「追加コメント」に次のように入力します。
- 影響を受けるアカウントの総数
 - 請求書の合計数
 - 返金総額
 - 返金に関するビジネス上の根拠
 - AWS Marketplace 返金チケットリファレンス ID
7. CSV ファイルをアタッチし、サポートケースを作成します。
8. サポートケースは、サポートのために次に利用可能なエージェントにルーティングされます。

AWS Marketplace 製品のリージョンと国

で製品を作成するときは AWS Marketplace、利用可能な AWS リージョンを選択します。購入者が製品を購入できる国も選択します。これら 2 つのプロパティは似ていますが、同じではありません。例として、購入者が米国に居住し、米国から購入していても、製品を欧州 (フランクフルト) リージョンにインストールする場合があります。この購入者が製品を購入するには、国のリストに米国を、リージョンのリストに欧州 (フランクフルト) を含める必要があります。このトピックを使用して、AWS Marketplace 製品のリージョンと国の詳細を確認できます。

AWS リージョン

サーバーまたは機械学習製品情報を作成または編集するときは、ユーザーが製品をインストールして使用できる特定の AWS リージョンに製品を制限できます。

Amazon マシンイメージ (AMI)、コンテナ、および AWS CloudFormation ベースの製品を含むサーバー製品の場合、製品が利用可能な特定のリージョンを選択できます。また、製品が利用可能になりしだい、新しい米国リージョン、米国以外のリージョン、またはすべてのリージョンで製品を自動的に利用可能になるように選択することもできます。

機械学習製品の場合は、特定のリージョンを選択することも、今後利用可能になるリージョンを含むすべてのリージョンを選択することもできます。

AWS リージョンの詳細については、AWS 「全般のリファレンス」の[AWS 「サービスエンドポイント」](#)を参照してください。

Countries (国)

デフォルトでは、が利用可能なすべての国の購入者 AWS Marketplace が製品を入手できます。新規および既存のサーバーと Software as a Service (SaaS) 製品については、税務、コンプライアンス、サポート、またはマーケティングを目的として、特定の国での製品の使用可否を管理できます。

この機能には例外があります。

- 以前の購入 - 新しい国のリストと共に製品を更新した後も、既に製品をサブスクライブしている購入者は、サブスクリプションが有効である間、引き続きアクセスできます。
- プライベートオファー - 製品を特定の国の購入者に限定しても、プライベートオファーは制限されません。特定の購入者に対してプライベートオファーを作成すると、指定した国に含めなかった国にその購入者がいる場合でも利用できます。

Note

顧客の適格性は、AWS リンクされたアカウントレベルで決定されます。詳細については、[「が AWS アカウントの場所を決定する方法」](#)を参照してください。

使用権限を共有する顧客は、許可したリージョンでのみその使用権限を有効化できます。使用権限の管理について詳しくは、「AWS Marketplace 購入者ガイド」の[「組織内でのサブスクリプションの共有」](#)を参照してください。

翻訳と言語

デフォルトでは、AWS Marketplace は製品リストとオファーを次の言語に変換します。

- フランス語 (fr-FR)
- 日本語 (ja-JP)
- 韓国語 (ko-KR)
- スペイン語 (es-ES)

AWS Marketplace は、参照のみを目的として、これらの言語での一連の標準契約例も提供します。契約には法的拘束力はありません。契約の詳細については、「AWS Marketplace 購入者ガイド」の[「サポートされている言語」](#)を参照してください。

以下のトピックでは、翻訳の仕組み、翻訳された標準契約のダウンロード方法、翻訳のオプトアウト方法について説明します。

トピック

- [入門](#)
- [翻訳のオプトアウト](#)

入門

AWS Marketplace 翻訳については、次の点に注意してください。

- グローバル市場へのリーチを拡大するために、AWS Marketplace はお客様から提供された英語コンテンツを翻訳します。翻訳に問題がある場合は、AWS Marketplace 管理ポータル[のお問い合わせページ](#)を使用してください。製品出品カテゴリとローカル言語オプトアウトカテゴリを選択し、問題に関するフィードバックを提供します。

AWS Marketplace は、技術用語や製品名を翻訳しないなど、限定的な変更を行うことができます。

- リストの翻訳をオプトアウトできます。その場合、購入者はコンテンツを英語で表示し、関連する AWS Marketplace コンテンツは任意の言語で表示します。オプトアウトの詳細については、[翻訳のオプトアウト](#) 以下を参照してください。
- 販売者の登録情報と製品出品情報を英語で送信する必要があります。
- EULAs、サポートされている言語で送信できます。関連する製品リストはジオターゲットでなければならない、言語は国のプライマリ言語である必要があります。

AWS Marketplace は、参照のみを目的として、上記の言語に翻訳された一連の標準サンプル契約を提供します。購入者 AWS Marketplace ガイド [のサポートされている言語](#) から契約をダウンロードできます。

- ベストプラクティスとして、新しい製品を公開するときは、製品を限定した状態で翻訳を確認します。製品を公開する前に、問題があれば [お問い合わせください](#)。
- 翻訳プロセスは、製品を限定された状態に移行した後に開始されます。購入者は、翻訳が利用可能になるまで英語のテキストを表示します。

翻訳のオプトアウト

次の手順では、製品リストの翻訳をオプトアウトする方法について説明します。一部またはすべての言語をオプトアウトできます。たとえば、リストをフランス語と日本語にのみ翻訳できます。

オプトアウトが有効になるまで 3~5 日かかります。

オプトアウトするには

1. [AWS Marketplace 管理ポータル](#) に移動し、お問い合わせを選択します。
2. 次のカテゴリを一覧表示された順序で選択します。
 - コマーシャルマーケットプレイス
 - 製品出品
 - ローカル言語のオプトアウト
3. 出品 ID とオプトアウトする言語を含めます。
4. ベストプラクティスとして、購入者が英語のテキストが表示される理由を理解するのに役立ちます。例:

- a. すべての翻訳をオプトアウトする場合は、製品名の末尾**English**に を追加します。
- b. 1 つ以上の翻訳をオプトアウトする場合は、製品説明の最後に説明テキストを追加します。

例: **We have declined automatic translation services for this product.**

5. リクエストを送信します。

での標準化された契約の使用 AWS Marketplace

AWS Marketplace 販売者は[製品を準備](#)する際、製品の使用に対してどのエンドユーザーライセンス契約書 (EULA) を適用するか決定する必要があります。以下を使用できます。

- 独自の EULA。
- の標準契約 AWS Marketplace (SCMP)。調達ワークフローの合理化とトランザクションの高速化に役立つ契約テンプレートです。SCMP の連邦補遺を使用して、米国政府の調達を迅速化することもできます。
- ISVs使用できる標準化リセラー契約テンプレートである AWS Marketplace (RCMP) のリセラー契約。

以下のトピックでは、標準化された AWS Marketplace 契約テンプレートを使用する方法について説明します。

トピック

- [の標準契約 AWS Marketplace](#)
- [のリセラー契約 AWS Marketplace](#)

の標準契約 AWS Marketplace

AWS Marketplace は、購入者コミュニティおよび販売者コミュニティと協力して[AWS Marketplace](#)、[\(SCMP\) の標準契約](#)を作成しました。SCMP は、デジタルソリューションの使用方法を規定し、購入者と販売者の義務を定めています。デジタルソリューションの例としては、サーバーソフトウェア、Software as a Service (SaaS)、人工知能と機械学習 (AI/ML) アルゴリズムなどがあります。

SCMP では、使用、保証、補償、準拠法などの主要な契約条項に共通する点が既に定義されています。販売者は、購入者がソリューションを検索、購入、迅速にデプロイできるセルフサービス取引の

EULA として SCMP 条件を提示できます。[プライベートオファー](#)の場合、購入者は販売者に SCMP テンプレートをリクエストでき、両当事者の合意に従ってカスタム取引要件に対応するように条件を修正できます。

セルフサービスまたはプライベートオファーについては、以下の補遺を SCMP にオプションで付けることもできます。

- [セキュリティ強化補遺](#) - データセキュリティ要件が厳しい取引をサポートします。
- [HIPAA ビジネスアソシエイト補遺](#) - HIPAA (Health Insurance Portability and Accountability Act of 1996: 医療保険の携行性と責任に関する法律) に準拠している取引をサポートします。
- [Federal Addendum](#) - 米国政府が関与するソフトウェア購入をサポートします。

免責事項

EULA はお客様と購入者の間で締結されます。SCMP を EULA として使用するかどうかは、お客様の裁量に委ねられます。SCMP を製品リストに適用することにより、SCMP プログラムに参加したことになります。このプログラムでは、SCMP テンプレートを定期的に AWS 更新し、条件を含む製品リストを最新バージョンで更新する場合があります。SCMP テンプレートを独自の EULA に置き換えることにより、いつでも SCMP プログラムから脱退することができます。

SCMP の利用方法

EULA を SCMP に更新して、新規および既存の Amazon マシンインスタンス (AMI) および Software as a Service (SaaS) の単一製品の購入者に提示することができます。更新をリクエストする手順は、製品がセルフサービスリスト (SSL) を通じて出品されているかどうかによって異なります。

ご質問がある場合は、の AWS Marketplace 標準契約チームに E メールメッセージを送信してください aws-mp-standardcontract@amazon.com。

SSL から出品されている AMI 製品および SaaS 製品の EULA を SCMP に更新するには

1. の [標準契約の AWS Marketplace](#) 条項を確認してください。
2. [AWS Marketplace 管理ポータル](#) にサインインします。
3. 製品については [製品タイプ] を選択し、契約書を更新する [製品リスト] を選択します。
4. [変更をリクエスト] を選択し、[リージョンと料金の更新] を選択します。

5. まだ選択されていない場合は、EULA の の標準契約 AWS Marketplaceを選択します。
6. [メモと通知] に **Please update this product to SCMP V2** と入力します。
7. 変更内容を確認し、[レビューのために送信] をクリックします。

SSL から出品されていない AMI 製品および SaaS 製品の EULA を SCMP に更新するには

1. AWS Marketplace 管理ポータルの[お問い合わせ](#)ページを開きます。
2. AWS Marketplace 販売者アカウントにサインインします。
3. 以下の手順に従ってフォームに入力します。
 - 質問の件名として、[Commercial Marketplace] を選択します。
 - カテゴリとして、[Product Listing] を選択します。
 - サブカテゴリとして、[Standard Contract Request] を選択します。
 - 説明に、**Please update these products to SCMP V2** と入力し、SCMP に更新する製品のタイトルと ID を入力します。

のリセラー契約 AWS Marketplace

リセラー契約 AWS Marketplace (RCMP) は、ISV がチャネルパートナーに ISVs 製品を AWS Marketplace 購入者に再販することを許可する際に使用できる標準化されたリセラー契約テンプレートです。この契約書は、ISV とチャネルパートナーがリセラー関係を締結したり、[チャネルパートナープライベートオファー \(CPPO\)](#) のリセラー条件を使用したりする際に、法的契約レビューの重複を減らし、市場投入までの時間を短縮するのに役立ちます。CPPO を作成すると、ISVs は契約をオポチュニティにアップロードし (再販認可)、チャネルパートナーは契約を表示して承諾できます。AWS Marketplace 購入者は RCMP を表示できません。

免責事項

RCMP は ISV 向けのオプション契約です。ISV がチャネルパートナーを通じて製品を再販することを決定した場合、オポチュニティを作成する際に RCMP を添付することも、既存または事前に交渉した独自の契約条件を添付することもできます。

RCMP の利用方法

このセクションでは、条件を確認して RCMP を利用する方法について説明します。

AWS Marketplace オポチュニティの作成中に RCMP を使用するには

1. [リセラー契約の AWS Marketplace](#) 条項を確認してください。
2. 「[RCMP ガイド](#)」を参照しながら、「[チャネルパートナー向けの再販のオポチュニティの作成](#)」の手順に従います。

AWS Marketplace 製品のメタデータの提供

販売者は、製品を追加するときに AWS Marketplace 製品メタデータを指定します。製品メタデータには、お客様の製品の説明に使用する名前、説明、カテゴリ、キーワードが含まれます。AWS Marketplace は、品質保証とエラー修正のみを目的として製品メタデータを改訂します。このトピックでは、AWS Marketplace 販売者による製品の効果的な命名、説明、分類に使用できる情報とベストプラクティスについて説明します。

Note

料金、使用状況、類似製品、製品比較など、すべての製品詳細ページのタブとセクションは動的で、によって管理されます AWS Marketplace。販売者は、セクションを追加または削除するために製品詳細ページをカスタマイズすることはできません。

製品の名前付けと説明

販売者が製品について提供する情報は、購入者に表示されます。潜在的な購入者が製品に関して十分な情報を得たうえで購入の決定ができるようにしてください。

製品名の決定

製品名を作成する際は、次のガイドラインに注意してください。

- タイトルケースを使用する (重要語それぞれの最初の文字が大文字になっている)
- 購入者が製品を名前のみで識別できるようにする
- ブランドまたは製造元の名前を使用する
- 説明的なデータや誇張を避ける

製品名の例: Smart Solution ロードバランサー - プレミアムエディション。

サポートされている文字

メタデータフィールドでサポートされている文字は次のとおりです。

- ASCII 文字コード 0 ~ 126
 - 詳細については、「[標準 ASCII テーブル](#)」を参照してください。
- Copyright ©
- 登録済み ®
- 商標 "
- 通貨記号 ¢ £ ¨ ¥

製品説明を書く

製品の説明では、製品の機能、利点、使用方法を示します。その他、製品固有の関連情報も提供できます。説明は最大 350 文字とすることができます。

製品説明を記述する際は、次のガイドラインに注意してください。

- 不必要な大文字を避ける
- 不要な句読点を避ける
- リダイレクト情報を含めない
- スペルと文法を確認する
- 重要で有益な情報のみを含める
- 説明的なデータや誇張を避ける

製品ソリューションの例: Smart Solution は、アプリケーションの着信トラフィックを複数の Amazon EC2 インスタンスに自動的に分散します。これにより、アプリケーションの着信トラフィックに応じて必要な量の負荷分散能力を提供し、アプリケーションの耐障害性をさらに高めることができます。Smart Solution はプール内の異常なインスタンスを検出すると、この異常なインスタンスが回復するまで、自動的にトラフィックを正常なインスタンスに振り分けます。1 つの AWS アベイラビリティーゾーンまたは複数のアベイラビリティーゾーンでスマートソリューションを有効にして、より一貫したアプリケーションパフォーマンスを確保できます。

製品ハイライトを書く

製品情報ページは、最大 3 つの製品ハイライトの箇条書きを表示します。これらの箇条書きを使用して、製品の主なセールスポイントを簡単に説明します。

製品ハイライトの例: コストの見積り - Smart Solution では、使用した分のみ支払います。Smart Solution が実行されている 1 時間 (または 1 時間未満) ごとに請求されます。

サポート情報を書き込む

顧客は、サービスの使用、トラブルシューティング、返金のリクエスト (該当する場合) などの問題について、簡単にサポートを受けることができる必要があります。E メール、電話番号、サポートウェブフォームリンクなどのサポート連絡先オプションは、フルフィルメントランディングページで指定する必要があります。

カテゴリとキーワードの選択

製品を出品するとき、製品に最大で 3 つのソフトウェアカテゴリと、対応するサブカテゴリを選択できます。これにより、購入者は製品を閲覧または検索するときに製品を検出できます AWS Marketplace。製品に関連するカテゴリのみを選択してください。ほとんどの場合、該当するカテゴリは 1 つのみです。製品ロードフォームおよび [製品] タブに、カテゴリの完全なリストが含まれます。

カテゴリはキーワードと同一ではありません。使用可能なカテゴリとサブカテゴリは 用に事前定義されており AWS Marketplace、製品リクエストプロセス中にリストから選択して、製品に適用するカテゴリとサブカテゴリを決定します。キーワードは事前定義されていません。プロセス中に作成されます。カテゴリをキーワードとして追加する必要はありません。

検索キーワードの作成

製品リクエストプロセス中に、購入者がサイトを検索して製品を見つけやすくするために、最大 3 つのキーワード (単語またはフレーズ) を入力できます。キーワードのフィールドには最大 250 文字含めることができます。

次のヒントを活用して、関連する一連の検索キーワードを作成します。

- 関連する用語を使用します。
- 他の販売者が公開している製品名を使用したり、他の販売者の名前を使用したりしないでください。
- 購入者のボキャブラリーからキーワードを選択します。つまり、この種の製品について購入者が考えるときに使用する可能性が高い単語やフレーズを選択します。
- 製品の具体的な特徴にもとづいてキーワードを作成する。
- 製品タイトルをキーワードとして使用しない。製品タイトルは検索のインデックスに登録済みです。

Note

キーワードはソフトウェアカテゴリと同一ではありません。キーワードは、製品に関連する、より具体的な用語です。

の AMI およびコンテナ製品の使用手順の作成 AWS Marketplace

AWS Marketplace 販売者は、さまざまな方法で製品を購入者に配送します。Amazon マシンイメージ (AMIs) とコンテナイメージを使用して配信する場合、ユーザーにはそれらの使用方法を記述する責任があります。以下のセクションでは、これらの製品タイプの使用手順の記述に関する要件と推奨事項について説明します。

詳細については、「[の AMI ベースの製品 AWS Marketplace](#)」を参照してください。

トピック

- [要件](#)
- [リリースノート](#)の記述
- [使用手順](#)の記述
- [アップグレード手順](#)の記述
- [CloudFormation 配信手順](#)の記述

要件

製品の使用手順を作成する際には、以下の情報を含める必要があります。

- 顧客が保存したすべての機密情報の場所。
- すべてのデータ暗号化設定について説明します。アプリケーションが暗号化手法を使用する場合に必要なデータを復号するために、ユーザーがアプリケーションを操作する方法に関する詳細な手順を提供します。
- 製品に暗号化マテリアルが含まれている場合は、使用手順にローテーション要件を含める必要があります。認証情報と暗号化キーを使用するリストの基本要件[the section called “AMI ベースの製品要件”](#)については、「」を参照してください。
- 製品内のデータストアのいずれかが専有である場合は、設定、バックアップ、および復旧に関する step-by-step の手順を提供します。

- アプリケーションの正常性と適切な機能を評価およびモニタリングする手順について説明します。
例：
 - [Amazon EC2 コンソール](#)に移動し、正しいリージョンにいることを確認します。
 - [インスタンス] を選択し、起動しているインスタンスを選択します。
 - サーバーを選択してメタデータページを表示し、ページ下部の [ステータスチェック] タブを選択してステータスチェックに合格したか失敗したかを確認します。
- AWS サービスクォータの管理に関する規範的なガイダンス。詳細については、「[AWS 全般のリファレンスガイド](#)」を参照してください。
- 標準クォータを超えて追加された AWS リソースの実行コストを含む料金内訳。この情報は、製品の使用手順に記載することも、Service Quotas (階層) の管理と増額のリクエストに関する詳細情報を含む[ドキュメント](#)リンクすることもできます。

リリースノートの記述

製品を更新するたびに、変更の内容をリリースノートで説明する必要があります。リリースノートには、ユーザーが更新をインストールするかどうかを判断できる具体的な情報を記載する必要があります。セキュリティ更新には「重大」、または他の種類の更新には「重要」や「オプション」といった、明確なラベルを使用します。

使用手順の記述

購入者がソフトウェアを正常に設定して実行するために役立つ使用手順を提供します。提供した使用手順は、設定プロセス中に表示されます。

効果的な使用手順を記述するには、次のガイドラインに従ってください。

- たとえば、ユーザーは興味はあるが、何も知らないと仮定します。
- ユーザーに必要なすべての情報を提供し、製品を起動および使用してもらいます。これには、構成設定や特別な手順が含まれます。

使用手順の例:

1. 1-Click を使用して製品を起動します。
2. ウェブブラウザを使用して、でアプリケーションにアクセスしますhttps://<EC2_Instance_Public_DNS>/index.html。
3. 次の認証情報を使用してサインインします。

- ユーザー名: user
- パスワード: インスタンス ID (instance_id)

アップグレード手順の記述

製品を旧バージョンからアップグレードする方法について購入者に詳しく説明します。別のインスタンスの作成時にデータと設定を保存する方法についての情報を含めます。アップグレードパスがない場合は、このフィールドを編集して具体的に説明してください。

アップグレード手順の例:

1. **** を実行してから、**** します。
2. *** を実行することによって、プロジェクトで使用されるすべてのプラグインが、バージョン *.* と互換性があることを確認します。互換性がない場合は *** を実行します。
3. *** を実行して、データのバックアップを作成します。

CloudFormation 配信手順の記述

CloudFormation 配信を使用するときは、以下も含める必要があります。

- テンプレートによって作成された各 AWS Identity and Access Management (IAM) ロールと AWS CloudFormation IAM ポリシーの目的
- AWS CloudFormation テンプレートによって作成された各キーの目的と場所
- 複数の要素を含むデプロイにおけるネットワーク設定の詳細
- アプリケーションの起動方法と、デプロイに複数の AWS リソースが含まれている場合に通信するように設定する方法に関する詳細なガイド
- 標準制限を超えて追加された AWS リソースの実行コストを含む料金内訳。AWS サービス制限の管理に関する規範的なガイダンスを提供します。
- すべてのデータ暗号化設定。例: Amazon S3 サーバー側の暗号化、Amazon Elastic Block Store (Amazon EBS) 暗号化、Linux Unified Key Setup (LUKS) など)

検索用に AWS Marketplace 製品を最適化する

AWS Marketplace 販売者は、購入者がより簡単に製品を検索できるように、検索用に製品を最適化できます。検索は、購入者のジャーニーで重要なツールであり、検索によって購入者は自身の二

ズを満たす最適な製品を見つけることができます。AWS Marketplace 顧客の場合、検索は検索エンジン (Google や Bing など) と検索関数の 2 つの主要な場所で行われます AWS Marketplace。このトピックでは、両方の場所で出品を最適化する方法について説明します。

トピック

- [検索エンジンの最適化](#)
- [AWS Marketplace 検索](#)

検索エンジンの最適化

関連するキーワードのランクを上げるように製品詳細ページを最適化することは、検索エンジンを通じてより多くのユニークビジターを製品詳細ページに誘導するために必須です。

製品の詳細ページに対するオーガニックサーチを向上させるために基礎的かつ影響力のある 3 つの主要なページ要素にキーワード、タイトルタグ、H1 見出しタグがあります。

キーワード

キーワードは、検索エンジンの最適化の中心的な要素です。これは、検索エンジンの結果を推進する集中的なクエリにトピックを抽出するためです。ページに最も関連性の高いキーワードを特定するプロセスには、キーワード調査が含まれます。検索エンジン最適化ツールは、キーワード検索ボリューム (Google でキーワードが検索される 1 か月あたりの回数)、現在のランキング、検索傾向、キーワードの競争力、関連するキーワードなどの貴重な情報を提供します。この調査から、プライマリキーワードとセカンダリキーワードを特定できます。

プライマリ検索エンジン最適化キーワードは、ページの主要トピックを表す一意の 1 つの単語またはフレーズである必要があります。このプライマリキーワードは、製品タイトル、簡単な説明、ハイライトセクションのコピーに自然に織り込まれている必要があります。セカンダリキーワードは、残りのページの内容に含まれる関連性の高い用語である必要があります。

タイトルタグ

検索エンジンの結果ページに表示され、ブラウザウィンドウまたはタブのページのタイトルとして表示されるタイトルタグは、ページの内容をリーダーと検索エンジンのウェブクロラーの両方に知らせます。AWS Marketplace 製品詳細ページでは、製品タイトルがタイトルタグとして機能するため、ランキングの可能性を向上させるには、検索エンジン最適化キーワードを使用して製品タイトルを最適化することが重要です。検索エンジンの結果ページで高ランクを達成する可能性を高めるには、ブランド名、製品名、関連するキーワードをタイトルタグに組み込みます。

H1 見出しタグ

H1 見出しタグには 3 つの役割があります。

- これは、訪問者がページの概要を把握し、必要な情報を確認するのに役立ちます。
- これにより、スクリーンリーダーを使用してページの内容を理解する視覚障害のある訪問者のアクセシビリティが向上します。
- これらはページ見出しにあるキーワードを提供します。これにより、次のページコンテンツでサポートされている場合、検索エンジンの最適化の関連性の重みが付加されます。

AWS Marketplace 検索

AWS Marketplace ウェブサイトでは、業界全体で使用されているのと同様の検索最適化手法を使用して、検索クエリの結果をランク付けします。が検索結果を AWS Marketplace ランク付けして返す方法を理解することで、AWS Marketplace 検索エンジン用に最適化された製品の詳細を作成できます。製品詳細ページを作成するときは、このガイダンスを考慮することをお勧めします。

キーワード

製品作成プロセス中に、顧客がサイト検索を通じてお客様の製品を見つける手助けをするため、最大 3 つのキーワード (単語またはフレーズ) を送信できます。キーワードテキストボックスに最大 250 文字を含めることができます。

検索キーワードを作成するために以下のヒントを使用します。

- 顧客が簡単にお客様の製品を見つけられるよう、関連する用語を使用する。
- 顧客のボキャブラリーにあるキーワード、つまり、製品タイプのことを考えるときに使用する可能性が高い単語やフレーズを選択する。
- 製品の具体的な特徴にもとづいてキーワードを作成する。
- 送信する用語に製品タイトルを含めない。製品タイトルは検索用にすでにインデックス化されています。

Note

キーワードはソフトウェアカテゴリと同一ではありません。キーワードは、製品に関連する、より具体的な用語です。

製品を作成した後に、製品のメタデータを編集してキーワードを編集できます。AWS Marketplace 管理ポータルの [製品] タブを使用して作成した製品の場合は、[製品] タブを使用して変更を加えます。詳細については、「[Product changes and updates](#)」を参照してください。

AWS Marketplace Seller Operations チームは、似たような言葉や似た意味の単語でクエリをリダイレクトするのに役立ちます。例えば、顧客が自動車を検索すると予想していたときに、車を検索した場合などです。

ソフトウェアカテゴリ

製品を出品するとき、製品に最大で 3 つのソフトウェアカテゴリと、対応するサブカテゴリを選択できます。これにより、顧客は製品を閲覧または検索するときに製品を見つけることができます AWS Marketplace。製品に関連するカテゴリのみを選択します。多くの場合、該当するのは 1 つのカテゴリのみです。Product Load Form および [製品] ページの両方に、カテゴリの完全なリストが含まれます。

Note

カテゴリはキーワードと同一ではありません。使用可能なカテゴリとサブカテゴリは事前定義されています AWS Marketplace。リストからカテゴリを選択して、製品に適用するカテゴリの種類を決定します。キーワードは事前定義されていません。プロセス中に作成されます。

ハイライトセクション

製品詳細ページでは、最大 3 つの製品ハイライトを箇条書きで表示します。顧客はハイライトで製品を検索できるため、製品を作成するときにハイライトを含めます。ハイライトは、製品の主なセールスポイントを簡潔にわかりやすい言葉で記述する必要があります。

Example [ハイライト]

- 費用の見積もり: AnyCompany では、使用したものに対してのみ支払います。AnyCompany が実行されている 1 時間 (または 1 時間未満) ごとに請求されます。

簡単な説明

製品説明は、製品の特徴、利点、使用方法を一覧表示し、他の関連する特定の製品情報を提供します。製品説明を作成する際は、次のガイドラインに注意してください。

- 必要以上に大文字や句読点を使わない
- リダイレクト情報を含めない
- スpellと文法を確認する
- 重要で有益な情報のみを含める

Example 簡単な説明

AnyCompany の製品は、アプリケーションへの着信トラフィックを複数の Amazon EC2 インスタンスに自動的に分散します。これにより、アプリケーションへの受信トラフィックに応じて必要なロードバランシング容量がシームレスに確保され、アプリケーションの耐障害性を高めることが可能です。AnyCompany の製品はプールに正常ではないインスタンスを検出すると、その異常なインスタンスが回復するまで、自動的にトラフィックを正常なインスタンスに振り分けます。お客様は、単一の AWS アベイラビリティゾーンまたは複数のアベイラビリティゾーンで有効にして、より一貫したアプリケーションパフォーマンスを実現できます。

プロモーションメディアで AWS Marketplace 製品を強化する

AWS Marketplace 販売者は、プロモーションメディアで製品出品を強化することで、顧客が製品をよりよく発見して評価できるように支援できます。プロモーションメディアは、製品ページに目立つように表示される動画や画像で、製品について簡単に知ることができます。以下のセクションでは、プロモーションメディアのベストプラクティスと、() で AWS Marketplace Management Portal プロモーションメディアを追加および管理するためのヒントについて説明します AMMP。

トピック

- [プロモーションメディアのベストプラクティス](#)
- [プロモーションメディアを追加および管理するためのヒント](#)

プロモーションメディアのベストプラクティス

最大 5 つの動画と 10 のイメージがサポートされています。各プロモーションメディア項目には、必要なタイトルとオプションの説明を含める必要があります。説明はメディアの代替テキストとして使用され、視覚的なアクセシビリティと検索エンジンの最適化を向上させるために強く推奨されます。

Note

プロモーションメディアをアップロードまたは追加するには、適切な権限とアクセス許可が必要です。製品に追加されたメディアは、すべてのユーザーが閲覧できるように公開されます AWS Marketplace。

動画

このアセットタイプは、簡潔な概要を通じて製品と会社を紹介する機会を提供します。顧客のインタビュー、見積り、関連する利点、データポイントなどのコンテンツを含めることもできます。さらに、ビデオは、録画されたデモやウォークスルー、特に主要な機能やユースケースで製品を紹介する優れた方法です。短い動画の方が影響が大きいいため、動画の再生時間は 2~5 分にするをお勧めします。

仕様：

- AMMP またはパブリックにアクセス可能な S3 リンクを介した動画の直接アップロードをサポートしています。
- 推奨されるビデオ解像度は 1080p (1920x1080 ピクセル) で、アスペクト比は 16:9 が推奨されます。
- ビデオ形式は .mp4 で、最大ファイルサイズは 50MB である必要があります。
- (オプション) 自動生成されたカバーの代わりにカバーイメージを追加できます。
 - カバーイメージの解像度は 500 x 281 ピクセル (アスペクト比 16:9) である必要があります。
 - イメージ形式は、透明性のない .png (推奨)、.jpg、または .svg である必要があります。

Note

外部でホストされたビデオは直接サポートされていません。パブリック S3 リンクは追加時によって AWS Marketplace キャッシュされ、追加後の変更はメディアが再度追加されるまで反映されません。

YouTube 動画は、限られた機能でサポートされています。

- 製品動画 URL フィールド (プロモーションメディアとは別のフィールド) に YouTube 動画への直接リンクが含まれている場合、埋め込みが動画に対して有効になっていると、動画は最後のプロモーションメディアアイテムとして直接埋め込まれます。
- 1 つの製品ビデオ URL のみがサポートされ (つまり、1 つの YouTube ビデオのみ)、YouTube 埋め込みメディアの順序変更はサポートされていません。

イメージ

このアセットタイプは、スクリーンショットと主要な概念のコールアウトを通じて製品を宣伝する機会を提供します。イメージは、簡潔な図またはフローチャートを使用して複雑な製品機能を説明するのに最適です。さらに、イメージを使用して、料金ディメンションの説明では対処できない料金階層とグループ化を説明できます。

仕様：

- イメージの直接アップロードは、を通じて、AMMPまたはパブリックにアクセス可能なS3リンクを通じてサポートされています。
- 推奨される画像解像度は 780 x 439 ピクセルで、アスペクト比は 16:9 が推奨されます。最大 3480x3480 ピクセル。
- イメージ形式は、透明性のない .png (推奨)、.jpg、または .svg である必要があります。
- サムネイルは元のソースイメージから自動生成されます。

プロモーションメディアを追加および管理するためのヒント

メディアの追加

- プロモーションメディアは、にログイン[AWS Marketplace Management Portal](#)し、新しい製品を作成するか、既存の製品を更新することで追加できます。画像と動画セクションは、プロモーションメディアの追加に使用され、製品情報エリア内にあります。
- はプロモーションメディアの追加APIにも使用できますが、直接アップロードは AMMP を通じてのみサポートされます。パブリックS3リンクは、APIと の両方でサポートされていますAMMP。
- メディアを表示する順序でプロモーションメディアを追加します。最初の項目は、製品ページの注目のメディアです。
- イメージアセットタイプを使用する場合、オプションのカバーイメージは使用できません。イメージが追加されると、サムネイルが自動生成されます。

メディアの更新または削除

- プロモーションメディアが製品に追加されると、製品を編集することで、タイトル、説明、カバー (利用可能な場合) を必要に応じて調整できます。
- プロモーションメディアは、送信前に AMMP でそのアセットの削除をクリックして追加すると削除できます。

メディアの注文

- デフォルトでは、新しいプロモーションメディアアセットがアセット注文の最後に追加されます。
- メディアの順序を変更するには、順序が間違っているアセットを削除し、送信する正しい順序で再度追加します。

AWS Marketplace 製品のプライベートオファーの準備

プライベートオファーは、製品の購入に使用される交渉済みの用語です AWS Marketplace。これには、カスタム料金プラン、エンドユーザーライセンス契約 (EULA)、またはカスタムソリューションが含まれる場合があります。販売者と購入者は、パブリックオファーとは異なるプライベートオファーにコミットする前に交渉します。複数のプライベートオファーを作成して、1 人の購入者に提供することができます。プライベートオファーが適用された購入者は、プライベートオファーとパブリックオファーのどちらかを選択できます。購入者が一度に購読できるオファーは 1 つのみです。プライベートオファーとパブリックオファーの両方を同時に購読することはできません。このトピックでは、特別な考慮事項、購入者エクスペリエンス、販売者レポートなど、プライベートオファーの仕組みについて説明します。

Note

AWS は、で購入者に一意のユースケースまたはエンタープライズユースケースを提供し AWS Marketplace、製品の詳細ページから直接製品のプライベートオファーをリクエストします。APN カスタマーエンゲージメント (ACE) の対象となる AWS パートナーネットワーク (APN) パートナーで、このオプションを購入者に提供する場合は、[プライベートオファーとデモリクエストボタンの追加](https://aws.amazon.com/partners/programs/ace/)「」を参照してください。 <https://aws.amazon.com/partners/programs/ace/>

トピック

- [プライベートオファーの仕組み](#)
- [プライベートオファーに関する考慮事項](#)
- [プライベートオファーの購入者エクスペリエンス](#)
- [プライベートオファーのレポート](#)
- [AWS Marketplace プライベートオファーでサポートされている製品タイプ](#)
- [プライベートオファーの作成と管理](#)
- [AWS Marketplace チャンネルパートナーとしてのプライベートオファーの作成](#)
- [プライベートオファーの分割プラン](#)
- [プライベートオファーの将来の日付の契約の作成](#)
- [プライベートオファーに関するよくある質問](#)

プライベートオファーの仕組み

プライベートオファーは、[AWS Marketplace 管理ポータル](#) の オファーページから作成および管理できます。オファー製品を指定して、一意の ID と URL を生成します。プライベートオファーの料金プランを作成し、法的条件と販売ドキュメントを追加し、オファーを特定の購入者 AWS アカウントに拡張します。オファーは、オファーを作成したアカウントにのみ表示されます。

プライベートオファーを作成し、購入者になる可能性がある相手に通知すると、相手はオファーを表示して受け入れることができます。オファーを表示するには、購入者がオファー AWS アカウントを受け取った にサインインする必要があります。

Note

リンクされたアカウントまたは管理者アカウントにオファーを拡張しない限り、購入者はオファーを表示できません。オファーでサービスの制限を強制することはできないため、製品に制限がない限り、購入者は交渉された価格で製品を好きなだけ使用できます。

プライベートオファーの作成については、「[Creating and managing private offers](#)」を参照してください。

プライベートオファーは販売者レポートで追跡されます。詳細については、「[Reporting for private offers](#)」および「[Seller reports guide](#)」を参照してください。

プライベートオファーに関する考慮事項

プライベートオファーを利用する際は、以下の点を考慮してください。

- 新しいインスタンスタイプまたは のサポートを追加すると AWS リージョン、既に製品のプライベートオファーをサブスクライブしているお客様は、新しく追加されたインスタンスまたはリージョンに自動的にアクセスできなくなります。お客様がアクセスするインスタンスとリージョンを使用して、別のプライベートオファーを作成する必要があります。新しいオファーを承諾すると、お客様は新しく追加されたインスタンスとリージョンにアクセスできます。将来の日付に製品をサブスクライブするお客様は、プライベートオファーに含まれている限り、その製品にアクセスすることもできます。新しいプライベートオファーを作成する方法の詳細については、「」を参照してください [での契約の修正 AWS Marketplace](#)。

- セカンドパーティー、Amazon マシンイメージ (AMI) の毎月のプライベートオファー、製品を使用したmulti-AMI-based配信 AWS CloudFormation、または顧客の使用制限のためのプライベートオファーを作成することはできません。
- 分割払いプランのプライベートオファーでは、前払いを時間の経過とともに複数の支払いに分割できます。詳細については、「[プライベートオファーの分割プラン](#)」を参照してください。
- プライベートオファーの購入者アカウントがプライベートマーケットプレイスで管理されている場合は、購入者のアカウントと、プライベートマーケットプレイス管理者が含まれるアカウントの両方をオファーに含める必要があります。
- プライベートオファーは、Bring-Your-Own-License (BYOL) はサポートしていません。
- プライベートオファーを作成する場合は、[カスタム EULA] オプションを使用して、独自の交渉による契約条件をプライベートオファーに含めてください。最大 5 つのドキュメントをアタッチできます。
- Software as a Service (SaaS) 契約製品および従量課金制の SaaS 契約製品については、購入者がプライベートオファーを承諾して締結した契約に対してアップグレードや更新を提供できます。これにより、新しい権利の付与、料金割引の提供、支払いスケジュールの調整、またはエンドユーザーライセンス契約 (EULA) の変更を行い、標準化されたライセンス条項を使用できます。詳細については、「[での契約の修正 AWS Marketplace](#)」を参照してください。

プライベートオファーの購入者エクスペリエンス

顧客が購入者のサブスクリプションページに移動すると、プライベートオファーが利用可能であることを示すバナーが表示されます。購入者がオファーを承諾すると、すべての AWS Marketplace トランザクションに使用したのと同じポータルツールを使用して購入に対して請求されます。承諾されたオファーは契約になります。購入者は の「サブスクリプションの管理」セクションで契約の詳細を検索でき AWS Management Console、販売者は の「契約」タブで詳細を確認できます AWS Marketplace 管理ポータル。

AWS Marketplace 購入者は、プライベートオファーのサードパーティーの資金調達にアクセスできます。詳細については、「[Customer Finance is now available in AWS Marketplace](#)」を参照してください。

Note

オファーは有効期限が切れる前にのみ受け入れることができます。オファーの有効期限が切れると、[受け売れたオファーと期限切れのオファー] タブに移動します。

プライベートオファーを表示して受け入れるには	購入者は次のことができます。
AWS Marketplace コンソールから	AWS Marketplace コンソールでプライベートオファーに移動し、利用可能なオファータブからオファー ID を選択します。 プライベートオファーの購入者体験の詳細については、「AWS Marketplace 購入者ガイド」の「 プライベートオファー 」を参照してください。
販売者が提供したリンクを使用する	販売者から送られたリンクに従って、プライベートオファーに直接アクセスします。 詳細については、「 購入者へのプライベートオファーの送信 」を参照してください。
製品ページから	製品の製品ページに移動して、バナーのリンクを選択しプライベートオファーを表示します。 プライベートオファーの購入者体験の詳細については、「AWS Marketplace 購入者ガイド」の「 プライベートオファー 」を参照してください。

プライベートオファーのレポート

プライベートオファーは、既存の販売者レポートとオファーに関連するレポートに表示されます。[月別請求済み収益レポート](#) が毎月生成され、オファーの可視性とオファー ID 情報が記載されます。購入者の請求書が生成されたときに、該当する請求期間をカバーするレポートに表示されます。ダッシュボードの詳細については、「[Seller dashboards](#)」を参照してください。

[Offer ID (オファー ID)] フィールドには、プライベートオファー用に生成された一意のオファー ID が記載されています。このレポートのエントリがプライベートオファー用でない場合、このフィールドは空白になります。[Offer Visibility (オファーの可視性)] フィールドには、レポートのエントリがパブリックオファーかプライベートオファーかが表示されます。プライベートオファーの場合はすべて、エントリがプライベートと記載されます。

AWS Marketplace プライベートオファーでサポートされている製品タイプ

AWS Marketplace 販売者は、プライベートオファーを使用して、Amazon マシンイメージ (AMIs)、コンテナ、プロフェッショナルサービス、機械学習 (ML)、Software as a Service (SaaS) 製品を販売できます。プライベートオファーは、製品の購入に使用される交渉された用語です AWS Marketplace。このトピックでは、AMI、コンテナ、SaaS、ML、プロフェッショナルサービス製品のプライベートオファーについて説明します。

詳細については、「[AWS Marketplace 製品のプライベートオファーの準備](#)」を参照してください。

トピック

- [AMI 製品のプライベートオファー](#)
- [コンテナ製品のプライベートオファー](#)
- [プロフェッショナルサービス製品のプライベートオファー](#)
- [SaaS 製品のプライベートオファー](#)
- [ML 製品のプライベートオファー](#)

AMI 製品のプライベートオファー

AMI 製品のプライベートオファーの料金を提供できます。

オファーは、以下の期間であれば自由にカスタマイズできます。

- AMI (時間単位) または年間プライベートオファー付きの AMI (時間単位): 最長 3 年間 (1,095 日)。柔軟な支払いスケジュールをサポートしているのは、年間プライベートオファー付きの AMI (時間単位)、または AMI 契約のみです。
- AMI 契約のプライベートオファー: 最長 5 年 (60 か月)

AMI 契約の場合、プライベートオファーによって使用状況が監視されることはありません。

購入者はいつでも手動で新しい契約レベルにアップグレードできます。ただし、契約階層を定義して、サービス制限を適用し、購入者に手動で契約をアップグレードしユニット数を増やすよう勧めるのは、独立系ソフトウェアベンダーの責任です。現時点でアップグレードをサポートしているのは、段階的でない価格ベースの契約のみです。プライベートオファーの契約期間は、公開されている製品リスト、または月単位 (最大 60 か月) のカスタム期間と一致する場合があります。

ライセンス使用権限は、購入者がプライベートオファーを承諾した日付から始まります。

柔軟な支払いスケジュールを持つ AMI プライベートオファーでは、契約期間中、契約で合意した年間インスタンスタイプの数を設定できます。

 Note

プライベートオファーは月単位の請求契約ではご利用いただけません。

コンテナ製品のプライベートオファー

コンテナベースの製品契約には、プライベートオファー価格を提供できます。

オファーは、以下の期間であれば自由にカスタマイズできます。

- コンテナ (時間単位) または長期プライベートオファー付きのコンテナ (時間単位) - 最長 3 年間 (1,095 日)。柔軟な支払いスケジュールをサポートしているのは、年間プライベートオファー付きのコンテナ (時間単位)、またはコンテナ契約のみです。
- コンテナ契約のプライベートオファー: 最長 5 年 (60 か月)

コンテナ契約の場合、プライベートオファーによって、使用状況が監視されることはありません。コンテナ契約のアップグレードは段階的でない価格を使用している場合にのみ可能です。

購入者はいつでも手動で新しい契約レベルにアップグレードできます。ただし、独立系ソフトウェアベンダーは契約階層を定義して、サービス制限を強制し、購入者に手動で契約をアップグレードしてユニット数を増やすよう勧めます。現時点でアップグレードをサポートしているのは、段階的でない価格ベースの契約のみです。プライベートオファーの契約期間は、公開されている製品リスト、または月単位 (最大 60 か月) のカスタム期間と一致する場合があります。

ライセンス使用権限は、購入者がプライベートオファーを承諾した日付から始まります。柔軟な支払いスケジュールを持つコンテナプライベートオファーでは、契約期間中、契約で合意したユニットの数を設定できます。また、購入者がそれ以上を使用する場合に、同じユニットに対してカスタムの時間料金を定義することもできます。

 Note

プライベートオファーは月単位の請求契約ではご利用いただけません。

プロフェッショナルサービス製品のプライベートオファー

プロフェッショナルサービス製品の提供はすべてプライベートオファーを通じて行われます。詳細については、「[プライベートオファーを作成する](#)」を参照してください。

SaaS 製品のプライベートオファー

Software as a Service (SaaS) のプライベートオファー製品は、特定の価格帯の価格レベルをタイミングに基づいて変更することはできません。例えば、あるオファーで 3 か月間は 1 時間あたり 0.80 USD を請求し、その後、同じ価格帯で 1 時間あたり 0.60 USD に変更することはできません。SaaS 契約の場合、プライベートオファーによって、使用状況が監視されることはありません。

購入者はいつでも手動で新しい契約レベルにアップグレードできます。ただし、独立系ソフトウェアベンダーは契約階層を定義して、サービス制限を強制し、手動で契約をアップグレードして必要に応じて契約階層を上げるよう購入者に勧めます。プライベートオファーの契約期間は、公開されている製品リスト、または月単位 (最大 60 か月) のカスタム期間と一致する場合があります。

ML 製品のプライベートオファー

機械学習 (ML) プライベートオファー商品では、特定の購入者に、公開されている価格とは異なる価格を提示します。プライベートオファーにおける出品者と購入者との間の一連の条件と契約は、パブリックオファーや他のプライベートオファーの条件と異なる場合があります。

プライベートオファーは、以下のいずれかの方法を使用します。

- 時間単位 - プライベートオファーは、公開されている時間単位の料金とは異なる時間単位の場合があります。機械学習製品のプライベートオファーには有効期限がないため、この時間単位の料金は永続的です。将来料金の変更が必要な場合は、購入者は新しいプライベートオファーに切り替える必要があります。製品で実行中の既存のインスタンスまたはエンドポイントには、新しい承認されたオファーで設定された時間単位の料金が自動的に請求されます。プライベートオファー内の契約コンポーネントの有効期限が切れたら、製品の時間単位の料金に設定してください。この時間レートを 0 USD に設定すると、購入者はソフトウェア料金なしで製品を無期限に使用できます。
- 推論単位 - 製品をエンドポイントとしてデプロイする場合の[推論価格](#)を設定している場合、プライベートオファーの推論レートは、公開されている推定レートとは異なる場合があります。
- 契約 - プライベートオファーは、指定された日数の固定前払い料金が適用される契約にできます。購入者は、契約期間中、インスタンス数の制限なく使用できます。契約の終了時に、継続して実行するすべてのインスタンスには、プライベートオファーで設定した時間単位の料金で請求されます。例えば、固定前払い料金で 365 日間無制限に使用できる契約を作成できます。プライベートオファーには時間単位の料金も設定できます。購入者がこのプライベートオファーを受け入れ

ると、購入者はその前払い料金を支払うことになります。契約が終了すると、まだ実行中のインスタンスはすべてその時間単位で請求されます。無料のプライベートトライアルを提供している場合は、無料の永久ライセンスが利用されないように、無料トライアル期間終了後の正しい時間単価を設定していることを確認してください。

複数のプライベートオファーを作成して、1人の購入者に提供することができます。プライベートオファーが適用された購入者は、プライベートオファーとパブリックオファーのどちらかを選択できます。購入者が一度に購読できるオファーは1つのみです。プライベートオファーとパブリックオファーの両方を同時に購読することはできません。

SageMaker 製品の特定の購入者のプライベートオファーを作成するには、「」を参照してください [the section called “プライベートオファーの作成”](#)。

プライベートオファーの作成と管理

AWS Marketplace 販売者は、プライベートオファーを作成および管理できます。プライベートオファーは、製品の購入に使用される交渉済みの用語です AWS Marketplace。これには、カスタム料金プラン、エンドユーザーライセンス契約 (EULA)、またはカスタムソリューションが含まれる場合があります。以下のセクションでは、プライベートオファーを作成して管理する方法を説明します。

Note

プライベートオファーを発行する資格を得るには、少なくとも1つのアクティブなパブリックリストが必要です。公開リストがあり、プライベートオファータブにアクセスできない場合は、[「IAM アクセス許可」](#)を参照するか、[サポートにお問い合わせください AWS Marketplace](#)。

トピック

- [新しいプライベートオファーを開始する](#)
- [オファーのステータスについて](#)
- [プライベートオファーの草案作成と公開](#)
- [プライベートオファーとデモリクエストボタンの追加](#)
- [購入者へのプライベートオファーの送信](#)
- [プライベートオファーのクローン作成](#)
- [オファーの詳細のダウンロード](#)

- [プライベートオファーの進行状況を保存する](#)
- [プライベートオファーの有効期限の更新](#)
- [プライベートオファーをキャンセルする](#)

新しいプライベートオファーを開始する

以下のプロセスを使用してオファーを作成し、CreateOffer API 変更リクエストを使用してオファー ID を生成します。ドラフト状態で空白のオファーを作成します。

新しいプライベートオファーの作成を開始する

1. [AWS Marketplace 管理ポータル](#) にサインインして [オファー] を選択します。
2. [オファー] ページで、[オファーを作成] を選択します。
3. [オファーを作成] ページで、プライベートオファーを作成する製品タイプと製品を選択します。処理には最大 30 秒かかります。処理が完了するまで、ページを閉じたり更新したりしないでください。

Note

オファーの作成後に製品タイプと製品を変更することはできません。製品タイプごとのプライベートオファーの詳細については、「[サポートされている製品タイプ](#)」を参照してください。

AWS Marketplace チャンネルパートナーの場合は、再販認可から独自の製品のオファーを作成するか、AWS Marketplace チャンネルパートナープライベートオファー (CPPO) を作成するかを選択します。CPPO の場合は、独立系ソフトウェアベンダー (ISV)、製品、認可を選択します。

4. [オファー詳細に移動] を選択します。段階的なエクスペリエンスが開くので、プライベートオファーの作成を続行します。

オファーのステータスについて

オファーには、ライフサイクルに応じて 3 つのステータスのいずれかが割り当てられます。

- 草案 – オファーは不完全であり、まだ準備中の状態です。ドラフトステータスのプライベートオファーは、保持スケジュールの対象ではありません。オファーを公開して購入者に提供するには、必要なすべての詳細を完了して送信する必要があります。

- アクティブ – オファーは公開され、購入者に提供された状態です。オファーの有効期限が切れていないため、購入者はオファーをサブスクライブできます。
- 期限切れ – オファーは公開され、購入者に提供された状態です。オファーの有効期限が切れているため、購入者はオファーをサブスクライブできません。有効期限を更新して、購入者がオファーを承認するまでの時間を増やします。オファーの有効期限を更新するには、「[プライベートオファーの有効期限の更新](#)」を参照してください。

 Note

オファーが承認されると、[契約] タブに契約が表示されます。オファーのステータスは変更されません。

プライベートオファーの草案作成と公開

以下のプロセスを使用して、プライベートオファーの草案と公開を行います。

プライベートオファーを起草して公開するには

1. [オファー情報の提供] ページで、オファー名、オファーの詳細、更新タイプ、オファーの有効期限を指定します。更新オファーの場合は、「」で作成された既存の契約を更新するための既存顧客 AWS Marketplace AWS Marketplace 「」、または「既存の顧客を移行するための更新 AWS Marketplaceのための既存顧客」のいずれかを選択する必要があります AWS Marketplace。

 Note

オファーの有効期限は、オファーが null になり無効になった日付です。この日付の 23:59:59 UTC 以降、購入者はこのプライベートオファーを表示および承認できなくなります。

2. [次へ] を選択します。
3. [オファーの料金と期間の設定] ページで、料金モデル、契約期間または使用期間、料金、通貨、支払いスケジュールを選択します。分割プランがある料金モデルについては、「[分割プラン](#)」を参照してください。

 Note

USD 以外のプライベートオファーは、契約価格商品に限定されます。また、USD 以外の支払いが設定されていることを確認してます。詳細については、「[the section called “支払い設定”](#)」を参照してください。

すべてのパブリックオファーと消費料金によるプライベートオファーは、USD でのみ作成できます。

4. 購入者の追加ページで、プライベートオファーを拡張する AWS Marketplace 購入者ごとに AWS アカウント ID を指定します。選択した各購入者には、選択したオファー通貨 AWS リージョン がサポートされている AWS アカウント に が必要です。別の AWS アカウント ID を追加するには、別の購入者を追加するを選択します。各プライベートオファーには最大 24 人の購入者を追加できます。
5. [次へ] を選択します。
6. [法的条件とオファードキュメントを設定] ページで、次のいずれかのオプションを選択します。
 - パブリックオファーエンドユーザーライセンス契約 (EULA) – パブリックオファーの EULA を使用します。
 - AWS Marketplace (SCMP) の標準契約 – が提供する標準契約を使用します AWS Marketplace。
 - カスタム法的条件 – 法的条件、作業明細書、部品表 (Bill of Materials)、料金表、その他の付録など、プライベートオファーに関連するファイルを最大 5 つアップロードします。これらのファイルは、オファーの作成時に 1 つのドキュメントにまとめられます。
7. [確認して作成] ページで、プライベートオファーの詳細を確認します。確認したら、[オファーを作成] を選択してオファーを公開し、選択した購入者に提供します。オファーの公開には AWS Marketplace Catalog API へのリクエストが含まれているため、オファーの検証と処理に最大 1 時間かかる場合があります。このリクエストは、[リクエスト] ページで表示できます。

 Note

オファーは、リクエストが成功した場合にのみ公開および提供されます。リクエストが失敗した場合、顧客には提供されません。失敗とは、システムエラーが発生したか、再送信する前に修正する必要があるエラーが発生したことを意味します。

プライベートオファーとデモリクエストボタンの追加

販売者は、製品詳細ページにcall-to-actionボタンを追加できます。ボタンを使用すると、購入者はプライベートオファーとガイド付き製品デモをリクエストできます。製品詳細ページに 1 つまたは両方のボタンを追加できます。

ボタンは、次の製品タイプで使用できます。

- Amazon マシンイメージ
- Software as a service (SaaS)
- コンテナ
- CloudFormation テンプレート

ボタンを使用するには、APN カスタマーエンゲージメントプログラム (ACE) に属している必要があります。購入者がオファーまたはデモをリクエストすると、問い合わせデータをフォームに入力し、詳細をリクエストします。次に AWS、需要生成チームはリクエストを認定し、それらの認定されたリクエストを Partner Central の ACE を通じて AWS 発信された機会として転送します。次に、顧客とフォローアップしてオファーの詳細について話し合ったり、ガイド付きデモをスケジュールしたりします。ACE の詳細については、[APN カスタマーエンゲージメントプログラム](#)のウェブサイトと、APN カスタマーエンゲージメント (ACE) に関するよくある質問の「[リードと機会](#)」を参照してください。 FAQs

以下のトピックのステップでは、製品の詳細ページにボタンを追加する方法について説明します。

トピック

- [ボタンの前提条件](#)
- [ボタンの有効化](#)

ボタンの前提条件

製品詳細ページcall-to-actionボタンを追加する前に、次の前提条件が必要です。

- AWS Partner Central で AWS 紹介されたリードと機会を受け取れることを確認します。詳細については、[APN カスタマーエンゲージメントプログラム](#)のウェブサイトを参照してください。

Note

ACE プログラムに登録すると、ステータスの更新は 2 週間ごとに行われます。プライベートオファーとデモリクエストボタンは、ステータスの更新が完了した後にのみ表示されます。これらのオプションを選択してメッセージを受信した場合、アクセスは次の隔週更新を保留中です。

- AWS Partner Central と AWS Marketplace アカウントをリンクします。これを行うには、以下を行う必要があります。
- IAM: CreatePartnerCentralCloudAdminRole ポリシーを作成します。詳細については、[「Partner Central 入門ガイド」の「アカウントリンクの前提条件AWS」](#)を参照してください。
- AWS Partner Central と AWS Marketplace アカウントをリンクします。詳細については、[AWS「Partner Central 入門ガイド」の「Partner Central アカウントを自分の AWS Marketplace アカウントにリンクする」](#)を参照してください。AWS

AWS Partner Central と AWS Marketplace アカウントをリンクすると、Partner Central ホームページに次のステータスメッセージが表示されます。

AWS Marketplace account linking	
Status	AWS account ID
✔ Account linked	933780032658

詳細については、Partner Central にサインインし、以下を参照してください。

- [AWS Partner Central & Marketplace アカウントリンクガイド](#)
- [AWS Partner および Marketplace アカウントリンクデモビデオ](#)

Note

これらのリソースを使用するには、サインインする必要があります。

ボタンの有効化

ACE が AWS 紹介を受け取る資格を得たら、AWS Marketplace 管理ポータルを使用して、一方または両方のcall-to-actionボタンを有効にします。

新しい製品リストを作成するか、現在のリストを更新するかに応じて、個別のプロセスに従ってボタンを有効にします。

新製品のボタンを有効にするには

1. を使用して、次のタイプの製品[AWS Marketplace 管理ポータル](#)を作成し、公開します。
 - AMI
 - SaaS
 - コンテナ
 - Cloud Front テンプレート
2. 製品の作成の一環として、ガイド付きデモとプライベートオファーリクエストで、購入者のガイド付きデモリクエストを有効にすると購入者のプライベートオファーリクエストを有効にするの任意の組み合わせを選択します。

Note

ボタンは、製品を公開した後、プライベートオファターの製品詳細ページにのみ表示されます。

既存の製品のボタンを有効にするには

1. [AWS Marketplace 管理ポータル](#)の「製品」タブで、変更する製品を選択します。
2. リクエスト変更リストを開き、製品情報の更新を選択します。
3. 購入者のガイド付きデモリクエストを有効にすると、購入者のプライベートオファーリクエストを有効にするの任意の組み合わせを選択します。

ボタンは、変更を保存した後にのみ製品詳細ページに表示されます。

購入者へのプライベートオファーの送信

プライベートオファーが公開されると、購入者は AWS Marketplace 管理ポータル の [プライベートオファー] ページにある [ご利用可能なプライベートオファー] タブで、オファーを確認できます。利用可能なプライベートオファータブで、購入者は登録販売者列で AWS Marketplace チャンネルパートナーによって拡張されたオファーを確認できます。独立系ソフトウェアベンダー (ISV) が [パブリッシャー] 列に表示されます。購入者はオファーリストで適切な [オファー ID] を選択すると、プライベートオファーに移動できます。

購入者は、[承諾済みまたは有効期限切れのオファー] タブで承認済みまたは有効期限切れのオファー ID を確認できます。

また、プライベートオファーが公開されると、そのオファーの [フルフィルメント] ページの URL を購入者に送信できます。

プライベートオファーを購入者に送信する

1. [AWS Marketplace 管理ポータル](#) にサインインして [オファー] を選択します。
2. 結果の横にあるラジオボタンを選択します。
3. [アクション]、[オファー URL をコピー] の順に選択します。
4. URL を購入者に送信します。

プライベートオファーのクローン作成

AWS Marketplace チャンネルパートナーのプライベートオファーなど、プライベートオファーのクローンを作成できます。クローン作成を使用して、テンプレートを使用して新しいオファーを作成するか、既存のオファーを更新して置き換えます。

プライベートオファーのクローンを作成するには

1. [AWS Marketplace 管理ポータル](#) にサインインして [オファー] を選択します。
2. オファー テーブルで、クローンを作成するオファーの横にあるオプションを選択します。
3. オファーのクローンを選択します。
4. 新しいオファー作成エクスペリエンスが開き、選択したオファーの情報が事前に入力されます。必要に応じてオファーの詳細を確認して変更します。
5. (オプション) クローンを作成して既存のオファーを置き換える場合は、既存のオファーをキャンセルを選択します。選択すると、元のオファーは自動的に期限切れになり、この新しいオ

ファーが公開されると購入者はアクセスできなくなります。これはオファーのアクセシビリティにのみ影響し、購入者が元のオファーを既に承諾している場合、既存のサブスクリプションには影響しません。

6. プライベートオファーのクローンを選択します。これにより、オファーが公開され、以前に選択した購入者に拡張されます。

オファーの詳細のダウンロード

.pdf ファイルにオファーの詳細をダウンロードするには、次の手順に従います。

オファーの詳細をダウンロードするには

1. [AWS Marketplace 管理ポータル](#) にサインインして [オファー] を選択します。
2. オファーテーブルで、オファーの横にあるオプションを選択し、詳細の表示を選択します。または、オファー ID 列でオファーのリンクを選択することもできます。
3. オファーの詳細ページで、PDF のダウンロードを選択します。

プライベートオファーの進行状況を保存する

以下のプロセスを使用して進行状況を保存し、後で再開します。

作業を保存して再開するには

1. 完了した任意のステップで、[保存して終了] を選択します。ダイアログボックスで、コンテンツを草案状態で保存し、検証エラーを確認します。検証エラーがあったり、詳細が欠落している場合は、[修正] を選択してステップに移動し、問題を解決します。準備が整ったら、[保存して終了] を選択して変更を保存します。

[保存して終了] すると、リクエストは処理中にレビューされます。処理の終了には、数分または数時間かかることがあります。リクエストが成功するまでステップを続行したり、リクエストを修正できません。リクエストが成功したら、保存は完了です。リクエストが失敗した場合、システムエラーまたは再送信する前に修正する必要があるエラーがあることを意味します。

2. オファーの作業を再開するには、[オファー] ページを開き、オファーを選択し、[オファーの作成を再開] を選択します。
3. 完了したら、[保存して終了] して進捗状況を保存するか、[オファーを作成] を選択して、プライベートオファーを公開して選択した購入者に提供するかを選択できます。

プライベートオファ어의有効期限の更新

プライベートオファ어의有効期限を更新するには、次のプロセスを使用します。

プライベートオファ어의有効期限を更新する

1. [AWS Marketplace 管理ポータル](#) にサインインして [オファ어] を選択します。
2. [オファ어] ページで、更新するオファ어を選択します。
3. [編集] を選択します。
4. 新しいオファ어의有効期限を指定します。
5. [Submit] を選択してください。

更新が完了すると、オファ어は [アクティブ] ステータスに変更され、購入者はオファ어를承認できます。

プライベートオファ어를キャンセルする

プライベートオファ어를キャンセルするには、次のプロセスを使用します。

1. [AWS Marketplace 管理ポータル](#) にサインインして [オファ어] を選択します。
2. [オファ어] ページで、更新するオファ어を選択します。

Note

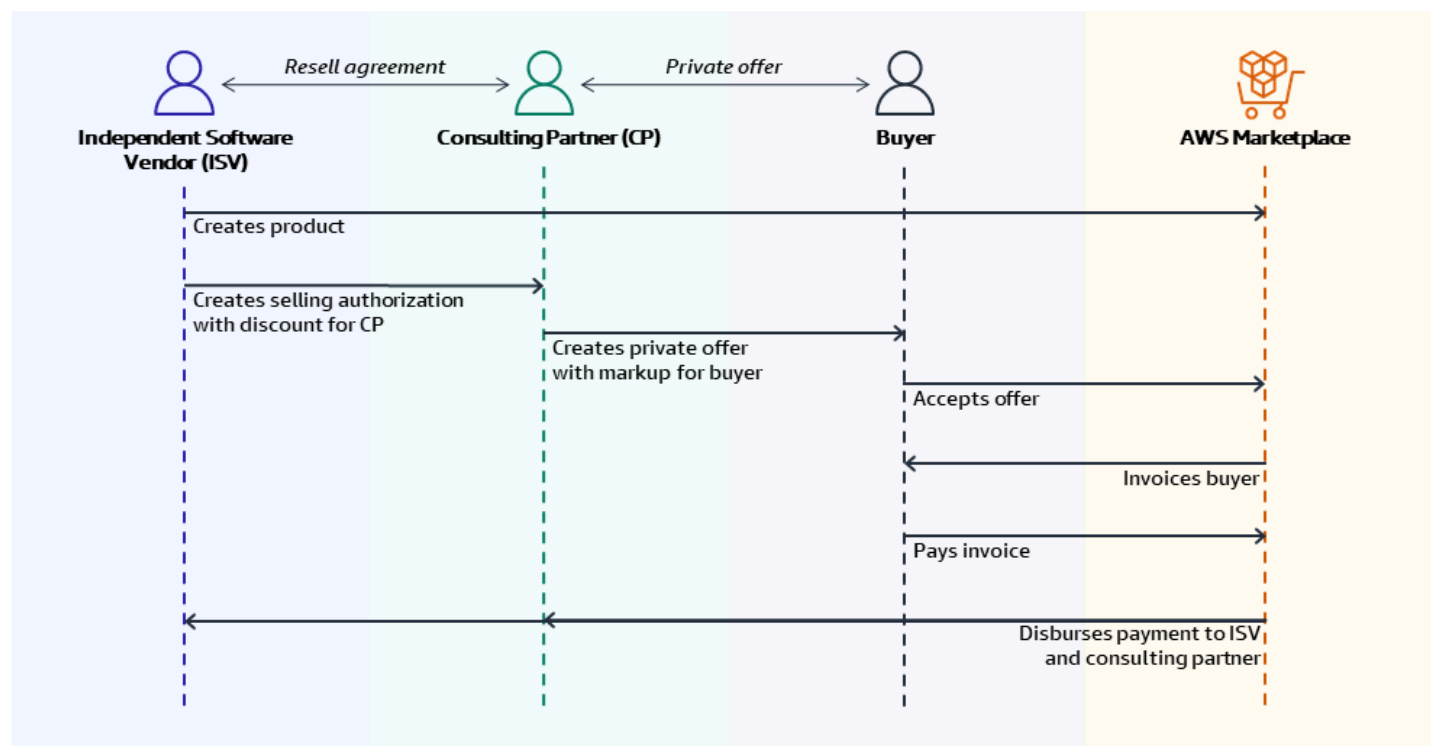
オファ어를キャンセルするとオファ어의有効期限が変更されるため、このオファ어를延長した購入者には有効期限が切れたオファ어として表示されます。

3. [アクション]、[オファ어를キャンセル] の順に選択します。

AWS Marketplace チャネルパートナーとしてのプライベートオファ어의作成

AWS Marketplace チャネルパートナーのプライベートオファ어は、チャネルパートナーが独立系ソフトウェアベンダー (ISVs) 製品を再販する機会を提供します AWS Marketplace。AWS Marketplace チャネルパートナーと ISV は、ISV の製品を 1 つ以上再販する契約を結び、チャネルパートナーはその製品の購入者にプライベートオファ어를延長します。

次の図は、ISV、チャネルパートナー、購入者の関係を示しています。



Note

ISV であるチャネルパートナーに販売認証を作成する方法の詳細については、「[ISV としての AWS Marketplace チャネルパートナーの販売承認の作成](#)」を参照してください。

各 AWS Marketplace チャネルパートナーのプライベートオファーは 1 人の購入者にのみ表示され、その購入者のニーズを満たすためにカスタマイズされた料金と固有の取引条件が適用されます。プライベートオファーを作成する場合は、ISV が設定した卸売価格から開始します。次に、その価格を割り増しして購入者のオファー価格を作成します。

Note

プライベートオファーを作成する場合、チャネルパートナーは ISV が販売承認で定義する通貨を使用する必要があります。

卸売コストは、次のいずれかの方法で決定します。

- 経常割引 - ISV は、AWS Marketplace チャンネルパートナーが、経常販売認証を用いて定価から合意された割引価格で製品を再販することを認可します。AWS Marketplace チャンネルパートナーは、この割引を活用して、ISV との価格交渉を行わずに製品の再販を継続できます。この割引は、指定した日付まで有効になるように設定することも、ISV またはチャンネルパートナーが終了するまで無期限に継続するように設定することもできます。
- 非定期割引 - ISV が AWS Marketplace チャンネルパートナーに付与する販売認可は、特定の購入者のみで使用することを目的とした 1 回限りの割引です。

いずれの場合も、購入者がプライベートオファーに対して支払いを行った後、は標準プロセス AWS Marketplace を使用して、合意された料金に基づいて AWS Marketplace チャンネルパートナーと ISV に資金を分配します。

Tip

ISVs とチャンネルパートナーは、のパートナーメニューを使用して販売承認 [AWS Marketplace 管理ポータル](#) を表示できます。

プライベートオファーの作成に関する詳細な手順については、[AWS Marketplace 「チャンネルパートナーのプライベートオファー - オファーの作成」](#) を参照してください。

プライベートオファーの第三者融資の詳細については、「[顧客向け融資が AWS Marketplace で利用可能になりました](#)」を参照してください。

追加情報

追加情報と質問については、ISVs とチャンネルパートナーが AWS Marketplace チャンネルチームに接続することをお勧めします。具体的な問い合わせ先が不明な場合は、aws-mp-channel@amazon.com まで E メールメッセージを送信してください。1 営業日以内にご連絡を差し上げます。

ISV としての AWS Marketplace チャンネルパートナーの販売承認の作成

独立系ソフトウェアベンダー (ISV) として、そのパートナーの販売承認を作成することで、AWS Marketplace チャンネルパートナーが製品を再販することを許可できます。チャンネルパートナーは、販売承認を使用して、最終購入者へのチャンネルパートナープライベートオファー (CPPO) を作成できます。サポートされている製品タイプは次のとおりです。

- AMI ベースの製品

- コンテナベースの製品
- SaaS ベースの製品
- プロフェッショナルサービス製品

次の手順では、ISVs AWS Marketplace チャンネルパートナーの販売承認を作成する方法の概要を説明します。この機能を使用するには、の販売承認タブを使用するためのアクセス許可が必要です AWS Marketplace 管理ポータル。詳細については、「[AWS Marketplace 販売者のポリシー](#)」を参照してください。

販売承認を作成する

1. AWS Marketplace 販売者アカウント [AWS Marketplace 管理ポータル](#)で にサインインします。

Tip

Seller アカウントでサインイン AWS アカウント する前に、別の AWS Marketplace からサインアウトしていることを確認してください。

2. 販売承認タブを選択し、販売承認の作成を選択します。
3. 販売承認の作成ページで、次の詳細を入力します。
 - 販売承認名には、承認の名前を入力します。

Note

販売承認名に入力した情報は、販売者レポートでチャンネルパートナーに表示されます。

- リセラーの場合は、ドロップダウンリストから認可する AWS Marketplace チャンネルパートナー (リセラー) を選択します。リセラーは名前またはアカウント ID で選択できます。
 - 製品タイプで、製品のタイプを選択し、販売承認を作成する製品のいずれかを選択します。
 - 「認可の詳細に進む」を選択します。
4. 詳細の指定 ページで、次の詳細を入力します。
 - 販売承認の説明には、説明を入力します。

Note

販売承認の説明に入力した情報は、販売者レポートでチャネルパートナーに表示されます。

- 更新では、この認可が同じ製品の既存の顧客で既存の有料サブスクリプションを更新する意図があるかどうかを指定します。
 - (オプション) [購入者アカウント ID] を 1 つ以上設定して、販売認証がそれらの購入者のみを対象とするように指定します。
 - 保存して終了するか、次へを選択します。プロセスの任意のステップで保存と終了を選択すると、販売承認が下書きとして保存されます。Next を選択すると、期間と料金の選択ページが表示されます。
5. 期間と料金の選択ページで、次の詳細を入力します。
- 料金モデルでは、次のいずれかのオプションを選択します。
 - 分割プランによる契約料金 - デイメンションタイプごとに固定数量、超過時間料金、分割プランを指定します。超過料金は時間単位の料金で請求され、個別に請求されます。
 - 前払いによる契約料金 - デイメンションタイプごとに契約料金を指定し、追加の使用には時間単位の料金を指定します。購入者は、各デイメンションタイプの契約価格でコミットする数量を選択し、承諾時に全額の請求を受けることができます。追加の使用量は時間単位の料金で請求され、個別に請求されます。
 - 使用料金 - デイメンションタイプごとに時間料金を指定します。購入者は、使用量の時間料金に基づいて課金されます。
 - 通貨 で、販売承認の通貨を選択します。

Note

米ドル以外の通貨は、契約料金オファーでのみ使用できます。

- Duration で、販売承認の期間を選択します。

Note

リセラーの開始日は、メーカーが再販認可に記載した日付より前でなければなりません。

- Product Dimension セクションには、製品で現在利用可能な契約ディメンションが表示されます。販売承認に含めるディメンションを選択し、数量または価格を指定できます。分割プランによる契約料金ではディメンションあたりの数量を入力でき、前払いによる契約料金ではディメンションあたりの価格を入力できます。
 - カスタムディメンションの管理を選択して、将来のオファーと販売承認のためにこの製品で使えるディメンションを製品に追加します。製品に追加されたカスタムディメンションは削除できません。1つの製品で最大 100 個のディメンションを持つことができます。
 - 使用ディメンションあたりの料金 には、ディメンションタイプごとに時間料金を指定します。購入者は、使用量の時間料金に基づいて課金されます。公開オファーの料金は、デフォルトでここに入力されます。
 - 購入者分割払いプランの場合は、契約合計を入力し、希望する頻度に基づいて分割払いプランを生成します。これは、分割プランによる契約料金に必要です。
 - インスタンスタイプあたりの料金については、インスタンスタイプごとに使用料金を設定します。オプションの料金ツールを使用すると、公開価格に割引を適用するか、すべてのインスタンスに同じ価格を適用して、価格を一括更新できます。
 - 販売承認の可用性については、次のいずれかのオプションを選択して、プライベートオファーの作成数を制限するか、この販売承認を使用してプライベートオファーを作成できる特定の時間に制限します。
 - 単一使用 – 指定された終了日までリセラーが単一のオファーを作成できるようにします。
 - 特定の期間 – 指定された終了日までリセラーが複数のオファーを作成できるようにします
 - 期間を設定しない – 販売承認が手動で非アクティブ化されるまで、リセラーが複数のオファーを作成できるようにします
 - 保存して終了、前へ、または次へを選択できます。Next を選択すると、法律用語の設定ページが表示されます。
6. 法的用語の設定ページで、次の詳細を入力します。
- エンドユーザーライセンス契約では、パブリック EULA または標準契約 AWS Marketplace (SCMP) を選択するか、カスタム EULA をアップロードします。

 Note

プロフェッショナルサービス販売者の場合、カスタム EULA のみがサポートされています。

- (オプション) AWS Marketplace (RCMP) のリセラー契約を選択するか、販売承認に含めるカスタム契約をアップロードします。
 - 保存して終了、前へ、または次へを選択できます。Next を選択すると、レビューと作成ページが表示されます。
7. 確認と作成ページで、すべての情報が正しいことを確認します。販売承認が発行されると、変更することはできません。
 8. 販売承認の作成を選択して、販売承認をリセラーに発行します。

販売認可の管理

- 販売承認作成テーブルが更新され、販売承認名、製品名、リセラー名、作成日、有効期限、ステータスなど、関連する販売承認の詳細が表示されます。
- 販売承認を作成した後、有効期限を延長したり、その他の詳細を変更したりすることはできません。
- リセラーによる販売承認の使用が不要になった場合は、販売承認を無効にすることができます。販売承認を無効にすると、その販売承認を使用して新しいオファーを作成することはできません。既に作成されたオファーは影響を受けません。
- 販売認証のクローンを作成するには、販売認証を選択して、[クローン] を選択します。これにより、すべてのフィールドに値が事前入力され、編集が可能になります。

販売承認ステータスとアクション

次のリストでは、販売承認ステータス値とその意味について説明します。

- 下書き - 販売承認は作成されていますが、まだアクティブ化されていません。
- 承認済み - 販売承認はアクティブであり、チャネルパートナーのプライベートオファー (CPPOs) の作成に使用できます。CPPOs はまだ作成されていません。
- 認可 (再利用可能) - 販売認可は、少なくとも 1 つの CPPO の作成に使用され、追加の CPPOs の作成に使用できます。このステータスは、特定の期間を持つ認可、または設定された期間を持たない認可に共通です。
- 認可 (消費) - 販売認可は完全に利用されており、追加の CPPOs の作成には使用できません。これは通常、リセラーがプライベートオファーを作成した後、シングルユース認可で発生します。
- 有効期限切れ - 販売承認の可用性の終了日が過ぎています。CPPOs の作成には使用できません。

- 非アクティブ化 - 独立系ソフトウェアベンダー (ISV) が認可を手動で非アクティブ化しました。CPPOs の作成には使用できません。このステータスは、API では「制限済み」と呼ばれます。

Note

システムは、販売承認の内部ステータス、可用性終了日、オファー延長ステータスなど、いくつかの要因に基づいてステータスを決定します。ステータス値は、重複の可能性を解決するための優先順位に基づいて割り当てられます。

プライベートオファーの分割プラン

AWS Marketplace 販売者は、分割プラン (フレキシブル支払いスケジュールとも呼ばれます) を使用して、カスタム支払いスケジュールでプライベートオファーを拡張できます。分割払いプランは、特定の製品および料金タイプのプライベートオファーで利用できます。詳細については、「[プライベートオファーに適した製品のタイプ](#)」を参照してください。支払いスケジュールは、購入者が通常の分割払いで支払いを行うことで、承認された契約期間に分散できます。

複数年およびカスタム期間の Amazon マシンイメージ (AMI) 製品の場合、オファーに含まれる各インスタンスタイプのインスタンス数と、起動した追加のインスタンスの時間単位の料金を設定します。購入者が指定された数のインスタンスを起動すると、起動された追加のインスタンスは、プライベートオファーで指定された時間単位の料金で課金されます。

拡張済みであり購入者がサブスクライブ済みのプライベートオファーの支払いスケジュールを変更することはできません。変更するには、[新しいオファーを作成](#)する必要があります。

プライベートオファーの分割プランの作成

プライベートオファーを作成するときは、分割プランを使用してカスタム支払いスケジュールを設定できます。

プライベートオファーの分割プランを作成するには

1. オファーの料金と期間の設定ページで、製品の料金について、分割プランで契約料金を選択します。
2. このオファーの契約期間を選択し、オファーの詳細を指定します。詳細については、「[AWS Marketplace 製品のプライベートオファーの準備](#)」を参照してください。

3. 購入者分割払いプランで、次のパラメータを入力します。

- 契約合計
- (オプション) 最初の支払いを他の支払いと異なるものにする場合は、初期支払いを入力します。残りの残高は、後続の支払いに均等に分割されます。
- [Frequency] (頻度)

毎月、四半期ごと、毎年、またはカスタムを選択します。Custom を選択した場合は、分割払いの数も入力します。

最大で 60 回の支払いを追加することができます。また、各支払い明細項目を調整するオプションもあります。支払い明細項目を調整するたびに、[購入者の合計支払い金額] が更新されます。

- 初回請求日

4. 「分割プランの生成」を選択します。請求書の日付が契約期間外になると、エラーメッセージが表示されます。
5. すべての請求書の金額と日付を確認したら、購入者から支払う合計金額が、プライベートオファアの過程で購入者に支払う合計価格と一致していることを確認します。プライベートオファアの作成を完了するには、「」の残りのステップを完了します [the section called “プライベートオファアの草案作成と公開”](#)。

購入者がプライベートオファアを承諾すると、支払いスケジュールで定義した請求日に 00:00 UTC に請求されます。が購入者から支払いを受け取った後、各請求書の支払い AWS Marketplace を受け取ります。オファアの承諾日より前に実行できる請求書の日付は 1 つだけです。プライベートオファアが支払いスケジュールの最初の請求日以降に承諾された場合、最初の請求書はオファアが承諾された直後に生成されます。購入者がサブスクライブすると、スケジュールと AWS 請求書にすべての支払いが表示され、支出を追跡するのに役立ちます。

分割プランレポート

分割プランを使用したプライベートオファアのレポートは、月別請求収益レポートのセクション 4 に表示されます。詳細については、「[セクション 4: Flexible Payment Schedule による契約](#)」を参照してください。

プライベートオファ어의将来の日付の契約の作成

AWS Marketplace 販売者は、将来の日付の契約を使用して、購入者が事前に決められた将来の日付に受け取る製品を販売できます。一般的な AWS Marketplace 取引では、購入者はオファーが承諾された後または契約が作成された後すぐに、製品のライセンスまたは権利を受け取ります。一方、将来の日付の契約 (FDA) では、購入者は事前に決められた将来の日付に、製品のライセンスまたは権利を受け取ります。FDA を使用して、購入者との既存の取引の更新を設定できます。FDA は、software as a service (SaaS) 製品契約、および Flexible Payment Schedule の有無にかかわらず使用料金付き契約 (CCP) をサポートしています。以下のセクションでは、将来の日付の契約の使用について説明します。

FDA では、購入者が製品の使用を開始するときではなく、購入者が希望するときに取引を終了できます。FDA を使用して、AWS Marketplaceでのトランザクションに基づいて以下のアクションを実行できます。

- 販売ニーズに基づいて取引を予約する (購入者がオファーを承諾する)。
- 財務または会計上のニーズに基づいて購入者に請求する。
- 購入者のニーズに基づいて、ライセンスや権利の有効化などの製品へのアクセスを購入者に提供する。

購入者は、現在の契約が有効な場合でも、次の場合、FDA のオファーを受け入れることができます。

- FDA の開始日は、既存の契約が終了した後に発生します。
- 自動更新は、FDA を受け入れる前に無効になります。

Important

FDA が承認されると、自動更新を再度有効にすることはできません。

- サービスの日付は、他の承認された FDAs と重複しません。

トピック

- [将来の日付の契約に関する考慮事項](#)
- [将来の日付の契約の作成](#)
- [将来の日付の契約での分割プランの使用](#)

- [将来の日付の契約の通知を受け取る](#)
- [チャネルパートナーのプライベートオファーの再販に関する将来の日付の契約を使用する](#)

将来の日付の契約に関する考慮事項

将来の日付の契約を使用する場合は、以下の日付に注意します。

契約署名日

購入者がオファーを承諾し、契約が作成された日付。

契約開始日

購入者の製品に対するライセンスまたは権利が有効になり、購入者が製品の使用を開始できる日付。

契約終了日

契約が終了する日付。契約と、購入者のライセンスまたは権利は、この日に失効します。

Note

エンドユーザーライセンス契約、注文フォーム、またはサブスクライバーとのその他の契約で指定された契約期間は、サブスクライバーの AWS Marketplace 請求書で指定された期間と矛盾するかどうかを制御します。

将来の日付の契約の作成

将来の開始日を設定したプライベートオファーを作成する場合、記録上の販売者が契約開始日を設定します。購入者は開始日を変更することはできませんが、AWS Marketplaceでプライベートオファーを承諾する前に開始日を確認することはできます。

将来の開始日のあるプライベートオファーを作成する

1. プライベートオファーを作成する際、[契約期間] で [将来の日付で開始] を選択します。
2. [サービス日] セクションに、[サービスの開始日] と [サービスの終了日] を入力します。ここで選択したサービスの開始日は、購入者がオファーを承諾したときの、将来の日付の契約の契約開始日になります。

Note

更新に FDA を使用するには、サービス開始日は、更新する契約の終了日から 1 日以上経過している必要があります。たとえば、契約終了日が 12/31/2024 の場合、サービス開始日を 1/1/2025 に設定する必要があります。

販売者は、サービスの開始日を 3 年先までの日付から選択できます。

将来の日付の契約での分割プランの使用

FDA で分割プランを使用すると、購入代金の支払い時期は、契約署名日と契約終了日の間の任意の時期に設定できます。これには、契約開始日の前後の支払いも含まれます。

プライベートオファーの支払い日と金額は、記録上の販売者が選択します。分割払いプランの設定の詳細については、「[the section called “プライベートオファーの分割プランの作成”](#)」を参照してください。

将来の日付の契約の通知を受け取る

将来の日付の契約に対して実行された次のアクションについて、指定されたルートアカウントに[電子メール通知](#)が届きます。

- オファーの承認/契約の作成 (契約署名日)
- ライセンスまたは権利のアクティベーション時 (契約開始日)
- 30 日後、60 日後、または 90 日後に失効する契約に関する事前のリマインダー
- 契約失効日 (契約終了日)
- 契約の修正または置き換え時

Note

SaaS 向けの既存の Amazon Simple Notification Service (Amazon SNS) 通知はすべて、FDA でも機能します。FDA では、両方の Amazon SNS トピックは契約開始日に開始されます (契約署名日ではありません)。詳細については、「[the section called “SaaS 製品に関する Amazon SNS 通知”](#)」を参照してください。

チャネルパートナーのプライベートオファ어의再販に関する将来の日付の契約を使用する

製造元とリセラーは、AWS Marketplace チャネルパートナーのプライベートオファ어に将来の日付の契約を使用できます。

製造元として:

- 標準の AWS Marketplace チャネルパートナープライベートオファ어 (CPPOs) と同様に、製造元は AWS Marketplace チャネルパートナーに再販承認を延長して、将来の開始日で CPPOs を作成することをチャネルパートナーに許可する必要があります。

再販認可を作成する方法の詳細については、[the section called “ISV として販売認証を作成する”](#) ページの手順に従ってください。

- 再販認可を作成する際、製造元はオプションでサービス開始日として認められる最大値を指定することができます。これは、対応するチャネルパートナーのプライベートオファ어を作成するときに AWS Marketplace チャネルパートナーが指定できる最大サービス開始日 AWS Marketplace になります。

Note

製造元が最大日付を指定しない場合、AWS Marketplace チャネルパートナーは最大 3 年先までの将来のサービス日付を指定できます。

リセラーとして:

- リセラーとチャネルパートナーの場合、将来の日付のチャネルパートナーのプライベートオファ어を作成する手順と、将来の日付の通常のプライベートオファ어を作成する手順は同じですが、重要な違いが 1 つあります。リセラーが指定できる契約開始日は、製造元による再販認可で許可されるサービス開始日の最大値として指定されている日付よりも早い日付でなければなりません。
- チャネルパートナーのプライベートオファ어を作成する方法については、「[the section called “チャネルパートナーのプライベートオファ어”](#)」を参照してください。

プライベートオファーに関するよくある質問

このよくある質問は、プライベートオファーの作成、管理、トラブルシューティングに関するよくある質問に回答します。

トピック

- [Private Offer Success Team \(POST\) とは何ですか？ 販売者はどのように連絡すればよいですか？](#)
- [販売者はオファータブにアクセスするときにどのようにエラーを解決できますか？](#)
- [分割払いプランまたは柔軟な支払いスケジュールとは](#)
- [プライベートオファーにアクセスするときに 404 エラーを受け取った購入者を支援するにはどうすればよいですか？](#)
- [プライベートオファーを承諾しようとする、「既にアクティブな契約がある」というエラーが表示されるのはなぜですか？](#)
- [販売者または購入者はプライベートオファーをキャンセルできますか？](#)
- [返金または契約のキャンセルをリクエストするにはどうすればよいですか？](#)
- [購入者に請求されるのはいつですか？](#)
- [オファーが承諾されたら、販売者はどのような手順を実行する必要がありますか？](#)
- [販売者やパートナーへの AWS 支払い方法](#)
- [は税金をどのように AWS 評価しますか？](#)
- [リソースとサポート](#)

Private Offer Success Team (POST) とは何ですか？ 販売者はどのように連絡すればよいですか？

POST は AWS Marketplace 、プライベートオファーエクスペリエンスで外部オーディエンスを有効にします。販売者は、AWS Marketplace 管理ポータルで [サポートフォーム](#) を使用できます。サポートフォームに関するヘルプについては、[プライベートオファーサポートフォームガイド](#) PDF をダウンロードして参照してください。

販売者はオファータブにアクセスするときにどのようにエラーを解決できますか？

AWS Marketplace 管理ポータルでオファータブを選択する際にエラーが発生した場合は、次の前提条件を満たしていることを確認してください。

ソフトウェアまたはサービスを直接販売するためのプライベートオファーを作成するための前提条件

1. AWS アカウントに適切な Identity and Access Management (IAM) ポリシーがあることを確認します。必要なポリシーの詳細については、[プライベートオファーの IAM ポリシー](#)を参照してください。

Note

IAM ポリシーまたはアクセス許可の変更についてサポートが必要な場合は、内部 AWS 管理者にお問い合わせください。は、IAM ポリシーまたはアクセス許可をサポート AWS できません。アクセスは、[責任共有モデル](#)に従ってお客様が管理するためです。

2. 公開されていない製品が少なくとも 1 つある。これを確認するには、[デベロッパー](#)を検索します AWS Marketplace。製品が表示されない場合は、製品を公開していないか、出品が制限されている可能性があります。

ソフトウェアまたはサービスを再販するためのチャネルパートナープライベートオファー (CPPOs) を作成するための前提条件

- 以下の「販売者」になるための地理的前提条件を満たします AWS Marketplace。
 - [対象となる管轄区域](#)の法人、またはそれらの地域の 1 つに法人化された事業体を使用します。
 - 銀行口座に関する必要な情報を入力します。以下を含む 1 つ以上の銀行口座を指定できます。
 - 米国自動決済機関 (ACH) アカウント
 - 適格な管轄区域からの世界銀行間金融通信協会 (SWIFT) 銀行口座
 - [Hyperwallet](#) アカウント

分割払いプランまたは柔軟な支払いスケジュールとは

Flexible Payment Schedule (FPS) と呼ばれる分割プランを使用すると、カスタム支払いスケジュールでプライベートオファーを拡張できます。これらのプランは、特定の製品および料金タイプのプライベートオファーで利用できます。詳細については、[「プライベートオファーの対象となる製品タイプ」](#)を参照してください。

支払いスケジュールは、購入者が通常の分割払いで支払いを行うことで、承認された契約期間に分散できます。サブスクライブ後、顧客はスケジュールと AWS 請求書ですべての支払いを表示できるため、支出を追跡できます。

分割プランと FPSs により、販売者は以下を指定できます。

- デイメンションまたはインスタンスタイプあたりのユニット数
- 契約の支払い条件 (前払い、遅延、または複数の請求書)

分割プランの作成

Note

ソフトウェア再販シナリオでは、独立系ソフトウェアベンダー (ISV) が分割プランを決定します。

分割プランを作成するには

1. オファ어의料金と期間の設定ページで、製品の料金について、分割プランで契約料金を選択します。
2. 契約期間を選択し、オファ어의詳細を指定します。
3. 購入者分割払いプランで、必要なパラメータを入力します。
 - 承諾時の前払い請求については、ドル金額を入力し、請求日をプライベートオファ어を作成している日付に設定します。
 - 遅延請求の場合は、ドル金額を入力し、将来の請求日を設定します。
 - 分割請求の場合は、支払いの追加を選択して、金額と請求日を含む複数の支払い明細項目を入力します。

詳細については、[「Installment Plans の作成」](#)を参照してください。[「Installment Plans」のビデオチュートリアル](#)を見することもできます。

固定 SKU またはインスタンスタイプの設定

オファ어作成ページのデイメンションセクションで、次の操作を行います。

1. 「Buyer can choose one or more options offered」を選択します。

- 異なるディメンションとそれぞれの数量を入力します。
- 複数のディメンションを含めるには「ディメンションの追加」を選択し、このプライベートオファークスタムディメンションを作成するには「新しいディメンションの作成と追加」を選択します。

プライベートオファークアクセスするときに 404 エラーを受け取った購入者を支援するにはどうすればよいですか？

404 エラーの一般的な理由とその解決策は次のとおりです。

アカウントの関連付けが正しくない

- [AWS コンソール](#)の[プライベートオファーク](#)タブにアクセスするように購入者に依頼します。
- オファークが「利用可能なプライベートオファーク」または「承諾済みオファークと期限切れオファーク」タブに表示されない場合：
 - 購入者が正しいアカウントにサインインしていることを確認します。AWS コンソールの右上隅でアカウント ID を確認できます。
 - 正しいアカウントにサインインしている場合は、プライベートオファークが Amazon マシンイメージ (AMI) の制限付き出品用であることを確認します (以下の AMI の制限付き出品セクションを参照)。

解決策： 購入者は正しいアカウントにサインインするか、正しい AWS アカウント ID にプライベートオファークを発行する必要があります。

期限切れのプライベートオファーク

- オファークが承諾済みおよび期限切れのオファークタブに表示される場合、オファークは期限切れです。

解決策： 有効期限を延長するか、新しいプライベートオファークを発行します。

AMI 限定出品製品

- オファークがどちらのタブにも表示されず、購入者が正しいアカウントにサインインしている場合、製品は制限付き出品ステータスになる可能性があります。

解決策:

- 単一 AMI の制限付き出品の場合: 購入者アカウントの許可出品手順に従って、[購入者のアカウントを許可出品](#)します。
- その他のサーバータイプの制限付き出品については、[プライベートオファー成功チーム \(POST\)](#) に連絡して、購入者のアカウントを許可リストに登録してください。

Note

許可リストは、製品ごとに 1 回限りのアクティビティです。

プライベートマーケットプレイスの制限

- 購入者にプライベートマーケットプレイスがある場合、製品がプライベートカタログで利用できないというエラーが表示されることがあります。

解決策: 購入者は許可リストに製品を追加する必要があります。 [「プライベートマーケットプレイスへの製品の追加」](#)を参照してください。

問題が解決しない場合は、購入者に以下を試してもらいます。

- サインアウトして再度サインインする
- ブラウザキャッシュをクリアする
- Cookie を削除する
- シークレットウィンドウにサインインする
- 別のブラウザを使用する (Internet Explorer ではない)

プライベートオファーを承諾しようとする、「既にアクティブな契約がある」というエラーが表示されるのはなぜですか？

このエラーは、購入者アカウントに既に製品へのアクティブなサブスクリプションがある場合に発生します。解決方法は製品タイプによって異なります。

Software as a Service (SaaS) 契約の場合

各購入者アカウントは、SaaS 契約または消費製品との契約へのアクティブなサブスクリプションを 1 つだけ持つことができます。アクティブなサブスクリプションを更新または展開するには：

1. [AWS Marketplace 管理ポータル \(AMMP\)](#) の契約タブから契約ベースのオファーを作成します。
2. 元のオファーの保留中の支払い条件は、まだ請求されていない残りの支払いを上書きするため、契約ベースのオファーに含めます。

契約ベースのオファーの作成

Marketplace プライベートオファー (MPPO) の場合：

1. [AMMP](#) の契約タブに移動します。
2. 購入者のアカウント ID を検索します。
3. 購入者の現在の契約を選択します。
4. 「契約ベースのオファーの作成」を選択します。

チャネルパートナープライベートオファー (CPPO) の場合：

1. ISV は、ライセンスユニット数、CP への支払い条件、契約期間など、パートナータブの新しい料金でチャネルパートナーに販売承認を付与します。
2. チャネルパートナーは、[AMMP](#) の契約タブに移動します。
3. 購入者のアカウント ID を検索します。
4. 購入者の現在の契約を選択します。
5. 契約ベースのオファーの作成を選択します。
6. 販売承認リストを開き、ステップ 1 から新しいオポチュニティを選択します。これにより、チャネルパートナーは契約ベースのオファーの条件を入力できます。

詳細については、[「契約ベースのオファーの作成」](#)を参照してください。

SaaS 契約および従量制契約 (CCPsその他のオプション)：

- 別の購入者アカウントを対象とする同じ製品 ID の新しいプライベートオファーを作成します。
- 同じ購入者アカウントに別の製品 ID の新しいプライベートオファーを発行します。
- アクティブな契約のキャンセルをリクエストし、同じ製品 ID と購入者 ID に対して新しいプライベートオファーを発行します。

AMI の場合、時間単位および年単位

各購入者アカウントは、アクティブなサブスクリプションを 1 つだけ持つことができます。更新または展開するには：

1. オファータブから新しいオファーを発行します。
2. 購入者が承諾すると、新しい条件は以前の契約条件よりも優先されます。
3. 前のオファーの支払いスケジュールからの保留中の料金は、キャンセルしない限り続行されます。

AMI 契約とプロフェッショナルサービスの場合

新しいプライベートオファーを受け入れる前に、購入者の現在のサブスクリプションをキャンセルする必要があります。[返金/キャンセルフォーム](#)を使用してキャンセルリクエストを開始します。

販売者または購入者はプライベートオファーをキャンセルできますか？

- 購入者がサブスクライブしていない場合：販売者は、AWS Marketplace 管理ポータルのおファータブに移動し、オファーを選択し、キャンセルを選択してキャンセルできます。
- 購入者がサブスクライブしている場合：登録販売者は、AWS Marketplace 管理ポータルの[返金/キャンセルフォームを使用してキャンセル](#)リクエストを開始する必要があります。詳細な手順については、この[ビデオチュートリアル](#)を参照してください。

返金または契約のキャンセルをリクエストするにはどうすればよいですか？

返金と契約のキャンセルは AWS、カスタマーサービスチームが処理します。

販売者—ISVs for Marketplace プライベートオファー、チャネルパートナープライベートオファーのチャネルパートナー—は、[返金/キャンセルフォーム](#)を使用して返金またはキャンセルを開始する必要があります。

[この動画](#)では、プロセス全体について説明します。

販売者は、次のデータを入力する必要があります。

- 購入者アカウント ID: この情報は、オファーの詳細または請求収益ダッシュボードで確認できます。これはサブスクライバーアカウント ID である必要があります。

- 販売者アカウント ID: これは、プライベートオファーの作成に使用される販売者の AWS アカウント ID です。
- 製品 ID: この情報は、オファーの詳細またはレガシー製品 ID 列 [請求済み収益ダッシュボード](#) のにあります。
- 請求日: この情報は、オファーの詳細または使用状況の開始期間列 [請求済み収益ダッシュボード](#) のにあります。
- 返金額: 返金が不要な場合、販売者はこれを 0 USD に設定できます。
- その他の詳細: 以下の注意事項を参照してください。

Important

- 契約のキャンセルを含むリクエストの場合は、次のテキストを含めます」

offer-X へのアカウント X のサブスクリプションをキャンセルしてください。

返金の場合、このセクションで購入者の AWS 請求書 ID を指定すると便利ですが、必須ではありません。

- 送信後に提供された参照 ID を保存して、フォローアップ時に後で参照できるようにします。

リクエストを送信したら、[AWS サポートコンソール](#)でステータスの更新を確認します。

購入者に請求されるのはいつですか？

- 承諾時: 請求書は、サブスクリプション直後に請求コンソールに作成されます。
- 柔軟な支払いスケジュール: 請求書は、販売者と購入者の間で交渉されたカスタム支払いスケジュールに基づいています。
- AWS 一括請求 (月の 2 日/3 日): パブリックオファーの購入と計測付き製品を含めることができます。

オファーが承諾されたら、販売者はどのような手順を実行する必要がありますか？

1. オファーの承諾を追跡します。

2. 財務チームの場合：販売者は、未払いの支払いを作成し、AWS Marketplace 注文に対する顧客への請求書の作成を抑制することで、購入の追跡を処理することが標準プラクティスです。

販売者やパートナーへの AWS 支払い方法

- 支払いは、サブスクライバーから資金が正常に回収された後にのみ開始されます。
- 支払いは、毎月 7 日から 10 日の間のデフォルトの支払い頻度で行われます。または、ISV とチャネルパートナーは、毎日または毎月の支払いスケジュールを選択できます。ISV またはチャネルパートナーが毎月のオプションを選択した場合、支払いを受け取る日付を選択できます。
- 支払いは、ISV またはチャネルパートナーのアカウントの米国の銀行口座に、出品料金を引いた金額で入金されます。資金が一般的な銀行 ACH SLAs に到達するまでに 1~3 営業日かかる場合があります。米国の銀行口座をお持ちでない場合は、[Hyperwallet](#) を使用して Amazon の売上の支払いを預金口座に受け取り、現地通貨で現地の銀行に直接送金できます。

は税金をどのように AWS 評価しますか？

AWS Marketplace は、以下に基づいて税金を請求します。

- [製品サブスクライバーの納税者住所](#)
- 製品タイプ
- Marketplace ファシリテーター法

Marketplace ファシリテーター：マーケットプレイスオペレーターは、税務機関に税金を請求、徴収、送金する必要があります。

マーケットプレイス以外のファシリテーター：責任は販売者にあります。

納税の詳細については、[AWS Marketplace 「 - 販売者の税務ヘルプ」](#) を参照してください。

リソースとサポート

特定のリクエストがある場合は、AWS Marketplace 管理ポータルから次のいずれかの AWS Marketplace チームにお問い合わせください。

Private Offers Success Team (POST): チームは、プライベートオファーの運用の有効化とサポートにより、販売者 (ISV およびチャネルパートナー) と購入者をサポートします。[サポートフォーム](#)からお問い合わせください。

Vendor Finance Success Team (VFS): チームは、ベンダー財務のキャッシュアプリケーション、調整、関連レポートに影響を与えるプロセスを改善し、MPPOs と CPPOs ではないという財務固有の質問で、販売者 (ISV とチャネルパートナー) のオンボーディングをサポートします。[サポートフォーム](#)からお問い合わせください。

Managed Catalog Operations (MCO): チームは、ソフトウェアのサードパーティー販売者を AWS Marketplace プラットフォームにオンボーディングし、ポリシーコンプライアンスと購入者エクスペリエンスのためにソフトウェア製品を確認して処理し、販売者との運用関係を管理します。[サポートフォーム](#)からお問い合わせください。

プライベートオファープロセスを初めて使用する場合は、この[ビデオライブラリ](#)を使用して、一般的な質問の概要、概要、回答の使用を開始してください。[購入者の請求](#)、[支払い](#)、[AWS 税ルール](#)に関する質問に対する回答は、リンクされた動画または [AWS Marketplace Vendor Finance Success](#) PDF でも確認できます。

契約の使用

AWS Marketplace 販売者契約は、販売されている製品の販売者と購入者間の契約です AWS Marketplace。契約は、購入者がパブリックまたはプライベートのオファーを受け入れるときに形成されます。このセクションのトピックでは、契約タイプ、ステータス、および契約の変更または修正方法について説明します。

トピック

- [契約タイプ](#)
- [契約の使用](#)
- [での契約の修正 AWS Marketplace](#)

契約タイプ

AWS Marketplace には、次のタイプの契約があります。

- 公開オファー契約 – 購入者が販売者から公開されている製品を購入するときに承諾する、交渉されていない標準的な法的契約 AWS Marketplace。
- プライベートオファー契約 – 購入者が承諾するプライベートオファーが契約になります。詳細については、「[プライベートオファーの作成と管理](#)」を参照してください。
- 将来の日付の契約 – 購入者は、事前に決められた将来の日付に製品ライセンスまたは使用権限を受け取ります。詳細については、「[プライベートオファーの将来の日付の契約の作成](#)」を参照してください。

契約の使用

の契約ページから契約を表示および管理します [AWS Marketplace 管理ポータル](#)。契約ページで、契約表に販売する製品の契約が表示されます AWS Marketplace。契約テーブルでは、契約のステータスは次のいずれかになります。

- アクティブ – 契約の条件はアクティブです。
- 有効期限切れ – 契約は 90 日以内に期限切れになります。
- 有効期限切れ – 契約は、定義された契約終了日に終了します。
- 更新済み – 契約が更新され、更新された条件を持つ新しい契約が作成されました。

- キャンセル済み – アクセプタは契約を早期に終了しました。
- 置き換え済み – 契約は、契約置き換えオファーを使用して置き換えられました。
- 終了 – 支払いの失敗などの理由で、契約は元の契約終了日 AWS より前に によって終了されました。
- アーカイブ済み – 契約は、指定された理由なしで終了しました。

契約の検索

契約を検索するには、フィルターを設定します。契約の検索バーを選択し、フィルタープロパティと値を選択します。たとえば、特定の契約を検索するには、契約 ID を選択し、ID を選択します。フィルターを保存するには、フィルターをクリアで、新しいフィルターセットとして保存を選択します。

契約のソート

テーブルは、最初は契約終了日 (UTC) 列でソートされます。契約が降順に表示され、将来の最も遠い末尾がリストの上部に表示されます。契約終了日 (UTC) 列見出しを選択してソート順序を変更できます。

契約の詳細の表示

契約の詳細を表示するには、契約 ID 列のリンクを選択します。または、契約の横にあるオプションを選択し、詳細の表示を選択します。

Note

契約を修正するには、「」を参照してください [での契約の修正 AWS Marketplace](#)。

契約の詳細のダウンロード

.pdf ファイルで契約の詳細をダウンロードするには、次の手順に従います。

契約の詳細をダウンロードするには

1. にサインインし [AWS Marketplace 管理ポータル](#)、契約を選択します。
2. 契約テーブルで、契約の横にあるオプションを選択し、詳細の表示を選択します。または、契約 ID 列で契約のリンクを選択することもできます。

3. 契約の詳細ページで、PDF のダウンロードを選択します。

での契約の修正 AWS Marketplace

AWS Marketplace 販売者、チャネルパートナー、ISVsは、アクティブな契約を置き換えるためのアップグレード、更新、修正を提供できます。たとえば、新しい使用権限を付与したり、料金割引を提供したり、支払いスケジュールを調整したり、[標準化されたライセンス条件](#)を使用するようにエンドユーザーライセンス契約 (EULA) を変更したりできます。ユニット数や支払いスケジュールを変更したり、カスタム終了日を追加したりすることもできます。

Software as Service (SaaS) 契約と消費製品との SaaS 契約は、パブリックオファーとプライベートオファーの修正をサポートしています。独立系ソフトウェアベンダー (ISVs) やチャネルパートナーなど、すべての AWS Marketplace 販売者は、これらの製品タイプのプライベートオファーをアップグレード、更新、または修正できます。以下のセクションでは、プロセスについて説明します。

オファーと契約の違いは、購入者がその条件を承諾済みであるかどうかです。

- オファーは、購入者による製品の使用に関する一連の条件です。オファーは、パブリックまたはプライベートのいずれかです。
- 契約は、購入者が承諾したオファーです。契約には、販売者がパブリックオファーまたはプライベートオファーを使用して利用可能にした有料製品と無料製品が含まれます。

契約を修正して、元の契約の登録販売者とは異なる登録販売者を指定することはできません。この機能を使用するには、AWS Marketplace 管理ポータル [契約] タブを使用する権限が必要です。詳細については、「[AWS Marketplace 販売者のアクセス許可](#)」を参照してください。

トピック

- [パブリックオファーとプライベートオファーの修正でサポートされている製品タイプ](#)
- [パブリックおよびプライベートオファーのアップグレード、更新、修正の作成](#)
- [アップグレード、更新、および修正のレポート](#)

パブリックオファーとプライベートオファーの修正でサポートされている製品タイプ

オファーの修正をサポートしているのは、次の製品タイプのみです。

- SaaS 契約
- SaaS 契約 (追加使用量あり)

以下の追加の製品タイプは、AWS Marketplace 管理ポータル の [契約] タブで確認できます。ただし、これらの製品タイプは修正をサポートしていません。

- 使用量ベースの SaaS 製品
- AMI ベースの製品
- コンテナベースの製品
- サーバー契約
- プロフェッショナルサービス製品

パブリックおよびプライベートオファ어의アップグレード、更新、修正の作成

オファ어의アップグレード、更新、修正は、次の手順 AWS Marketplace 管理ポータル を使用してから作成できます。チャンネルパートナーのプライベートオファ어의場合、チャンネルパートナーは修正を作成するときに販売承認で定義された通貨を使用する必要があります。

Note

- 変更されたオファ어は、将来の契約開始日を持つことはできません。修正されたオファ어はターゲットのアクティブな契約のみを提供し、承認されると、修正された契約はすぐにアクティブになり、将来のデートができなくなります。
- 承認されたパブリックオファ어를修正すると、プライベートオファ어になり、自動更新されなくなります。自動更新を維持するには、購入者がパブリックオファ어를サブスクライブする必要があります。詳細については、「[プライベートオファ어의仕組み](#)」を参照してください。

オファ어의アップグレード、更新、修正を作成するには

1. [AWS Marketplace 管理ポータル](#) にサインインし、[Agreements (契約)] を選択します。
2. [契約] ページで、契約の横にあるチェックボックスを選択して、[詳細の表示] を選択します。
3. 契約の表示ページで、契約の修正を選択します。

4. 契約の詳細の修正ページで、サービス日、製品ディメンション、オファー通貨 (MP direct PO AWS の場合)、支払いスケジュール、使用ディメンション、更新ステータス、EULA、オファーの有効期限を変更することもできます。
5. オファーを確認し、オファーの作成を選択します。

 Tip

わかりやすいカスタムオファー名を入力すると、[Offers (オファー)] ページで有効なオファー間の区別に役立ちます。カスタムオファー名は購入者にも表示されます。AWS では、独自の IDs や発注書番号など、追加の識別情報を含むカスタムオファー名を使用することをお勧めします。**upgrade** または **renewal** やカスタム会社名などの大まかな説明を使用することもお勧めします。個人を特定できるデータ (姓名、電話番号、住所など) は使用しないでください。このフィールドには最大 150 文字を入力できます。

修正されたオファーの詳細

変更されたオファーは、約 45 分以内にプライベートオファーページに表示されます。オファーを表示するには、にサインイン AWS Marketplace 管理ポータルしてオファーを選択します。プライベートオファーページで、購入者はオファーを受け入れるか、元の契約を続行するかを選択できます。

購入者がパブリックまたはプライベートオファーのアップグレードまたは更新を受け入れると、新しい契約はすぐに有効になり、契約は「」の「契約」ページに表示されます AWS Marketplace 管理ポータル。以前の契約の残存する支払いスケジュールはキャンセルされます。購入者は、プライベートオファーを受け入れるのと同じ方法で修正を受け入れます。プライベートオファーの購入者体験の詳細については、「AWS Marketplace 購入者ガイド」の「[プライベートオファー](#)」を参照してください。

購入者が有効期限が切れる前にパブリックまたはプライベートオファーのアップグレードまたは更新を受け入れない場合、元の契約は変更されません。

 Note

[SaaS 製品の Amazon SNS 通知の場合](#)、購入者が修正を承諾offer-identifierすると、新しいとともにsubscribe-successメッセージが表示されます。

アップグレード、更新、および修正のレポート

パブリックオファーとプライベートオファーのアップグレードと更新は、既存の販売者レポートとオファーに関連するレポートに表示されます。[日次顧客サブスクライバーレポート](#) と [日別ビジネスレポート](#) は毎日生成されます。[月別請求済み収益レポート](#) は毎月生成されます。

日別顧客サブスクリプションレポートの [Subscription intent (サブスクリプションの目的)] フィールドは、レポートエントリが新しいプライベートオファーであるかどうかを示します。以前のオファー ID フィールドには、前述のオファーの ID が表示されます。すべてのプライベートオファーは、レポートで「プライベート」というラベルが付けられます。

Important

修正により、購入者の現在のサブスクリプションが置き換えられます。既存の請求書は変更されません。ただし、修正案の支払いスケジュールは、以前のサブスクリプションの保留中の請求書を置き換えます。

の AMI ベースの製品 AWS Marketplace

AWS Marketplace 販売者は、[Amazon マシンイメージ \(AMIs\)](#) を使用して購入者に製品を配送できます。AMI は Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを起動するために必要な情報を提供します。製品用のカスタム AMI を作成すると、購入者はそれを使用して、製品がインストール済みですぐに使用できる EC2 インスタンスを作成できます。このトピックでは、AMIs を使用して AWS Marketplace 製品を配信する方法について説明します。

購入者が提供した AMI を使用すると、製品用に作成した料金設定と計測オプションに従って、作成したインスタンスに対して請求されます。購入者は、AMI の新しいカスタムバージョンの作成など AWS、で他の AMIs を使用するのと同じ方法で製品 AMI を使用できます。AMI から作成された EC2 インスタンスは、引き続き AMI 製品コードに基づいて製品として請求されます。

AMI ベースの製品配信方法

AMI ベースの製品は、次の 3 つの方法のいずれかで配信できます。

- 単一 AMI - 購入者は AMI を EC2 インスタンスのテンプレートとして選択して使用します。購入者は、Amazon マシンイメージ の配信方法フィルターを使用してこれらの製品を見つけることができます。詳細については、「[AMI ベースの製品の作成](#)」を参照してください。
- AWS CloudFormation テンプレート - 購入者が、異なるロールを持つ複数のインスタンスのシステムを単一のユニットとしてインストールできるようにするテンプレートを作成します。購入者は、CloudFormation の配信方法フィルターを使用してこれらの製品を見つけることができます。詳細については、「[CloudFormation テンプレートを製品に追加する](#)」を参照してください。

追加リソース

AMI 製品の詳細については、次のトピックを参照してください。

AWS Marketplace

- [の製品料金 AWS Marketplace](#)
- [AWS Marketplace Metering Service を使用した AMI 製品のカスタム計測を構成する](#)

AMI ベースの製品

- [での AMI ベースの製品について AWS Marketplace](#)

- [AMI ベースの製品の作成](#)
- [販売者として AWS Marketplace の AMI ベースの製品の管理](#)
- [CloudFormation テンプレートを製品に追加する](#)
- [で使用する AMIs を構築するためのベストプラクティス AWS Marketplace](#)
- [の AMI 製品の料金 AWS Marketplace](#)
- [での AMI 製品の Amazon SNS 通知の受信 AWS Marketplace](#)
- [の AMI 製品チェックリスト AWS Marketplace](#)
- [の AMI ベースの製品要件 AWS Marketplace](#)

での AMI ベースの製品について AWS Marketplace

AWS Marketplace 販売者は、[Amazon マシンイメージ \(AMIs\)](#) を使用して製品を購入者に配信できます。AMI は Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを起動するために必要な情報を提供します。以下のセクションでは、AMI ベース製品を使用するための主要な概念について説明します。

Note

AMI 製品で利用できる AMI は 1 つだけですが、その AMI のバージョンを製品に追加できます。

トピック

- [製品のライフサイクル](#)
- [AMI 製品コード](#)
- [変更リクエスト](#)
- [製品ロードフォーム](#)
- [年次契約の変更](#)

製品のライフサイクル

AMI ベースの製品には、1 つ以上のバージョンのソフトウェアと、製品全体に関するメタデータが含まれます。製品を作成するときは、製品名、説明、料金 AWS Marketplace など、でそのプロパティ

を設定します。また、関連する検索で商品が表示されるように、商品に適したカテゴリを決定し、キーワードを追加します。

また、ソフトウェアの最初のバージョンが作成されます。これは単なる AMI でも、購入者が独自の AMIs を作成するために使用できる AWS CloudFormation テンプレートやソフトウェアパッケージを含めることもできます。詳細については、「[AMI ベースの製品配信方法](#)」を参照してください。

有料製品の場合、購入者はインストールされたインスタンスの数に応じて課金されます。製品ユーザー数など、ソフトウェアが追跡する別のディメンションを計測するには、製品を AWS Marketplace Metering Service と統合します。詳細については、「[AWS Marketplace Metering Serviceを使用した AMI 製品のカスタム計測を構成する](#)」を参照してください。

製品とソフトウェアの最初のバージョンを作成すると、最初は限られた範囲で公開され、自分のアカウントだけがアクセスできるようになります。準備ができたら、AWS Marketplace カタログに公開して、購入者が製品をサブスクライブして購入できるようにします。

[\[サーバー製品\]](#) ページでは、製品のリストを表示できます。製品のステータスは次のいずれかになります。

- [ステージング] - 情報を追加中の、不完全な製品。セルフサービス作成エクスペリエンスの最初の [保存して終了] で、変更リクエストが成功すると、送信したすべてのステップの情報を含む未公開の製品が作成されます。このステータスから、引き続き商品に情報を追加したり、変更リクエストを通じて送信済みの情報を変更したりできます。
- [制限あり] - 製品がシステムに送信され、システム内のすべての検証に合格すると完成します。その後、製品は [制限あり] 状態にリリースされます。この時点で、製品にはアカウントと許可リストに登録したユーザーだけがアクセスできる詳細ページがあります。商品は詳細ページからテストできます。必要に応じて、詳細やサポートが必要な場合は、[AWS Marketplace Seller Operations チーム](#)にお問い合わせください。
- [パブリック] - 購入者が商品を閲覧してサブスクライブできるように商品を公開する準備ができたら、[可視性を更新] 変更リクエストを使用します。これにより、AWS Marketplace Seller Operations チームが当社の[ポリシー](#)に照らして製品を確認して監査するワークフローが開始されます。製品が承認され、変更リクエストが処理されると、製品のステータスが [制限あり] から [パブリック] に移行します。
- [制限付き] - 新規ユーザーによる製品のサブスクライブを停止する場合は、[可視性を更新] 変更リクエストを使用して製品を制限できます。制限付き ステータスは、既存のユーザーが引き続き製品を使用できることを意味します。ただし、製品は一般には公開されなくなり、新規ユーザーも利用できなくなります。

の AMI ベースの製品のライフサイクル AWS Marketplace は、最初のバージョンを公開した後に終了しません。新しいバージョンのソフトウェアと、基本オペレーティングシステムのセキュリティパッチを使用して、製品を最新の状態に保つ必要があります。

完全な AMI ベースの製品のライフサイクルの例として、販売者が AMI ベースの製品を AWS Marketplace で販売することを考えているとします。以下は、販売者が長年にわたって製品を作成し、保守する方法です。

1. 製品の作成 – 販売者は製品を作成し、バージョン 1.0.0 を公開します AWS Marketplace。購入者は、バージョン 1.0.0 のインスタンスを作成して使用できます。
2. 新しいバージョンを追加 – その後、販売者は製品に新しい機能を追加し、その機能を含む新しいバージョン 1.1.0 を追加します。購入者は、元のバージョン 1.0.0 を引き続き使用することも、新しいバージョン 1.1.0 を選択することもできます。

 Note

新製品とは異なり、新バージョンは完全に公開されます。製品全体が限定リリースの場合にのみ、顧客が気付か AWS Marketplace ずに でテストできます。

3. 製品情報の更新 – バージョン 1.1.0 がリリースされたため、販売者は新機能を説明する新しいハイライトテキストで製品情報を更新して、購入者に新機能について知らせます。
4. マイナーバージョンの追加 – 販売者がバージョン 1.1.0 のバグを修正すると、新しいバージョン 1.1.1 を追加してリリースします。購入者は、バージョン 1.0.0、1.1.0、1.1.1 のいずれかを使用できるようになりました。
5. バージョンを制限する – 販売者は、購入者にバージョン 1.1.0 を使用させたくないほどバグが深刻であると判断し、そのバージョンを制限します。その場合、新規顧客は 1.1.0 を購入できなくなります (1.0.0 または 1.1.1 しか選択できません)。ただし、既存の購入者は引き続き 1.1.0 にアクセスできます。
6. バージョン情報の更新 – 既存の購入者を支援するために、販売者は 1.1.0 のバージョン情報を更新し、バージョン 1.1.1 へのアップグレードを提案します。
7. 使用状況のモニタリング – 購入者が製品を購入して使用すると、販売者は を使用して売上、使用状況、およびその他のメトリクスをモニタリングします AWS Marketplace [の販売者レポート、データフィード、ダッシュボード AWS Marketplace](#)。
8. 製品を削除する – 製品が不要になると、販売者は製品を削除します AWS Marketplace。

この例では、販売者は製品に 3 つの異なるバージョンの AMI を作成しましたが、(製品を削除する前は) 新規購入者が利用できたのは 2 つのみでした。

バージョンや製品情報を変更するには、AWS Marketplace 管理ポータルで[変更リクエスト](#)を作成します。

AMI ベースの製品を作成および管理するステップの詳細については、「[AMI ベースの製品の作成](#)」を参照してください。

AMI 製品コード

AWS Marketplaceで製品を作成すると、固有の製品コードが割り当てられます。その製品コードは製品の AMI に関連付けられ、製品の使用状況を追跡するために使用されます。製品コードは、購入者がソフトウェアを使用している間に自動的に伝播します。たとえば、お客様は AMI にサブスクライブして起動し、それを構成したり、新しい AMI を生成します。新しい AMI には元の製品コードが含まれているため、正しい使用状況の追跡とアクセス許可は維持されます。

Note

製品コードは製品の製品 ID とは異なります。の各製品には、一意の製品 ID AWS Marketplace が割り当てられます。この製品 ID は、AWS Marketplace カタログ、顧客の請求、販売者レポートで製品を識別するために使用されます。製品コードは、AMI から作成されたインスタンスにインスタンスメタデータとして添付されます。その製品コードを含む AMI を使用してインスタンスを作成すると、顧客には関連する製品 ID が記載された請求書が届きます。製品を作成したら、製品の AWS Marketplace 管理ポータル ページで製品コードと製品 ID を見つけます。

販売者は、ソフトウェアが実行中の Amazon Elastic Compute Cloud (Amazon EC2) インスタンスの製品コードをランタイムにインスタンスのメタデータから取得できます。製品の開始時に製品コードを検証するなど、セキュリティを強化するために製品コードを使用できます。AMI の製品コードへの API コールは、製品がテストのために限定された状態で発行されるまでできません。製品コードの検証の詳細については、「[AWS Marketplace AMI でソフトウェアが実行されていることを確認する](#)」を参照してください。

変更リクエスト

で製品またはバージョンを変更するには AWS Marketplace、 を通じて変更リクエストを送信します AWS Marketplace 管理ポータル。変更リクエストはキューに追加され、リクエストのタイプ

によっては解決までに数分から数日かかる場合があります。リクエストのステータスは、AWS Marketplace 管理ポータルで確認できます。

 Note

に加えて AWS Marketplace 管理ポータル、[AWS Marketplace Catalog API](#) を使用して変更リクエストを作成することもできます。

AMI ベースの製品についてリクエストできる変更のタイプは次のとおりです。

- 購入者に表示される製品情報を更新する。
- 購入者に表示されるバージョン情報を更新する。
- 製品の新しいバージョンを追加する。
- 新しい購入者がそのバージョンにアクセスできないようにバージョンを制限する。
- 製品が利用可能な AWS リージョン を更新します。
- 製品の料金設定とインスタンスタイプを更新する。
- から製品を削除します AWS Marketplace。

 Note

一部の変更リクエストでは、製品ロードフォームを使用してリクエストを作成する必要があります。詳細については、「[製品ロードフォーム](#)」を参照してください。

変更リクエストを更新する

更新で始まる変更リクエストには、現在の詳細が読み込まれます。その後、更新を行うと、既存の詳細が上書きされます。

変更リクエストを追加または制限する

追加リクエストと制限リクエストペアは、リクエストが成功するたびにプロビジョニングされるステップとアップデート専用です。セルフサービスエクスペリエンスで [保存して終了] とアクションを [送信] を選択すると、リクエストは成功します。

例えば、追加したインスタンスとリージョンに AMI アセットをプロビジョニングした場合、それらは完全に削除されるのではなく、制限することしかできません。つまり、既存のサブスクライバーと

ユーザーは、サブスクリプションまたは契約が終了するまで製品を使い続けることができます。ただし、[制限付き] ステータスの製品には新しいサブスクライバーを追加することはできません。

製品ロードフォーム

通常、[AWS Marketplace 管理ポータル](#)を使用して製品を作成または編集します。ただし、いくつかのオペレーションでは、Product Load Form (PLF) を使用する必要があります。

PLF は、製品に関するすべての情報を含むスプレッドシートです。PLF を取得するには、次の操作を行います。

- の製品の詳細ページから、既存の製品の PLF をダウンロードします AWS Marketplace 管理ポータル。
- PLF を必要とするアクションのメニュー項目を選択します。例えば、新しい月額課金サーバー製品を作成すると、適切な PLF をダウンロードするように求められます。

作業内容が既存製品の編集である場合、PLF にはその製品の情報が事前に入力されるため、更新する詳細を変更するだけで済みます。

- 新しい空の PLF が必要な場合は、AWS Marketplace 管理ポータル [ファイルのアップロード](#) ページに移動します。このページには、作成する製品タイプの PLFs へのリンクが含まれています。

PLF が完了したら、AWS Marketplace 管理ポータル [ファイルのアップロード](#) ページにアップロードします。PLF 自体の [手順] タブには、より詳細な説明があります。

年次契約の変更

時間単位の年次 (年次) プランの修正により、ユーザーとその購入者は既存のプランに以下の変更を加えることができます。

- Amazon EC2 インスタンスタイプファミリー間の切り替え
- Amazon EC2 インスタンスタイプサイズ間の切り替え
- 新しいインスタンスタイプを追加する
- 契約内の既存のインスタンスタイプの数量を増やす

購入者は、変更の日割額が 0 より大きい (サブスクリプションの値を下げるできない) 限り、変更を行うことができます。新しく追加された Amazon EC2 インスタンスの日割額は、契約の残りの期間に合わせて調整されたインスタンスタイプの年次コストに基づいています。インスタンスタイ

プを切り替える場合、削除された Amazon EC2 インスタンスタイプの比例配分コストは、新しく追加された Amazon EC2 インスタンスタイプの日割額から差し引かれます。

AMI 年間製品の修正を有効にするための追加のアクションは必要ありません。修正は、一般オファーから行われたすべての契約と、分割プランを使用しないプライベートオファーからの契約でサポートされます。

購入者が行った修正は、次のダッシュボードで確認できます。

- [契約と更新のダッシュボード](#) – 修正された契約のリスト。
- [請求済み収益ダッシュボード](#) – 顧客への請求。
- [回収と支払いダッシュボード](#) – 支出。

AMI ベースの製品の作成

Amazon マシンイメージ (AMI) セルフサービスエクスペリエンスは、製品リストの作成や変更リクエストを行う際に役立ちます。セルフサービスエクスペリエンスを使用すると、AWS Marketplace Seller Operations チームによる処理に必要な時間を短縮して、製品リストを直接更新できます。セルフサービスエクスペリエンスの多くのステップは、のカatalogシステムと一致します。これにより AWS Marketplace、AWS Marketplace Seller Operations チームによる処理と検証を待つ代わりに、直接検証が容易になります。このトピックでは、AMI セルフサービスエクスペリエンスを使用して単一 AMI 製品リストを作成する方法について説明します。顧客は AMI を使用して、製品をインストールして設定済みの Amazon EC2 インスタンスを作成できます。

トピック

- [前提条件](#)
- [セルフサービスエクスペリエンスを理解する](#)
- [リストを作成する](#)
- [追加リソース](#)

前提条件

AMI 製品リストを作成するには、最初に次の前提条件を満たす必要があります。

1. AWS Marketplace 管理ポータルにアクセスできます。これは、販売者として登録し、販売する製品を管理するために使用するツールです AWS Marketplace。へのアクセスの詳細については

AWS Marketplace 管理ポータル、「」を参照してください[AWS Marketplace 販売者のポリシーとアクセス許可](#)。

2. 販売者として登録し、製品の代金を請求する場合は、税金と銀行情報を送信してください。販売者になる方法については、「[AWS Marketplace 販売者としての開始方法](#)」を参照してください。
3. 売りたい製品がある。AMI ベース製品の場合、これは通常、サーバーソフトウェアを作成または変更し、顧客が使用できる AMI を作成したことを意味します。で使用する AMI の準備の詳細については AWS Marketplace、「」を参照してください[で使用する AMIs を構築するためのベストプラクティス AWS Marketplace](#)。

セルフサービスエクスペリエンスを理解する

セルフサービスエクスペリエンスでは、で製品を作成する手順を説明します AWS Marketplace。手順を進める際には、製品情報と、インスタンスタイプ AWS リージョンや AMI の詳細などの AMI デプロイ設定を指定します。また、料金、販売可能な国、EULA、返金ポリシーなどの取引の詳細も設定します。オプションとして、制限付きステータスの製品にアクセスしてテストするための AWS アカウント IDs の許可リストを指定できます。

始める前に、セルフサービスエクスペリエンスに関する以下の重要な点を確認してください。

- 次のステップに進むには、現在のステップで必須フィールドに入力する必要があります。この要件は、各ステップの最後にページレベルの検証が行われるためです。不完全なステップは保存または送信できません。
- プロセスのすべてのステップを完了する前にセッションを終了する必要がある場合は、[保存して終了] を選択して、完了したステップをステージングエリアに送信できます。
- 不完全で検証に合格しなかったステップはシステムに送信されません。部分的に完了したステップは無効であり、保存することもできません。
- [保存して終了] を選択すると、[保存して終了] ダイアログボックスに検証チェックに合格したステップが表示されます。最後に完了して検証されたステップまで確認し、保存することを選択できます。検証エラーや詳細が不足している場合は、[修正] を選択してそのステップに戻ることができます。
- [保存して終了] すると、リクエストは処理中にレビューされます。処理の終了には、数分または数時間かかることがあります。リクエストが成功するまで、ステップを続行したり、変更を加えたりすることはできません。最初の [保存して終了] のリクエストは、完了したステップと並行して製品を作成します。
- リクエストが成功したら、保存は完了です。[製品の概要] ページで変更を再開するには、[製品の作成を再開する] を選択するか、[変更をリクエスト] を使用して前回のセッションで送信した

詳細を更新します。再開すると、完了した手順に緑色の [成功] というラベルが付いていることに注目してください。以前に送信したステップを更新するには、[変更をリクエスト] を使用してください。このステップを続行するには、前回の [保存して終了] のリクエストを完了する必要があります。

- すべてのステップを完了したら、[次へ] を選択してレビューを確認できます。[送信] を選択すると、システムから最終検証の実行をリクエストできます。[成功] というレスポンスを受け取ると、製品は [制限あり] ステータスに移行します。詳細ページを確認すると、許可リストに登録されている人なら誰でもその製品を利用できるようになったことがわかります。リクエストが失敗した場合、製品は [ステージング] ステータスのままになり、再提出する前に修正を加える必要があります。

リストを作成する

このセクションのステップでは、単一 AMI 製品のリストを作成する方法について説明します。

Note

次のステップに進むには、現在のステップで必須フィールドに入力する必要があります。不完全なステップは保存または送信できません。プロセスのすべてのステップを完了する前にセッションを終了する必要がある場合は、[保存して終了] を選択すると、完了したステップをステージングエリアに送信できます。詳細については、「[セルフサービスエクスペリエンスを理解する](#)」を参照してください。

単一 AMI 製品を作成する

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [製品] メニューの [サーバー] を選択します。または、[\[サーバー製品\]](#) ページに直接移動することもできます。
3. [サーバー製品] タブから [サーバー製品の作成] を選択し、[Amazon マシンイメージ (AMI)] を選択して、単一 AMI 製品のライセンスタイプから 1 つを選択します。
 - Bring your own license (BYOL) – ユーザーが外部からライセンスを取得する製品 AWS Marketplace。有料ライセンスでも無料ライセンスでもかまいません。

- [無料] - サブスクリバラーが無料で使用できる製品。(関連する Amazon Elastic Compute Cloud (Amazon EC2) インスタンスやその他の AWS リソースに対しては引き続き料金が発生します)。
 - 時間単位または時間単位の年間料金 – 購入者が時間単位または年間契約で時間単位で支払う製品。AWS は、AMI の製品コードに基づいて計測を行います。
 - [月額] - 購入者が AWSによって毎月請求される製品。
 - 有料使用料 - ソフトウェアの料金は、販売者がユーザー、データ、帯域幅、またはホストの 4 つの使用カテゴリのいずれかと併せて提供する価値に対して直接請求されます。製品に対して最大 24 のディメンションを定義できます。すべての料金は引き続きお客様のご負担となります。
 - 契約料金の AMI – 購入者が前払い料金を支払う AWS CloudFormation スタックを持つシングル AMI 製品またはシングル AMI。
4. セルフサービスエクスペリエンスでは、AWS Marketplace リストを作成する手順を案内します。製品情報 (メタデータ)、製品デプロイの詳細 (AWS リージョン、インスタンス、AMI の詳細)、およびパブリックオファーの詳細 (料金、EULA、国別の在庫状況、EULA、返金) を入力する必要があります。オプションとして、許可リストにアカウントを追加して製品をテストできます。各ステップを完了すると、プロセスの次のステップに進みます。

 Note

プロセスのすべてのステップを完了する前にセッションを終了する必要がある場合は、[保存して終了] を使用して、完了したステップをステージングエリアに送信できます。これにより、入力した情報の検証を求めるリクエストが作成されます。リクエストが処理されている間は、製品を編集することはできません。リクエストが成功したら、[製品の作成を再開する] を選択して製品の作成を続行できます。

リクエストが失敗した場合は、検証エラーにより製品が更新されなかったことを意味します。これは製品のリクエストログに表示されます。リクエストを選択してエラーを表示し、[アクション] の [新規にコピー] を使用してエラーを修正し、リクエストを再送信できます。ステップを再開すると、前回のセッションで保存したステップの後のステップから続行できます。以前のステップを更新するには、[製品概要] ページに移動し、以前に送信したステップを更新する変更リクエストを送信します。

5. すべての変更リクエストステップに必要な情報を入力したら、[送信] を選択します。この送信により、情報を検証し、検証に合格した場合に製品を [制限あり] 状態にリリースするよう AWS Marketplace カタログシステムにリクエストが作成されます。リクエストが処理されている間

は、製品の編集を続けることはできません。リクエストが成功すると、製品は [制限あり] 状態に移行します。

- 製品が最初に公開されると、(製品の作成 AWS アカウント に使用したもの) と AWS Marketplace Seller Operations チームのテストアカウントのみがアクセスできます。サーバー製品ページから製品を表示する場合は、で表示 AWS Marketplace を選択して、購入者 AWS Marketplace に表示される製品の詳細を表示できます。この詳細リストは AWS Marketplace 他のユーザーには表示されません。
 - この機能により、製品を公開する前にテストできます (複数のバージョンをテスト用に公開することもできます)。
6. 限定状態で製品をテストし、AWS Marketplace [AMI ベースの製品要件](#)と[製品チェックリスト](#)に従っていることを確認します。次に、製品を [パブリック] に公開するようにリクエストするには、[可視性を更新] を選択します。AWS Marketplace Seller Operations チームは、公開を承認する前に製品を確認する必要があります。

 Note

製品の検証と公開は手動プロセスであり、AWS Marketplace Seller Operations チームによって処理されます。エラーがない場合は、最初の製品バージョンを公開するまでに 7~10 営業日かかることがあります。タイミングについては、「[タイミングと心構え](#)」を参照してください。

単一 AMI 製品と製品情報の両方を準備して提出する方法の詳細については、「[追加リソース](#)」を参照してください。

追加リソース

製品に関する情報を準備し、公開用に送信する方法の詳細については、次のリソースを参照してください。

- [用の製品の準備 AWS Marketplace](#)
- [での公開用に製品を送信する AWS Marketplace](#)

単一 AMI 製品を AWS Marketplace に送信する準備の詳細については、以下のリソースを参照してください。

- [で使用する AMIs を構築するためのベストプラクティス AWS Marketplace](#)
- [の AMI 製品チェックリスト AWS Marketplace](#)
- [の AMI ベースの製品要件 AWS Marketplace](#)

CloudFormation テンプレートを製品に追加する

AWS Marketplace 販売者は、AWS CloudFormation テンプレートを使用して AWS Marketplace 購入者に配信される AMI ベースの製品を一覧表示できます。AMI ベースの製品に CloudFormation テンプレートを追加すると、購入者はリソースと依存関係を手動で設定しなくてもソリューションをデプロイできます。テンプレートを使用して、各製品の分散アーキテクチャまたはクラスターを定義する、または異なる AMI の組み合わせや製品設定を選択できます。単一の AMI ソリューションには、最大 3 つの CloudFormation テンプレートを含めることができます。

CloudFormation テンプレートは、単一の Amazon マシンイメージ (AMI) を、関連する構成ファイルと Lambda 関数と共に配信するように設定できます。さらに、各テンプレートのアーキテクチャ図を含める必要があります。

トピック

- [CloudFormation テンプレートの準備](#)
- [アーキテクチャ図](#)
- [既存の製品の CloudFormation テンプレートを変換する](#)
- [サーバーレスアプリケーションコンポーネントの追加](#)

CloudFormation テンプレートの準備

CloudFormation テンプレートを構築するには、テンプレート的前提条件を満たし、必要な入力とセキュリティパラメータを用意する必要があります。CloudFormation テンプレートを送信する場合は、以下のセクションのガイドラインに従ってください。

テンプレート的前提条件

- テンプレートが、製品に対して AWS リージョン 有効になっているすべての AWS CloudFormation コンソールから正常に起動されていることを確認します。[TaskCat ツール](#)を使用して、テンプレートをテストできます。
- CloudFormation テンプレートに含まれる AMI は、公開する製品の AMI であるか、最新の Amazon Linux 2 などの AWS 管理の AMI である必要があります。コミュニティ AMI または ユーザー

所有またはユーザー共有またはサードパーティ共有の AMI は含まないでください。AWS マネージド AMI を使用するには、AMI ID をハードコーディングする代わりに、[AWS Systems Manager パラメータストアでパブリックパラメータ](#)を使用します。IDs 例えば、AMI ID を指定する CloudFormation テンプレート内では、動的リファレンス ImageId: `'{{resolve:ssm:/aws/service/ecs/optimized-ami/amazon-linux-2/recommended/image_id}}'` を使用します。

- 特定のアベイラビリティーゾーン (AZ) の使用に依存しないようにテンプレートを構築します。すべてのアベイラビリティーゾーンにすべての顧客がアクセスできるとは限りません。AZ はさまざまなアカウントに別々にマッピングされます。
- Auto Scaling グループを使用してクラスター化されたソリューションを構築している場合は、スケールイベントを考慮することをお勧めします。新規ノードは、自動的に動作クラスターを結合する必要があります。
- 単一ノードの製品であっても、[Auto Scaling グループ](#)を使用することをお勧めします。
- ソリューションに、複数のインスタンスのクラスターが含まれる場合で、インスタンス間でネットワークレイテンシーが低い、またはネットワークスループットが高い、あるいはその両方の場合、プレイスメントグループを使用することを検討してください。
- AWS Marketplace チームによるレビューを容易にし、顧客に透明性を持たせるために、UserData セクションにコメントを追加することをお勧めします。

AMI の詳細の要件

Note

販売者ポータル[のサーバー製品](#)ページで CloudFormation を使用して Amazon マシンイメージ (AMI) または AMI を作成し、[製品ロードフォーム](#)のダウンロードを求めるプロンプトが表示された場合は、[the section called “製品ロードフォームを使用した AMI の詳細の要件”](#)代わりに「」を参照してください。

AMI を

[AWS::EC2::Instance](#)、[AWS::AWS::AutoScaling::LaunchConfiguration](#)、[AWS::EC2::LaunchTemplate](#) リソースなどの EC2 インスタンスにデプロイするリソースの ImageId プロパティを指定する場合は、[テンプレートパラメータを参照する必要があります](#)。[AWS::EC2::LaunchTemplate](#) パラメータタイプは、`AWS::EC2::Image::Id`、`AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`、またはのいずれかである必要があります String。

このテンプレートパラメータには、任意の有効なパラメータ名を指定できます。は、テンプレートを独自の Amazon S3 バケット AWS Marketplace にコピーし、指定されたパラメータを [AWS Systems Manager Parameter Store](#) パラメータに置き換えます。AWS Marketplace また、は説明と制約テキストを更新して、テンプレートをデプロイする購入者に正しい値を明確にします。購入者がテンプレートをデプロイすると、そのパラメータは公開済み製品の AWS リージョン固有の AMI ID に解決されます。

次のテンプレート例は、組み込み関数 [Ref](#) を使用してテンプレートパラメータを参照する ImageId プロパティを示しています。

YAML の例 :

```
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example1234567890
Resources:
  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
```

JSON の例 :

```
{
  "Parameters": {
    "ImageId": {
      "Type": "AWS::EC2::Image::Id",
      "Default": "ami-example1234567890"
    }
  },
  "Resources": {
    "MyInstance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": {
          "Ref": "ImageId"
        }
      }
    }
  }
}
```

ルートスタックではなくネストされたスタック内に EC2 インスタンスをデプロイする場合、AMI ID はルートスタックからその値を動的に継承する必要があります。ルートスタックとネストされたスタックを編集して、ルートスタックでテンプレートパラメータの値を設定すると、このネストされたスタックで使用する AMI ID が上書きされるようにします。

製品ロードフォームを使用した AMI の詳細の要件

Note

販売者ポータル[のサーバー製品](#)ページで CloudFormation を使用して Amazon マシンイメージ (AMI) または AMI を作成し、[製品ロードフォーム](#)のダウンロードをすぐに求められない場合は、[the section called “AMI の詳細の要件”](#)「」を参照してください。

AMI は各リージョンのマッピングテーブルに存在する必要があります。AWS Marketplace チームはクローン作成後に AMI IDs を更新します。ソース AMI は us-east-1リージョンにある必要があります。他のリージョンではプレースホルダーを使用できます。

YAML の例 :

```
Mappings:
  RegionMap:
    us-east-1:
      ImageId: ami-0123456789abcdef0
    us-west-1:
      ImageId: ami-xxxxxxxxxxxxxxxxxxx
    eu-west-1:
      ImageId: ami-xxxxxxxxxxxxxxxxxxx
    ap-southeast-1:
      ImageId: ami-xxxxxxxxxxxxxxxxxxx
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap
        - RegionMap
        - !Ref AWS::Region
        - ImageId
```

ネストされたスタックテンプレートの要件

Note

このセクションは、[製品ロードフォーム](#)を使用しない料金モデルにのみ適用されます。製品ロードフォームを使用する料金モデルの場合、ネストされたスタックTemplateURLプロパティには固定文字列のみが許可されます。

テンプレートに[ネストされたスタック](#)が含まれている場合、ネストされたスタックリソースのTemplateURLプロパティは、Amazon S3 バケット名、バケットリージョン、Amazon S3 オブジェクトキープレフィックスのテンプレートパラメータを参照する必要があります。バケット名のパラメータ名は MPS3BucketName、バケットリージョンは MPS3BucketRegion、オブジェクトキープレフィックスは である必要がありますMPS3KeyPrefix。

ネストされたテンプレートが保存されている Amazon S3 バケットに対応するように、これらのパラメータのデフォルト値を設定します。ネストされたテンプレートはすべてパブリックにアクセス可能である必要があります。公開用のテンプレートを送信すると、 はテンプレートを独自の Amazon S3 バケット AWS Marketplace にコピーし、これら 3 つのパラメータのプロパティを変更して、コピーの保存先に対応するようにデフォルト値と許容値を設定します。AWS Marketplace また、 は説明と制約テキストを更新して、テンプレートをデプロイする購入者に正しい値を明確にします。

複数のレベルのネストされたスタックがある場合、追加のネストされたスタックを作成するすべてのネストされたスタックは、TemplateURLプロパティがルートスタックから Amazon S3 バケット名、Amazon S3 バケットリージョン、および Amazon S3 オブジェクトキーの値を動的に継承するように設定する必要があります。ルートスタックとネストされたスタックを編集して、ルートスタックでテンプレートパラメータ MPS3BucketName、 、 の値を設定して、このネストされたスタックで使用する URL 内のそれぞれの値をMPS3KeyPrefix上書きしMPS3BucketRegion、追加のネストされたスタックを作成します。

次のテンプレート例は、組み込み関数 [Fn::Sub](#) を使用してテンプレートパラメータを参照するTemplateURLプロパティを示しています。

YAML の例：

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
```

```

- Label:
  default: AWS Marketplace Parameters
Parameters:
  - ImageId
  - MPS3BucketName
  - MPS3BucketRegion
  - MPS3KeyPrefix
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example1234567890
    Description: The AMI that will be used to launch EC2 resources.
  MPS3BucketName:
    Type: String
    Default: sellerbucket
    Description: Name of the S3 bucket for your copy of the nested templates.
  MPS3BucketRegion:
    Type: String
    Default: us-east-1
    Description: AWS Region where the S3 bucket for your copy of the nested templates
is hosted.
  MPS3KeyPrefix:
    Type: String
    Default: sellerproductfolder/
    Description: S3 key prefix that is used to simulate a folder for your copy of the
nested templates.
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml

```

JSON の例 :

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Metadata": {
    "AWS::CloudFormation::Interface": {

```

```

        "ParameterGroups": [
            {
                "Label": {
                    "default": "AWS Marketplace Parameters"
                },
                "Parameters": [
                    "ImageId",
                    "MPS3BucketName",
                    "MPS3BucketRegion",
                    "MPS3KeyPrefix"
                ]
            }
        ],
        "Parameters": {
            "ImageId": {
                "Type": "AWS::EC2::Image::Id",
                "Default": "ami-example1234567890",
                "Description": "The AMI that will be used to launch EC2 resources."
            },
            "MPS3BucketName": {
                "Type": "String",
                "Default": "sellerbucket",
                "Description": "Name of the S3 bucket for your copy of the nested
templates."
            },
            "MPS3BucketRegion": {
                "Type": "String",
                "Default": "us-east-1",
                "Description": "AWS Region where the S3 bucket for your copy of the nested
templates is hosted."
            },
            "MPS3KeyPrefix": {
                "Type": "String",
                "Default": "sellerproductfolder/",
                "Description": "S3 key prefix that is used to simulate a folder for your
copy of the nested templates."
            }
        },
        "Resources": {
            "EC2Instance": {
                "Type": "AWS::EC2::Instance",
                "Properties": {

```

```
        "ImageId": {
            "Ref": "ImageId"
        }
    },
    "NestedStack": {
        "Type": "AWS::CloudFormation::Stack",
        "Properties": {
            "TemplateURL": {
                "Fn::Sub": "https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml"
            }
        }
    }
}
```

Note

[AWS::CloudFormation::Interface](#) は、購入者がテンプレートをデプロイするときに AWS CloudFormation コンソールでパラメータをグループ化およびソートする方法を定義するために使用されます。

テンプレート入力パラメータ

- テンプレートへの入力パラメータには、AWS Marketplace 顧客の AWS 認証情報 (パスワード、パブリックキー、プライベートキー、証明書など) を含めることはできません。
- パスワードなどの機密入力パラメータには、NoEcho プロパティを選択し、強力な正規表現を有効にします。他の入力パラメータでは、最も一般的な入力を適切なヘルパーテキストで設定します。
- 使用可能な入力には AWS CloudFormation パラメータタイプを使用します。
- `AWS::CloudFormation::Interface` を使用し入力パラメータをグループ化およびソートします。
- 以下の入力パラメータにはデフォルト値を設定しないでください。

Note

顧客は、これらを入力パラメータとして指定する必要があります。

- パブリックインターネットからリモートアクセスポートへの進入を許可するデフォルトの CIDR 範囲
- パブリックインターネットからデータベース接続ポートへの進入を許可するデフォルトの CIDR 範囲
- ユーザーまたはデータベースのデフォルトパスワード

ネットワークおよびセキュリティパラメータ

- デフォルトの SSH ポート (22) または RDP ポート (3389) が 0.0.0.0 に対して開かれていないことを確認します。
- デフォルトの VPC (仮想プライベートクラウド) を使用する代わりに、適切な ACL (アクセスコントロールリスト) とセキュリティグループを設定した VPC を作成することをお勧めします。
- テンプレートは、ユーザーに長期的なアクセスキーをリクエストしたり、AWS リソースにアクセスするために作成したりすることはできません。AMI アプリケーションが購入者のアカウント内の AWS サービスにアクセスする必要がある場合は、[Amazon EC2 の IAM ロール](#)を使用する必要があります。
- IAM ロールとポリシーに[最小特権を付与](#)し、絶対に必要な場合のみに書き込みアクセスを有効にします。たとえば、アプリケーションで S3:GET、PUT および DELETE オペレーションのみが必要な場合に、これらのアクションのみを指定します。この場合に、S3:* の使用はお勧めしません。

テンプレートを受信すると、は製品設定と情報 AWS Marketplace を検証し、必要なリビジョンに関するフィードバックを提供します。

アーキテクチャ図

各テンプレートのアーキテクチャ図を用意する必要があります。ダイアグラムの詳細については、「[アーキテクチャダイアグラムとは](#)」を参照してください。

図は次の基準を満たしている必要があります。

- での標準デプロイの例を示します AWS。
- リソースがデプロイされる場所を論理的に表現します。例えば、Amazon EC2 インスタンスなどのリソースは正しいサブネットにあります。

- AWS CloudFormation テンプレートを通じて AWS のサービス デプロイされた各 の最新の AWS 製品アイコンを使用します。現在のアーキテクチャアイコンのセットをダウンロードするには、「[AWS アーキテクチャアイコン](#)」を参照してください。
- AWS CloudFormation テンプレートによってデプロイされたすべてのサービスのメタデータを含めます。
- AWS CloudFormation テンプレートによってデプロイされたすべてのネットワーク、VPCs、サブネットを含めます。
- サードパーティーのアセット、API およびオンプレミス、ハイブリッドアセットを含む、統合ポイントを表示します。
- この図のサイズは 1100 x 700 ピクセルである必要があります。元の図の比率は、伸縮やトリミングを行わずに維持します。

既存の製品の CloudFormation テンプレートを変換する

Note

このセクションは、製品[ロードフォーム](#)を使用してテンプレートを発行し、製品ロードフォームを使用せずにそのテンプレートを更新する CloudFormation 製品を持つ既存の AMI を持つ販売者を対象としています。新しい製品を公開する場合は、[CloudFormation テンプレートの準備](#)」を参照してください。

販売者ポータル[のサーバー製品](#)ページで CloudFormation を使用して Amazon マシンイメージ (AMI) または AMI を作成し、[製品ロードフォーム](#)のダウンロードを求めるプロンプトが表示された場合は、「」を参照してください[the section called “製品ロードフォームを使用した AMI の詳細の要件”](#)。

セルフサービスエクスペリエンスを使用して、以前に製品[ロードフォーム](#)を使用して公開した既存の製品を更新する場合は、既存の CloudFormation テンプレートを変更する必要があります。

次の表に、製品ロードフォームとセルフサービスエクスペリエンスの使用の違いを示します。

	製品ロードフォーム	セルフサービスエクスペリエンス
EC2 リソースの ImageId プロパティの値	AMI ID のマッピングテーブルを参照します。詳細について	AMI ID のテンプレートパラメータを参照します。詳細につ

	製品ロードフォーム	セルフサービスエクスペリエンス
	は、「 the section called “製品ロードフォームを使用したAMIの詳細の要件” 」を参照してください。	いては、「 the section called “AMIの詳細の要件” 」を参照してください。
ネストされたスタックの TemplateURL プロパティの値	固定文字列でなければならず、組み込み関数を使用できません。	組み込み関数を使用して動的にすることができます。一連のテンプレートパラメータを参照する必要があります。詳細については、 the section called “ネストされたスタックテンプレートの要件” を参照してください。

次のサンプルテンプレートは、製品ロードフォームを使用してテンプレートを発行した既存の製品の例を示しています。この例では、AMI ID は `ami-example123456` で、ネストされたテンプレートは の場所にある販売者の S3 バケットにあります `https://sellerbucket.s3.us-east-1.amazonaws.com/sellerproductfolder/nested-template.yaml`。

製品ロードフォームで公開された YAML の例：

```
AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-example123456
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap
        - RegionMap
        - !Ref AWS::Region
        - AMI
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
```

```
TemplateURL: https://sellerbucket.s3.us-east-1.amazonaws.com/sellerproductfolder/nested-template.yaml
```

製品ロードフォームで公開された JSON の例 :

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Mappings": {
    "RegionMap": {
      "us-east-1": {
        "AMI": "ami-example123456"
      }
    }
  },
  "Resources": {
    "EC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": {
          "Fn::FindInMap": [
            "RegionMap",
            {
              "Ref": "AWS::Region"
            }
          ],
          "AMI"
        }
      }
    },
    "NestedStack": {
      "Type": "AWS::CloudFormation::Stack",
      "Properties": {
        "TemplateURL": "https://sellerbucket.s3.us-east-1.amazonaws.com/sellerproductfolder/nested-template.yaml"
      }
    }
  }
}
```

次のテンプレート例は、セルフサービスエクスペリエンスを使用して製品を更新するために必要な変更を示しています。

セルフサービスエクスペリエンスで公開された YAML の例 :

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: AWS Marketplace Parameters
        Parameters:
          - ImageId
          - MPS3BucketName
          - MPS3BucketRegion
          - MPS3KeyPrefix
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example123456
    Description: The AMI that will be used to launch EC2 resources.
  MPS3BucketName:
    Type: String
    Default: sellerbucket
    Description: Name of the S3 bucket for your copy of the nested templates.
  MPS3BucketRegion:
    Type: String
    Default: us-east-1
    Description: AWS Region where the S3 bucket for your copy of the nested templates
is hosted.
  MPS3KeyPrefix:
    Type: String
    Default: sellerproductfolder/
    Description: S3 key prefix that is used to simulate a folder for your copy of the
nested templates.
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml
```

セルフサービスエクスペリエンスで公開された JSON の例 :

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {
            "default": "AWS Marketplace Parameters"
          },
          "Parameters": [
            "ImageId",
            "MPS3BucketName",
            "MPS3BucketRegion",
            "MPS3KeyPrefix"
          ]
        }
      ]
    },
    "Parameters": {
      "ImageId": {
        "Type": "AWS::EC2::Image::Id",
        "Default": "ami-example123456",
        "Description": "The AMI that will be used to launch EC2 resources."
      },
      "MPS3BucketName": {
        "Type": "String",
        "Default": "sellerbucket",
        "Description": "Name of the S3 bucket for your copy of the nested
templates."
      },
      "MPS3BucketRegion": {
        "Type": "String",
        "Default": "us-east-1",
        "Description": "AWS Region where the S3 bucket for your copy of the nested
templates is hosted."
      },
      "MPS3KeyPrefix": {
        "Type": "String",
        "Default": "sellerproductfolder/",
        "Description": "S3 key prefix that is used to simulate a folder for your
copy of the nested templates."
      }
    }
  }
}
```

```

    },
    "Resources": {
      "EC2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
          "ImageId": {
            "Ref": "ImageId"
          }
        }
      },
    },
    "NestedStack": {
      "Type": "AWS::CloudFormation::Stack",
      "Properties": {
        "TemplateURL": {
          "Fn::Sub": "https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml"
        }
      }
    }
  }
}

```

サーバーレスアプリケーションコンポーネントの追加

Important

AWS Marketplace は、リソースをデプロイする CloudFormation テンプレートを使用した新製品の公開をサポートしなくなりました AWS Serverless Application Repository。販売者は、 から AWS Serverless Application Repository 将来の発表日までリソースをデプロイする CloudFormation テンプレートを使用して既存の製品を引き続き公開できます。

AWS CloudFormation テンプレートを使用して配信される Amazon マシンイメージ (AMI) を含む製品を作成し、サーバーレスコンポーネントを製品に組み込むことができます。たとえば、コントローラサーバーとして設定された AMI を使用して製品を作成し、AWS CloudFormation スタックとして配信できます。スタックの作成に使用される AWS CloudFormation テンプレートには、サーバー内のイベントによってトリガーされる AWS Lambda 関数を設定する定義を含めることができます。このアプローチを使用して製品を設計すると、アーキテクチャを簡素化し、購入者が簡単に起動できるようになります。また、このアプローチにより、製品の更新が容易になります。以下のセクションでは、このタイプの製品を作成して提供する方法を示します。

製品の AMI の作成については、「」を参照してくださいの [AMI ベースの製品 AWS Marketplace](#)。
製品の AWS CloudFormation テンプレートを完了する方法については、「」を参照してください [CloudFormation テンプレートを製品に追加する](#)。

サーバーレスアプリケーションを定義するときは、に保存する AWS Serverless Application Model (AWS SAM) テンプレートを使用します AWS Serverless Application Repository。AWS SAM は、サーバーレスアプリケーションを構築するためのオープンソースフレームワークです。デプロイ中、は AWS Serverless Application Model 構文を AWS SAM 変換して AWS CloudFormation 構文に拡張します。AWS Serverless Application Repository は、サーバーレスアプリケーション用のマネージド型のリポジトリです。再利用可能なアプリケーションを保存して共有できるため、購入者はサーバーレスアーキテクチャを構築してデプロイできます。

Note

- AWS Marketplace は、出品が作成される前に製品を確認して検証します。オファーがリスト化される前に解決しなければならない問題がある場合、メールメッセージでお知らせします。
- サブスクリプションの受理の一環として、AMIs、サーバーレスアプリケーション、テンプレート AWS CloudFormation を各 の AWS Marketplace 所有リポジトリにコピーします AWS リージョン。購入者が製品をサブスクライブすると、購入者にアクセス権が付与され、ソフトウェアの更新時にも通知されます。

トピック

- [ステップ 1: サーバーレスアプリケーションを作成する](#)
- [ステップ 2: アプリケーションをリポジトリに公開する](#)
- [ステップ 3: CloudFormation テンプレートの作成](#)
- [ステップ 4: CloudFormation テンプレートと設定ファイルの送信](#)
- [ステップ 5: AWS Serverless Application Repository アプリケーションのアクセス許可を更新する](#)
- [ステップ 6: AMI を共有する](#)
- [ステップ 7: AMI とサーバーレスアプリケーションを使用する CloudFormation 製品を送信する](#)

ステップ 1: サーバーレスアプリケーションを作成する

最初のステップは、サーバーレスアプリケーションの作成に使用される AWS Lambda 関数をパッケージ化することです。アプリケーションは、Lambda 関数、イベントソース、その他のリソースを組み合わせたもので、協調して動作することによりタスクを実行します。サーバーレスアプリケーションは、1 つの Lambda 関数と同じくらいシンプルにすることも、APIs、データベース、イベントソースマッピングなどの他のリソースを含む複数の関数を含めることもできます。

を使用して AWS SAM、サーバーレスアプリケーションのモデルを定義します。プロパティ名とタイプの詳細については、GitHub の AWS「Labs」の [AWS::Serverless::Application](#) を参照してください。以下は、単一の Lambda 関数と AWS Identity and Access Management (IAM) ロールを持つ AWS SAM テンプレートの例です。

```
AWS::Serverless::Application
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: An example of SAM template with Lambda function and IAM role

Resources:
  SampleFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: 'com.sampleproject.SampleHandler::handleRequest'
      Runtime: java8
      CodeUri: 's3://amzn-s3-demo-bucket/2EXAMPLE-1234-4b12-ac37-515EXAMPLEe5-lambda.zip'
      Description: Sample Lambda function
      Timeout: 120
      MemorySize: 1024
      Role:
        Fn::GetAtt: [SampleFunctionRole, Arn]

  # Role to execute the Lambda function
  SampleFunctionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: 'Allow'
            Principal:
              Service:
                - 'lambda.amazonaws.com'
            Action: 'sts:AssumeRole'
```

```
ManagedPolicyArns:
  - "arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
Policies:
  - PolicyName: SFNXDeployWorkflowDefinitionPolicy
    PolicyDocument:
      Statement:
        - Effect: "Allow"
          Action:
            - "s3:Get*"
          Resource: "*"
    RoleName: "SampleFunctionRole"
```

ステップ 2: アプリケーションをリポジトリに公開する

アプリケーションを公開するには、最初にアプリケーションコードをアップロードします。アカウントが所有する Amazon S3 バケットにコードアーティファクト (Lambda 関数、スクリプト、構成ファイルなど) を保存します。アプリケーションをアップロードすると、最初はプライベートに設定されます。つまり、アプリケーションを AWS アカウント 作成した のみ使用できます。アップロードしたアーティファクトにアクセスするための AWS Serverless Application Repository アクセス許可を付与する IAM ポリシーを作成する必要があります。

サーバーレスアプリケーションをサーバーレスアプリケーションリポジトリに公開するには

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. アプリケーションのパッケージ化に使用した Amazon S3 バケットを選択します。
3. [アクセス許可] タブを選択します。
4. [バケットポリシー] を選択します。
5. 次のポリシーステートメントの例をコピーして貼り付けます。

Note

ポリシーステートメントの例では、次の手順で `aws:SourceAccount` および `Resource` の値が更新されるまでエラーが生成されます。

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "serverlessrepo.amazonaws.com"
    },
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
]
```

- a. Resource プロパティ値の amzn-s3-demo-bucket をバケットのバケット名に置き換えます。
 - b. Condition 要素の 123456789012 を AWS アカウント ID に置き換えます。Condition 要素により、AWS Serverless Application Repository にのみ、指定された からアプリケーションにアクセスするアクセス許可が付与されます AWS アカウント。
6. [保存] を選択します。
 7. で AWS Serverless Application Repository コンソールを開きます<https://console.aws.amazon.com/serverlessrepo>。
 8. [My Applications (マイアプリケーション)] ページで、[Create application (アプリケーションの作成)] を選択します。
 9. 必要に応じて、必須フィールドと任意のフィールドに入力します。必須フィールドは次のとおりです。
 - アプリケーション名
 - 筆者
 - 説明
 - ソースコード URL
 - SAM template
 10. [Publish application (アプリケーションの公開)] を選択します。

アプリケーションの後続バージョンを公開するには

1. で AWS Serverless Application Repository コンソールを開きます <https://console.aws.amazon.com/serverlessrepo>。
2. [ナビゲーションペイン] で、[My Applications (マイアプリケーション)] からアプリケーションを選択します。
3. [新しいバージョンを発行] を選択します。

詳細については、[「AWS SAM CLI を使用したサーバーレスアプリケーションの発行」](#)を参照してください。

ステップ 3: CloudFormation テンプレートの作成

CloudFormation テンプレートを構築するには、テンプレートの前提条件を満たし、必要な入力とセキュリティパラメータを用意する必要があります。詳細については、「AWS CloudFormation ユーザーガイド」の[「テンプレートの構造分析」](#)を参照してください。

CloudFormation テンプレートで、サーバーレスアプリケーションと AMI を参照できます。ネストされた CloudFormation テンプレートを使用して、ルートテンプレートとネストされたテンプレートの両方でサーバーレスアプリケーションを参照することもできます。サーバーレスアプリケーションを参照するには、AWS SAM テンプレートを使用します。からアプリケーションの AWS SAM テンプレートを自動的に生成できます AWS Serverless Application Repository。以下はテンプレートの例です。

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: An example root template for a SAR application

Resources:
  SampleSARApplication:
    Type: AWS::Serverless::Application
    Properties:
      Location:
        ApplicationId: arn:aws:serverlessrepo:us-east-1:1234567890:applications/
TestApplication
  SemanticVersion: 1.0.0
  SampleEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: "ami-79fd7eee"
```

```
KeyName: "testkey"
BlockDeviceMappings:
  - DeviceName: "/dev/sdm"
    Ebs:
      VolumeType: "io1"
      Iops: "200"
      DeleteOnTermination: "false"
      VolumeSize: "20"
  - DeviceName: "/dev/sdk"
    NoDevice: {}
```

AWS SAM テンプレートには以下の要素が含まれています。

- **ApplicationID** - アプリケーションの Amazon リソースネーム (ARN)。この情報は、AWS Serverless Application Repository の [マイアプリケーション] セクションにあります。
- **SemanticVersion** - サーバーレスアプリケーションのバージョン。このフォームは、AWS Serverless Application Repository の [マイアプリケーション] セクションにあります。
- **Parameter (オプション)** - アプリケーションのパラメータ。

Note

ApplicationID および SemanticVersion では、[組み込み関数](#)はサポートされていません。これらの文字列はハードコーディングする必要があります。ApplicationID は、クローンの作成時に更新されます AWS Marketplace。

CloudFormation テンプレートで設定ファイルおよびスクリプトファイルを参照する場合は、次に示す形式に従ってください。ネストされたテンプレート (AWS::CloudFormation::Stack) では、組み込み関数がない TemplateURLs でのみサポートされます。テンプレートの Parameters の内容に注意してください。

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  Name: Seller test product
Parameters:
  CFRefFilesBucket:
    Type: String
    Default: "seller-bucket"
  CFRefFilesBucketKeyPrefix:
```

```

    Type: String
    Default: "cftsolutionFolder/additionCFfiles"
Resources:
  TestEc2:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        addCloudAccount:
          files:
            /etc/cfn/set-aia-settings.sh:
              source:
                Fn::Sub:
                  - https://${CFTRefFilesBucket}.${S3Region}amazonaws.com/
                    ${CFTRefFilesBucketKeyPrefix}/sampleScript.sh
                  - S3Region:
                      !If
                        - GovCloudCondition
                        - s3-us-gov-west-1
                        - s3
                      owner: root
                      mode: '000700'
                      authentication: Amazon S3AccessCreds
          ..
          ..
          ..
  SampleNestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: 'https://sellerbucket.s3.amazonaws.com/sellerproductfolder/
nestedCft.template'
      Parameters:
        SampleParameter: 'test'
Transform: AWS::Serverless-2016-10-31

```

ステップ 4: CloudFormation テンプレートと設定ファイルの送信

CloudFormation テンプレートと設定ファイルおよびスクリプトファイルを送信するには、これらのファイルが保存されている Amazon S3 バケットを読み取るためのアクセス許可を AWS Marketplace に付与します。これを行うには、バケットポリシーを更新して以下のアクセス許可を含めます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "assets.marketplace.amazonaws.com"
      },
      "Action": ["s3:GetObject", "s3:ListBucket"],
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"]
    }
  ]
}
```

ステップ 5: AWS Serverless Application Repository アプリケーションのアクセス許可を更新する

AWS Serverless Application Repository アプリケーションを送信するには AWS Marketplace、アプリケーションを読み取るアクセス AWS Marketplace 許可を付与する必要があります。これを行うには、サーバーレスアプリケーションに関連付けられているポリシーにアクセス許可を追加します。アプリケーションポリシーを更新するには 2 つの方法があります。

- [AWS Serverless Application Repository](#) に移動します。リストからサーバーレスアプリケーションを選択します。[共有] タブを選択し、[ステートメントを作成] を選択します。[ステートメントの設定] ページの [アカウント ID] フィールドに、次のサービスプリンシパル **assets.marketplace.amazonaws.com** を入力します。次に、[保存] を選択します。
- 次の AWS CLI コマンドを使用して、アプリケーションポリシーを更新します。

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements Principals=assets.marketplace.amazonaws.com,Actions=Deploy
```

ステップ 6: AMI を共有する

に構築および送信されるすべての AMIs AWS Marketplace は、すべての製品ポリシーに従う必要があります。AWS Marketplace 管理ポータルではセルフサービス AMI スキャンを利用できます。この機能により、AMI のスキャンを開始できます。スキャン結果を迅速に (通常は 1 時間以内に) 受信でき、1 か所で明確なフィードバックを得ることができます。AMI が正常にスキャンされたら、製品ロードフォームをアップロードして AWS Marketplace Seller Operations チームによる処理のために AMI を送信します。

ステップ 7: AMI とサーバーレスアプリケーションを使用する CloudFormation 製品を送信する

製品を送信する前に、次の点に注意してください。

- 各テンプレートのアーキテクチャ図を用意する必要があります。この図では、AWS CloudFormation テンプレートを介してデプロイされた各 AWS サービスの製品アイコンを使用する必要があります。また、ダイアグラムにはサービスのメタデータが含まれている必要があります。公式 AWS アーキテクチャアイコンをダウンロードするには、[AWS「アーキテクチャアイコン」](#)を参照してください。
- 購入者に表示される各テンプレートのインフラストラクチャの料金見積もりは、[AWS 料金計算ツール](#)を使用して提示した見積もりに基づきます。見積もりには、一般的なデプロイ向けのデフォルト値と共にテンプレートの一部としてデプロイされるサービスのリストを含めます。
- 製品ロードフォームに入力します。AWS Marketplace 管理ポータルから製品ロードフォームを見つけることができます。単一 AMI 製品と複数 AMI 製品には、異なる製品ロードフォームが必要です。製品ロードフォームで、CloudFormation テンプレートのパブリック URL を指定します。CloudFormation テンプレートは、パブリック URL の形式で送信する必要があります。
- AWS Marketplace 管理ポータルを使用して出品を送信します。[Assets (アセット)] から [File upload (ファイルのアップロード)] を選択し、ファイルをアタッチして、[Upload (アップロード)] を選択します。テンプレートとメタデータを受信すると、はリクエストの処理 AWS を開始します。

出品を送信すると、は製品のロードフォーム AWS Marketplace を確認して検証します。さらに、は AMIs とサーバーレスアプリケーションを AWS Marketplace リージョン化し、ユーザーに代わって AWS CloudFormation テンプレートのリージョンマッピングを更新します。問題が発生した場合は、AWS Marketplace Seller Operations チームが E メールで連絡します。

販売者として AWS Marketplace の AMI ベースの製品の管理

AWS Marketplace 販売者は、単一の [Amazon マシンイメージ \(AMIs\)](#) 製品を管理および更新できます。AMI ベースの製品には、ソフトウェアの 1 つ以上のバージョンのセットと、製品全体に関するメタデータが含まれます。製品を作成するときは、製品名、説明 AWS Marketplace、料金など、でそのプロパティを設定します。また、関連する検索で商品が表示されるように、商品に適したカテゴリを決定し、キーワードを追加します。単一 AMI 製品を作成したら、変更リクエストを送信すると、製品またはバージョンを変更することができます。

AMI ベースの製品についてリクエストできる変更のタイプは次のとおりです。

- 購入者に表示される製品情報を更新する。
- 購入者に表示されるバージョン情報を更新する。
- 製品の新しいバージョンを追加する。
- 新しい購入者がそのバージョンにアクセスできないようにバージョンを制限する。
- 製品が利用可能な AWS リージョン を更新します。
- 製品の料金設定とインスタンスタイプを更新する。
- から製品を削除します AWS Marketplace。

以下のトピックでは、単一 AMI 製品を管理および更新する方法について説明します。

トピック

- [での AMI ベースの製品の変更リクエストの作成 AWS Marketplace](#)
- [AMI ベースの製品の可視性の更新](#)
- [の AMI インスタンスの追加と制限 AWS Marketplace](#)
- [での AMI ベースの製品のバージョン管理 AWS Marketplace](#)
- [での AMI ベースの製品情報の更新 AWS Marketplace](#)
- [および国別の AWS リージョン AMI ベースの製品の可用性の管理](#)
- [AMI ベースの AWS Marketplace 製品のエンドユーザーライセンス契約 \(EULA\) の更新](#)
- [での AMI ベースの製品の返金ポリシーの更新 AWS Marketplace](#)
- [AMI AWS Marketplace へのアクセスの許可](#)
- [AWS Marketplaceからの製品の削除](#)
- [AWS Marketplace の変更リクエストの一般的なエラーのトラブルシューティング](#)

での AMI ベースの製品の変更リクエストの作成 AWS Marketplace

で製品またはバージョンを変更するには AWS Marketplace、を通じて変更リクエストを送信します AWS Marketplace 管理ポータル。変更リクエストはキューに追加され、リクエストのタイプによっては解決までに数分から数日かかる場合があります。リクエストのステータスは、AWS Marketplace 管理ポータルで確認できます。このトピックでは、セルフサービスエクスペリエンスを使用するオプションなど AWS Marketplace、で単一 AMI 製品の変更リクエストを作成するために使用できる手順について説明します。

以下の状況では、変更リクエストを作成できます。

- セルフサービスエクスペリエンスを使用して単一 AMI 製品リストを作成しているときに、進行中のステップを保存したが、プロセス全体を完了していません。残りのステップを完了するには、変更リクエストを作成します。
- [制限あり] または [パブリック] 状態の製品の製品情報を変更します。情報を更新するには、変更リクエストを作成します。AMI ベースの製品に対してリクエストできる変更のタイプの詳細については、「[変更リクエストを作成する](#)」を参照してください。

Note

に加えて AWS Marketplace 管理ポータル、[AWS Marketplace Catalog API](#) を使用して変更リクエストを作成することもできます。

トピック

- [セルフサービスを使用して変更リクエストを作成します。](#)
- [変更リクエストを作成する](#)
- [変更リクエストのステータスを取得する](#)
- [追加リソース](#)

セルフサービスを使用して変更リクエストを作成します。

バージョンや製品情報を変更するには、AWS Marketplace 管理ポータルで変更リクエストを作成します。変更リクエストは、製品に変更を加える際に使用するセルフサービスリストの構成要素です。ステップから [保存して終了] を選択して手順を終了するか、更新の [送信] を選択するたびに、変更

リクエストを送信することになります。リクエストは AWS Marketplace 管理ポータル [Request](#) タブで確認できます。

セルフサービスを使用して変更リクエストを作成するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインして、[サーバー製品](#) ページに移動します。
2. [サーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンからオプションを選択します。
4. 変更リクエストを行った後、システムがリクエストを処理するまで待ち時間があり、[レビュー中] と表示されます。リクエストが完了すると、[成功] または [失敗] と表示されます。
 - リクエストが送信されると、[レビュー中]、[変更を準備中]、[変更を適用中] のステータスで処理が開始されます。
 - [成功] とは、リクエストした変更が処理され、変更がシステムに反映されたことを意味します。
 - [失敗] とは、リクエストに何か問題が生じたため、変更が処理されなかったことを意味します。ステータスが、[失敗] の場合は、リクエストを選択して、エラーの修正方法に関する推奨事項を示すエラーコードを検索できます。この時点で、エラーのトラブルシューティングを行い、変更の新しいリクエストを作成できます。処理を速めるには、[新しいリクエストにコピー] を選択して失敗したリクエストの詳細をコピーできます。その後、調整を行ってリクエストを再送信できます。

変更リクエストを作成する

Important

2023 年 6 月 15 日、AWS Marketplace は以下の手順を中止します。2023 年 6 月 15 日以降は、[the section called “セルフサービスを使用して変更リクエストを作成します。”](#) の手順を使用してください。

バージョンや製品情報を変更するには、AWS Marketplace 管理ポータルで変更リクエストを作成します。

変更リクエストを作成するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインして、[サーバー製品](#) ページに移動します。
2. [\[サーバー製品\]](#) タブで、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストからオプションを選択します。

ほとんどの変更リクエストでは、ユーザーインターフェイスのフォームに記入して送信するだけです。ただし、変更によっては、製品積載フォーム (PLF) をダウンロードして記入し、アップロードする必要があります。これは必要な情報を記入するためのフォームが入ったスプレッドシートです。これらの変更リクエストのいずれかを選択すると、作成しようとしているリクエストに対応する正しい PLF をダウンロードするように求められます。PLF には、既存の製品詳細の情報が事前に入力されています。完成した PLF を AWS Marketplace 管理ポータル [ファイルのアップロード](#) ページにアップロードできます。

Note

最新の PLF をダウンロードして使用することを強くお勧めします。フォームは、インスタンスタイプや AWS リージョン などの新しい情報が利用可能になりしだい、定期的に更新されます。製品の最新の PLF は、[\[サーバー製品\]](#) ページで製品を選択し、[\[製品積載フォームをダウンロード\]](#) を選択すると確認できます。

変更リクエストのステータスの詳細については、「[the section called “変更リクエストのステータスを取得する”](#)」を参照してください。変更リクエストで発生する可能性のある問題については、「[AWS Marketplace の変更リクエストの一般的なエラーのトラブルシューティング](#)」を参照してください。

変更リクエストのステータスを取得する

Important

2023 年 6 月 15 日、AWS Marketplace は以下の手順を中止します。この手順は、セルフサービスエクスペリエンスでは不要になります。

変更リクエストを送信すると、AWS Marketplace 管理ポータルの [\[サーバー製品\]](#) ページの [\[リクエスト\]](#) タブでリクエストのステータスを確認できます。ステータスは、次のいずれかになります。

- [レビュー中] とは、リクエストがレビュー中であることを意味します。一部のリクエストでは、AWS Marketplace チームによる手動レビューが必要ですが、ほとんどのリクエストはシステムで自動的にレビューされます。
- [成功] とは、リクエストが完了したことを意味します。製品またはバージョンがリクエストどおりに更新されました。
- [アクションが必要] とは、問題を解決したり、リクエストに関する質問に回答したりするために、リクエストを更新する必要があるということを意味します。リクエストを選択すると、問題を含む詳細が表示されます。
- [失敗] とは、リクエストに何か問題があったことを意味し、同じデータを使用して新しい変更リクエストを作成する必要があります。

追加リソース

特定のタイプの更新の変更リクエストの詳細については、以下のリソースを参照してください。

- [での AMI ベースの製品情報の更新 AWS Marketplace](#)
- [バージョン情報を更新する](#)
- [新しいバージョンを追加する](#)
- [バージョンを制限する](#)

AMI ベースの製品の可視性の更新

AWS Marketplace 販売者は、製品の可視性を更新して、単一の AMI 製品を表示できる購入者を変更できます AWS Marketplace。可視性ステータスが Public に設定されている場合、製品はすべての AWS Marketplace 購入者に表示されます。製品の可視性が Limited に設定されている場合、製品は許可リストに登録した AWS アカウント IDs にのみ表示されます。製品を表示できる AWS アカウント IDs のこの許可リストを管理および更新することもできます。以下のセクションでは、製品の可視性と [限定] 可視性の許可リストを更新する方法を示します。

トピック

- [製品の可視性を更新する](#)
- [許可リスト \(プレビューアカウント\) を更新する](#)

製品の可視性を更新する

どの購入者が製品を表示できるかを変更するには AWS Marketplace、可視性の更新を使用できます。

可視性を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[\[現在のサーバー製品\]](#) タブで、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンから [\[可視性を更新\]](#) を選択します。

Note

この変更リクエストを使用して、製品を [\[制限あり\]](#) ステータスから [\[パブリック\]](#) ステータスに移行するようリクエストできます。ただし、変更リクエストを公開に移行するには、AWS Marketplace Seller Operations チームの承認プロセスを実行する必要があります。

4. [\[変更リクエストを送信\]](#) を選択して、レビューのリクエストを送信します。
5. [\[リクエスト\]](#) タブの [\[リクエストステータス\]](#) が [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスが [\[成功\]](#) になります。

許可リスト (プレビューアカウント) を更新する

限定状態の製品を表示できる AWS アカウント IDs のリストを変更するには、Update allowlist を使用します。

許可リストを更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[\[現在のサーバー製品\]](#) タブで、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンから [\[許可リストを更新\]](#) を選択します。現在のリストには、現在、許可リストに登録されているアカウントのリストが表示されます。
4. 可視性に適した AWS アカウント IDs を追加し、IDs をカンマで区切ります。
5. [\[変更リクエストを送信\]](#) を選択して、レビューのリクエストを送信します。

6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

の AMI インスタンスの追加と制限 AWS Marketplace

AWS Marketplace 販売者は、購入者が単一の Amazon Machine Image (AMI) 製品に使用できるインスタンスを管理できます。購入者が単一 AMI を使用できる新しいインスタンスを追加できます。同様に、新規購入者が特定のインスタンスの単一の AMI 製品を使用できないようにする場合は、インスタンスを制限できます。

Amazon EC2 インスタンスタイプの詳細については、Amazon EC2 ユーザーガイド」の「[使用可能なインスタンスタイプ](#)」を参照してください。

以下のセクションでは、インスタンスを追加および制限する方法について説明します。

トピック

- [インスタンスの追加](#)
- [インスタンスの制限](#)

インスタンスの追加

購入者が単一の AMI として使用できる新しいインスタンスを追加できます。

インスタンスを追加するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [インスタンスを追加] を選択します。
4. インスタンスアーキテクチャを選択します。
5. 使用可能なインスタンスのリストから、追加するインスタンスタイプを選択します。
6. [リクエストを送信] を選択して、レビューのリクエストを送信します。
7. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

 Note

- 現在の料金モデルが無料ではないか、Bring Your Own License (BYOL) モデルを使用している場合は、料金も追加する必要があります。
- 新しいインスタンスの料金でインスタンスを追加を作成した場合、または料金を更新して料金を引き上げる場合、変更を行った日から 90 日以内にセルフサービスを使用してインスタンスを追加することはできません。これらの変更を行うには、[AWS Marketplace Seller Operations チーム](#)に連絡してください。
- 新しいインスタンスタイプのサポートを追加すると、既に製品のプライベートオファーをサブスクライブしている顧客は、新しく追加されたインスタンスを自動的に起動できなくなります。顧客にアクセスさせたいインスタンスを使用して、別のプライベートオファーを作成する必要があります。新しいオファーを承諾すると、お客様は新しく追加されたインスタンスを起動できます。将来の日付に製品をサブスクライブするお客様は、インスタンスがプライベートオファーに含まれている限り、そのインスタンスを起動することもできます。新しいプライベートオファーを作成する方法の詳細については、このガイドの[での契約の修正 AWS Marketplace](#)後半の「」を参照してください。

インスタンスの制限

新しい購入者が AMI 製品のインスタンスを使用できないようにするには、インスタンスを制限できます。インスタンスは、必要に応じて後で追加し直すことができます。制限されたインスタンス上の単一の AMI の既存のユーザーは、サブスクリプション期間中、リージョンからの製品を引き続き使用できます。

インスタンスを制限するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [インスタンスの制限] を選択します。
4. 制限するインスタンスを選択し、[制限] を選択します。
5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。

6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

 Note

チェックボックスがグレー表示されている場合は、そのインスタンスが推奨インスタンスタイプとして 1 つまたは複数のバージョンに関連付けられていることを意味します。このようなインスタンスを制限するには、[バージョンの更新] を使用して別の推奨インスタンスタイプを選択してください。変更リクエストが完了し、制限するインスタンスが推奨インスタンスタイプではなくなったら、[インスタンスの制限] に戻って選択したインスタンスを制限できます。

での AMI ベースの製品のバージョン管理 AWS Marketplace

Amazon マシンイメージ (AMI) ベースの製品を作成するときは AWS Marketplace、特定のバージョンのソフトウェアを含めます。の AMI ベースの製品のライフサイクル AWS Marketplace は、最初のバージョンを公開した後に終了しません。販売者は、ソフトウェアの新しいバージョンで製品を最新の状態に保つ必要があります。以下のセクションでは、バージョン情報 (説明や日付など) の更新、新しいバージョンの追加、以前のバージョンへのアクセスの制限など、バージョンを管理する方法を示します。

トピック

- [バージョン情報を更新する](#)
- [新しいバージョンを追加する](#)
- [バージョンを制限する](#)

バージョン情報を更新する

バージョンが作成されたら、そのバージョンに関連する情報を変更して、更新された情報を購入者に提供すると便利です。例えば、バージョン 1.1 がリリースされた後にバージョン 1.0 を制限する予定であれば、バージョン 1.0 の説明を更新して、そのバージョンが制限される日付を指定して購入者をバージョン 1.1 に誘導できます。バージョン情報は AWS Marketplace 管理ポータルから更新します。

バージョン情報を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[サーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [バージョン情報を更新] を選択します。
4. [バージョンを更新] ページで、更新するバージョンを選択します。
5. 変更する必要がある次の情報を更新します。
 - リリースノート
 - [使用手順]
 - 64 ビット (x86) Amazon マシンイメージ (AMI) - 使用状況とセキュリティグループの詳細
6. [Submit] (送信) を選択します。
7. リクエストが [リクエスト] タブに [レビュー中] ステータスで表示されていることを確認します。

Note

この手順を使用して、バージョンタイトルやバージョンに関連付けられている AMI を更新することはできません。代わりに、[新しいバージョンを作成し](#)、[以前のバージョンを制限してください](#)。

リクエストのステータスは、[サーバー製品](#) ページの [リクエスト] タブからいつでも確認できます。詳細については、「[変更リクエストのステータスを取得する](#)」を参照してください。

新しいバージョンを追加する

製品、ベースイメージに変更を加えたとき、または製品の AMI を変更する必要があるときはいつでも、製品の新しいバージョンを追加できます。から製品の新しいバージョンを追加します AWS Marketplace 管理ポータル。

Note

の AMI の作成については AWS Marketplace、「」を参照してください [で使用する AMIs を構築するためのベストプラクティス AWS Marketplace](#)。

新しいバージョンを追加するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [新しいバージョンを追加] を選択します。新しいバージョンの追加フォームが表示され、最新バージョンの情報が入力されます。

Note

「バージョンの追加」のテストを選択して、新しいバージョンを追加せずに AMI と AWS CloudFormation テンプレート (複数可) をスキャンすることもできます。詳細については、[「AMI のスキャン」を参照して公開要件](#)を確認してください。

4. [バージョン情報] セクションで、次の情報を入力します。
 - [バージョンタイトル] - 有効な文字列 (**1.1** や **##### 2.0** など) を入力します。製品全体で一意である必要があります。
 - [リリースノート] - このバージョンに関する詳細を説明するテキストを入力します。
5. 配信オプションセクションで、購入者がソリューションをデプロイする方法、AMI (スタンドアロン)、CloudFormation を使用した AMI、またはその両方を選択します。CloudFormation 配信オプションでは、最大 3 つの AMI を選択できます。

Note

新しい配信オプションを既存のバージョンに追加することはできません。1 つのバージョンのすべての配信オプションは、同じリクエストで送信する必要があります。

6. Amazon マシンイメージ (AMI) セクションで、次の情報を入力します。
 - [Amazon マシンイメージ ID] - このバージョンに使用する AMI の AMI ID を入力します。AMI ID は、[コンソールの AMI のリスト](#)から確認できます。AMI は、米国東部 (バージニア北部) リージョンと AWS Marketplace 販売者アカウントに存在する必要があります。この AMI に関連付けられているスナップショットは暗号化できません。
 - IAM アクセスロール ARN - に AMI AWS Marketplace へのアクセスを許可する AWS Identity and Access Management (IAM) ロールの Amazon リソースネーム (ARN) を入力します。IAM ロールの作成手順については、[「AMI AWS Marketplace へのアクセスの許可」](#)を参照してく

ださい。IAM ARN には、標準形式 (例: `arn:aws:iam::123456789012:role/RoleName`) を使用します。ARN は AWS Marketplace 販売者アカウントに存在する必要があります。

- オペレーティングシステム (OS) – AMI のベースオペレーティングシステムを入力します。
- OS バージョン – AMI が使用しているオペレーティングシステムのバージョンを入力します。
- [OS ユーザー名] – Linux ベースの AMI の場合は、インスタンスへのサインインに使用できるユーザーの名前を入力します。ec2-user を使用することをお勧めします。
- [スキャンポート] – オペレーティングシステムへのログインに使用できるポート番号 (Linux AMI の場合は SSH ポート、Windows AMI の場合は RDP ポート) を入力します。

7. 該当する場合は、AMI 配信オプションセクションに次の設定を指定します。

- [推奨インスタンスタイプ] – 購入者がデフォルトで利用できるインスタンスタイプを選択します。
- [使用手順] – AMI の使用方法、または AMI の使用に関する詳細情報へのリンクを入力します。
例: #####<https://example.com/usage.htm> #####
- [エンドポイント URL] – 購入者がインスタンスを作成した後にソフトウェアにアクセスする方法についての情報を入力します。購入者が製品へのアクセスに使用できる [プロトコル] (https または http)、相対 URL (例: [/index.html](#))、およびポート (例: [443](#)) を入力します。(ホスト名は EC2 インスタンスによって異なるため、相対パスのみを指定する必要があります)。
- [セキュリティグループの推奨事項] – プロトコル (TCP または UDP)、許可するポートの範囲、IPv4 CIDR IP のリスト (xxx.xxx.xxx.xxx/nn の形式、例えば [192.0.2.0/24](#)) など、1 つ以上の推奨事項の情報を入力します。

8. 該当する場合は、CloudFormation 配信オプションを使用して AMI に次の設定を指定します。

- 配信オプションのタイトル – これは、この Single AMI CloudFormation テンプレート配信オプションを識別する一意の名前です。
- 簡単な説明 – CloudFormation テンプレートの簡単な説明。
- Long Description – CloudFormation テンプレートの詳細な説明。
- 推奨インスタンスタイプ – 購入者がデフォルトで取得するインスタンスタイプ。
- 使用手順 – ソリューションを使用する手順、またはソリューションの使用に関する詳細情報へのリンク。
- CloudFormation テンプレートリンク – Amazon S3 の CloudFormation テンプレートの場所への URL。ネストされたスタックまたはテンプレートの場合は、エントリポイントまたは親テンプレートを指定します。ファイルが存在する Amazon S3 バケットはパブリックにアクセス

可能である必要があります。詳細については、[「製品に CloudFormation テンプレートを追加する」](#)を参照してください。

- AMI パラメータ名 – テンプレート内の EC2 インスタンスリソースが ImageId プロパティ値で参照している CloudFormation テンプレートにパラメータの名前を追加します。は、指定されたパラメータを [AWS Systems Manager パラメータストア](#) パラメータ AWS Marketplace に置き換えます。購入者がテンプレートをデプロイすると、そのパラメータは公開された製品の AWS リージョン固有の AMI ID に解決されます。詳細については、[「AMI の要件」](#)を参照してください。
- アーキテクチャ図リンク – Amazon S3 のアーキテクチャ図の場所への URL。最大イメージファイルは、サイズが 1560x878 または 1560x3120 ピクセルで、比率が 16:9 または 1:2 である必要があります。イメージファイルが存在する Amazon S3 バケットは、パブリックにアクセス可能である必要があります。図の要件については、[「アーキテクチャ図」](#)を参照してください。

9. [送信] を選択し、新しいバージョンを追加するリクエストを送信します。

 Note

新しいバージョンを追加すると、AMI がスキャンされます。詳細については、[「AMI のスキャン」](#)を参照して公開要件を確認してください。

10. リクエストが [リクエスト] タブに [レビュー中] ステータスで表示されていることを確認します。修正する必要があるエラーがある場合、ページ上部の表にエラーが表示され、更新が必要な特定のフィールドは赤で表示されます。

リクエストのステータスは、[\[サーバー製品\]](#) ページの [リクエスト] タブからいつでも確認できます。新しいバージョンはレビューされ、承認された場合は製品の新しい公開バージョンとして公開されます。問題がある場合は、ステータスが [アクションが必要] になる場合があります。リクエストを選択すると、問題を含む詳細が表示されます。

リクエストが成功すると、既存のユーザーには以下の E メールメッセージが届きます。このメッセージは、新しいバージョンが利用可能になったことをユーザーに通知し、そのバージョンのリリースノートにリンクし、最新バージョンへのアップグレードを提案します。AWS アカウント ルートユーザーには、に関連付けられている E メールアカウントにも E メールメッセージのコピーが送信されます AWS アカウント。

Greetings from AWS Marketplace,

Thank you for subscribing to <product-title>

We are writing to inform you that <seller-name> has added a new version to <product-title> on AWS Marketplace.

As an existing customer, your subscription to the product, any running instances and access to previous versions are unaffected. However, <seller-name> does recommend you to update to the latest version, <product-title>/<version-title> by visiting <product-detail-page-of-new-listing>.

For additional questions or upgrade information, please contact <seller-name> directly.

Click here <link of seller page on MP>

to visit the seller's profile page on AWS Marketplace.

Release notes for <product-title>/<version-title>:

<release-notes>

Thank you,

The AWS Marketplace Team

<https://aws.amazon.com/marketplace>

AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of Amazon.com, Inc.

This message was produced and distributed by AWS Inc., 410 Terry Ave. North, Seattle, WA 98109-5210

バージョンを制限する

購入者が公開している製品の特定のバージョンにアクセスできないようにする場合は、そのバージョンを制限できます。

Note

制限状況にかかわらず、すべてのサブスクライバーが現在のバージョンを使用できます。AWS Marketplace ガイドラインでは、バージョンを制限してから 90 日間、既存の購入者へのサポートを継続することが義務付けられています。バージョンが制限されると、AMI は非推奨としてマークされます。詳細については、「Windows インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[AMI を非推奨にする](#)」を参照してください。

バージョンを制限するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [バージョンを制限] を選択します。
4. [バージョンを制限] ページで、制限するバージョン (1 つまたは複数) を選択します。
5. [送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

Note

製品のすべてのバージョンを制限することはできません。最後に残っているパブリックバージョンの製品を制限しようとする、エラーが発生します。製品を完全に削除する方法については、「[the section called “製品を削除する”](#)」を参照してください。

リクエストのステータスは、[\[サーバー製品\]](#) ページの [リクエスト] タブからいつでも確認できます。詳細については、「[変更リクエストのステータスを取得する](#)」を参照してください。

Note

バージョンの制限には、最大で 3 日かかることがあります。

リクエストが成功すると、既存のユーザーには、バージョン制限を通知し、入手可能な最新バージョンを使用することを勧める次の E メールメッセージが届きます。AWS アカウント ルートユーザーは、に関連付けられている E メールアカウントにも E メールメッセージのコピーを受け取ります AWS アカウント。

Greetings from AWS Marketplace,

Thank you for subscribing to <product-title>.

We are writing to inform you that, as of <Version-Restriction-Date>, <Seller Name> will no longer offer version(s) "<version-title>" to new subscribers. Your use and

subscription is unaffected for this version(s), however it is recommended that users upgrade to the latest version on AWS Marketplace.

For additional questions or upgrade information, please contact <seller-name> directly. Click here<link of seller page on MP> to visit the seller's profile page on AWS Marketplace.

Thank you,

The AWS Marketplace Team

<https://aws.amazon.com/marketplace>

AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of Amazon.com, Inc. This message was produced and distributed by AWS Inc., 410 Terry Ave. North, Seattle, WA 98109-5210

での AMI ベースの製品情報の更新 AWS Marketplace

単一の Amazon マシンイメージ (AMI) 製品を作成したら、それに関連する情報の一部を AWS Marketplace で変更できます。例えば、新しいバージョンで製品の説明やハイライトが変更された場合は、新しいデータで製品情報を編集できます。製品のタイトル、SKU の説明、カテゴリ、キーワードなど、他の製品情報を更新することもできます。詳細については、以下の手順を参照してください。

製品情報を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[サーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [製品情報を更新] を選択します。
4. 変更する必要がある次のフィールドのいずれかを更新します。
 - [製品のタイトル]
 - [SKU]
 - 短い説明
 - [詳しい説明]
 - [製品ロゴイメージ URL]
 - [ハイライト]
 - [製品カテゴリ]
 - キーワード

- [製品動画 URL]
- リソース
- [サポート情報]

 Note

ロゴフォーマットの詳細については、「[会社と製品のロゴ要件](#)」を参照してください。

5. [Submit] (送信) を選択します。
6. リクエストが [リクエスト] タブに [レビュー中] ステータスで表示されていることを確認します。リストにリクエストが表示されるようにするには、ページの更新が必要になる場合があります。

リクエストのステータスは、[サーバー製品](#) ページの [リクエスト] タブからいつでも確認できます。詳細については、「[変更リクエストのステータスを取得する](#)」を参照してください。

および国別の AWS リージョン AMI ベースの製品の可用性の管理

で製品を作成するときは AWS Marketplace、利用可能な AWS リージョンを選択します。購入者が製品を購入できる国も選択します。これら 2 つのプロパティは似ていますが、同じではありません。例として、購入者が米国に居住し、米国から購入していても、製品を欧州 (フランクフルト) リージョンにインストールする場合があります。この購入者が製品を購入するには、国のリストに米国を、リージョンのリストに欧州 (フランクフルト) を含める必要があります。以下のセクションを使用して、リージョンおよび国ごとに製品の可用性を更新できます。

トピック

- [を追加する AWS リージョン](#)
- [の制限 AWS リージョン](#)
- [将来の AWS リージョンのサポートの更新](#)
- [国別の可用性の更新](#)

を追加する AWS リージョン

購入者が製品を使用できるリージョンを追加できます。

リージョンを追加するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [リージョンを追加] を選択します。
4. 使用可能なリージョンのリストから、追加するリージョンを選択します。
5. [リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

Note

新しい のサポートを追加すると AWS リージョン、既に製品のプライベートオファーをサブスクライブしているお客様は、新しく追加されたリージョンに自動的にアクセスできなくなります。顧客がアクセスするリージョンを使用して、別のプライベートオファーを作成する必要があります。新しいオファーを承諾すると、顧客は新しく追加されたリージョンにアクセスできます。将来の日付に製品をサブスクライブする顧客は、リージョンがプライベートオファーに含まれている限り、そのリージョンにアクセスすることもできます。新しいプライベートオファーを作成する方法の詳細については、「[Private offer upgrades, renewals, and amendments](#)」を参照してください。

の制限 AWS リージョン

新しい購入者が特定の で製品を使用できないようにするには AWS リージョン、リージョンを制限できます。リージョンは後で追加し直すことができます。リージョン内の製品の既存のサブスクライバーは、サブスクライブしている限り、そのリージョンの製品を引き続き使用できます。

リージョンを制限するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [リージョンを制限] を選択します。

4. ドロップダウンメニューを選択すると、製品が現在販売されているリージョンのリストが表示されます。
5. 制限するリージョンを選択します。
6. 選択したリージョンはトークンとして表示されます。制限するリージョンのリストを確認し、制限しないリージョンには X を入力します。
7. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
8. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

リクエストが成功すると、既存のユーザーには、制限されるリージョンを通知する次の E メールメッセージが届きます。登録している間は製品を使い続けることができますが、サブスクライブをキャンセルすると再サブスクライブすることはできません。

Greetings from AWS Marketplace,

This message is a notification detailing a recent change for <ProductName>. {{{sellerName}}} has opted to restrict the <ProductType> product in <Restricted Region(s)> beginning <DateOfChange>.

This impacts you in the following ways:

1. As long as you're subscribed to the product, you can continue using the software product in the restricted Region.
2. You can't begin new instances of the software product in the restricted Region.
3. You can continue using the software product in all available AWS Regions.

Regards,

The AWS Marketplace Team

AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com (<http://amazon.com/>) is a registered

trademark of Amazon.com, Inc. This message was produced and distributed by Amazon Web Services Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.

将来の AWS リージョンのサポートの更新

製品を新しく起動した にオンボードする場合は AWS リージョン、今後のリージョンの更新サポートを使用できます。

将来のリージョンのサポートを更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [将来のリージョンのサポートの更新] を選択します。
4. 将来のリージョンサポートを有効にして、AWS Marketplace が AWS リージョン ユーザーに代わって新しく起動された に製品をオンボードできるようにすることができます。
5. この機能を有効にすると、将来リリースされるすべてのリージョンを選択するか、米国リージョンのみに限定するかを選択できます。
6. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
7. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

国別の可用性の更新

製品をサブスクライブして提供できる国を変更する場合は、[可用性の更新] を使用してください。

国別の可用性を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [可用性の更新] を選択します。
4. 以下のオプションのいずれかを選択してください：
 1. [すべての国] - サポートされているすべての国で利用できます。
 2. [除外対象のすべての国] - 選択した国を除くすべてのサポート対象国で利用できます。
 3. [カスタムリスト] - 製品が販売されている特定の国のリスト。
5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

AMI ベースの AWS Marketplace 製品のエンドユーザーライセンス契約 (EULA) の更新

AWS Marketplace 販売者は、単一の Amazon マシンイメージ (AMI) 製品の使用を管理するエンドユーザーライセンス契約 (EULA) を更新できます。EULA は、AWS Marketplace に公開されているソフトウェアリストの製品リストページに記載されています。独自の EULA を適用することも、[AWS Marketplace の標準契約書 \(SCMP\)](#) を使用することもできます。以下の手順は、単一の AMI 製品の EULA を更新する方法を示しています。

EULA の詳細については、「[での標準化された契約の使用 AWS Marketplace](#)」を参照してください。

EULA を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) タブを選択し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから、[エンドユーザー使用許諾契約の更新] を選択します。
4. [AWS Marketplace \(SCMP\) の標準契約](#) を選択するか、独自のカスタム EULA を送信できます。カスタム EULA の場合は、Amazon S3 バケットからのカスタム契約を提供する必要があります。

Note

Amazon S3 バケットでパブリックアクセシビリティを有効にする必要があります。

5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

での AMI ベースの製品の返金ポリシーの更新 AWS Marketplace

AWS Marketplace 販売者は、単一の Amazon マシンイメージ (AMI) 製品の返金ポリシーを設定できます。製品の返金ポリシーを変更する場合は、[返金ポリシーを更新] を使用できます。以下の手順では、返金ポリシーを更新する方法を示します。

返金ポリシーを更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [返金ポリシーを更新] を選択します。
4. 現在の返金ポリシーの詳細がテキストボックスに表示され、編集できます。リクエストを送信すると、現在の返金ポリシーが上書きされます。
5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

AMI AWS Marketplace へのアクセスの許可

への新しい Amazon マシンイメージ (AMI) の追加を含むリクエストを作成する場合 AWS Marketplace、AMI を AWS Marketplace システムにコピーし、セキュリティ上の問題がないかスキャンする必要があります。AMI でアクションを実行するアクセス許可を持つ AWS Identity and Access Management (IAM) ロールと、 がロールを引き受け AWS Marketplace ることを許可する信頼ポリシーを作成して、AMI AWS Marketplace へのアクセスを許可する必要があります。IAM ロールは一度作成するだけで済みます。次の手順では、AMI AWS Marketplace へのアクセスを許可する AWS Marketplace アセット取り込み用のロールを作成する方法を示します。

AMI AWS Marketplace アセットの取り込み用のロールを作成するには

1. にサインインし AWS Management Console、IAM コンソールを開き、[ロールページ](#)に移動します。
2. [Create role] (ロールの作成) を選択します。
3. [ロールを作成] ページで、以下の選択を行います。
 - 信頼されたエンティティのタイプを選択する – AWS サービスを選択します。
 - ユースケースを選択する – を選択しますAWS Marketplace。
 - [ユースケースの選択] - [Marketplace - AMI アセットの取り込み] を選択します。
 - 次のページに移動して [次: アクセス] ページに移動します。
4. [AWSMarketplaceAmiIngestion] ポリシーを選択します。必要に応じてアクセス許可の境界を追加し、[次へ: タグ] を選択して続行します。

Note

アクセス許可の境界を使用して、このロールで AWS Marketplace に与えるアクセスを制限できます。詳細については、「AWS Identity and Access Management ユーザーガイド」の「[IAM エンティティのアクセス許可の境界](#)」を参照してください。

5. 続行するには、[次へ: レビュー] を選択します。
6. ロールの名前を入力し、[ロールを作成] を選択します。
7. ページの上部に「ロールの####が作成されました」と表示され、ロールがロールのリストに表示されます。

このページで、作成したロールを選択すると、その ARN が `arn:aws:iam::123456789012:role/exampleRole` という形式で表示されます。製品に[新しいバージョンを追加する](#)場合など、変更リクエストを作成するときは、[IAM アクセスロール ARN] に ARN を使用します。

AWS Marketplaceからの製品の削除

製品を公開したら、から削除できます AWS Marketplace。これは、日没とも呼ばれます。製品を削除するには、製品を特定して削除する理由と連絡先の E メールアドレスと一緒に削除リクエストを送信します。また、現在の製品を新しい製品と交換する場合は、代替製品 ID を提供することもできます。

製品の削除をリクエストすると、新規顧客は製品をサブスクライブできなくなります。既存のお客様は、最低 90 日間サポートする必要があります。

Note

AWS Marketplace 管理ポータルから制限付き製品を削除することはできません。ポータルは、アカウントの発行履歴の一部として保持します。

からの製品削除のリクエスト AWS Marketplace は、以下の条件で処理されます。

- 製品は、AWS Marketplace 検索、参照、その他の検出ツールから削除されます。[サブスクライブ] ボタンまたは機能は無効になっており、ページ上のメッセージは製品が使用できなくなったことを明確に示しています。製品の詳細ページは引き続き URL を使用してアクセスでき、公開されている検索エンジンでインデックスに登録されている可能性があります。

- 削除の理由を指定する必要があります (サポートの終了、製品の更新の終了、交換製品など)。削除された製品のサポートを継続するための要件については、「[AWS Marketplace 販売者の利用規約](#)」を参照してください。
- AWS Marketplace は、製品の削除、削除の理由、販売者の連絡先情報の提供を通知する E メールメッセージを通じて、現在の購入者に連絡します。
- 現在の購入者がサブスクリプションをキャンセルするまでソフトウェアへのアクセスは維持されます。これらのお客様は、製品の削除による影響を受けません。

を使用して作成された製品を削除するには AWS Marketplace 管理ポータル

1. AWS Marketplace 管理ポータル で を開き<https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [製品] タブを選択し、次に [サーバー] を選択します。
3. [現在のサーバー製品] の下にある製品ページで、削除する製品を探します。[変更のリクエスト] ドロップダウンリストで、[可視性の更新] を選択します。
4. [製品可視性の更新] ページで、[制限付き] を選択します。
5. (オプション) 削除する製品の代わりとなる別の製品がある場合は、[代替製品 ID] を入力します。
6. 情報が正しいことを確認して、[送信] を選択します。

製品の削除リクエストを送信した後、[What's next (最新情報)] 情報ページが表示されます。Seller AWS Marketplace Operations は、リクエストを確認して処理します。[リクエスト] を表示して、送信のステータスを確認します。

製品が削除されると、製品は AWS Marketplace 管理ポータルの [現在の製品] リストに表示されます。[Current Products (現在の製品)] で利用できる唯一のアクションは製品のスプレッドシートをダウンロードすることです。別のサンセットリクエストの編集または送信はできません。

製品の削除についてご不明な点がございましたら、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

AWS Marketplace の変更リクエストの一般的なエラーのトラブルシューティング

AWS Marketplace で製品に関する情報を変更すると、エラーが発生する可能性があります。このトピックでは、一般的なエラーとその修正方法について説明します。

- AMI のスキャン AMI のスキャン中にいくつかの問題が発生する可能性があります。
- AMI をスキャンする AWS Marketplace アクセス許可が付与されていません。AWS Marketplace にアクセス許可を付与してください。または、アクセス許可を付与しましたが、アクセス許可の境界が制限されすぎています。詳細については、「[AMI AWS Marketplace へのアクセスの許可](#)」を参照してください。
- スキャンで AMI にセキュリティ上の問題や一般的な脆弱性と漏えい (CVE) が見つかった場合は、イメージ内のオペレーティングシステム用の最新のパッチを使用していることを確認してください。詳細については、「[の AMI ベースの製品要件 AWS Marketplace](#)」を参照してください。

AMI の構築に関する一般的なガイドラインについては、「[で使用する AMIs を構築するためのベストプラクティス AWS Marketplace](#)」を参照してください。

- AWS Marketplace 管理ポータル フィールド - AWS Marketplace 管理ポータル の一部のフィールドには、非常に具体的な情報が必要です。
- フィールドが何をリクエストしているのかわからない場合は、コンソールで詳細を確認してみてください。ほとんどのフィールドには、フィールドの上にはテキストによる説明があり、フィールドの下にはフォーマット要件があります。
- 1 つ以上の無効なフィールドを含むフォームを送信しようとすると、問題のリストが表示されます。問題の解決に役立つ推奨アクションが表示されます。
- ARN の入力を求められた場合、通常、これはコンソールの別の場所にあります。例えば、AWS Marketplace に AMI へのアクセスを付与するために作成した IAM ロールの ARN は、IAM コンソールの [ルールページ](#) にあります。ARN はすべて同様の形式です。例えば、IAM ロール ARN は、arn:aws:iam::123456789012:role/exampleRole というような形式になります。
- ロゴと動画は、コンテンツへの URL として直接提供する必要があります。ロゴ形式の詳細については、「[会社と製品のロゴ要件](#)」を参照してください。

製品およびバージョン変更リクエストの送信について詳しくは、「[での公開用に製品を送信する AWS Marketplace](#)」を参照してください。

- 製品積載フォーム (PLF) の問題 - PLF には、スプレッドシートに含まれる指示が含まれています。全体的な説明は説明表に記載されています。各フィールドには入力方法が記載されています。フィールドを選択すると指示が表示されます。
- 進行中のリクエスト - 一部のリクエストは並行して実行できません。製品の特定の処理中の情報を更新するリクエストは、一度に 1 つしか送信できません。レビュー中のすべてのリクエストは、AWS Marketplace 管理ポータル の [サーバー製品] ページの [リクエスト] タブで確認できま

す。意図しない保留中のリクエストがある場合は、そのリクエストをキャンセルして、必要な変更を加えた新しいリクエストを送信できます。

- バージョンを更新 (追加または制限する) が進行中の場合は、バージョン情報を更新できません。
- AWS Marketplace Seller Operations チームから保留中のリクエストがある場合、新しい変更を送信することはできません。
- 原因不明のエラー - 説明もなく送信が失敗した場合は、もう一度行ってください。サーバーの負荷により送信が失敗することがあります。

それでも変更リクエストに問題がある場合は、[AWS Marketplace Seller Operations](#) チームに連絡してください。

で使用する AMIs を構築するためのベストプラクティス AWS Marketplace

このトピックでは、で使用する Amazon マシンイメージ (AMIs) の構築に役立つベストプラクティスとリファレンスについて説明します AWS Marketplace。に構築および送信される AMIs AWS Marketplace は、すべての AWS Marketplace 製品ポリシーに従う必要があります。詳細については、次のセクションを参照してください。

トピック

- [再販権の確保](#)
- [AMI の構築](#)
- [の AMI の準備と保護 AWS Marketplace](#)
- [公開要件を確認するための AMI のスキャン](#)
- [AWS Marketplace AMI でソフトウェアが実行されていることを確認する](#)

再販権の確保

無料以外の Linux ディストリビューションについては、AWS提供された Amazon Linux、RHEL、SUSE を除き、再販権を保護する責任があります。Windows AMIs の再販権限を保護する必要はありません。

AMI の構築

AMI の構築に関する以下のガイドラインを使用してください。

- AMI がすべての[AWS Marketplace ポリシー](#)を満たしていることを確認します。
- 米国東部 (バージニア北部) リージョンで AMI を作成します。
- AWS Marketplaceなどの信頼できる評判の高いソースから入手した、明確に定義されたライフサイクルを持つ、Amazon Elastic Block Store (Amazon EBS) にサポートされた、よく管理された既存の AMI から製品を作成します。
- 最新のオペレーティングシステム、パッケージ、およびソフトウェアを使用して AMI を構築します。
- AMI が、ハードウェア仮想マシン (HVM) 仮想化と 64 ビットアーキテクチャを使用するパブリック Amazon EC2 AMI に基づいていることを確認します。
- AMI の構築、更新、再公開のための繰り返し可能なプロセスを開発します。
- すべてのバージョンと製品で一貫したオペレーティングシステム (OS) ユーザー名を使用します。推奨されるデフォルトのユーザー名は、ec2-userLinux およびその他の Unix に似たシステムの場合は、Windows Administratorの場合は です。
- 公開する最終的な AMI AWS Marketplace を送信する前に、AMI からインスタンスを起動してテストし、意図したエンドユーザーエクスペリエンスを検証します。このインスタンスですべてのインストール方法、機能、パフォーマンスをテストします。
- 以下のようにポート設定を確認してください。
 - オープンファイアウォール、リバースプロキシ、SSRF の脆弱性に対する[ベストプラクティスのセキュリティ設定](#)として、IMDS サポートオプションは IMDSv2 のみに設定する必要があります。次の CLI は、最終ビルドフェーズで新しい AMI を登録するときに使用できます。
 - `aws ec2 register-image --name my-image --root-device-name /dev/xvda --block-device-mappings DeviceName=/dev/xvda,Ebs={SnapshotId=snap-0123456789example} --architecture x86_64 --imds-support v2.0`

AMI の作成の詳細については、以下の関連リソースを参照してください。

- 「Amazon EC2 ユーザーガイド」の「[Amazon EBS-backed AMI を作成する](#)」
- 「Amazon EC2 ユーザーガイド」の「[Windows Sysprep を使用して Amazon EC2 AMI を作成する](#)」

- [EBS-backed インスタンスから Amazon マシンイメージ \(AMI\) を作成する方法。](#)
- [Amazon Linux AMI](#)
- [Amazon EC2 インスタンスタイプ](#)および[インスタンスタイプ](#)
- [IMDS V2 用にデフォルトで AMI を設定する](#)

の AMI の準備と保護 AWS Marketplace

セキュアな AMI を作成するために、次のガイドラインをお勧めします。

- 「Amazon EC2 ユーザーガイド」の「[共有 Linux AMI のガイドライン](#)」を使用する
- 攻撃の危険性を減らすため、最低限のインストールとしてデプロイするように AMI を設計します。不要なサービスやプログラムを無効にするか、削除します。
- 可能な限り、ネットワークトラフィックにエンドツーエンドの暗号化を使用します。例えば、Secure Sockets Layer (SSL) を使用して、購入者との間の HTTP セッションを保護します。サービスが有効なかつ最新の証明書のみを使用するようにしてください。
- AMI 製品を文書化するときは、購入者がインスタンスへのインバウンドトラフィックアクセスを制御するためのセキュリティグループの推奨事項を提供します。レコメンデーションでは、以下を指定する必要があります。
 - サービスが機能するために必要なポートの最小セット。
 - 管理アクセスに推奨されるポートと送信元 IP アドレスの範囲。

これらのセキュリティグループの推奨事項は、購入者が適切なアクセスコントロールを実装するのに役立ちます。AMI 製品に新しいバージョンを追加する方法の詳細については、「[新しいバージョンを追加する](#)」を参照してください。

- AWS コンピューティング環境に対して定期的にペネトレーションテストを実施することを検討してください。または、ユーザーに代わってそのようなテストを実行するためにサードパーティーを採用することを検討してください。ペネトレーションテストリクエストフォームなど詳細は、「[AWS ペネトレーションテスト](#)」を参照してください。
- ウェブアプリケーションのトップ 10 の脆弱性に気づき、必要に応じてアプリケーションを構築してください。詳細については、「[Open Web Application Security Project \(OWASP\) - Top 10 Web Application Security Risks](#)」を参照してください。新しいインターネットの脆弱性が発見された場合は、AMI に同梱されているウェブアプリケーションを速やかに更新してください。この情報を含むリソースの例として、「[SecurityFocus](#)」および「[NIST 全国脆弱性データベース](#)」があります。

セキュリティの詳細については、以下の関連リソースを参照してください。

- [AWS クラウドセキュリティ](#)
- [インターネットセキュリティセンター \(CIS\): セキュリティベンチマーク](#)
- [オープンなウェブアプリケーションセキュリティプロジェクト \(OWASP\): セキュアなコーディングプラクティスクイックリファレンスガイド](#)
- [OWASP トップ 10 ウェブアプリケーションのセキュリティリスク](#)
- [SANS \(SysAdmin、監査、ネットワーク、およびセキュリティ\) 共通短所列挙 \(CWE\) トップ 25 最も危険なソフトウェアエラ](#)
- [セキュリティフォーカス](#)
- [NIST 全国脆弱性データベース](#)

公開要件を確認するための AMI のスキャン

AMI を AWS Marketplace カタログに公開するには、AMI スキャンを完了する必要があります。AMI スキャンは、パッチが適用されていない一般的な脆弱性と露出 (CVEs) をチェックし、AMI がセキュリティのベストプラクティスに従っていることを確認します。詳細については、「[用の AMI の準備と保護 AWS Marketplace](#)」を参照してください。

AMI スキャンを実行するには、次のいずれかのオプションを選択します。

オプション 1: アセットメニュー

この方法では、製品作成フローの外部で AMIs をスキャンできます。また、AMI 製品を作成せずにアセットをスキャンする必要がある SAAS Quick Launch を使用する SaaS 販売者にも役立ちます。

1. から[AWS Marketplace 管理ポータル](#)、アセットメニューに移動し、Amazon マシンイメージを選択します。
2. スキャンプロセスを開始するには、AMI の追加を選択します。
3. このページに戻ると、AMIs のステータスを表示できます。

オプション 2: 変更メニューをリクエストする

このオプションは、AMI 製品を既に作成している販売者が使用できます。詳細については、「[AMI ベースの製品の作成](#)」を参照してください。

1. から[AWS Marketplace 管理ポータル](#)、製品メニューに移動し、サーバーを選択します。

2. Server 製品から製品を選択します。これは AMI ベースの製品である必要があります。製品はどの状態でもかまいません。次のステップでは、公開状態である必要はありません。
3. 変更のリクエストメニューに移動し、バージョンの更新を選択します。
4. テスト「バージョンの追加」を選択します。プロンプトに従って、AMI の詳細を含むリクエストを送信します。リクエストが成功すると、AMI がスキャンに合格したことを示します。新しいバージョンの追加オプションとは異なり、スキャンが成功した場合、テスト「バージョンの追加」は AMI ベースの製品に新しいバージョンを追加しません。

Note

AMI AWS Marketplace へのアクセスを許可する方法については、「」を参照してください
[AMI AWS Marketplace へのアクセスの許可](#)。

AWS Marketplace AMI でソフトウェアが実行されていることを確認する

AMI 製品から作成された Amazon EC2 インスタンスで実行されていることを、ランタイムにソフトウェアに確認させたい場合があります。

Amazon EC2 インスタンスが AMI 製品から作成されていることを確認するには、Amazon EC2 に組み込まれているインスタンスメタデータサービスを使用します。以下のステップでは、この検証を順を追って説明します。メタデータサービスの使用の詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[インスタンスメタデータとユーザーデータ](#)」を参照してください。

1. インスタンスアイデンティティドキュメントの取得

実行中の各インスタンスには、インスタンスからアクセス可能な、インスタンス自体に関するデータを提供するアイデンティティドキュメントが用意されています。次の例は、インスタンスから curl を使用してインスタンスのアイデンティティドキュメントを取得する方法を示しています。

IMDSv2: (推奨)

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/instance-identity/document
{
```

```

    "accountId" : "0123456789",
    "architecture" : "x86_64",
    "availabilityZone" : "us-east-1e",
    "billingProducts" : null,
    "devpayProductCodes" : null,
    "marketplaceProductCodes" : [ "0vg0000000000000000000000000" ],
    "imageId" : "ami-0123456789abcdef1",
    "instanceId" : "i-0123456789abcdef0",
    "instanceType" : "t2.medium",
    "kernelId" : null,
    "pendingTime" : "2020-02-25T20:23:14Z",
    "privateIp" : "10.0.0.2",
    "ramdiskId" : null,
    "region" : "us-east-1",
    "version" : "2017-09-30"
  }

```

IMDSv1:

```

curl http://169.254.169.254/latest/dynamic/instance-identity/document{
  "accountId" : "0123456789",
  "architecture" : "x86_64",
  "availabilityZone" : "us-east-1e",
  "billingProducts" : null,
  "devpayProductCodes" : null,
  "marketplaceProductCodes" : [ "0vg0000000000000000000000000" ],
  "imageId" : "ami-0123456789abcdef1",
  "instanceId" : "i-0123456789abcdef0",
  "instanceType" : "t2.medium",
  "kernelId" : null,
  "pendingTime" : "2020-02-25T20:23:14Z",
  "privateIp" : "10.0.0.2",
  "ramdiskId" : null,
  "region" : "us-east-1",
  "version" : "2017-09-30"
}

```

2. インスタンスアイデンティティドキュメントの検証

署名を使用して、インスタンスアイデンティティが正しいことを確認できます。このプロセスの詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[インスタンスアイデンティティドキュメント](#)」を参照してください。

3. 製品コードの検証

AMI 製品を最初に公開用に提出すると、製品には AWS Marketplace によって [製品コード](#) が割り当てられます。製品コードは、インスタンスアイデンティティドキュメントの `marketplaceProductCodes` フィールドを確認すると検証できます。または、メタデータサービスから直接取得することもできます。

IMDSv2:

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/product-codes
```

製品コードが AMI 製品のコードと一致する場合、インスタンスは製品から作成されたものです。

`instanceId` やインスタンス `privateIp` など、インスタンスアイデンティティドキュメントから他の情報を確認したい場合もあります。

の AMI 製品の料金 AWS Marketplace

AWS Marketplace には、Amazon マシンイメージ (AMI) 製品の複数の料金モデルがあります。販売者のプライベートオファーでは、複数年契約とカスタム期間契約のオプションがあります。複数年契約とカスタム期間契約の詳細については、[AWS Marketplace 製品のプライベートオファーの準備およびプライベートオファーの分割プラン](#)を参照してください。以下のセクションでは、AMI ベースの製品の料金モデルに関する一般的な情報を示します。

Note

[AWS Marketplace 販売者としての登録](#) に記載されているように、W-9 フォーム (米国居住者) または W-8 フォーム (米国非居住者) を提出する必要があります。

トピック

- [AMI 料金モデル](#)
- [AWS 料金とソフトウェア料金](#)
- [AWS Marketplace Metering Service による AMI 製品のカスタム計測価格](#)

- [での AMI 製品の契約料金 AWS Marketplace](#)
- [を使用したライセンスと AMI ベースの製品の関連付け AWS License Manager](#)

AMI 料金モデル

次の表に、AMI ベースの製品の料金モデルに関する一般的な情報を示します。

料金モデル	説明
無料	お客様は Amazon Elastic Compute Cloud (Amazon EC2) でサポートされている量のインスタンスを実行することができます。ソフトウェアの追加料金はかかりません。  Note 無料トライアルと年間料金は、月額料金と組み合わせることはできません。
自分のライセンス使用 (BYOL)	AWS Marketplace はソフトウェアの使用に対して顧客に課金しませんが、お客様は製品をアクティブ化するためにライセンスキーを指定する必要があります。このキーは の外部で購入されます AWS Marketplace。資格とライセンスの実施、およびすべての料金と課金は、販売者によって処理されます。
時間単位料金または時間あたりの年間料金	[時間単位] - ソフトウェアの料金は、時間単位で請求されます。各インスタンスタイプの料金は個々に異なり (ただし、必須ではない)、その使用量は最も近い時間に切り上げられます。 [時間単位 (無料トライアルあり)] - お客様は、ソフトウェアのインスタンスを 1 つだけ実行できるという制限があり、料金は発生しません。期間は 5~30 日の間で定義します。無料トライアルは、実行中の最も高価なインスタンスタイプに適用され、1 インスタンスを超える同時使用はすべて、時間料金で請求されます。注: これは Amazon EC2 の AWS 無

料金モデル	説明
	料利用枠とは別のモデルです。これにより、お客様には毎月 750 時間の無料利用が提供されます。 [時間単価と月額料金] - 時間単価と月額料金の両方が別々に適用されます。月額料金は使用量にかかわらず毎月適用され、時間料金は時間単位の使用量のみに基づいて適用されます。 [年間で毎時]: お客様は、1 つのインスタンスタイプの 1 つの Amazon EC2 インスタンスに対して 1 年間の使用量を購入することができます。販売者は、インスタンスタイプごとの料金を設定し、時間単価よりも価格を抑えることができます。購入された年間サブスクリプションを超えるお客様の使用量は、そのインスタンスタイプの販売者によって設定された時間単価で請求されます。 [時間単位 (複数年間およびカスタム期間)] - このタイプのオファーは、販売者のプライベートオファーを通じてのみ利用できます。販売者のプライベートオファーを使用して、カスタム契約期間 (最大 3 年) を指定します。前払い料金を指定することも、柔軟な支払いスケジュールを含めることもできます。各インスタンスタイプの料金を設定します。オファーに柔軟な支払いスケジュールがある場合は、オファーに含まれる各インスタンスタイプの請求日、支払額、インスタンス数も設定します。柔軟な支払いスケジュールを持つアクティブな販売者のプライベートオファーの場合、お客様が指定された数のインスタンスを起動した後、起動された追加のインスタンスには、販売者のプライベートオファーで指定された時間料金で課金されます。複数年契約とカスタム期間契約の詳細については、AWS Marketplace 製品のプライベートオファーの準備およびプライベートオファーの分割プランを参照してください。 [時間単位 (無料トライアル付き)] - 年間オプション付きの時間単位モデルと同じです。ただし、販売者が設定した日数で、お客様がインスタンスタイプのインスタンスを無料で 1

料金モデル	説明
	つ実行できる無料トライアルを含む点を除きます。年間サブスクリプションはいつでも購入することができます。また、無料トライアルサブスクリプションと組み合わせることができます。 [年間料金 (時間単位)] 時間単位の料金モデルと同じ。お客様は、1つのインスタンスタイプの1つの Amazon EC2 インスタンスに対して1年間の使用量を購入することができます。販売者は、インスタンスタイプごとの料金を設定し、時間単価よりも料金を抑えることができますが、減額は必須ではありません。購入された年間サブスクリプションを超えるお客様の使用量は、そのインスタンスタイプの販売者によって設定された時間単価で請求されます。 [複数年間およびカスタム期間 (時間単位)] - これは、AWS Marketplace 製品のプライベートオファーの準備 を通じてのみ利用できます。販売者のプライベートオファーを使用して、最長3年のカスタム期間契約を指定できます。前払いを要求することも、顧客に柔軟な支払いスケジュールを提供することもできます。契約期間中の各インスタンスタイプの料金と、起動される追加のインスタンスの時間料金を設定します。柔軟な支払いスケジュールを提供する場合は、オファーに含まれる各インスタンスタイプの請求日、支払額、インスタンス数も設定します。柔軟な支払いスケジュールを持つアクティブなプライベートオファーの場合、指定された数のインスタンスが起動された後、お客様が起動する追加のインスタンスには、プライベートオファーで指定された時間料金で課金されます。複数年契約とカスタム期間契約の詳細については、AWS Marketplace 製品のプライベートオファーの準備 および プライベートオファーの分割プラン を参照してください。

料金モデル	説明
	i Note 無料トライアルと年間料金は、月額料金と組み合わせることはできません。

AWS 料金とソフトウェア料金

Amazon マシンイメージ (AMI) ベースの製品料金は 2 つのカテゴリに分類されます。

- インフラストラクチャ料金の詳細 – すべての AMI ベースの製品には、使用されるサービスと AWS インフラストラクチャに応じて、関連するインフラストラクチャ料金が発生します。これらのレートと料金は、によって定義および管理され AWS、によって異なる場合があります AWS リージョン。詳細については、[Amazon EC2 の料金表](#)を参照してください。
- ソフトウェアの料金の詳細 - 有料製品の場合、販売者はソフトウェアの使用料金を定義します。

これら 2 つの製品料金カテゴリは、購入者が製品使用の潜在的なコストを理解するのに役立つように、AWS Marketplace 詳細ページに個別に表示されます。

AMI 時間単位製品の無料トライアル

AMI 時間単位の製品は、オプションの無料トライアルプログラムの対象となります。無料トライアルでは、お客様は製品のソフトウェア料金を支払うことなく、製品をサブスクライブし、単一のインスタンスを最大 31 日間使用できます。適用される AWS インフラストラクチャ料金が引き続き適用されます。無料トライアルは、有効期限が切れると自動的に有料サブスクリプションに切り替わります。お客様には、提供された無料単位を超える追加使用分が課金されます。時間単位の製品無料トライアルを提供するには、トライアル期間を定義し、[AWS Marketplace Seller Operations](#) チームに通知してください。試用期間は 5~31 日です。

無料トライアル製品をサブスクライブしている場合、お客様は、無料トライアルの期間、計算された有効期限、およびアンスクライブに関する詳細を含むようこそメールメッセージを受け取ります。有効期限が切れる 3 日前に通知メールが送信されます。

で無料トライアル製品を提供する場合 AWS Marketplace、返金ポリシーで説明されている特定の返金ポリシーに同意することになります。

追加リソース

AMI 製品の料金の詳細については、以下のトピックを参照してください。

- [AWS Marketplace Metering Service による AMI 製品のカスタム計測価格](#)
- [での AMI 製品の契約料金 AWS Marketplace](#)

AMI ベースの製品の請求、計測、ライセンスの統合については、以下のトピックを参照してください。

AMI 製品の料金の詳細については、以下のトピックを参照してください。

- [AWS Marketplace Metering Serviceを使用した AMI 製品のカスタム計測を構成する](#)
- [を使用したライセンスと AMI ベースの製品の関連付け AWS License Manager](#)

AWS Marketplace Metering Service による AMI 製品のカスタム計測価格

AWS Marketplace Metering Service を使用すると、使用量を取得するエンドポイントに計測レコードを送信するようソフトウェアを改修できます。使用カテゴリを選択して、その 1 つのカテゴリに対して最大 24 のディメンションを定義します。これらのディメンションは 1 時間に 1 回計測および集計され、定義した料金プランに持つについて請求されます。販売者は、使用するディメンションを決定する必要があります。AMI が発行された後は、ディメンションを変更できません。以下のセクションでは、AWS Marketplace Metering Service について説明します。

トピック

- [Metering Service の概念](#)
- [ソフトウェアの料金](#)
- [への製品の追加 AWS Marketplace](#)
- [Metering Service を使用するためのソフトウェアの変更](#)
- [ベンダー計測タグ付け \(オプション\)](#)
- [カスタム計測を設定する](#)
- [AWS Marketplace Metering Serviceを使用した AMI 製品のカスタム計測を構成する](#)

Metering Service の概念

AWS Marketplace Metering Service を使用する前に、以下の重要なサービス概念に注意してください。

- [使用カテゴリ] - Metering Service を使用して価格が設定されるソフトウェア製品はすべて、1 つの使用カテゴリに従って分類され、それによって顧客への適切な請求方法が決まります。使用カテゴリには次のものがありますが、これらに限定されるものではありません。
- [ユーザー] - 単一の ID に関連付けられた、定義済みのアクセス許可のセット。このカテゴリは、顧客のユーザーが直接ソフトウェアに接続するソフトウェア (たとえば、顧客関係管理またはビジネスインテリジェンス報告) に適切です。

- ホスト - サーバー、ノード、インスタンス、エンドポイント、またはコンピューティングシステムの他の部分。このカテゴリは、多くの顧客所有インスタンス (たとえば、パフォーマンスまたはセキュリティモニタリング) を監視またはスキャンするソフトウェアに適切です。
- データ - MB、GB、または TB で測定される、ストレージまたは情報。このカテゴリは、保存されたデータを管理する、またはバッチでデータを処理するソフトウェアに適切です。
- [帯域幅] - Mbps または Gbps 単位で測定されます。このカテゴリは、お客様がプロビジョニングする帯域幅を指定できるソフトウェアに適しています。
- [単位] - 測定単位。次に説明する例を参照してください。
- 使用量の単位 - 選択された使用カテゴリに対応するソフトウェア製品の特定の使用量の単位。この使用量の単位はソフトウェアの課金の単位について説明するものです。以下に例を示します。
 - [NodesHrs] (ホストカテゴリに対応)
 - [UserHrs] (ユーザーカテゴリに対応)
 - [GBStored] (データカテゴリに対応)
- 消費量 - Metering Service を使用して価格設定されているソフトウェアは、次の 3 つのいずれかの方法で消費量に対して課金されます。
 - プロビジョニング - ソフトウェアで、リソースの特定の量を使用するようにお客様が設定できます (例えば、ユーザー数や固定量の帯域幅)。お客様は、プロビジョニングした量に対して 1 時間ごとに支払います。
 - 同時実行 - ソフトウェアで、任意の数のホストまたはユーザーがそのソフトウェアに接続できるようにします。お客様は、ソフトウェアにアクセスしたユーザー数またはホスト数に基づいて 1 時間ごとに支払います。
 - 累積 - ソフトウェアで、任意の量のデータを処理または保管に使用できるようにします。お客様は、集計された量に対して 1 時間ごとに支払います。
- 料金表 - Metering Service を使用して価格設定したソフトウェア製品では、単一の料金を指定するか、または独自の料金がある最大 24 個のディメンションを定義する必要があります。価格設定オプションの詳細には以下が含まれます。
 - 単一ディメンション - 最もシンプルな料金オプションです。顧客は、サイズまたはボリュームに関わらず、1 時間あたりリソース単位あたりの単一の料金を支払います (たとえば、1 時間あたりユーザーあたり 0.014 USD、または 1 時間あたりホストあたり 0.070 USD)。
 - 複数ディメンション - この料金オプションは、選択した使用カテゴリが複数の軸で変化する場合に適しています。たとえば、ホストモニタリングでは、ホストのサイズに応じて異なる料金を設定できます。または、ユーザーベースの料金では、ユーザーのタイプ (管理者、パワーユーザー、読み取り専用ユーザー、など) に応じて異なる料金を設定できます。

- 計測 - 1 時間に 1 回、すべての使用量が計測イベントとして記録されます。適切なディメンションおよび使用量を AWS Marketplace Metering Service に送信するようにソフトウェアが設定されている必要があります。
- 配分 - オプションで、追跡する施設ごとに分けて使用量を配分できます。これらの配分は、購入者にはタグとして表示されます。タグを使うと、購入者はコストをタグ別の使用量に分けて表示できます。例えば、ユーザーごとに課金し、ユーザーに「Department」プロパティがある場合、「Department」のキーを持つタグを使用して使用量の割り当てを作成し、値ごとに 1 つの割り当てを行うことができます。この方法では、レポートする価格、ディメンション、または合計使用量は変わりません。ただし、お客様は商品に適したカテゴリ別にコストを確認できます。

ソフトウェアの料金

AWS Marketplace Metering Service でソフトウェアの料金を設定するには、まず使用カテゴリとその使用方法を指定する必要があります。このサービスは 6 つの異なる個別料金シナリオをサポートしています。これらのシナリオのうちいずれかを製品向けに選択する必要があります。

- プロビジョニングされたユーザー (1 時間あたり)
- 同時ユーザー (1 時間あたり)
- プロビジョニングされたホスト (1 時間あたり)
- 同時ホスト (1 時間あたり)
- プロビジョニングされた帯域幅 (1 時間あたり)
- 蓄積されたデータ (1 時間あたり)

次に、選択した使用カテゴリの料金を指定する方法を判断する必要があります。

- 単一価格
- 複数のディメンション (最大 24 個)

「[への製品の追加 AWS Marketplace](#)」では、ディメンションと料金についてわかりやすい説明を提供する方法について説明します。

例: 非線形料金設定によりプロビジョニングされた帯域幅

ネットワークアプライアンスソフトウェアを提供するとします。プロビジョニングされた帯域幅で請求します。使用カテゴリで、[帯域幅] を選択します。帯域幅による請求に加えて、購入者のスケール

アップに応じて別料金を請求するとします。帯域幅カテゴリ内で複数のディメンションを定義することができます。25 Mbps、100 Mbps、および 1 Gbps 向けに個別の料金を定義できます。

例: 複数のディメンションにおける同時ホスト

他の Amazon EC2 インスタンスを監視するソフトウェアを提供するとします。監視中のホスト数で請求するとします。使用カテゴリで、[ホスト] を選択します。ホストごとの請求に加えて、規模の大きなホストを監視する場合は特別料金を請求するとします。ホストカテゴリ内で複数のディメンションを使用することができます。マイクロ、スモール、ミディアム、ラージ、x ラージ、2XL、4XL、8XL インスタンス向けに個別の料金を定義することができます。お客様のソフトウェアを使用して、特定の各ホストを定義されたディメンションのいずれかにマッピングします。また、該当する場合は、使用カテゴリのディメンションごとに個別にレコードを計測します。

への製品の追加 AWS Marketplace

Metering Service を利用するには、が一覧表示 AWS Marketplace する新しい製品を作成する必要があります。製品がすでに にはある場合は AWS Marketplace、現在の製品に加えて新しい AWS Marketplace Metering Service 製品が利用可能になるか、新しいユーザーが利用できる唯一のバージョンとして現在の製品を置き換えるかを決定する必要があります。置き換えを選択すると、既存の製品が から削除 AWS Marketplace され、新しい購入者が利用できなくなります。既存の顧客は、引き続き古い製品とインスタンスにアクセスできますが、必要に応じて新しい製品に移行することができます。新製品では、「[Metering Service を使用するためのソフトウェアの変更](#)」に説明しているように、AWS Marketplace Metering Service の使用料金を測定する必要があります。

AMI を取得したら、標準プロセスに従い、セルフサービスツールを使用して AMI を共有およびスキャンします。さらに、管理ポータルで使用可能なテンプレートを使用して製品ロードフォームに記入後、アップロードして処理を開始します。

AWS Marketplace Metering Service の製品ロードフォームのフィールドに入力するには、次の定義を使用します。製品ロードフォームでは、これらのフィールドは、時間単位の製品および月額の商品と区別するために [Flexible Consumption Pricing (FCP)] というラベルが付けられています。

- タイトル – にすでに製品があり AWS Marketplace 、 で同じ製品を追加する場合は AWS Marketplace Metering Service、括弧内に FCP カテゴリとディメンションを含めてそれらを区別します (例: 「製品タイトル (データ)」)。
- [料金モデル] - ドロップダウンリストより、[使用量] を選択します。
- [FCP カテゴリ] - [使用量] 料金コンポーネントを含む出品製品の請求対象となるカテゴリ。ドロップダウンリストより、[ユーザー]、[ホスト]、[データ]、または [帯域幅] を選択します。

- [FCP 単位] - [使用量] 料金コンポーネントを含む出品製品の請求対象となる測定の単位。選択した FCP カテゴリに基づき、ドロップダウンリストにオプションが表示されます。カテゴリごとに有効な単位を次のテーブルに示します。

カテゴリ	有効な単位
[ユーザー]	UserHrs
[ホスト]	HostHrs
[データ]	MB、GB、TB
[帯域幅]	Mbps、Gbps

- [FCP デイメンション名] - MeterUsage オペレーションを呼び出して計測レコードを送信する際に使用される名前。これは請求レポートに記載されます。ただし、外部向けではないため、わかりやすい名前である必要はありません。名前は 15 文字以内でなければならず、英数字とアンダースコアのみ含めることができます。名前を設定して製品を公開した後は、変更できません。この名前を変更するには、新しい AMI が必要です。
- [FCP デイメンションの説明] - 製品のデイメンションを記述した顧客向けステートメント。説明は 70 文字以内でなければならず、ユーザーフレンドリである必要があります。説明の例としては、「1 時間あたりの管理者数」や「プロビジョニングされた Mbps あたりの帯域幅」などがあります。製品が発行された後に、この説明を変更することはできません。
- [FCP レート] - この製品のユニットあたりのソフトウェア料金。このフィールドは小数点以下 8 桁をサポートしています。

注記:

- 時間単位および年間単位の料金フィールドを入力する必要はありません。
- 無料トライアルと年間料金は互換性がありません。
- AMI、クラスター、AWS リソース機能を使用する製品は、AWS Marketplace Metering Service を使用できません。
- 料金、インスタンスタイプ、または AWS リージョン 変更は、他の AWS Marketplace 製品と同じプロセスに従います。

- AWS Marketplace Metering Service を使用した製品は、時間単位、月別、または Bring-Your-Own-License (BYOL) などの他の料金モデルに変換することはできません。
- AWS Marketplace では、使用手順またはドキュメントに IAM ポリシー情報を追加することをお勧めします。
- 合計で最大 24 個の FCP デイメンションを含めることができます。作成して公開すると、既存のデイメンションを変更することはできませんが、新しいデイメンションを追加できます (最大 24 個)。

ご質問がある場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

Metering Service を使用するためのソフトウェアの変更

顧客の使用状況の記録、Metering Service への時間単位の使用状況レポートの送信、新しい障害モードの処理を行うには、ソフトウェアを変更する必要があります。このソフトウェアは料金とは関係なく使用できますが、ソフトウェアの使用カテゴリ、使用方法、デイメンションについて把握する必要があります。

使用量の測定

ソフトウェアを使用して、選択された使用カテゴリと顧客が使用したデイメンションを判断する必要があります。この値は、Metering Service に 1 時間ごとに送信されます。いずれの場合も、Metering Service に時間単位で送信する目的で、リソースの消費量の測定、記録、読み取りの機能がソフトウェアに搭載されていることを前提としています。

プロビジョニングされた消費の場合、この値は通常、サンプル値としてソフトウェア構成から読み取られますが、1 時間ごとに記録される最大設定値でもあります。同時消費の場合、この値は、定期的なサンプルか、1 時間ごとに記録される最大値のいずれかになります。累積消費の場合、この値は 1 時間ごとに累積されます。

複数のデイメンションの料金については、複数の値を測定し、デイメンションごとに 1 つずつ Metering Service に送信する必要があります。そのためには、AMI を指定する際に、既知のデイメンションセットを使用してソフトウェアをプログラミングまたは設定する必要があります。製品が作成された後は、一連のデイメンションを変更することはできません。

次のテーブルには、1 時間あたりの消費量を測定するための推奨方法を料金シナリオごとに示しています。

シナリオ	測定方法
プロビジョンドユーザー	現在のプロビジョンドユーザー数 (サンプリング)。 -または- プロビジョンドユーザーの最大数 (その時間を参照)
同時ユーザー	現在の同時ユーザー数 (サンプリング)。 -または- 同時ユーザーの最大数 (その時間を参照) -または- 個々のユーザーの合計数 (その時間を参照)
プロビジョンドホスト	現在のプロビジョンドホスト数 (サンプリング)。 -または- プロビジョンドホストの最大数 (その時間を参照)
同時ホスト	現在の同時ホスト数 (サンプリング)。 -または- 同時ホストの最大数 (その時間を参照) -または- 個々のホストの合計数 (その時間を参照)
プロビジョンド帯域幅	現在のプロビジョニングされた帯域幅設定 (サンプリングされた)。 -または-

シナリオ	測定方法
	その時間に発生するプロビジョニングされた最大帯域幅 (その時間を参照)。
蓄積されたデータ	現在の GB に格納されたデータ (サンプリングされた)。 -または- 格納されたデータの最大 GB (その時間を参照) -または- その時間に追加または処理されたデータの合計 (GB)。 -または- その時間に処理されたデータの合計 (GB)。

ベンダー計測タグ付け (オプション)

ベンダー計測タグ付けは、独立系ソフトウェアベンダー (ISV) がソフトウェアの使用状況についてより詳細な情報を購入者に提供し、コスト配分を行うのに役立ちます。

購入者のソフトウェア使用状況をタグ付けするには、コストの配分方法を決定する必要があります。まず、コスト配分に何を期待するかを購入者にたずねます。次に、購入者のアカウントで追跡するプロパティ全体で使用量を分割できます。プロパティの例には、Account ID、Business Unit、Cost Centers、および商品に関連する他のメタデータが含まれます。これらのプロパティはタグとして購入者に公開されます。タグを使用すると、購入者は AWS 請求コンソール (<https://console.aws.amazon.com/costmanagement/>) のタグ値によってコストを使用量に分割して表示できます。ベンダー計測タグ付けでは、報告する価格、ディメンション、合計使用量は変わりません。お客様は商品に適したカテゴリ別にコストを確認できます。

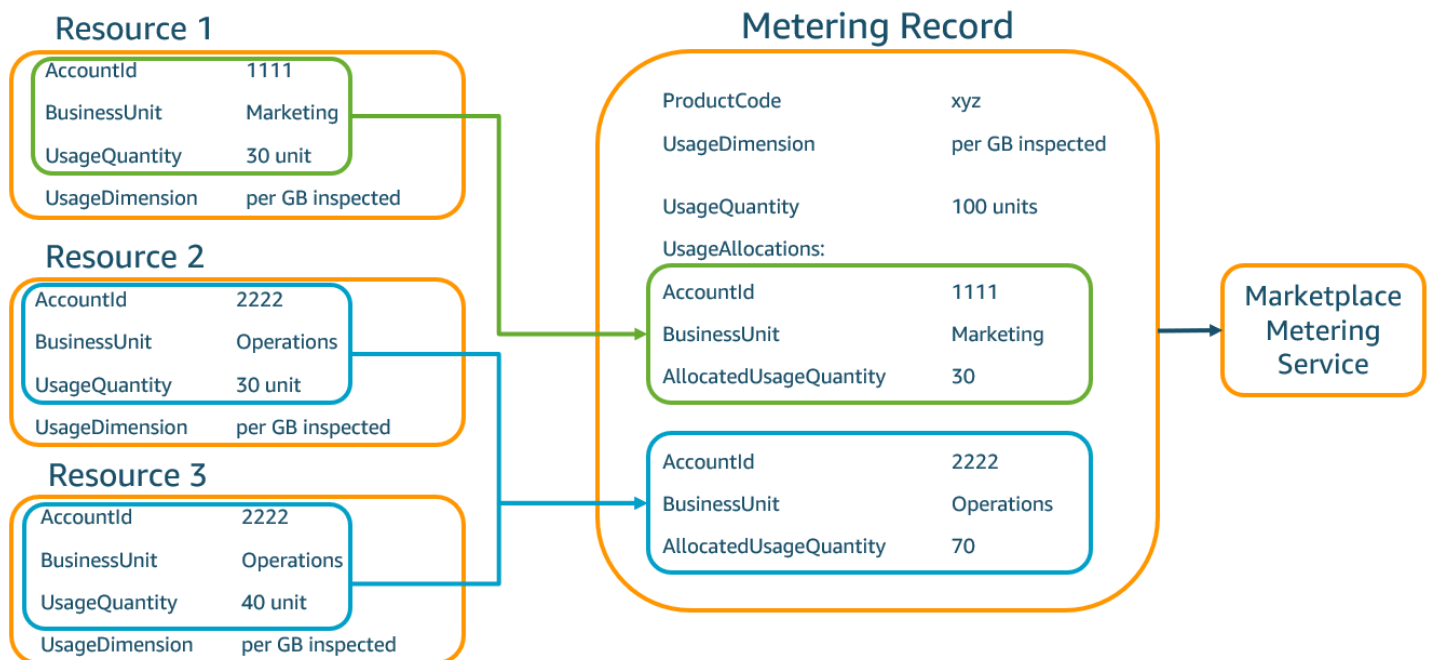
一般的なユースケースでは、購入者が 1 つの AWS アカウントを使って商品を購入します。また、購入者には同じ製品サブスクリプションに関連するユーザーが多数います。Account ID のキーを持つタグを使用して使用量の割り当てを作成し、各ユーザーに使用量を割り当てることができます。この場合、購入者は Billing and Cost Management コンソールで Account ID タグを有効にして、個々のユーザーの使用状況を分析できます。

販売者のエクスペリエンス

販売者は、すべてのリソースの使用量を集計する代わりに、同じタグセットを使用してリソースの計測レコードを集計できます。例えば、販売者は UsageAllocations の異なるバケットを含む計測レコードを作成できます。各バケットは、AccountId や BusinessUnit などの、タグのセットの UsageQuantity を表します。

以下の図では、[リソース 1] には固有の AccountId と BusinessUnit タグセットがあり、[計測レコード] に 1 つのエントリとして表示されます。

[リソース 2] と [リソース 3] にはどちらも同じ AccountId タグ、2222、および同じ BusinessUnit タグ Operations があります。その結果、これらは [計測レコード] の 1 つの UsageAllocations エントリにまとめられます。



また、販売者は、タグのないリソースを割り当てられた使用量の単一の UsageAllocation にまとめて、UsageAllocations のエントリの 1 つとして送信することもできます。

制限には以下が含まれます。

- タグの数 - 5
- UsageAllocations のサイズ (カーディナリティ) - 2,500

検証には次の項目が含まれます。

- タグのキーと値に使用できる文字 - a-zA-Z0-9+ -=._:\@
- UsageAllocation リスト全体の最大タグ数 - 5
- 2 つの UsageAllocations タグに同じタグ (つまり、同じタグキーと値の組み合わせ) を持つことはできません。その場合は、同じ UsageAllocation を使用する必要があります。
- UsageAllocation の AllocatedUsageQuantity の合計は、総使用量である UsageQuantity と等しくなる必要があります。

購入者エクスペリエンス

次の表は、購入者が AccountId および BusinessUnit ベンダータグを有効化した後の購入者エクスペリエンスの例を示しています。

この例では、購入者は [コスト使用状況レポート] で割り当てられた使用量を確認できます。ベンダー計測タグにはプレフィックス “aws:marketplace:isv” が使用されます。購入者は、請求情報とコスト管理の、[コスト配分タグ]、[AWS生成コスト配分タグ] でそれらを有効化できます。

[コスト使用状況レポート] の最初と最後の行は、出品者が Metering Service に送信する内容に関係します ([販売者のエクスペリエンス](#) 例を参照)。

コスト使用状況レポート (簡略版)

ProductCode	購入者	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	70	2222	オペレーション
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	30	3333	財務
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	20	4444	IT

ProductCode	購入者	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	20	5555	マーケティング
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	30	1111	マーケティング

コード例については、「[使用量配分タグ付き MeterUsage \(オプション\)](#)」を参照してください。

カスタム計測を設定する

AWS Marketplace Metering Service の使用の詳細については、「[AWS Marketplace Metering Service を使用した AMI 製品のカスタム計測を構成する](#)」を参照してください。

AWS Marketplace Metering Service を使用した AMI 製品のカスタム計測を構成する

Note

カスタム計測料金の AMI ベースの製品の場合、ソフトウェアは Amazon EC2 インスタンスにアタッチされた Amazon Elastic Compute Cloud の IAM ロールの一時的な AWS 認証情報を使用して [MeterUsage API](#) を呼び出す必要があります。長期アクセスキーの使用はサポートされていません。

AWS Marketplace Metering Service は、使用カテゴリごとにソフトウェアを直接請求するために使用できる価格および計測機能です。ユーザー、データ、帯域幅、ホスト、ユニットの 5 つの使用カテゴリがあります。Metering Service は、Amazon マシンイメージ (AMI) ベース、コンテナベース、および Software as a Service (SaaS) ベースの製品で利用できます。以下のセクションでは、AWS Marketplace Metering Service でカスタム計測を設定する方法について詳述します。

AWS Marketplace Metering Service では、いくつかの新しいシナリオが利用可能になります。たとえば、ソフトウェアでホストをモニタリングしている場合は、モニタリングされたホストごとに料金

を請求できます。ホストのサイズに基づいて異なる料金を設定でき、モニタリングされる 1 時間ごとの同時実行ホスト数に対して課金できます。同様に、組織全体で多数のユーザーがソフトウェアにサインインできる場合は、ユーザー数によって課金できます。1 時間ごとに、プロビジョニングされたユーザーの合計数が請求されます。

詳細については、「[AWS Marketplace Metering Service API Reference](#)」を参照してください。

AMI ベースの製品の AWS Marketplace Metering Service API をカスタム計測料金と統合する方法の詳細については、AWS Marketplace 販売者ワークショップの「[カスタムユニットラボで価格が設定されている AMI 製品のリスト](#)」を参照してください。

トピック

- [要件](#)
- [AWS Marketplace Metering Service の呼び出し](#)
- [障害処理](#)
- [制限](#)
- [コード例](#)

要件

Metering Service を使用するすべての AMI ベースのソフトウェアは次の要件を満たしている必要があります。

- ソフトウェアは、Amazon マシンイメージ (AMI) AWS Marketplace を介して から起動する必要があります。
- に既存の製品がある場合は AWS Marketplace、新しい AMI を送信し、この機能を有効にする新しい製品を作成する必要があります。
- すべてのソフトウェアは AWS Identity and Access Management (IAM) ロールでプロビジョニングする必要があります。エンドカスタマーは、ユーザーがそのソフトウェアでプロビジョニングしている Amazon Elastic Compute Cloud (Amazon EC2) インスタンスに IAM ロールを追加する必要があります。ソフトウェアをデプロイする場合は、IAM ロールの使用はオプションです AWS Marketplace。これは AWS Marketplace Metering Service ソフトウェアをデプロイするときには必要です。
- ソフトウェアでは、何らかの方法で消費量を決定できる必要があります。

AWS Marketplace Metering Service の呼び出し

ソフトウェアを使用して、時間単位で Metering Service を呼び出し、その時間の消費値を記録する必要があります。

ソフトウェアが起動したら、起動した時間 (分/時間) を記録する必要があります。これは、start-minute と呼ばれます。起動時間の 1 時間ごとに、ソフトウェアはその時間の消費量を判断し、Metering Service を呼び出します。この値を取得する方法の詳細については、「[Metering Service を使用するためのソフトウェアの変更](#)」を参照してください。

1 時間ごとの開始時間に起動するには、以下のいずれかのアプローチをソフトウェアで使用する必要があります。

- ソフトウェア内のスレッド。
- インスタンスまたはソフトウェアで起動するデーモンプロセス。
- アプリケーションのスタートアップ時に設定される cron ジョブ。

Note

お客様のソフトウェアで、顧客のインスタンスに設定された IAM ロールを使用して AWS Marketplace Metering Service を呼び出し、消費のディメンションと量を指定する必要があります。

次の実装例のように、ソフトウェアは AWS SDK を使用して AWS Marketplace Metering Service を呼び出すことができます。

1. サービスクライアントを作成するにはインスタンスプロファイルを使用します。そのためには、EC2 インスタンス用に設定されたロールが必要です。このロールの認証情報は、SDK によって自動的に更新されます。
2. 1 時間ごとに、ソフトウェア設定とステータスを確認し、その時間の消費値を決定します。これには、ディメンションごとの値の収集が含まれる場合があります。
3. 次のパラメータを指定して、SDK クライアントの meterUsage メソッドを呼び出します (使用量があるディメンションごとに追加で呼び出す)。
 - timestamp - 記録中の時間のタイムスタンプ (UTC で)。
 - productCode - ソフトウェアに割り当てられた製品コード。

- dimension - ソフトウェアに割り当てられた 1 つのディメンション (または複数のディメンション)
- quantity - ある時間の消費値。
- allocations - (オプション) 追跡するプロパティ全体にわたって使用量を割り当てることができます。これらの割り当てを合計すると、レコード内の総消費量になる必要があります。購入者にとって、これらは請求ツール (AWS Billing and Cost Management コンソールなど) に潜在的なコスト配分タグとして表示されます。購入者がこれらのタグを使用してコストを追跡するには、アカウントでタグを有効にする必要があります。

さらに、ソフトウェアを使用して、リージョン内の AWS Marketplace Metering Service エンドポイントを呼び出す必要があります。製品に適切なリージョンのエンドポイントが設定されている必要があります。そのため、記録は us-east-1 より us-east-1 エンドポイント、us-west-2 より us-west-2 エンドポイントに送信されます。リージョン内呼び出しを行うと、購入者にはより安定したエクスペリエンスが提供され、無関係なリージョンのアベイラビリティが別のリージョンで実行されているソフトウェアに影響を及ぼすことを防ぐことができます。

サービスに計測レコードを送信するときは、リージョンの AWS Marketplace Metering Service に接続する必要があります。getCurrentRegion() ヘルパーメソッドを使用して、EC2 インスタンスが実行されているリージョンを特定し、このリージョンの情報を MeteringServiceClient コンストラクタに渡します。SDK コンストラクタ AWS リージョン で を指定しない場合、デフォルトの us-east-1 リージョンが使用されます。アプリケーションでサービスへのクロスリージョンコールを行うと、拒否されます。詳細については、[Determining an Application's Current Region](#) および [getCurrentRegion\(\)](#) を参照してください。

障害処理

製品は、公共のインターネットエンドポイントであるサービスに計測レコードを送信するため、使用状況をキャプチャして請求することができます。顧客は計測レコードの配信を妨げる方法でネットワーク設定を変更する可能性があるため、製品は障害モードを選択することで考慮する必要があります。

Note

一部の計測の失敗は、への接続における一時的な問題である場合があります AWS Marketplace Metering Service。AWS Marketplace では、短期的な停止やネットワークの問題を回避するために、エクスポネンシャルバックオフで最大 30 分間の再試行を実装することを強くお勧めします。

通常、ソフトウェアはフェイルオープン (警告メッセージを表示しながら完全な機能を維持する) か、またはフェイルクローズ (接続が再確立されるまでアプリケーション内のすべての機能を無効にする) を行うことができます。フェイルオープン、フェイルクローズ、またはアプリケーション固有の機能を選択できます。2 時間以内の計測に失敗した場合は、フェイルクローズを行わないことをお勧めします。

部分的なフェイルオープンの場合の一例として、ソフトウェアへのアクセスを引き続き許可することはできますが、購入者がソフトウェア設定を変更することはできません。または、購入者はソフトウェアに引き続きアクセスできますが、追加のユーザーを作成することはできません。ソフトウェアを使用して、この障害モードを定義して実行する必要があります。AMI の送信時、ソフトウェアの障害モードを含む必要があります。また、後に変更することはできません。

制限

Metering Service 対応ソフトウェアを設計および送信する場合、次の制約事項に留意してください。

- お客様の IAM ロールとインターネットゲートウェイの要件 - お客様はインターネットゲートウェイを持ち、特定のアクセス許可を持つ IAM ロールでソフトウェアを起動する必要があります。詳細については、「[AWS Marketplace 計測と使用権限 API アクセス許可](#)」を参照してください。これらのうち 2 つの条件が満たされない場合、ソフトウェアは Metering Service に接続できません。
- 既存の Metering Service product 製品への新しい使用カテゴリの追加および変更の不可 - お客様は、ソフトウェア製品をサブスクライブすると、利用規約に同意したものと見なされます。Metering Service を使用して製品の使用カテゴリを変更するには、新しい製品と新しいサブスクリプションが必要です。
- 既存の Metering Service 製品のディメンションの変更の不可 - お客様は、ソフトウェア製品をサブスクライブすると、利用規約に同意したものと見なされます。Metering Service を使用して製品のディメンションを変更するには、新しい製品と新しいサブスクリプションが必要です。既存の製品には、最大 24 個まで新しいディメンションを追加できます。
- 無料トライアルおよび年間サブスクリプションは利用不可 - Metering Service 製品は、起動時に無料トライアルおよび年間サブスクリプションをサポートしていません。
- マルチインスタンスまたはクラスターベースのデプロイメントに関する考慮事項 - 一部のソフトウェアは、マルチインスタンスデプロイメントの一部としてデプロイされます。ソフトウェアの設計時には、消費量を計測する方法と場所、および計測レコードが出力される場所を検討してください。

コード例

次のコード例は、製品の公開およびメンテナンスに必要な AWS Marketplace API と AMI 製品を統合するのに役立てることを目的として提供されています。

使用量配分タグ付き **MeterUsage** (オプション)

次のコード例は、消費料金モデルを持つ AMI 製品に関連しています。Python の例では、従量制料金を顧客に請求するために、適切な使用量割り当てタグを含む計測レコードを AWS Marketplace に送信します。

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited

# Import AWS Python SDK
import boto3
import time

usageRecord = [
    {
        "AllocatedUsageQuantity": 2,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "IT" },
                { "Key": "AccountId", "Value": "123456789" },
            ]
    },
    {
        "AllocatedUsageQuantity": 1,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "Finance" },
                { "Key": "AccountId", "Value": "987654321" },
            ]
    }
]
```

```
marketplaceClient = boto3.client("meteringmarketplace")

response = marketplaceClient.meter_usage(
    ProductCode="testProduct",
    Timestamp=int(time.time()),
    UsageDimension="Dimension1",
    UsageQuantity=3,
    DryRun=False,
    UsageAllocations=usageRecord
)
```

MeterUsage の詳細については、「AWS Marketplace Metering Service API リファレンス」の「[MeterUsage](#)」を参照してください。

レスポンスの例

```
{ "MeteringRecordId": "string" }
```

での AMI 製品の契約料金 AWS Marketplace

Amazon マシンイメージ (AMI) ベースの製品の契約料金とは、購入者が 1 つの AMI 製品または 1 つの AMI と AWS CloudFormation スタックに対して前払い料金を支払うことを意味します。契約料金の AMI ベースの製品の場合、は、お客様とお客様の間の契約に基づいて、事前に、またはお客様が定義した支払いスケジュールに従って顧客に AWS Marketplace 請求します。その時点で、それらのリソースの使用権限が付与されます。このトピックでは、契約料金について詳述します。

料金を設定するには、顧客に提供する 1 つ以上の契約期間を選択します。契約期間ごとに異なる価格を入力できます。契約期間は 1 か月、12 か月、24 か月、36 か月です。プライベートオファーでは、カスタム期間を月単位 (最大 60 か月) で指定できます。

Note

無料トライアルは、契約料金の AMI 製品ではご利用いただけません。

製品の価格を最もよく表すカテゴリを選択します。料金カテゴリは、AWS Marketplace ウェブサイトの顧客に表示されます。[帯域幅] (GB/s、MB/s)、[データ] (GB、MB、TB)、[ホスト]、[リクエスト]、[階層]、[ユーザー] のいずれかを選択できます。事前定義カテゴリのどれもニーズに適合しない場合は、汎用的な [単位] カテゴリを選択できます。

このオファーには最大 24 のディメンションを追加できます。各ディメンションには次のデータが必要です。

- 契約カテゴリ - 消費ベースの料金設定のない契約製品の場合は、契約内のディメンションのカテゴリに最も近いカテゴリを選択するか、契約内のディメンションの単位に似た値がない場合は [単位] を選択できます。
- 契約単位 - 選択したカテゴリに基づいて、ディメンションに最も近い単位の値を 1 つ選択します。
- 契約ディメンションによる複数購入が可能 - このフィールドは、オファーが階層型の料金設定オファーか非階層型のオファーかを示すために使用されます。

階層型オファー - 購入者がオファー内の利用可能なディメンションのうちの 1 つだけをサブスクライブできるようにします。階層型オファーのディメンションには数量という概念はありません。特定のディメンションで契約を結ぶことは、基本的に、購入者がそのディメンションが示す特定の機能を選択したことを意味します。

非階層型オファー - 顧客は契約の一環として複数のディメンションを調達でき、各ディメンションの複数のユニットを調達できます。

このフィールドの値を true に設定すると、そのオファーは非階層型オファーであることを示します。このフィールドに false を設定すると、そのオファーは階層型オファーであることを示します。

製品積載フォーム (PLF) を使用して AMI 製品の契約を作成する場合、料金ディメンションに次のフィールドを定義する必要があります。

- Contracts DimensionX API Name – 購入者の AWS License Manager アカウントで生成されたライセンスに表示される名前。この名前は Checkoutlicense API コールの Entitlement Name の値としても使用されます。
- [契約 DimensionX 表示名] - AWS Marketplace ウェブサイトの製品詳細ページと調達ページに表示される、顧客向けのディメンション名です。わかりやすい名前を付けてください。名前の最大長は 24 文字です。リストが公開されたら、Name の値を変更することはできません。
- [契約 DimensionX 説明] - 特定のディメンションが提供する機能など、製品のディメンションに関する追加情報を提供する、顧客向けのディメンションの説明です。説明の最大長は 70 文字です。
- [契約 DimensionX 数量] - 商品の契約が修正された場合の比例配分の計算に使用されます。このフィールドのこの値は、すべての契約オファーで 1 に設定する必要があります。この値は編集しないでください。

- [契約 DimensionX 1 か月料金] - このディメンションの 1 か月分の使用権限に対して請求される契約レート。非階層型オファースの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは小数点以下 3 桁がサポートされています。
- [契約 DimensionX 12 か月料金] - ディメンションに対する 12 か月間の使用権限に対して請求される契約レート。非階層型オファースの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは小数点以下 3 桁がサポートされています。
- [契約 DimensionX 24 か月料金] - ディメンションに対する 24 か月間の使用権限に対して請求される契約レート。非階層型オファースの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは小数点以下 3 桁がサポートされています。
- [契約 DimensionX 36 か月料金] - ディメンションに対する 36 か月間の使用権限に対して請求される契約レート。非階層型オファースの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは小数点以下 3 桁がサポートされています。

例: データストレージアプリケーション

	1 か月料金	12 か月料金	24 か月料金	36 か月料金
暗号化されていないデータ (GB)	1.50 USD/GB	16.00 USD/GB	30.00 USD/GB	60.00 USD/GB
暗号化されているデータ (GB)	1.55 USD/GB	16.60 USD/GB	31.20 USD/GB	61.20 USD/GB

例: ログモニタリング製品

	1 か月料金	12 か月料金	24 か月料金	36 か月料金
Basic (10 ホストのモニタリング、5 コンテナのモニタリング)	100 USD	1,000 USD	2,000 USD	4000 USD
Standard (20 ホストのモニタリング、10 コンテナのモニタリング)	200 USD	2,000 USD	4000 USD	8,000 USD

	1 か月料金	12 か月料金	24 か月料金	36 か月料金
Pro (40 ホスト のモニタリング、20 コンテナ のモニタリング)	400 USD	4000 USD	8,000 USD	16,000 USD
1 時間あたりの モニタリング対 象追加ホスト	10 USD	100 USD	\$200	400 USD
1 時間あたりの モニタリング対 象追加コンテナ	10 USD	100 USD	\$200	400 USD

Note

価格は、1 か月、12 か月、24 か月、36 か月の期間に対して定義できます。製品にこれらのオプションを 1 つ以上提供することができます。期間は各ディメンション間で同じである必要があります。

Example

たとえば、ReadOnlyUsers ディメンションと AdminUsers ディメンションがある場合、ReadOnlyUsers に年間料金を提供する場合は、AdminUsers の年間料金も提供する必要があります。

自動更新

お客様が AMI 契約 AWS Marketplace を使用して を通じて製品を購入すると、契約条件を自動的に更新することに同意できます。顧客は 1 年、2 年、または 3 年の期間中に、使用権限に対して毎月支払い続けます。

お客様はいつでも更新の設定を変更できます。詳細については、「AWS Marketplace 購入者ガイド」の「[既存の契約の変更](#)」を参照してください。

を使用したライセンスと AMI ベースの製品の関連付け AWS License Manager

契約料金の Amazon Machine Image (AMI) ベースの製品の場合、AWS License Manager を使用してライセンスを製品に関連付けることができます。AWS License Manager は、顧客が購入したライセンス (使用権限とも呼ばれます) をアプリケーションが追跡および更新できるようにするライセンス管理ツールです。統合が完了したら、AWS Marketplace に製品リストを公開できます。次のセクションでは、AWS License Manager を使用したライセンスと AMI ベース商品の関連付けに関して説明します。

詳細については AWS License Manager、[AWS License Manager 「ユーザーガイド」](#) および AWS CLI 「コマンドリファレンス」の[AWS License Manager 「」](#) セクションを参照してください。

AMI ベースの製品 AWS License Manager と契約料金との統合の詳細については、AWS Marketplace 販売者ワークショップの[前払いラボで料金が設定されている AMI 製品のリスト](#)を参照してください。

Note

- 契約の有効期限を過ぎると、顧客は AMI の新しいインスタンスを起動できなくなります。ただし、契約期間中は、インスタンスをいくつでも起動できます。これらのライセンスはノードロックされたり、特定のインスタンスに結び付けられたりすることはありません。
- [プライベートオファーの作成] - 販売者は、AWS Marketplace 管理ポータルプライベートオファー作成ツールを使用して、製品のプライベートオファーを作成できます。
- [レポート] - AWS Marketplace 管理ポータルの [レポート] セクションで Amazon S3 バケットを設定して、データフィードを設定できます。詳細については、「[の販売者レポート、データフィード、ダッシュボード AWS Marketplace](#)」を参照してください。

トピック

- [ライセンスモデル](#)
- [ワークフロー統合](#)
- [License Manager 統合の前提条件](#)
- [AMI ベースの製品をと統合する AWS License Manager](#)
- [ライセンスの更新とアップグレード](#)

ライセンスモデル

AWS Marketplace との統合では、次の 2 つのライセンスモデル AWS License Manager がサポートされています。

- [設定可能なライセンスモデル](#)
- [階層型ライセンスモデル](#)

設定可能なライセンスモデル

設定可能なライセンスモデル (定量化可能なライセンスモデルとも呼ばれます) では、購入者がライセンスを調達した後に、特定の量のリソースを購入者に付与できます。

料金設定ディメンションと単価を設定します。次に、購入者は購入するリソースの数量を選択できます。

Example 料金ディメンションと単価の

料金設定ディメンション (データバックアップなど) と単価 (1 ユニットあたり 30 USD など) を設定できます。

購入者は 5 単位、10 単位、または 20 単位の購入を選択できます。

製品は使用状況を追跡して測定し、消費されたリソースの量を測定します。

設定モデルでは、使用権限は次の 2 つのいずれかにカウントされます。

- [ドローダウンライセンス](#)
- [フローティングライセンス](#)

ドローダウンライセンス

ライセンスは、使用時に許可されたライセンス数のプールから引き出されます。その使用権限は永久にチェックアウトされ、ライセンスプールに戻すことはできません。

Example 限られた量のデータを処理する

ユーザーには 500 GB のデータを処理する権限があります。ユーザーがデータを処理し続けると、500 GB のライセンスがすべて消費されるまで 500 GB のプールからデータが引き出されます。

ドローダウンライセンスの場合は、CheckoutLicense API オペレーションを使用して、消費されたライセンスユニットを確認できます。

Example Amazon S3 へのバックアップの単位数/年

Amazon Simple Storage Service へのバックアップを 1 年間最大 1024 ユニットのデータに許可するストレージ製品があります。アプリケーションは、複数の Amazon EC2 インスタンスを使用して起動できます。アプリケーションには、データを追跡して集約するメカニズムがあります。ソフトウェアは、バックアップのたびに、または一定の間隔で、製品 ID を使用して CheckoutLicense API オペレーションを呼び出し、消費量を更新します。

ソフトウェアが CheckoutLicense を呼び出して 10 ユニットのデータをチェックアウトします。合計容量が顧客が購入したバックアップ制限に達すると、API コールは失敗します。

リクエスト

```
linux-machine ~]$ aws license-manager checkout-license\
--product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \
--checkout-type "PERPETUAL" \
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements "Name=DataConsumption, Value=10, Unit=Count" \
--client-token "AKIAIOSFODNN7EXAMPLE"
```

レスポンス

```
{
  "CheckoutType": "PERPETUAL",
  "EntitlementsAllowed": [
    {
      "Name": "DataConsumption",
      "Count": 10,
      "Units": "Count",
      "Value": "Enabled"
    }
  ],
  "Expiration": "2021-04-22T19:02:36",
  "IssuedAt": "2021-04-22T18:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

フローティングライセンス

ライセンスは、使用後に許可されたライセンス数のプールに返却されます。

Example 固定された上限を超えるユーザー数

1 人のユーザーには、アプリケーションで 500 人の同時ユーザーを割り当てることができます。ユーザーがログインしてログアウトすると、そのユーザーは引き出され、500 人のユーザープールに戻ります。ただし、500 人の同時ユーザーが固定の上限であるため、アプリケーションはプールから 500 人を超えるユーザーを引き出すことはできません。

フローティングライセンスの場合は、CheckInLicense API オペレーションを使用してライセンス単位を使用権限プールに戻すことができます。

Example 1 年間の同時ユーザー数

製品の料金は、同時接続ユーザー数に基づいて決定されます。顧客は 10 ユーザー分のライセンスを 1 年間購入します。顧客は AWS Identity and Access Management (IAM) アクセス許可を与えてソフトウェアを起動します。ユーザーがログインすると、アプリケーションが CheckoutLicense API オペレーションを呼び出して数量を 1 つ減らします。ユーザーがログアウトすると、アプリケーションは CheckInLicense API オペレーションを呼び出してそのライセンスをプールに戻します。CheckInLicense を呼び出さない場合、ライセンス単位は 1 時間後に自動的にチェックインされます。

Note

次のリクエストでは、key-fingerprint はプレースホルダー値ではなく、すべてのライセンスが公開されるフィンガープリントの実際の値です。

リクエスト

```
linux-machine ~]$ aws license-manager checkout-license \
--product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \
--checkout-type "PROVISIONAL" \
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements "Name=ReadOnlyUSers, Value=10, Unit=Count" \
--client-token "AKIAIOSFODNN7EXAMPLE"
```

レスポンス

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "ReadOnlyUsers",
      "Count": 10,
      "Units": "Count",
      "Value": "Enabled"
    }
  ],
  "Expiration": "2021-04-22T19:02:36",
  "IssuedAt": "2021-04-22T18:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

階層型ライセンスモデル

階層型ライセンスモデルでは、購入者がライセンスを調達した後に、特定のレベル (階層) のアプリケーション機能を利用できるようになります。

製品には、ベーシック、中級、プレミアムなどの階層を作成します。次に、購入者はあらかじめ定義されている階層の 1 つを選択します。

アプリケーションはアプリケーションの使用状況を追跡したり、測定する必要はありません。

階層型ライセンスモデルでは、使用権限はカウントされず、代わりに顧客が購入したサービスの階層が示されます。

バンドルされた機能を同時に提供する場合は、階層型ライセンスモデルを使用することをお勧めします。

Example ベーシック、中級、プレミアムの各レベル

お客様は、ソフトウェアの 3 つのレベル (ベーシック、中級、プレミアム) のいずれかの契約に署名できます。これらの階層にはそれぞれ独自の料金設定があります。CheckoutLicense API オペレーションを呼び出し、リクエストで可能なすべての階層を指定して、ソフトウェアが顧客がサインアップした階層を特定できます。

リクエストのレスポンスには、顧客が調達した階層に対応する使用権限が含まれています。この情報に基づいて、ソフトウェアは適切なカスタマーエクスペリエンスを提供できます。

リクエスト

```
linux-machine ~]$ aws license-manager checkout-license\
--product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \
--checkout-type "PROVISIONAL" \
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements "Name=BasicTier, Unit=None" "Name=IntermediateTier, Unit=None" \
"Name=PremiumTier, Unit=None"
```

レスポンス

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "IntermediateTier",
      "Units": "None"
    }
  ],
  "Expiration": "2021-04-22T19:02:36",
  "IssuedAt": "2021-04-22T18:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

ワークフロー統合

以下の手順は、AMI 製品を AWS License Manager と統合するためのワークフローを示しています。

1. 販売者は、AWS License Manager 統合した製品を作成します。
2. 販売者は製品を一覧表示します AWS Marketplace。
3. 購入者は で製品を見つけ AWS Marketplace で購入します。
4. ライセンスは、AWS アカウントの購入者に送付されます。
5. 購入者は、Amazon Elastic Compute Cloud (Amazon EC2) インスタンス、Amazon Elastic Container Service (Amazon ECS) タスク、または Amazon Elastic Kubernetes Service (Amazon EKS) ポッドソフトウェアを起動してソフトウェアを使用します。顧客は IAM ロールを使用してデプロイします。
6. ソフトウェアは購入者の AWS License Manager アカウントでライセンスを読み取り、購入した使用権限を検出し、それに応じて機能をプロビジョニングします。

Note

License Manager は追跡や更新を行いません。これは販売者のアプリケーションによって行われます。

License Manager 統合の前提条件

製品を公開する前に、次を行う必要があります。

1. で新しい AMI 製品を作成し AWS Marketplace 管理ポータル、その製品コードを書き留めます。
2. 必要な料金情報を製品積載フォーム (PLF) に記入し、処理のために当社に返送してください。
3. CheckoutLicense、ExtendLicenseConsumption、および CheckInLicense を呼び出すために必要な IAM アクセス許可を持つアプリケーションを実行するタスクまたはポッドの IAM ロールを使用します。

必要な IAM アクセス許可の詳細は、次の IAM ポリシーに記載されています。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "license-manager:CheckoutLicense",
        "license-manager:GetLicense",
        "license-manager:CheckInLicense",
        "license-manager:ExtendLicenseConsumption",
        "license-manager:ListReceivedLicenses"
      ],
      "Resource": "*"
    }
  ]
}
```

4. 定義したすべての料金ディメンションのレコードを使用して、RegisterUsage API オペレーションをテストコールします。

AMI ベースの製品を と統合する AWS License Manager

[AWS License Manager](#) API を使用して AMI ベースの製品を License Manager と統合できます。AMI ベースの製品を使用して Amazon EC2 AWS Marketplace インスタンスを起動します。

Note

次の手順を実行する前に、[the section called “License Manager 統合の前提条件”](#) を完了していることを確認します。

AMI ベースの製品を License Manager と統合するには

1. [the section called “License Manager でのテストライセンスの作成”](#) のステップを完了します。統合をテストするには、License Manager でテストライセンスを作成する必要があります。
2. ステップ 1 で取得したライセンスの Amazon リソースネーム (ARN) を使用して [GetLicense](#) API オペレーションを実行します。後でできるように、GetLicense レスポンスの KeyFingerprint 属性の値を書き留めておきます。
3. 最新のパブリック AWS SDK をダウンロードしてアプリケーションに含めます。
4. 購入者がアプリケーションのライセンスを使用する資格を持っていることを確認するには、[CheckoutLicense](#) API オペレーションを実行します。ステップ 1 で取得したテストライセンスの使用権限の詳細とキーフィンガープリントを使用します。

ライセンスに該当する使用権限が見つからないか、使用権限の最大数を越えた場合は、CheckoutLicense API オペレーションから NoEntitlementsAllowedException が返されます。使用権限が有効であるか、使用可能な場合、CheckoutLicense オペレーションはリクエストされた使用権限とその値を含む成功レスポンスを返します。

5. (フローティング使用権限の場合にのみ必須) CheckoutLicense レスポンスで受け取った LicenseConsumptionToken を使用して [CheckinLicense](#) API オペレーションを実行します。このアクションは、以前にチェックアウトした使用権限を、使用可能な使用権限のプールに戻します。
6. ステップ 1 で作成したテストライセンスと License Manager の統合を正常に確認したら、コード内のキーフィンガープリントを `aws:294406891311:AWS/Marketplace:issuer-`

fingerprint に更新します。これで、AWS Marketplaceによって発行されたライセンスを使用する準備ができました。

AMI 製品にアプリケーションを構築するリリースプロセスに従い、製品の公開プロセス AWS Marketplace に従って製品を に送信します。

License Manager でのテストライセンスの作成

(AWS Command Line Interface AWS CLI) のバージョン 2 を使用して、 でテストライセンスを作成します AWS License Manager。このテストライセンスは、 AWS License Manager 統合の検証とテストにのみ使用されます。テストが完了したら、テストライセンスを削除できます。実際のライセンスは、別のキーフィンガープリントを使用して AWS Marketplace によって生成されます。

AWS Marketplace は、 で 2 種類のエンタイトルメントをサポートしています AWS License Manager。ただし、1 つの製品に対して有効にできるのは 1 つのタイプだけです。テストライセンスを含むライセンスを作成するときは、次のどちらかのタイプの資格を指定する必要があります。

階層型ライセンス - 階層型ライセンスモデルでは、顧客に特定のアプリケーション機能の使用権が付与されます。顧客は購入するユニットの数量を定義することはできません。ただし、定義済みのパッケージまたは階層を 1 つ選択することはできます。顧客は後で契約を変更して別の階層にサブスクライブできます。

設定可能な使用権限 - 設定可能なライセンスモデルでは、顧客がライセンスを購入すると、一定量のリソースに使用権限が付与されます。お客様はサブスクリプションプロセス中に購するユニット数を選択し、そのユニット料金に基づいて請求されます。顧客は複数のディメンションをサブスクライブすることもできます。

CheckoutLicense API オペレーションで使用するために必要なパラメータは次のとおりです。

- CheckoutType - 有効な値は Perpetual または Provisional です。
 - Perpetual - チェックアウトされた使用権限の数がプールからなくなる場合に使用されます。
例: 購入者には 500 GB のデータを処理する権限があります。データの処理を続けると、500 GB のプールからその量が引き出され、使い果たされます。購入したライセンスの有効期限が切れている、または間もなく切れるかどうかのステータスを取得して、顧客に通知を送信します。
 - Provisional - 使用権限がプールからチェックアウトされ、使用後に返却されるフローティングライセンス使用権限に使用されます。例: ユーザーには、アプリケーション内で同時に 500 人のユーザーを使用する権利があります。ユーザーがログインしてログアウトすると、そのユーザーは引き出され、500 人のユーザープールに戻ります。フローティングライセンスの使用権限

の詳細については、[AWS「License Manager で販売者が発行したライセンス」を参照してください。](#)

- ClientToken - 大文字と小文字が区別される一意の識別子。これにより、何度試しても正確な結果が得られ、同じになります。リクエストごとにランダムな共通の一意の識別子 (UUID) を使用することをお勧めします。
- Entitlements - チェックアウトする使用権限のリスト。
 - 階層型使用権限の場合は、Name プロパティと Unit プロパティを次のように指定します。

```
{  
  
  "Name": "<Entitlement_Name>",  
  
  "Unit": "None"  
  
}
```

- 設定可能な使用権限には、Name プロパティ、Unit プロパティおよび Value プロパティを次のように指定します。

```
{  
  
  "Name": "<Entitlement_Name>",  
  
  "Unit": "<Entitlement_Unit>",  
  
  "Value": <Desired_Count>{  
  
  }
```

- KeyFingerprint - このキーフィンガープリントを使用して、ライセンスが AWS Marketplace によって発行されたものであることを確認します。によって発行されたライセンスのキーフィンガープリント AWS Marketplace は次のとおりです。

```
aws:294406891311:AWS/Marketplace:issuer-fingerprint
```

- Product SKU – 製品に関連付けられているグローバル一意識別子 (GUID) 形式の AWS Marketplace 製品 ID。

Example 設定可能な使用権限

以下は、CheckoutLicense API オペレーションを使用して PowerUsers という名前の設定可能な使用権限をチェックアウトするリクエストの例です。

```
aws license-manager checkout-license \  
  product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \  
  checkout-type "PROVISIONAL" \  
  client-token "79464194dca9429698cc774587a603a1" \"Statement\":[  
  entitlements "Name=PowerUsers,Value=1,Unit=Count" \  
  key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint"
```

Example 階層型使用権限

以下は、CheckoutLicense API オペレーションを使用して EnterpriseEdition という名前の機能使用権限をチェックアウトするリクエストの例です。

```
aws license-manager checkout-license \  
  --product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \  
  --checkout-type "PROVISIONAL" \  
  --client-token "79464194dca9429698cc774587a603a1" \  
  --entitlements "Name=EnterpriseEdition,Unit=None" \  
  --key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint"
```

AMI ベース製品のテストライセンスを作成するには

1. v2 AWS CLI がインストールされているローカル環境から、次のスクリプトを実行します。このスクリプトはテストライセンスを作成し、適切な製品詳細を設定します。

Note

ソフトウェアをデプロイしてテスト AWS アカウント するテスト AWS アカウント とは異なる を使用します。ライセンスの作成、付与、チェックアウトを同じ AWS アカウントで行うことはできません。

```
#!/bin/bash  
  
# Replace with intended product ID on AWS Marketplace  
PRODUCT_ID=<REPLACE-WITH-PRODUCT-ID>  
  
# Replace with license recipient's AWS Account ID  
BENEFICIARY_ACCOUNT_ID=<REPLACE-WITH-BENEFICIARY-ACCOUNT-ID>
```

```
# Replace with your product's name
PRODUCT_NAME="Test Product"

# Replace with your seller name on AWS Marketplace
SELLER_OF_RECORD="Test Seller"

# Replace with intended license name
LICENSE_NAME="AWSMP Test License"

# Replace the following with desired contract dimensions
# More info here: https://docs.aws.amazon.com/license-manager/latest/APIReference/API\_Entitlement.html
# Example "configurable entitlement"
ENTITLEMENTS='[
  {
    "Name": "ReadOnly",
    "MaxCount": 5,
    "Overage": false,
    "Unit": "Count",
    "AllowCheckIn": true
  }
]'
# Example "tiered entitlement"
# ENTITLEMENTS='[
#   {
#     "Name": "EnterpriseUsage",
#     "Value": "Enabled",
#     "Unit": "None"
#   }
# ]'

# Format "yyyy-mm-ddTHH:mm:ss.SSSZ"
# This creates a validity period of 10 days starting the current day
# Can be updated to desired dates
VALIDITY_START=$(date +%Y-%m-%dT%H:%M:%S.%SZ)
VALIDITY_END=$(date --date="+10 days" +%Y-%m-%dT%H:%M:%S.%SZ)

# Configuration for consumption of the license as set on Marketplace products
CONSUMPTION_CONFIG='{
  "RenewType": "None",
  "ProvisionalConfiguration": {
    "MaxTimeToLiveInMinutes": 60
  }
}'
```

```
# License's home Region
HOME_REGION=us-east-1

# License issuer's name
ISSUER=Self

# Run AWS CLI command to create a license
aws license-manager create-license \
  --license-name "${LICENSE_NAME}" \
  --product-name "${PRODUCT_NAME}" \
  --product-sku "${PRODUCT_ID}" \
  --issuer Name="${ISSUER}" \
  --beneficiary "${BENEFICIARY_ACCOUNT_ID}" \
  --validity 'Begin="`${VALIDITY_START}`",End="`${VALIDITY_END}`"' \
  --entitlements "${ENTITLEMENTS}" \
  --home-region "${HOME_REGION}" \
  --region "${HOME_REGION}" \
  --consumption-configuration "${CONSUMPTION_CONFIG}" \
  --client-token $(uuidgen)
```

2. AWS License Manager コンソールを使用してライセンスを付与します。詳細については、「License Manager ユーザーガイド」の「[使用権限の配布](#)」を参照してください。
3. ソフトウェアをデプロイしてテスト AWS アカウント する購入者アカウントとして機能する にサインインします。これは、ライセンスを作成して付与 AWS アカウント した AWS アカウントとは異なる必要があります。
4. AWS License Manager コンソールに移動して、付与されたライセンスを受け入れてアクティブ化します。詳細については、「AWS License Manager ユーザーガイド」の「[付与されたライセンスの管理](#)」を参照してください。
5. 環境で次のコマンドを実行します。

```
# The following example uses a key fingerprint that should match the test license
you created.
# When checking out an actual AWS Marketplace created license, use the following
fingerprint:
# aws:294406891311:AWS/Marketplace:issuer-fingerprint
aws license-manager checkout-license \
  --product-sku <REPLACE-WITH-PRODUCT-ID> \
  --checkout-type PROVISIONAL \
  --key-fingerprint "aws:<ACCOUNT-ID-WHERE-YOU-CREATED-TEST-LICENSE>:Self:issuer-
fingerprint" \
```

```
--entitlements "Name=ReadOnly,Value=1,Unit=Count" \  
--client-token $(uuidgen)
```

前のコマンドでは、PROVISIONAL を CheckoutType パラメータの値として使用しています。使用権限でダウンロードライセンスを使用している場合は、値には PERPETUAL を使用します。

License Manager API コール

顧客の License Manager アカウントに保存されているライセンスを管理するために、ソフトウェアは次の API コールを使用できます。

- GetLicense - ライセンスの有効期限が切れているか、もうすぐ期限切れになるかについて、購入したライセンスのステータスを取得して、顧客に通知を送信します。
- CheckoutLicense - ユーザーが購入したライセンスを検出します。ユーザーがいくらかのライセンスを消費した時に、API コールを使用してライセンス数量を更新することもできます。CheckoutLicense を使用すると、顧客が使用したライセンスの数量を継続的に確認できます。顧客がすべてのライセンスを使い果たすと、この呼び出しはエラーを返します。CheckoutLicense を実行するための推奨ケイデンスについては、「[the section called “ライセンスの更新とアップグレード”](#)」を参照してください。
- ExtendLicenseConsumption - フローティングディメンションの場合、ソフトウェアがライセンスをチェックアウトすると、60 分後に自動的にライセンスがプールに返却されます。ライセンスがチェックアウトされたままの期間を延長する場合は、ソフトウェアがライセンスをさらに 60 分間延長するように ExtendLicenseConsumption を呼び出すことができます。
- CheckInLicense - フローティングディメンションの場合、ライセンスを使用権限プールに戻す場合は、CheckInLicense を使用してください。
- ListReceivedLicenses - 購入者が購入したライセンスを一覧表示します。

ライセンスの更新とアップグレード

顧客は AWS Marketplace 管理ポータルでライセンスを更新またはアップグレードできます。追加購入を行うと、新しい使用権限を反映する新しいバージョンのライセンス AWS Marketplace を生成します。ソフトウェアは同じ API コールを使用して新しい使用権限を読み取ります。License Manager の統合に関しては、更新やアップグレードを処理するために特別なことをする必要はありません。

ライセンスの更新、アップグレード、キャンセルなどが発生するため、製品の使用中は定期的に CheckoutLicense API コールを実行することをお勧めします。CheckoutLicense API オペレー

ションを定期的に使用して、製品はアップグレードや有効期限などの使用権限の変更を検出できます。

CheckoutLicense API コールは 15 分ごとに実行することをお勧めします。

での AMI 製品の Amazon SNS 通知の受信 AWS Marketplace

製品のカスタマーサブスクリプションの変更に関する通知を受け取るには、製品の作成時に AWS Marketplace 提供される の Amazon Simple Notification Service (Amazon SNS) トピックをサブスクライブできます。例えば、顧客がプライベートオファーを承認するタイミングがわかります。[Amazon SNS トピック: aws-mp-subscription-notification](#) は、AMI 製品で利用できる Amazon SNS トピックです。このトピックは、購入者が製品をサブスクライブまたはサブスクライブを解除した時に通知します。この通知は、時間単位、時間単位、年間料金など、時間単位の料金モデルで利用できます。詳細については、次のセクションを参照してください。

Note

製品作成プロセス中に、製品の Amazon SNS トピックが作成されます。通知をサブスクライブするには、Amazon SNS トピックの Amazon リソースネーム (ARN) が必要です (例: `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`)。ARN は、サーバー製品の販売者ポータルでは使用できません。[AWS Marketplace オペレーションチーム](#) に連絡して ARN をリクエストします。

トピック

- [Amazon SNS トピック: aws-mp-subscription-notification](#)
- [Amazon SNS トピックへ Amazon SQS キューをサブスクライブする](#)

Amazon SNS トピック: aws-mp-subscription-notification

aws-mp-subscription-notification トピック内の subscribe-success および subscribe-fail アクションのメッセージは、それぞれ次の形式になっています。

```
{
  "action": "action-name",
  "customer-identifier": " X01EXAMPLEX",
  "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
  "offer-identifier": "offer-abcexample123"
}
```

```
}
```

`<action-name>` は、通知によって異なります。指定できるアクションは以下のとおりです。

- subscribe-success
- subscribe-fail
- unsubscribe-pending
- unsubscribe-success

アクション `offer-identifier` が `subscribe-success` または の場合にのみ、 が通知に含まれます `subscribe-fail`。アクションが `unsubscribe-pending` または の場合、通知には含まれません `unsubscribe-success`。2024 年 1 月より前に作成されたオファーの場合、この識別子はプライベートオファーの通知にのみ含まれます。2024 年 1 月以降に作成されたオファーの場合、この識別子はプライベートオファーとパブリックオファーの両方を含むすべてのオファーの通知に含まれます。

オファータイプの詳細については、[DescribeEntity API](#) からのレスポンスまたは契約 [更新ダッシュボードの契約](#) のオファーの可視性を参照してください。

Note

[DescribeEntity API](#) の場合、そのオファーのターゲットルールのファセットをターゲットとする がアカウント AWS アカウント で見つかった場合、これはプライベートオファーです。そのオファーのターゲットルールのファセットをターゲットとする が AWS アカウント アカウント にない場合は、パブリックオファーです。

Amazon SNS トピックへ Amazon SQS キューをサブスクライブする

Amazon SQS キューを指定された SNS トピックにサブスクライブすることを推奨します。SQS キューを作成し、そのキューをトピックにサブスクライブする方法の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[Amazon SNS トピックへ Amazon SQS キューをサブスクライブする](#)」を参照してください。

Note

製品を販売する AWS アカウント ために使用される AWS Marketplace からのみ SNS トピックをサブスクライブできます。ただし、メッセージを別のアカウントに転送することはでき

ます。詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[別のアカウントの Amazon SQS キューへ Amazon SNS メッセージを送信する](#)」を参照してください。

通知の SQS キューのポーリング

SQS キューを SNS トピックにサブスクライブすると、メッセージは SQS に保存されます。継続的にキューをポーリングし、メッセージを探して必要に応じて処理するサービスを定義する必要があります。

の AMI 製品チェックリスト AWS Marketplace

Amazon マシンイメージ (AMI) 製品を に送信する前に AWS、このチェックリストを使用して送信を検証します。これらの要件に従うことで、効率的な公開プロセスを実現できます。

製品の使用

- AMI は本番稼働対応です。
- 製品の使用量は、時間やその他の測定値によって制限されません。
- AMI は、ワンクリックフルフィルメントエクスペリエンスと互換性があります。
- 製品を使用するために必要なものはすべて、クライアントアプリケーションを含め、ソフトウェアに含まれます。ソフトウェアパッケージやクライアントアプリケーションなど、外部依存関係を必要とする製品は、適切な開示[製品使用ポリシー](#)を含む に従います。
- AMI は必須の を満たしています [の AMI ベースの製品要件 AWS Marketplace](#)。
- 製品を使用するために追加のライセンスは必要ありません。
- 購入者は、製品を使用するために個人を特定できる情報 (E メールアドレスなど) を提供する必要はありません。

AMI の準備

- 製品名と説明は、提供する AMI 製品の説明フィールドと一致する必要があります。
- ハードウェア仮想マシン (HVM) の仮想化と 64 ビットアーキテクチャを使用します。
- 製品には、既知の脆弱性、マルウェア、またはウイルスは含まれていません。
- お客様はネットワーク経由でインスタンスにアクセスし、管理アクセスを使用できます。

- AMI にはハードコードされたシークレットはありません。ハードコードされたシークレットの例には、システムユーザーとサービスのデフォルトパスワード、プライベートキー、認証情報などがあります。
- AMI にはハードコードされた SSH 認可のパブリックキーはありません。
- テスト「バージョンの追加」検証テストは問題なく正常に完了します。

Windows AMI

- Windows Server 2012 以降のオペレーティングシステムでは、AMI は最新バージョンの [EC2Launch v2](#) を使用します。
- EC2Launch v2 を使用している場合は、次の手順を実行します。
 - [Amazon EC2Launch 設定](#)で、「管理者アカウントを設定する」の「ランダム」を選択して、実行時に管理者パスワードを生成します。
 - [Amazon EC2Launch 設定](#)で、Sysprep の後に SSM サービスを再有効化して開始するを選択します。
 - [EC2 v2 タスク構成](#)に [UserData] を追加します。
- Windows Server 2012 以降のバージョンでは、[EC2Config](#) を使用しないでください。EC2Config が必要な場合は、必ず最新バージョンを使用してください。
- EC2Config を使用する場合は、AMI [の設定ファイル](#)で次のパラメータを有効にします。
 - Ec2SetPassword
 - Ec2WindowsActivate
 - Ec2HandleUserData
- ゲストアカウントまたはリモートデスクトップのユーザーが存在しないことを確認します。

Linux AMI

- スーパーユーザーとしてのリモートログインは禁止されています。
- パスワードベースのリモートアクセスは禁止されています。

製品ロードフォームまたは製品タブ

- すべての必須フィールドが入力されています。
- すべての値が指定された文字制限内にあります。

- すべての URL がエラーなしで読み込まれます。
- 製品イメージは、少なくとも 110 ピクセル幅で、1:1 ~ 2:1 の比率です。
- 有効なすべてのインスタンスタイプで料金が指定されています (料金モデルが、時間単位、時間ベースの月額、および時間ベースの年額の場合)。
- 月別料金が指定されています (時間ベースの月額および月額モデルの場合)。

の AMI ベースの製品要件 AWS Marketplace

AWS Marketplace は、すべての Amazon マシンイメージ (AMI) 製品およびサービスについて以下のポリシーを維持します。このセクションのポリシーは、お客様に安全で安全で信頼できるコンピューティングプラットフォームを提供することを目的としています。

すべての製品とその関連メタデータは、送信時にレビューされ、現在の AWS Marketplace ポリシーを満たしているか超えているかが確認されます。これらのポリシーは、進化するセキュリティガイドラインに合わせて定期的に更新されます。AWS Marketplace は製品を継続的にスキャンして、既存の出品がこれらの要件に対する変更を引き続き満たしていることを確認します。製品がコンプライアンス違反になった場合、AWS Marketplace は販売者に連絡して、新しい基準を満たすように製品を更新します。場合によっては、問題が解決されるまで、新しいサブスクライバーが製品を一時的に使用できなくなることがあります。このプロセスは、すべてのユーザーの AWS Marketplace プラットフォームのセキュリティと信頼性を維持するのに役立ちます。

製品を送信する前に、の「[バージョンの追加](#)」テスト機能を使用して AWS Marketplace 管理ポータル、現在のポリシーに準拠していることを強くお勧めします。

トピック

- [AMI 製品販売者ポリシー](#)
- [セキュリティポリシー](#)
- [アーキテクチャポリシー](#)
- [AMI 製品使用説明書](#)
- [AMI 製品バージョンポリシー](#)
- [カスタマー情報ポリシー](#)
- [製品使用ポリシー](#)

AMI 製品販売者ポリシー

すべての AMIs、次の販売者ポリシーに従う必要があります。

- デフォルトでは、AWS Marketplace 販売者は最大 75 のパブリック AMI 製品リストに制限されています。制限を超えるすべての販売者は、定期的なパフォーマンスレビューの対象となり、パフォーマンスの低い出品を制限する必要がある場合があります。は、独自の裁量で、この制限の引き上げを許可および取り消す AWS Marketplace ことができます。

セキュリティポリシー

一般的なポリシー

すべての AMIs、次のポリシーに準拠する必要があります。

- AMIs [AWS Marketplace AMI スキャンツール](#)によって実行されたすべてのセキュリティチェックに合格し、既知の脆弱性やマルウェアがないことを示す必要があります。
- AMIs 現在サポートされているオペレーティングシステムとソフトウェアを使用する必要があります。有効期限に達したオペレーティングシステムやソフトウェアは許可されません。
- インスタンスサービスのパスワードベースの認証は禁止されています。これは、起動時にユーザーがパスワードを生成、リセット、または定義した場合でも適用されます。null および空白のパスワードは使用できません。

例外:

- Windows インスタンス EC2Config/EC2Launch で によって生成された管理者パスワード。
- 他の認証方法がない場合のホストサービス (ウェブアプリケーションなど) への管理者以外のアクセス。強力なパスワードを使用する場合は、インスタンスごとにランダムに生成し、サービス管理者が最初の認証に 1 回使用し、最初のログインの直後に変更する必要があります。
- AMI には、システムユーザーやサービスパスワード (ハッシュパスワードを含む)、プライベートキー、認証情報などのハードコードされたシークレットを含めることはできません。
- AMIs は、AWS サービスにアクセスするための AWS 認証情報をリクエストしないでください。製品から AWS サービスにアクセスする必要がある場合、インスタンスには最小限の特権 AWS Identity and Access Management (IAM) ロールを割り当てる必要があります。ユーザーは、手動で、または AWS CloudFormation テンプレートを使用してロールを作成できます。CloudFormation 配信方法を使用する製品でシングル AMI 起動が有効になっている場合、使用手順には、最小限の特権を持つ IAM ロールを作成するための明確なガイダンスを含める必要があ

ります。詳細については、[AWS CloudFormation を使用した AMI ベースの製品の配信](#)」を参照してください。

- 販売者は、顧客が実行するインスタンスにアクセスすることはできません。このようなアクセスがサポートやその他の目的で必要な場合は、明示的に有効にするように顧客に指示できます。

SSH (Secure Shell) アクセスポリシー

[一般的なポリシー](#)に加えて、SSH (Secure Shell) アクセスを提供する AMIs、次のセキュリティポリシーに準拠する必要があります。

- AMIsは、SSH を使用したパスワードベースの認証を許可してはいけません。これを確実にするには、sshd_config ファイルで PasswordAuthenticationを に設定しますno。
- AMIsスーパーユーザーアカウントのパスワードベースのリモートログインを無効にする必要があります。詳細については、[「ルートユーザーのパスワードベースのリモートログインを無効にする」](#)を参照してください。
- AMIsSSH アクセス用に承認されたパブリックキーを含めることはできません。
- AMIs の SSH は、AWS Marketplace 内部審査手順にアクセスできる必要があります。
 - SSH サービスは、AMI スキャン用に指定された TCP ポートをリッスンする必要があります。詳細については、[「新しいバージョンの追加」](#)を参照してください。
 - SSH は、インスタンスの起動時に Amazon Elastic Compute Cloud (Amazon EC2) によって割り当てられたサブネット10.0.0.0/16と 10.2.0.0/16 IP アドレスからアクセスできる必要があります。

Linux およびその他の Unix に似たオペレーティングシステムに基づく AMIs のポリシー

[一般的なポリシー](#)に加えて、Linux やその他の Unix に似たオペレーティングシステムに基づく AMIs は、次のセキュリティポリシーに準拠する必要があります。

- AMIs、ユーザーにフル特権アクセス (sudoアクセスを許可するなど) を許可する必要があります。

Windows ベースの AMIsポリシー

[一般的なポリシー](#)に加えて、Windows ベースの AMIsセキュリティポリシーに準拠する必要があります。

- AMIsがゲストアカウントを含めることはできません。
- インスタンスへのリモートデスクトップアクセスは、管理者アカウントにのみ付与できます。
- Windows AMIs[EC2Launch](#) (または Windows 2016 以前の EC[EC2Config](#)) でこれらのオプションを有効にして管理者パスワードを生成する必要があります。
 - Ec2SetPassword
 - Ec2WindowsActivate
 - Ec2HandleUserData
- AMIs は自動審査に使用できる必要があります。次のいずれかの要件を実装する必要があります。
 - (推奨オプション) SSM エージェントがインストールされ、管理アクセス許可とアウトバウンドネットワークアクセスがあります。
 - [Windows リモート管理 \(WinRM\) サービス](#)は有効になっており、TCP ポート をリッスンし5985、インスタンスの起動時に Amazon Elastic Compute Cloud (Amazon EC2) によって割り当てられた IP アドレス10.0.0.0/16とサブネットからアクセスできます。
 - [Microsoft Server Message Block \(SMB\) Protocol および Common Internet File System \(CIFS\) Protocol サービス](#)は、インスタンスの起動時に Amazon Elastic Compute Cloud (Amazon EC2) によって割り当てられたサブネット10.0.0.0/16および 10.2.0.0/16 IP アドレスから有効、リッスン445、139およびアクセス可能です。

アーキテクチャポリシー

すべての AMI は、次のアーキテクチャポリシーを順守する必要があります。

- のソース AMIs は、米国東部 (バージニア北部) リージョンで提供 AWS Marketplace する必要があります。
- AMI は、HVM 仮想化を使用する必要があります。
- AMIs x86-64 または 64 ビット ARM アーキテクチャを使用する必要があります。
- AMI は、Amazon Elastic Block Store (Amazon EBS) によってサポートされる AMI である必要があります。Amazon Simple Storage Service がサポートする AMIs はサポートされていません。
- AMI は暗号化された EBS スナップショットを使用してはいけません。
- AMI は暗号化されたファイルシステムを使用してはいけません。
- AMIs は、すべての で実行でき AWS リージョン、リージョンに依存しないように構築する必要があります。リージョンごとに異なる構築をした AMI は許可されません。

AMI 製品使用説明書

AMI 製品の使用説明書を作成する際は、[the section called “AMI とコンテナ製品の使用手順”](#)に記載されている手順とガイダンスに従ってください。

AMI 製品バージョンポリシー

AWS Marketplace は、S-AMI、CloudFormation テンプレートを使用した AMI、およびコンテナ製品を使用して、AWS 顧客と販売者のバージョン管理エクスペリエンスを自動化します。自動バージョンアーカイブでは、販売者によって 2 年以上制限されているすべての製品バージョンが自動的にアーカイブされます。アーカイブされたバージョンは、新しいお客様 AWS Marketplace に対してから起動できなくなりましたが、既存のユーザーは AMI ID を指定することで、起動テンプレートと Amazon EC2 Auto Scaling グループを通じてアーカイブされたバージョンを引き続き使用できます。過去 13 か月間に新しいインスタンスの起動に使用されていないアーカイブバージョンは削除されます。アーカイブされたバージョンが削除されると、新規または既存のユーザーに対して起動できなくなります。

カスタマー情報ポリシー

すべての AMI は、次のお客様情報ポリシーに準拠する必要があります。

- ソフトウェアは、BYOL (Bring Your Own License) で要求される場合を除き、お客様の知識と明示的な同意なしにお客様のデータを収集、またはエクスポートしてはなりません。顧客データを収集またはエクスポートするアプリケーションは、以下のガイドラインに従う必要があります。
- 顧客データの収集は、セルフサービスで、自動化され、安全である必要があります。購入者は、販売者がソフトウェアの導入を承認するのを待つ必要はありません。
- 顧客データの収集は、[AWS Marketplace の利用規約](#) AWS、[AWS サービス条件](#)、[AWS プライバシー通知](#)、[AWS カスタマーアグリーメント](#)など、との契約と一致する必要があります。
- 支払情報は収集してはなりません。

製品使用ポリシー

すべての AMI は、次の製品使用ポリシーを遵守する必要があります。

- 製品は、製品または製品機能へのアクセスを時間、ユーザーの数または他の制限事項によって制限してはなりません。ベータ版およびプレリリース版の製品、あるいはトライアルまたは評価機能を提供することのみを目的とした製品はサポートされていません。同等の有料バージョンも AWS

Marketplaceで入手可能である場合、商用ソフトウェアの開発者、コミュニティ、および BYOL エディションがサポートされています。

- すべての AMI は、ウェブサイトからの起動または AWS CloudFormation を介した AMI ベースの配信のいずれかと互換性がある必要があります。ウェブサイトから起動する場合、AMI はインスタンス作成時にお客様データまたはユーザーのデータを正しく機能させることを要求できません。
- AMI とそのソフトウェアはセルフサービス方式でデプロイ可能である必要があります。追加の支払い方法や費用を必要としないものでなければなりません。デプロイ時に外部に依存する必要があるアプリケーションは、以下のガイドラインに従う必要があります。
- 要件は、リストの説明または使用説明書に明記する必要があります。例えば、この製品を正しくデプロイするにはインターネット接続が必要です。デプロイ時に以下のパッケージがダウンロードされます。<パッケージのリスト>。
- 販売者は、すべての外部依存関係を使用し、その可用性とセキュリティを確保する責任を負います。
- 外部依存関係が使用できなくなった場合は、製品 AWS Marketplace も から削除する必要があります。
- 外部依存関係によって追加の支払い方法や費用が必要になってはいけません。
- 購入者の直接管理下でない外部リソースへの継続的な接続を必要とする AMIs。例えば、外部 APIs 販売者またはサードパーティーが AWS のサービス 管理する AMI は、次のガイドラインに従う必要があります。
- 要件は、リストの説明または使用説明書に明記する必要があります。例えば、この製品には継続的なインターネット接続が必要です。正しく機能するには、以下の継続的な外部サービスが必要です。<リソースのリスト>。
- 販売者は、すべての外部リソースを使用し、その可用性とセキュリティを確保する責任を負います。
- 外部リソースが使用できなくなった場合は、製品 AWS Marketplace も から削除する必要があります。
- 外部リソースは追加の支払い方法や費用を必要とせず、接続の設定を自動化する必要があります。
- 製品ソフトウェアとメタデータには、AWS Marketplace では利用できない他のクラウドプラットフォーム、追加の製品、またはアップセルサービスにユーザーをリダイレクトする言葉を含めてはいけません。
- 製品が別の製品または別の ISV 製品のアドオンである場合、製品の説明には、それが他の製品の機能を拡張するものであり、これがないと製品の有用性が非常に限られることを明記する必要があります。例えば、この製品は <製品名> の機能を拡張するものであり、それがなければ、この製品

の実用性は非常に限られています。<製品名> は、このリストのすべての機能を利用するには、独自のライセンスが必要な場合がありますのでご注意ください。

の EC2 Image Builder コンポーネントベースの製品 AWS Marketplace

AWS Marketplace 販売者は、[Image Builder コンポーネント](#)を使用して製品を購入者に配信できます。Image Builder コンポーネントは、カスタマイズされた EC2 Image Builder コンポーネントを作成および維持するために Image Builder サービスで使用される基本的な構成要素です。コンポーネントは、イメージ作成プロセス中に実行する特定のタスクのセットを定義する、再利用可能なモジュラー構成です。

コンポーネントは、イメージ作成前にインスタンスをカスタマイズする（ビルドコンポーネント）か、作成されたイメージから起動されたインスタンスをテストする（テストコンポーネント）ために必要な一連の手順を定義します。Image Builder コンポーネントは、次の 3 つの項目で構成されます。

- コンポーネントドキュメント、宣言ファイル、YAML ファイル、または JSON ファイル。コンポーネントを使用してイメージを構築するときに実行するステップとアクションを指定します。
- コンポーネントの設定オプションを提供するコンポーネント属性。コンポーネントの特性、動作、互換性を定義するのに役立ちます。
- ソフトウェアパッケージ、コンポーネントが正しく機能するためにインストールする必要がある依存関係。

での Image Builder コンポーネントの公開 AWS Marketplace は、AWS Marketplace Catalog API を使用してサポートされています。Catalog API を使用してコンポーネントを発行する手順については、API AWS Marketplace リファレンスの[EC2 Image Builder コンポーネント製品の使用](#)」を参照してください。

での EC2 Image Builder 製品の変更リクエストの作成 AWS Marketplace

で製品またはバージョンを変更するには AWS Marketplace、を通じて変更リクエストを送信します AWS Marketplace 管理ポータル。変更リクエストはキューに追加され、リクエストのタイプによっては解決までに数分から数日かかる場合があります。リクエストのステータスは、AWS Marketplace 管理ポータルで確認できます。このトピックでは、セルフサービスエクスペリエンスを使用するオプションなど AWS Marketplace、で単一の製品の変更リクエストを作成するために使用できる手順について説明します。

以下の状況では、変更リクエストを作成できます。

- 進行中のステップを保存しましたが、セルフサービスエクスペリエンスを使用して単一の製品リストを作成している間、プロセス全体を完了しませんでした。残りのステップを完了するには、変更リクエストを作成します。
- [制限あり] または [パブリック] 状態の製品の製品情報を変更します。情報を更新するには、変更リクエストを作成します。EC2 Image Builder 製品にリクエストできる変更のタイプの詳細については、「」を参照してください[変更リクエストを作成する](#)。

Note

に加えて AWS Marketplace 管理ポータル、[AWS Marketplace Catalog API](#) を使用して変更リクエストを作成することもできます。

トピック

- [セルフサービスを使用して変更リクエストを作成します。](#)
- [変更リクエストを作成する](#)
- [変更リクエストのステータスを取得する](#)
- [追加リソース](#)

セルフサービスを使用して変更リクエストを作成します。

バージョンや製品情報を変更するには、AWS Marketplace 管理ポータルで変更リクエストを作成します。変更リクエストは、製品に変更を加える際に使用するセルフサービスリストの構成要素です。ステップから [保存して終了] を選択して手順を終了するか、更新の [送信] を選択するたびに、変更リクエストを送信することになります。リクエストは AWS Marketplace 管理ポータル [Request](#) タブで確認できます。

セルフサービスを使用して変更リクエストを作成するには

1. AWS Marketplace 管理ポータル で を開き<https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインしてから、[サーバー製品](#) ページに移動します。
2. [サーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンからオプションを選択します。

4. 変更リクエストを行った後、システムがリクエストを処理するまで待ち時間があり、[レビュー中] と表示されます。リクエストが完了すると、[成功] または [失敗] と表示されます。
- リクエストが送信されると、[レビュー中]、[変更を準備中]、[変更を適用中] のステータスで処理が開始されます。
 - [成功] とは、リクエストした変更が処理され、変更がシステムに反映されたことを意味します。
 - [失敗] とは、リクエストに何か問題が生じたため、変更が処理されなかったことを意味します。ステータスが、[失敗] の場合は、リクエストを選択して、エラーの修正方法に関する推奨事項を示すエラーコードを検索できます。この時点で、エラーのトラブルシューティングを行い、変更の新しいリクエストを作成できます。処理を速めるには、[新しいリクエストにコピー] を選択して失敗したリクエストの詳細をコピーできます。その後、調整を行ってリクエストを再送信できます。

変更リクエストを作成する

Important

2023 年 6 月 15 日、AWS Marketplace は以下の手順を中止します。2023 年 6 月 15 日以降は、[the section called “セルフサービスを使用して変更リクエストを作成します。”](#) の手順を使用してください。

バージョンや製品情報を変更するには、AWS Marketplace 管理ポータルで変更リクエストを作成します。

変更リクエストを作成するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインしてから、[サーバー製品](#) ページに移動します。
2. [サーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンリストからオプションを選択します。

ほとんどの変更リクエストについては、ユーザーインターフェイスでフォームに入力して送信します。

変更リクエストのステータスの詳細については、「[the section called “変更リクエストのステータスを取得する”](#)」を参照してください。変更リクエストで発生する可能性のある問題については、「[AWS Marketplace の変更リクエストの一般的なエラーのトラブルシューティング](#)」を参照してください。

変更リクエストのステータスを取得する

Important

2023 年 6 月 15 日、AWS Marketplace は以下の手順を中止します。この手順は、セルフサービスエクスペリエンスでは不要になります。

変更リクエストを送信すると、AWS Marketplace 管理ポータル内の [\[サーバー製品\]](#) ページの [リクエスト] タブでリクエストのステータスを確認できます。ステータスは、次のいずれかになります。

- [レビュー中] とは、リクエストがレビュー中であることを意味します。一部のリクエストでは、AWS Marketplace チームによる手動レビューが必要ですが、ほとんどのリクエストはシステムで自動的にレビューされます。
- [成功] とは、リクエストが完了したことを意味します。製品またはバージョンがリクエストどおりに更新されました。
- [アクションが必要] とは、問題を解決したり、リクエストに関する質問に回答したりするために、リクエストを更新する必要があるということを意味します。リクエストを選択すると、問題を含む詳細が表示されます。
- [失敗] とは、リクエストに何か問題があったことを意味し、同じデータを使用して新しい変更リクエストを作成する必要があります。

追加リソース

特定のタイプの更新の変更リクエストの詳細については、以下のリソースを参照してください。

- [での EC2 Image Builder 製品情報の更新 AWS Marketplace](#)

EC2 Image Builder 製品の可視性の更新

AWS Marketplace 販売者は、製品の可視性を更新して、どの購入者が製品を表示できるかを変更できます AWS Marketplace。可視性ステータスが Public に設定されている場合、製品はすべての

AWS Marketplace 購入者に表示されます。製品の可視性が Limited に設定されている場合、製品は許可リストにある AWS アカウント IDs にのみ表示されます。製品を表示できる AWS アカウント IDs のこの許可リストを管理および更新することもできます。以下のセクションでは、製品の可視性と [限定] 可視性の許可リストを更新する方法を示します。

トピック

- [製品の可視性を更新する](#)
- [許可リスト \(プレビューアカウント\) を更新する](#)

製品の可視性を更新する

どの購入者が製品を表示できるかを変更するには AWS Marketplace、Update visibility を使用できます。

可視性を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [可視性を更新] を選択します。

Note

この変更リクエストを使用して、製品を [制限あり] ステータスから [パブリック] ステータスに移行するようリクエストできます。ただし、変更リクエストを公開に移行するには、AWS Marketplace Seller Operations チームの承認プロセスを実行する必要があります。

4. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
5. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

許可リスト (プレビューアカウント) を更新する

限定状態の製品を表示できる AWS アカウント IDs のリストを変更するには、Update allowlist を使用します。

許可リストを更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [許可リストを更新] を選択します。現在のリストには、現在、許可リストに登録されているアカウントのリストが表示されます。
4. 可視性のために優先される AWS アカウント IDs を追加し、IDs をカンマで区切ります。
5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

の Amazon EC2 インスタンスの追加と制限 AWS Marketplace

AWS Marketplace 販売者は、購入者が使用できるインスタンスを管理できます。購入者が使用できる新しいインスタンスを製品に追加できます。同様に、新しい購入者が特定のインスタンスから製品を使用することを防ぐ場合は、インスタンスを制限できます。

Amazon EC2 インスタンスタイプの詳細については、Amazon EC2 ユーザーガイド」の「[使用可能なインスタンスタイプ](#)」を参照してください。

以下のセクションでは、インスタンスを追加および制限する方法について説明します。

トピック

- [インスタンスの追加](#)
- [インスタンスの制限](#)

インスタンスの追加

購入者が EC2 Image Builder コンポーネントとして使用できる新しいインスタンスを追加できます。

インスタンスを追加するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。

3. [変更をリクエスト] ドロップダウンから [インスタンスを追加] を選択します。
4. インスタンスアーキテクチャを選択します。
5. 使用可能なインスタンスのリストから、追加するインスタンスタイプを選択します。
6. [リクエストを送信] を選択して、レビューのリクエストを送信します。
7. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

Note

- 現在の料金モデルが無料ではないか、Bring Your Own License (BYOL) モデルを使用している場合は、料金も追加する必要があります。
- 新しいインスタンスの料金でインスタンスを追加を作成した場合、または料金を更新して料金を引き上げる場合、変更を行った日から 90 日以内にセルフサービスを使用してインスタンスを追加することはできません。これらの変更を行うには、[AWS Marketplace Seller Operations チーム](#)に連絡してください。
- 新しいインスタンスタイプのサポートを追加すると、既に製品のプライベートオファアをサブスクライブしている顧客は、新しく追加されたインスタンスを自動的に起動できなくなります。顧客にアクセスさせたいインスタンスを使用して、別のプライベートオファアを作成する必要があります。新しいオファアを承諾すると、お客様は新しく追加されたインスタンスを起動できます。将来の日付に製品をサブスクライブするお客様は、インスタンスがプライベートオファアに含まれている限り、そのインスタンスを起動することもできます。新しいプライベートオファアを作成する方法の詳細については、このガイドの[での契約の修正 AWS Marketplace](#)後半の「」を参照してください。

インスタンスの制限

新しい購入者がインスタンスを使用できないようにするには、インスタンスを制限できます。インスタンスは、必要に応じて後で追加し直すことができます。制限されたインスタンス上の EC2 Image Builder コンポーネントの既存のユーザーは、サブスクリプションの期間中、リージョンの製品を引き続き使用できます。

インスタンスを制限するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [インスタンスの制限] を選択します。
4. 制限するインスタンスを選択し、[制限] を選択します。
5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

Note

チェックボックスがグレー表示されている場合は、そのインスタンスが推奨インスタンスタイプとして 1 つまたは複数のバージョンに関連付けられていることを意味します。このようなインスタンスを制限するには、[バージョンの更新] を使用して別の推奨インスタンスタイプを選択してください。変更リクエストが完了し、制限するインスタンスが推奨インスタンスタイプではなくなったら、[インスタンスの制限] に戻って選択したインスタンスを制限できます。

での EC2 Image Builder 製品のバージョン管理 AWS Marketplace

EC2 Image Builder コンポーネントのバージョン管理は UI から実行できません。詳細については、次のカタログ API セクションを参照してください。

- [新しいバージョンを追加する](#)
- [既存のバージョンに関する情報を更新する](#)
- [バージョンを制限する](#)

での EC2 Image Builder 製品情報の更新 AWS Marketplace

単一の EC2 Image Builder コンポーネントを作成したら、そのコンポーネントに関連付けられている情報の一部を変更できます AWS Marketplace。例えば、新しいバージョンで製品の説明やハイライトが変更された場合は、新しいデータで製品情報を編集できます。製品のタイトル、SKU の説明、

カテゴリ、キーワードなど、他の製品情報を更新することもできます。詳細については、以下の手順を参照してください。

製品情報を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[サーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [製品情報を更新] を選択します。
4. 変更する必要がある次のフィールドのいずれかを更新します。

- [製品のタイトル]
- [SKU]
- 短い説明
- [詳しい説明]
- [製品ロゴイメージ URL]
- [ハイライト]
- [製品カテゴリ]
- キーワード
- [製品動画 URL]
- リソース
- [サポート情報]

 Note

ロゴフォーマットの詳細については、「[会社と製品のロゴ要件](#)」を参照してください。

5. [Submit] (送信) を選択します。
6. リクエストが [リクエスト] タブに [レビュー中] ステータスで表示されていることを確認します。リストにリクエストが表示されるようにするには、ページの更新が必要になる場合があります。

および国別の AWS リージョン EC2 Image Builder 製品の可用性の管理

で製品を作成するときは AWS Marketplace、利用可能な AWS リージョンを選択します。購入者が製品を購入できる国も選択します。これら 2 つのプロパティは似ていますが、同じではありません。例として、購入者が米国に居住し、米国から購入していても、製品を欧州 (フランクフルト) リージョンにインストールする場合があります。この購入者が製品を購入するには、国のリストに米国を、リージョンのリストに欧州 (フランクフルト) を含める必要があります。以下のセクションを使用して、リージョンおよび国ごとに製品の可用性を更新できます。

トピック

- [を追加する AWS リージョン](#)
- [の制限 AWS リージョン](#)
- [将来の AWS リージョンのサポートの更新](#)
- [国別の可用性の更新](#)

を追加する AWS リージョン

購入者が製品を使用できるリージョンを追加できます。

リージョンを追加するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [リージョンを追加] を選択します。
4. 使用可能なリージョンのリストから、追加するリージョンを選択します。
5. [リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

Note

新しい のサポートを追加すると AWS リージョン、既に製品のプライベートオファーをサブスクライブしているお客様は、新しく追加されたリージョンに自動的にアクセスできなく

なります。顧客がアクセスするリージョンを使用して、別のプライベートオファーを作成する必要があります。新しいオファーを承諾すると、顧客は新しく追加されたリージョンにアクセスできます。将来の日付に製品をサブスクライブする顧客は、リージョンがプライベートオファーに含まれている限り、そのリージョンにアクセスすることもできます。新しいプライベートオファーを作成する方法の詳細については、「[Private offer upgrades, renewals, and amendments](#)」を参照してください。

の制限 AWS リージョン

新しい購入者が特定の で製品を使用できないようにするには AWS リージョン、リージョンを制限できます。リージョンは後で追加し直すことができます。リージョン内の製品の既存のサブスクライバーは、サブスクライブしている限り、そのリージョンの製品を引き続き使用できます。

リージョンを制限するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [リージョンを制限] を選択します。
4. ドロップダウンメニューを選択すると、製品が現在販売されているリージョンのリストが表示されます。
5. 制限するリージョンを選択します。
6. 選択したリージョンはトークンとして表示されます。制限するリージョンのリストを確認し、制限しないリージョンには X を入力します。
7. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
8. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

リクエストが成功すると、既存のユーザーには、制限されるリージョンを通知する次の E メールメッセージが届きます。登録している間は製品を使い続けることができますが、サブスクライブをキャンセルすると再サブスクライブすることはできません。

Greetings from AWS Marketplace,

This message is a notification detailing a recent change for <ProductName>.

{{sellerName}} has opted to restrict the <ProductType> product in <Restricted Region(s)> beginning <DateOfChange>.

This impacts you in the following ways:

1. As long as you're subscribed to the product, you can continue using the software product in the restricted Region.
2. You can't begin new instances of the software product in the restricted Region.
3. You can continue using the software product in all available AWS Regions.

Regards,

The AWS Marketplace Team

AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com (<http://amazon.com/>) is a registered

trademark of Amazon.com, Inc. This message was produced and distributed by Amazon Web Services Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.

将来の AWS リージョンのサポートの更新

製品を新しく起動した にオンボードする場合は AWS リージョン、今後のリージョンの更新サポートを使用できます。

将来のリージョンのサポートを更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [将来のリージョンのサポートの更新] を選択します。
4. 将来のリージョンサポートを有効にして、AWS Marketplace が AWS リージョン ユーザーに代わって新しく起動された に製品をオンボードできるようにすることができます。
5. この機能を有効にすると、将来リリースされるすべてのリージョンを選択するか、米国リージョンのみに限定するかを選択できます。
6. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
7. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

国別の可用性の更新

製品をサブスクライブして提供できる国を変更する場合は、[可用性の更新] を使用してください。

国別の可用性を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [可用性の更新] を選択します。
4. 以下のオプションのいずれかを選択してください：
 1. [すべての国] - サポートされているすべての国で利用できます。
 2. [除外対象のすべての国] - 選択した国を除くすべてのサポート対象国で利用できます。
 3. [カスタムリスト] - 製品が販売されている特定の国のリスト。
5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

EC2 Image Builder AWS Marketplace 製品のエンドユーザーライセンス契約 (EULA) の更新

AWS Marketplace 販売者は、EC2 Image Builder コンポーネントの使用を管理するエンドユーザーライセンス契約 (EULA) を更新できます。EULA は、AWS Marketplace に公開されているソフトウェアリストの製品リストページに記載されています。独自の EULA を適用することも、[AWS Marketplace の標準契約書 \(SCMP\)](#) を使用することもできます。次の手順は、EC2 Image Builder コンポーネントの EULA を更新する方法を示しています。

EULA の詳細については、「[での標準化された契約の使用 AWS Marketplace](#)」を参照してください。

EULA を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) タブを選択し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから、[エンドユーザー使用許諾契約の更新] を選択します。

4. [AWS Marketplace \(SCMP\) の標準契約](#)を選択するか、独自のカスタム EULA を送信できます。カスタム EULA の場合は、Amazon S3 バケットからのカスタム契約を提供する必要があります。

 Note

Amazon S3 バケットでパブリックアクセシビリティを有効にする必要があります。

5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

での EC2 Image Builder 製品の返金ポリシーの更新 AWS Marketplace

AWS Marketplace 販売者は、EC2 Image Builder コンポーネントの返金ポリシーを設定できます。製品の返金ポリシーを変更する場合は、[返金ポリシーを更新] を使用できます。以下の手順では、返金ポリシーを更新する方法を示します。

返金ポリシーを更新するには

1. AWS Marketplace 管理ポータル でを開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページに移動し、[現在のサーバー製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [返金ポリシーを更新] を選択します。
4. 現在の返金ポリシーの詳細がテキストボックスに表示され、編集できます。リクエストを送信すると、現在の返金ポリシーが上書きされます。
5. [変更リクエストを送信] を選択して、レビューのリクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

AWS Marketplaceからの製品の削除

製品を公開したら、 から削除できます AWS Marketplace。これは日没とも呼ばれます。製品を削除するには、製品を特定して削除する理由と連絡先の E メールアドレスと一緒に削除リクエストを送

信します。また、現在の製品を新しい製品と交換する場合は、代替製品 ID を提供することもできます。

製品の削除をリクエストすると、新しいお客様は製品をサブスクライブできません。既存のお客様は、少なくとも 90 日間サポートする必要があります。

 Note

AWS Marketplace 管理ポータルから制限された製品を削除することはできません。ポータルは、アカウントの発行履歴の一部としてそれらを保持します。

からの製品削除のリクエストは、以下の条件で処理 AWS Marketplace されます。

- 製品は、AWS Marketplace 検索、参照、その他の検出ツールから削除されます。[サブスクライブ] ボタンまたは機能は無効になっており、ページ上のメッセージは製品が使用できなくなったことを明確に示しています。製品の詳細ページは引き続き URL を使用してアクセスでき、公開されている検索エンジンでインデックスに登録されている可能性があります。
- 削除の理由を指定する必要があります (サポートの終了、製品の更新の終了、交換製品など)。削除された製品のサポートを継続するための要件については、[AWS Marketplace 「販売者の利用規約」](#)を参照してください。
- AWS Marketplace は、製品の削除、削除の理由、販売者の連絡先情報を知らせる E メールメッセージを通じて現在の購入者に連絡します。
- 現在の購入者がサブスクリプションをキャンセルするまでソフトウェアへのアクセスは維持されます。これらのお客様は、製品の削除による影響を受けません。

を使用して作成された製品を削除するには AWS Marketplace 管理ポータル

1. AWS Marketplace 管理ポータル でを開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [製品] タブを選択し、次に [サーバー] を選択します。
3. [現在のサーバー製品] の下にある製品ページで、削除する製品を探します。[変更のリクエスト] ドロップダウンリストで、[可視性の更新] を選択します。
4. [製品可視性の更新] ページで、[制限付き] を選択します。
5. (オプション) 削除する製品の代わりとなる別の製品がある場合は、[代替製品 ID] を入力します。

6. 情報が正しいことを確認して、[送信] を選択します。

製品の削除リクエストを送信した後、[What's next (最新情報)] 情報ページが表示されます。Seller AWS Marketplace Operations は、リクエストを確認して処理します。[リクエスト] を表示して、送信のステータスを確認します。

製品が削除されると、製品は AWS Marketplace 管理ポータルの [現在の製品] リストに表示されます。[Current Products (現在の製品)] で利用できる唯一のアクションは製品のスプレッドシートをダウンロードすることです。別のサンセットリクエストの編集または送信はできません。

製品の削除についてご不明な点がございましたら、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

のコンテナベースの製品 AWS Marketplace

AWS Marketplace は、Docker コンテナを使用するソフトウェア製品をサポートしています。コンテナ製品は、コンテナイメージとデプロイテンプレートのセットである配信オプションで構成されます。製品に少なくとも 1 つの配信オプション (最大 4 つまで) を送信します。各配信オプションについて、コンテナイメージ、使用手順、購入者がその配信オプションを起動するためのデプロイテンプレートへのリンクを提供します。このトピックでは、AWS Marketplace のコンテナベースの製品について説明します。

AWS Marketplace 購入者は、公開されている製品詳細ページで利用可能な配信オプションを確認できます。製品をサブスクライブして必要な配信方法を選択すると、購入者には製品の発売および使用に関する情報と手順が表示されます。コンテナイメージの配信オプションの場合、購入者には利用可能なデプロイテンプレートとコンテナイメージ URL へのリンクが表示されます。また、個々のコンテナイメージを取得する方法についての説明も受け取ります。Helm チャートの配信オプションについては、購入者には Helm を使用して起動する手順が順を追って表示されます。

購入エクスペリエンスのチュートリアルについては、[「Amazon ECS クラスターに AWS Marketplace コンテナをデプロイする \(3:34\)」](#)を参照してください。

任意の環境の任意の Kubernetes クラスター AWS Marketplace で、からサードパーティーの Kubernetes アプリケーションを検索、サブスクライブ、デプロイできます。Amazon EKS Anywhere (EKS Anywhere) を使用して Amazon EKS Anywhere) AWS Fargate、およびオンプレミスにサードパーティーの Kubernetes アプリケーションをデプロイできます。オンプレミスまたは Amazon Elastic Compute Cloud (Amazon EC2) の自己管理型 Kubernetes クラスターにデプロイすることもできます。

Docker 互換のランタイムで Free および BYOL モデル (自分のライセンスを使用する) コンテナ製品を実行できます。

トピック

- [ヘルプの利用](#)
- [コンテナ製品の開始方法](#)
- [のコンテナベースの製品要件 AWS Marketplace](#)
- [のコンテナ製品の料金 AWS Marketplace](#)
- [コンテナ製品の請求、計測、ライセンスの統合](#)
- [コンテナ製品の Amazon SNS 通知](#)

ヘルプの利用

コンテナ製品に関するヘルプについては、AWS Marketplace の事業開発パートナーに連絡するか、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

コンテナ製品の開始方法

AWS Marketplace 販売者は、コンテナベースのソフトウェア製品を作成できます。コンテナ製品は、コンテナイメージと一緒に扱うデプロイテンプレートのセットである配信オプションで構成されます。次のトピックでは、コンテナ製品の使用を開始する方法について説明します。

- [製品のライフサイクル](#)
- [前提条件](#)
- [ステップ 1: コンテナ製品の製品 ID と製品コードを作成する](#)
- [ステップ 2: 初期リストを作成する](#)
- [ステップ 3: 製品の初期バージョンを追加する](#)
- [ステップ 4: \(有料製品のみ\) 計測または契約料金を統合する](#)
- [次のステップ](#)
- [コンテナ製品はセキュリティ上の問題をスキャンします。](#)

製品のライフサイクル

で製品を作成すると AWS Marketplace、最初に可視性が制限されて公開され、製品を作成したアカウントを含め、許可リストのアカウントがその製品を表示できます。準備ができれば、AWS Marketplace カタログに公開して、購入者が製品をサブスクライブして購入できるようにします。

[サーバー製品](#) ページでは、製品のリストを表示できます。ステージに応じて、製品のステータスは、次のいずれかです。

- ステージング - 情報を追加中の、不完全な製品。セルフサービスエクスペリエンスの最初の保存して終了で、変更リクエストが成功すると、送信した完了したステップの情報を含む未公開の製品が作成されます。このステータスから、引き続き商品に情報を追加したり、変更リクエストを通じて送信済みの情報を変更したりできます。
- 制限あり - 製品がシステムに送信され、システム内のすべての検証に合格すると完成します。その後、製品は制限あり状態にリリースされます。この時点で、製品にはアカウントと許可リストに登録したユーザーだけがアクセスできる詳細ページがあります。商品は詳細ページからテストできま

す。詳細またはヘルプについては、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。

- パブリック - 購入者が商品を開覧してサブスクライブできるように商品を公開する準備ができたら、可視性を更新する変更リクエストを使用します。このリクエストは、AWS Marketplace Seller Operations チームが AWS ポリシーに照らして製品を確認および監査するワークフローを開始します。製品が承認され、変更リクエストが処理されると、製品のステータスが制限ありからパブリックに移行します。AWS ガイドラインの詳細については、「」を参照してくださいの[コンテナベースの製品要件 AWS Marketplace](#)。
- 制限付き - 新規ユーザーによる製品のサブスクライブを停止する場合は、可視性の更新変更リクエストを使用して製品を制限できます。制限付きステータスは、既存の許可リストに登録されているユーザーが引き続き製品を使用できることを意味します。ただし、製品は一般には公開されなくなり、新規ユーザーも利用できなくなります。

前提条件

開始するには、以下の前提条件を満たす必要があります。

1. [AWS Marketplace 管理ポータル](#) にアクセスして使用する。これは、販売者として登録し、販売する製品を管理するために使用するツールです AWS Marketplace。詳細については、「[AWS Marketplace 管理ポータル](#)」を参照してください。
2. 販売者として登録し、税金および銀行の情報を提出します。詳細については、「[AWS Marketplace 販売者としての登録](#)」を参照してください。
3. Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS)、または AWS Fargate に少なくとも 1 つのコンテナを作成します。関連するイメージへのリンクがあることを確認します。
4. AWS Marketplace でコンテナ製品を作成および統合する方法を計画します。

製品を公開する前に、料金、使用権限管理、および計測戦略を十分に計画することをお勧めします。

- コンテナベースの製品の要件については、「[コンテナベースの製品要件 AWS Marketplace](#)」を参照してください。
- 製品の料金設定の詳細については、「[コンテナ製品の料金 AWS Marketplace pricing](#)」を参照してください。
- 有料のコンテナベース製品のカスタム計測については、「[AWS Marketplace Metering Service による時間単位計測およびカスタム計測](#)」を参照してください。

概要: コンテナ製品を作成する

コンテナ製品の作成には、以下のステップを実行します。

1. [ステップ 1: コンテナ製品の製品 ID と製品コードを作成する](#)
2. [ステップ 2: 初期リストを作成する](#)
3. [ステップ 3: 製品の初期バージョンを追加する](#)
4. [ステップ 4: \(有料製品のみ\) 計測または契約料金を統合する](#)
5. [製品の可視性を更新する](#)

製品のライフサイクルの詳細については、「[製品のライフサイクル](#)」を参照してください。

ステップ 1: コンテナ製品の製品 ID と製品コードを作成する

コンテナ製品の使用を開始するには、で製品 ID と製品コードレコードを作成する必要があります AWS Marketplace。製品 ID は、製品のライフサイクル全体を通じて製品を追跡するために使用されます。

で新しいコンテナ製品を作成し、製品 ID を生成するには AWS Marketplace 管理ポータル、次の手順に従います。

Note

このプロセスでは、製品とペアリングするコンテナのパブリックキーも作成されます。

コンテナ製品 ID を作成するには

1. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
2. メニューバーから [製品] を選択し、[サーバー] を選択します。
3. [サーバー製品を作成] を選択し、[コンテナ] を選択します。
4. コンテナ製品 ID とコードを生成します。

Note

(オプション) 商品にタグを付けてタグベースの認可を受けることができます。詳細については、「[AWS リソースのタグ付け](#)」を参照してください。

5. [続行] を選択して製品の作成を続行します。

ステップ 2: 初期リストを作成する

製品 ID、製品コード、パブリックキーを生成したら、ウィザードを使用して初期リストを作成します。

1. 製品リストの製品情報を入力します。
2. 製品の料金モデルを決定します。

Note

コンテナ製品の詳細については、「[コンテナ製品の料金](#)」を参照してください。

Note

有料製品の場合、AWS Marketplace 出品者と Seller Operations チームが高額な費用をかけずに製品をテストできるように、製品の料金は 0.01 USD から開始します。実際の料金は、公開時に提示します。

3. 返金ポリシー、EULA、オファ어의提供状況など、その他のオファ어情報を提供してください。
4. コンテナ製品の初期リポジトリを追加します。
5. 最後のステップで [送信] を選択し、製品を限定公開に移行します。

Note

コンテナ製品は最初にプレースホルダーバージョンを使用して作成されます。最終バージョンは、製品の公開範囲が限定されている場合に追加します。

ステップ 3: 製品の初期バージョンを追加する

製品には、その使用期間中に複数のバージョンがある場合があります。各バージョンには、そのバージョン固有のコンテナイメージセットがあります。製品の初期バージョンを追加するには、「[でのコンテナ製品の新しいバージョンの追加 AWS Marketplace](#)」を参照してください。

ステップ 4: (有料製品のみ) 計測または契約料金を統合する

使用料金を使用したコンテナベースの製品の場合、製品を使用する使用権限の確認および請求用使用計測の両方に対して、[AWS Marketplace Metering Service](#) を使用します。料金情報を設定するときに作成した料金モデルに合わせて計測する必要があります。詳細については、「[AWS Marketplace Metering Serviceによる時間単位計測およびカスタム計測](#)」を参照してください。

契約料金

契約料金のコンテナベースの製品の場合は、AWS License Manager を使用してライセンスを製品に関連付けます。

との統合の詳細については AWS License Manager、「」を参照してください[を使用したコンテナ製品の契約料金 AWS License Manager](#)。

ステップ 5: 製品の可視性を更新する

で製品を作成すると AWS Marketplace、最初に可視性が制限されて公開され、製品を作成したアカウントを含め、許可リストのアカウントがその製品を表示できます。製品の可視性を更新して、購入者が製品をサブスクライブして購入できるようにします。または、追加する製品許可リストを更新することもできます AWS アカウント。このトピックでは、AWS Marketplaceで製品を表示できる購入者を管理する方法について説明します。

製品の可視性とライフサイクルの詳細については、「[製品のライフサイクル](#)」を参照してください。

トピック

- [製品の可視性を更新する](#)
- [AWS アカウント ID の許可リストを更新します。](#)

製品の可視性を更新する

可視性を更新するには

1. AWS Marketplace 管理ポータル で を開き <https://aws.amazon.com/marketplace/management/tour/>、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページの [現在のサーバー製品] タブで、変更するコンテナベースの製品を選択します。
3. [変更をリクエスト] ドロップダウンから [可視性を更新] を選択します。

Note

この変更リクエストを使用して、製品を [制限あり] ステータスから [パブリック] ステータスに移行するようリクエストできます。ただし、変更リクエストをパブリックに移行するには、AWS Marketplace Seller Operations チームの承認プロセスを経る必要があります。

4. [送信] を選択して、リクエストを送信します。
5. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

AWS アカウント ID の許可リストを更新します。

限定された状態で製品を表示できる AWS アカウント IDs のリストを変更できます。許可リストに登録されているアカウントには、製品詳細ページの製品バージョンの横に Limited バッジが表示されます。

1. <https://aws.amazon.com/marketplace/management/tour/> AWS Marketplace 管理ポータル で を開き、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページから、変更するコンテナ製品を選択します。
3. [変更をリクエスト] ドロップダウンリストから [許可リストを更新] を選択します。現在許可リストに登録されているアカウントのリストが表示されます。
4. [許可リストに登録された AWS アカウント] フィールドに AWS アカウント ID を入力し、カンマで区切ります。
5. [送信] を選択して、リクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] または [失敗] に更新されます。

次のステップ

コンテナ製品を作成したら、次のトピックの情報を使用してコンテナ製品を設定および管理できます。

- [AWS Marketplace でのコンテナ製品の製品情報の更新](#)
- [でのコンテナ製品の新しいバージョンの追加 AWS Marketplace](#)

- [AWS Marketplace でコンテナ製品価格を管理する](#)
- [AWS Marketplace の国別にコンテナ製品の在庫状況を更新する](#)
- [でのコンテナ製品のエンドユーザーライセンス契約 \(EULA\) の更新 AWS Marketplace](#)
- [AWS Marketplace でのコンテナ製品のテストとリリース](#)

コンテナ製品はセキュリティ上の問題をスキャンします。

コンテナ製品に新しいバージョンを追加する変更リクエストを作成すると、その新しいバージョンに含まれるコンテナイメージがスキャンされ、セキュリティの脆弱性がチェックされます。これを行うため、イメージに対してレイヤーごとの静的スキャンを実行します。リモートで悪用可能なリスクベクトルがある重大な脆弱性が見つかった場合は、見つかった問題のリストが表示されます。取り込みと公開プロセスの遅延を避けるには、Clair、Twistlock、Aqua Security、Trend Micro などのコンテナイメージスキャナーを使用して、独自のセキュリティ分析を実行することを強くお勧めします。

コンテナイメージを構築するためのベースイメージの選択は、最終的なイメージのセキュリティプロファイルに大きな影響を与えます。既知の重大な脆弱性があるベースイメージを選択すると、アプリケーションソフトウェアのレイヤーがクリーンであってもベースレイヤーが原因でフラグ付けされます。イメージを構築して送信する前に、脆弱性のないベースコンテナから開始していることを確認することをお勧めします AWS Marketplace。

AWS Marketplace でのコンテナ製品の製品情報の更新

製品 ID を作成して料金を設定したら、AWS Marketplace でコンテナ製品について顧客に表示される内容など、製品情報を編集できます。例えば、新しいバージョンで製品の説明やハイライトが変更された場合は、新しいデータで製品情報を編集できます。製品のタイトル、SKU の説明、カテゴリ、キーワードなど、他の製品情報を更新することもできます。以下の手順では、製品の製品詳細を作成する手順を概説しています。

コンテナ製品の製品詳細を作成または更新するには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [製品] メニューで [サーバー] を選択します。
3. [サーバー製品] タブで、変更する製品を選択します。
4. [変更をリクエスト] ドロップダウンから [製品情報を更新] を選択します。
5. 変更する次のフィールドのいずれかを更新します。

- [製品のタイトル]

- [SKU]
- 短い説明
- [詳しい説明]
- [製品ロゴイメージ URL]
- [ハイライト]
- [製品カテゴリ]
- キーワード
- [製品動画 URL]
- リソース
- [サポート情報]

 Note

イメージ URL は、パブリックにアクセス可能な Amazon S3 バケットに含まれている必要があります。ロゴフォーマットの詳細については、「[会社と製品のロゴ要件](#)」を参照してください。

6. [送信] を選択します。
7. リクエストが [リクエスト] タブに [レビュー中] ステータスで表示されていることを確認します。リストにリクエストが表示されるようにするには、ページの更新が必要になる場合があります。

リクエストのステータスは、[サーバー製品](#) ページの [リクエスト] タブからいつでも確認できます。

でのコンテナ製品の新しいバージョンの追加 AWS Marketplace

AWS Marketplace 販売者は、コンテナ製品の新しいバージョンを追加したり、バージョンを管理したり、バージョン情報を更新したりできます。製品には、その使用期間中に複数のバージョンがある場合があります。各バージョンには、そのバージョン固有のコンテナイメージセットがあります。以下のトピックでは、コンテナ製品の製品バージョンを管理する方法について説明します。

Note

製品 ID と製品の料金を設定するまで、製品にバージョンを追加できません。これらの手順の詳細については、「[ステップ 1: コンテナ製品の製品 ID と製品コードを作成する](#)」を参照してください。

トピック

- [ステップ 1: リポジトリを追加する](#)
- [ステップ 2: コンテナイメージとアーティファクトをリポジトリにアップロードする](#)
- [ステップ 3: コンテナ製品に新しいバージョンを追加する](#)
- [ステップ 4: バージョン情報を更新する](#)
- [Amazon EKS アドオンのバージョンを制限する](#)

ステップ 1: リポジトリを追加する

製品のコンテナイメージやその他のアーティファクトは、AWS Marketplaceのリポジトリに保存されます。通常、必要なアーティファクトごとに1つのリポジトリを作成しますが、リポジトリにはアーティファクトの複数のバージョンを (異なるタグで) 保存できます。

Note

製品デプロイ内のすべてのイメージは、AWS Marketplace リポジトリのイメージを使用する必要があります。

次の手順では、必要なリポジトリを に追加する方法について説明します AWS Marketplace。

リポジトリを追加するには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [製品] メニューで [サーバー] を選択します。
3. [サーバー製品] タブで、変更する製品を選択し、[変更をリクエスト] ドロップダウンから [リポジトリを追加] を選択します。
4. 作成するリポジトリの名前を入力します。複数の新しいリポジトリを作成する場合：

- 追加するリポジトリごとに新しいリポジトリを追加するを選択します。
- 一意の名前を付けます。選択した一意の名前は、販売者アカウントのすべての製品にまたがっている必要があります。

 Note

リポジトリは次のような構造になります。<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>。リポジトリにアイテムを追加すると (以下の手順で)、アイテムにはタグが付けられ、この構造になります。<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>:<tag>。repositoryID は の内部 ID です AWS Marketplace。sellerName は、販売者アカウント用に作成した名前に基づいています。このステップで repositoryName を定義します。tag は、アーティファクトをリポジトリにアップロードするときに設定されます。

5. [Submit] (送信) を選択します。

 Note

製品ごとに最大 50 個のリポジトリを持つことができます。

新しいリクエストが作成され、[リクエスト] タブに表示されます。完了すると、数分以内に、作成したリポジトリにコンテナイメージやその他のアーティファクトを追加できるようになります。

ステップ 2: コンテナイメージとアーティファクトをリポジトリにアップロードする

コンテナイメージとアーティファクトをリポジトリにアップロードするには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [製品] メニューの [サーバー] を選択します。
3. [サーバー製品] タブから、変更する製品を選択します。
4. [変更をリクエスト] ドロップダウンから [リポジトリを追加] を選択します。
5. [既存のリポジトリを表示] を選択します。
6. アップロードするリポジトリを選択します。

7. [プッシュコマンドを表示] を選択すると、Docker コンテナイメージと Helm チャートをそのリポジトリにプッシュするために使用できるコマンドを含む手順一覧が開きます。

コンテナイメージやその他のアーティファクトをリポジトリにプッシュする方法に関する一般的な情報については、「Amazon Elastic Container Registry ユーザーガイド」の「[イメージのプッシュ](#)」を参照してください。

Note

`docker pull` または `docker push` を呼び出す時に、次の Amazon Elastic Container Registry (Amazon ECR) API オペレーションを使用できます。

- `DescribeImages` - これを使用して、リポジトリ内のイメージに関するメタデータを確認します。
- `GetAuthorizationToken` - アーティファクトをリポジトリにアップロードする前に認証を行い、`docker pull` または `docker push` コマンドを使用してください。
- `ListImages` - プッシュしたイメージのリストを表示する場合に使用します。

8. リストされているコマンドを使用して、必要なアーティファクトをローカルリポジトリから製品の AWS Marketplace リポジトリにプッシュします。

Note

`push` コマンドで指定した **tag** は、リポジトリにアップロードするアーティファクトのバージョンを区別するために使用されます。アーティファクトが含まれるバージョンに適したタグを使用してください。

9. 使用しているバージョンで必要なコンテナイメージまたはアーティファクトごとに同じ手順を繰り返します。

Note

バージョンには、各配信オプションに最大 50 個のコンテナイメージまたはアーティファクトを含めることができます。配信オプションの詳細については、以下の手順を参照してください。

アーティファクトをアップロードしたら、製品のバージョンを作成する準備が整います。

Note

コンテナイメージは自動的にスキャンされ、[のコンテナベースの製品要件 AWS Marketplace](#)を満たしているかどうかを確認されます。詳細については、「[コンテナ製品はセキュリティ上の問題をスキャンします。](#)」を参照してください。

新しい配信オプションの追加

コンテナ製品の各バージョンには、配信オプションが必要です。配信オプションは、購入者が利用できるデプロイオプションを指定します。以下の配信オプションのいずれかに応じて、適切なアーティファクトをリポジトリにアップロードする必要があります。

- コンテナイメージ配信オプションでは、製品のインストールに必要なすべてのコンテナイメージを、AWS Marketplace コンソールで作成された Amazon Elastic Container Registry (Amazon ECR) リポジトリにアップロードします。
- Helm チャート配信オプションでは、AWS Marketplace コンソールで作成された Amazon ECR リポジトリにHelmチャートイメージとコンテナイメージをアップロードします。
- Amazon EKS コンソールのアドオン配信オプションでは、Helmチャートイメージとコンテナイメージを AWS Marketplace コンソールで作成された Amazon ECR リポジトリにアップロードします。

ステップ 3: コンテナ製品に新しいバージョンを追加する

Note

コンテナに新しいバージョンを追加する際にエラーが発生した場合は、「AWS Marketplace Catalog API リファレンス」の「[新しいバージョンの非同期エラーテーブルを追加する](#)」を参照してください。

コンテナ製品に新しいバージョンを追加するには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [製品] メニューの [サーバー] を選択します。
3. [サーバー製品] タブで、バージョンを追加する製品を選択します。次に、[変更をリクエスト] ドロップダウンから [新しいバージョンを追加] を選択します。

4. [新しいバージョンを追加] ページで、使用しているバージョンの [バージョンタイトル] と [リリースノート] を入力します。
5. バージョンの詳細を入力したら、次のステップとして、配信オプションを追加します。配信オプションとは、購入者が製品バージョンからソフトウェアを起動するために使用できる指示と情報のセットです。配信オプションは、購入者へのフルフィルメントオプションと呼ばれます。

 Note

製品は、コンテナイメージが異なる複数のプラットフォーム (Kubernetes および Ubuntu デプロイ) をサポートできます。顧客が製品をセットアップする方法ごとに 1 つの配信オプションを作成できます。1 つの製品に対して最大 4 つの配信オプションを作成できます。

- a. 製品に他のバージョンで既に配信オプションがある場合は、既存のオプションをテンプレートとして使用して、新しいバージョンに配信オプションを追加できます。[配信オプション] で、追加する配信オプションをリストから選択します。オプションは、以降のステップの指示に従って編集できます。
 - b. 新しい配信オプションを追加するには、[新しい配信オプション] を選択します。オプションを追加したら、以下のステップの指示に従って設定します。
6. 配信オプションの配信方法を選択します。配信方法によって、購入者がソフトウェアをどのように起動するかが決まります。
 - [コンテナイメージ] 配信オプションの場合、AWS Marketplace コンソールで作成された Amazon Elastic Container Registry (Amazon ECR) 内のコンテナイメージへのパスを指定します。購入者はコンテナイメージパスを使用してイメージを自分の環境に直接取り込み、ソフトウェアを起動します。
 - [Helm チャート] の配信オプションでは、AWS Marketplace コンソールで作成された Amazon ECR リポジトリ内の Helm チャートへのパスを指定します。購入者はデプロイ環境に Helm チャートをインストールしてソフトウェアを起動します。
 - Amazon EKS コンソールアドオン配信オプションの場合、AWS Marketplace コンソールで作成された Amazon ECR リポジトリの Helm チャートへのパスが指定されます。購入者は、Amazon EKS コンソールまたはネイティブの Amazon EKS アドオン API を使用して、コンテナをインストールし、ソフトウェアを起動します。詳細については、「[Amazon EKS で使用できる Amazon EKS アドオン](#)」を参照してください。

- a. [コンテナイメージ] 配信オプションを追加するには、以下のステップを実行します。
 - i. [コンテナイメージ] で、製品バージョンのソフトウェアを含むコンテナイメージに Amazon ECR URL を追加します。
 - ii. [配信オプションのタイトル] と [デプロイオプションの説明] に、この配信オプションのタイトルと説明を入力します。
 - iii. [使用説明書] に、購入者がソフトウェアを起動した後に使用するのに役立つ詳細情報を入力します。
 - iv. [サポート対象のサービス] で、購入者がソフトウェアを起動できる環境を選択します。
 - v. [デプロイテンプレート] に、購入者がソフトウェアの起動に使用できるリソースを追加します。各テンプレートのタイトルとリソースの URL を入力します。
- b. Helm チャート配信オプションを追加するには、次の手順を実行します。
 - i. Helm チャートで、購入者がデプロイ環境にインストールする Helm チャートに Amazon ECR URL を追加して、ソフトウェアを起動します。
 - ii. [コンテナイメージ] で、製品バージョンのソフトウェアを含むコンテナイメージに Amazon ECR URL を追加します。
 - iii. [配信オプションのタイトル] と [デプロイオプションの説明] に、この配信オプションのタイトルと説明を入力します。
 - iv. [使用説明書] に、購入者がソフトウェアを起動した後に使用するのに役立つ詳細情報を入力します。
 - v. [サポート対象のサービス] で、購入者がソフトウェアを起動できる環境を選択します。
 - vi. オプション - [Helm リリース名] に、Helm チャートをインストールする Kubernetes 名前空間の名前を入力します。
 - vii. オプション - [Helm インストール名前空間] に、helm install コマンドが使用する Helm リリースの名前を入力します。
 - viii. オプション - [Kubernetes サービスアカウント名] に、AWS Identity and Access Management (IAM) の接続に使用する Kubernetes サービスアカウントの名前を入力します。Kubernetes サービスアカウントが、ライセンスや計測などの AWS サービスを呼び出します。
 - ix. この製品バージョンで [QuickLaunch] を有効にします。クイック起動は の機能です AWS Marketplace。購入者は、クイック起動を使用して Amazon EKS クラスターをすばやく作成し、 を使用してそのクラスターでソフトウェアを起動できます AWS

CloudFormation。詳細については、「[でのクイック起動 AWS Marketplace](#)」を参照してください。

- x. [パラメータをオーバーライドする] に、ソフトウェアを起動する Helm CLI コマンドで使用するパラメータを入力します。これらのパラメータを使用すると購入者は、指定されたデフォルト値をオーバーライドできます。クイック起動を有効にしている場合は、CloudFormation フォームのパラメータ名と説明も入力します。AWS Marketplace マネジメントコンソールを使用する場合は 15 個のパラメータに制限がありますが、を使用する場合の制限はありません AWS Marketplace Catalog API。詳細については、「[コンテナベースの製品への新しいバージョンの追加](#)」を参照してください。

 Note

一部の [パラメータの上書き] は必須です。Amazon EKS Anywhere の製品では、ライセンスシークレットに "\${AWSMP_LICENSE_SECRET}" の DefaultValue を持つ [上書きパラメータ] が必要です。有料製品では、"\${AWSMP_SERVICE_ACCOUNT}" の DefaultValue を持つサービスアカウントの設定に対して 1 つの [上書きパラメータ] を指定する必要があります。

- xi. コンソール、コマンドラインツール、および API の機密情報をマスクするには、[パスワードとシークレットを非表示] を選択します。詳細については、「AWS CloudFormation ユーザーガイド」の「[パラメータ](#)」にある NoEcho パラメータのドキュメントを参照してください。
- c. Amazon EKS コンソールのアドオン配信オプションを追加するには、アーティファクトが [Amazon EKS アドオン製品の要件](#) に準拠していることを確認し、次の手順を実行します。

 Note

バージョンごとにサポートされる Amazon EKS アドオン配信オプションは 1 つだけです。作業している現在のバージョンが Amazon EKS コンソールで公開されるまで、新しいバージョンを追加することはできません。

- i. Helm チャートで、購入者がデプロイ環境にインストールする Helm チャートに Amazon ECR URL を追加して、ソフトウェアを起動します。

- ii. [コンテナイメージ] で、製品バージョンのソフトウェアを含むコンテナイメージに Amazon ECR URL を追加します。Helm チャート内のすべてのイメージがリストされていることを確認します。
 - iii. [配信オプションのタイトル] と [デプロイオプションの説明] に、この配信オプションのタイトルと説明を入力します。
 - iv. [可視性] では、[選択した制限] のデフォルト値が維持されます。
 - v. [アドオン名] に、このアドオンの一意の名前を入力します。入力したアドオン名には、Amazon EKS コンソールに表示される間に販売者の名前が付加されます。
 - vi. [アドオンバージョン] で、このアドオンのインストールまたはアップグレード時に表示されるアドオンのバージョンを入力します。これはフォーマット `major.minor.patch` に従います。
 - vii. [アドオンタイプ] で、ドロップダウンリストからアドオンのカテゴリを選択します。
 - viii. [Kubernetes バージョン] で、アドオンがサポートするすべての Kubernetes バージョンを選択します。
 - ix. [アーキテクチャ] で、アドオンがサポートするプラットフォームアーキテクチャを選択します。オプションは [AMD64] と [ARM64] です。互換性を最大化するために、両方のアーキテクチャをサポートすることが推奨されます。アドオンが ARM64 デバイスをサポートしていない場合は、製品がすべての商用 AWS リージョンに公開される前に、サポートの追加を希望する日付を指定する必要があります。
 - x. [名前空間] に、アドオンをインストールする一意の Kubernetes 名前空間を入力します。default、kube-system、および kube-public 名前空間は、サードパーティーのアドオンのインストールではサポートされていません。
 - xi. Environment Override パラメータでは、Amazon EKS アドオンフレームワークから最大 2 つの環境パラメータを選択できます。values.yaml のパラメータ名を、`${AWS_REGION}` および `${AWS_EKS_CLUSTER_NAME}` の環境変数にマッピングできます。
7. 他の配信オプションを追加するには、[新しい配信オプション] を選択し、前の手順を繰り返して構成します。
 8. [Submit] を選択してください。

ステップ 4: バージョン情報を更新する

バージョンが作成されたら、そのバージョンに関連する情報を変更して、更新された情報を購入者に提供すると便利です。例えば、バージョン 1.1 がリリースされた後にバージョン 1.0 を制限する

予定であれば、バージョン 1.0 の説明を更新して、購入者をバージョン 1.1 に誘導することができます。バージョン 1.0 が制限される日付を入力してください。バージョン情報は AWS Marketplace 管理ポータルから更新します。

バージョン情報を更新するには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [製品] メニューで [サーバー] を選択します。
3. [サーバー製品] タブで、変更する製品を選択します。
4. [変更をリクエスト] ドロップダウンから [バージョン情報を更新] を選択します。
5. [バージョンを更新] ページで、更新するバージョンを選択します。
6. 選択したバージョンを更新します。更新できるフィールドは、製品バージョンまたは配信オプションのステータスによって異なります。
 - a. すべてのバージョンで、[リリースノート] を更新できます。
 - b. まだ公開されていないバージョンでは、[バージョンタイトル] を更新できます。
 - c. 制限されていない配信オプションについては、以下のフィールドを更新できます。
 - 説明
 - [使用手順]
 - [サポートされるサービス]
 - d. まだ公開されていないバージョンの配信オプションについては、以下のフィールドを更新できます。
 - [配信オプションタイトル]
 - Helm チャート (Helm チャート配信オプションのみ)
 - コンテナイメージ
 - [デプロイリソース]
 - アドオン名
 - アドオンバージョン
 - アドオンタイプ
 - Helm チャート URI
 - CompatibleKubernetesVersions

- 名前空間
 - EnvironmentOverrideParameters
- e. 公開されているバージョンの配信オプションについては、SupportedArchitectures を更新できます。
7. [Submit] を選択してください。
8. リクエストが [リクエスト] タブに [レビュー中] ステータスで表示されていることを確認します。

リクエストのステータスは、[サーバー製品](#) ページの [リクエスト] タブからいつでも確認できます。

Amazon EKS アドオンのバージョンを制限する

アドオンとして公開されているコンテナ製品のバージョンを制限するには、「[AWS Marketplace 管理ポータル](#)」の下部にある連絡先を参照して、AWS Marketplace 運用チームにお問い合わせください。

AWS Marketplace でコンテナ製品価格を管理する

AWS Marketplace 販売者は、コンテナ製品の価格を管理および更新できます。コンテナ製品の場合、Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS) および AWS Fargate の無料製品、所有しているライセンスの持ち込み (BYOL) モデル製品、有料製品を出品できます。1 つの製品につき 1 つの料金設定ができます。価格ディメンションを追加または更新できます。価格モデルに応じて、契約と使用状況のディメンションを追加できます。価格条件または返金ポリシーを更新することもできます。詳細については、次のセクションを参照してください。

AWS Marketplace のコンテナ製品の価格モデルの詳細については、「[のコンテナ製品の料金 AWS Marketplace](#)」を参照してください。

トピック

- [料金設定ディメンションの追加](#)
- [ディメンション情報の更新](#)
- [料金条件の更新](#)
- [製品の返金ポリシーの更新](#)

料金設定ディメンションの追加

製品請求に使用する料金モデルに料金設定ディメンションを追加できます。料金モデルの詳細については、「[コンテナ料金モデル](#)」を参照してください。

Note

AWS Marketplace 管理ポータル では、消費料金が設定された契約に料金設定ディメンション (例えば、追加使用分の従量制料金) を追加することはできません。
契約、使用量、消費料金が設定された契約の間で料金モデルを変更することはできません。
サポートが必要な場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

1. <https://aws.amazon.com/marketplace/management/tour/> で AWS Marketplace 管理ポータル を開き、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページから、変更するコンテナ製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストから [\[料金設定ディメンションを更新\]](#) を選択し、[\[料金設定ディメンションを追加\]](#) を選択します。
4. 料金モデルによっては、[\[API 識別子\]](#)、[\[表示名\]](#)、[\[説明\]](#) の情報を入力して、契約ディメンションや使用量ディメンションを追加できます。
5. [\[次へ\]](#) を選択し、契約ディメンション料金を入力します。
6. [\[送信\]](#) を選択して、リクエストを送信します。
7. [\[リクエスト\]](#) タブの [\[リクエストステータス\]](#) が [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスが [\[成功\]](#) または [\[失敗\]](#) に更新されます。

ディメンション情報の更新

製品のディメンション情報を変更できます。料金モデルの詳細については、「[コンテナ料金モデル](#)」を参照してください。

1. <https://aws.amazon.com/marketplace/management/tour/> で AWS Marketplace 管理ポータル を開き、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページから、変更するコンテナ製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストから [\[料金設定ディメンションを更新\]](#) を選択し、[\[ディメンション情報を更新\]](#) を選択します。

4. 更新するディメンションの [表示名] と [説明] に情報を入力して、ディメンション情報を追加できます。
5. [送信] を選択して、リクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] または [失敗] に更新されます。

料金条件の更新

製品の料金条件を変更できます。製品を提供する国を変更した場合、料金条件を更新する必要がある場合があります。

1. <https://aws.amazon.com/marketplace/management/tour/> で AWS Marketplace 管理ポータルを開き、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページから、変更するコンテナ製品を選択します。
3. [変更をリクエスト] リストで [パブリックオファーを更新] を選択し、[料金条件を更新] を選択します。
4. 現在の料金は編集できるように事前に入力されています。現在の料金を削除して、新しい料金を追加できます。レビューのリクエストを送信する前に、リクエストする料金を確認することをお勧めします。
5. [送信] を選択して、リクエストを送信します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] または [失敗] に更新されます。

Note

[料金条件を更新] を使用してディメンションの料金を引き上げると、90 日間は料金を更新できなくなります。料金を引き上げると、料金は、変更を開始した日から 90 日間固定されます。この料金ロックは料金の値上げにのみ有効で、値下げには適用されません。

製品の返品ポリシーの更新

製品の返品ポリシーを更新できます。返品ポリシーの更新はすべてのユーザーに有効になります。詳細については、「[での製品の返品 AWS Marketplace](#)」を参照してください。

1. <https://aws.amazon.com/marketplace/management/tour/> で AWS Marketplace 管理ポータル を開き、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページから、変更するコンテナ製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストから [\[パブリックオファーを更新\]](#) を選択し、[\[返金ポリシーを更新\]](#) を選択します。
4. 現在の返金ポリシーは編集できるように事前に入力されています。現在の返金ポリシーを削除して、新しい返金ポリシーを追加できます。レビューのリクエストを送信する前に、リクエストする返金ポリシーを確認することをお勧めします。リクエストを送信すると、現在の返金ポリシーが上書きされます。
5. [\[送信\]](#) を選択して、リクエストを送信します。
6. [\[リクエスト\]](#) タブの [\[リクエストステータス\]](#) が [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスが [\[成功\]](#) または [\[失敗\]](#) に更新されます。

AWS Marketplace の国別にコンテナ製品の在庫状況を更新する

AWS Marketplace 販売者は、コンテナ製品を提供およびサブスクライブできる国を変更することができます。詳細については、「[Countries \(国\)](#)」を参照してください。次の手順では、コンテナ製品の在庫状況を国ごとに更新する方法を示します。

1. <https://aws.amazon.com/marketplace/management/tour/> で AWS Marketplace 管理ポータル を開き、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページから、変更するコンテナ製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストから [\[パブリックオファーを更新\]](#) を選択し、[\[国別の可用性を更新\]](#) を選択します。
4. 次のオプションのいずれかを選択します。
 - [\[すべての国\]](#) - サポートされているすべての国で利用できます。
 - [\[除外対象のすべての国\]](#) - 選択した国を除くすべてのサポート対象国で利用できます。
 - [\[許可リストに登録されている国のみ\]](#) - 購入者は指定した国のみで購入できます。
5. [\[送信\]](#) を選択して、リクエストを送信します。
6. [\[リクエスト\]](#) タブの [\[リクエストステータス\]](#) が [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスが [\[成功\]](#) または [\[失敗\]](#) に更新されます。

でのコンテナ製品のエンドユーザーライセンス契約 (EULA) の更新 AWS Marketplace

AWS Marketplace 販売者は、[の標準契約 AWS Marketplace](#)またはカスタム EULA を使用するようにエンドユーザーライセンス契約 (EULA) を更新できます。EULA の更新は、製品をサブスクライブする新規ユーザーや製品の更新時に有効になります。EULA を更新するには、次の手順を実行します。

1. <https://aws.amazon.com/marketplace/management/tour/> AWS Marketplace 管理ポータル で を開き、販売者アカウントにサインインします。
2. [\[サーバー製品\]](#) ページから、変更するコンテナ製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストから [\[パブリックオファーを更新\]](#) を選択し、[\[EULA を更新\]](#) を選択します。
4. [の標準契約 AWS Marketplace](#)を選択するか、カスタム EULA を送信します。カスタム EULA の場合は、Amazon S3 バケットからの契約を提供する必要があります。
5. [\[送信\]](#) を選択して、リクエストを送信します。
6. [\[リクエスト\]](#) タブの [\[リクエストステータス\]](#) が [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスが [\[成功\]](#) または [\[失敗\]](#) に更新されます。

AWS Marketplace でのコンテナ製品のテストとリリース

コンテナ製品の新しいバージョンを AWS Marketplace に追加したら、製品をテストして公開できます。このトピックでは、製品をテストして公開するために必要な特定の手順とプロセスについて説明します。

トピック

- [コンテナイメージと Helm チャート配信オプション](#)
- [Amazon EKS アドオン配信オプション](#)

コンテナイメージと Helm チャート配信オプション

このセクションでは、コンテナイメージと Helm チャートのリリースに関するガイダンスを提供します。

新しいバージョンのリクエストが作成され、数分以内に完了します。[\[サーバー製品\]](#) ページの [\[リクエスト\]](#) タブからリクエストを追跡できます。アドオンのテストまたはリリース時にエラーが発生し

た場合は、「AWS Marketplace Catalog API リファレンス」の「[Add a new version](#)」にある非同期エラーの表を参照してください。

Note

現在、製品が限定販売に設定されている場合、その製品を購入できる一部の購入者のみが製品バージョンにアクセスできます。現在、製品が一般公開に設定されている場合は、すべての AWS Marketplace 購入者が製品バージョンにアクセスできます。

これが初めてのバージョンセットだった場合は、これで製品を公開する準備が整いました。

Amazon EKS アドオン配信オプション

このセクションでは、Amazon EKS アドオンのテストとリリースに関するガイダンスを提供します。

アドオンをテストする

- アドオンを送信すると、AWS Marketplace はリクエストを処理し、Amazon EKS アドオンカタログで検証するための制限付き状態でアドオンを公開します。AWS Marketplace 管理ポータル の [サーバー製品] ページの [リクエスト] タブからリクエストを追跡できます。取り込みには、処理するリクエストの量に応じて 5~10 営業日かかります。

リクエストのステータスがレビュー中の場合、アドオンは AWS Marketplace から AWS チームによって Amazon EKS アドオンカタログに公開されます。アドオンが制限付き状態で公開されると、リクエストステータスが成功に変更されます。この後、アドオンのテストを開始できます。

- アドオンが使用可能になったら、テスト目的でアジアパシフィック (ソウル) リージョンで確認できます。AWS Marketplace でのソフトウェア機能の検証は、お客様の専門知識に依存します。アドオンをテストするには、アドオンが許可リストに登録されている販売者アカウントのアジアパシフィック (ソウル) リージョンに Amazon EKS クラスターを作成する必要があります。アドオンをテストするには、[こちらの詳細な手順](#)に従います。ソフトウェアがサポートする各 Kubernetes バージョンで必ずテストを行います。
- 有料製品を提供する場合は、以下の内部 AWS アカウント にプライベートオファーを作成します。これらのアカウントは、すべての商用 AWS リージョン の Amazon EKS コンソールにソフトウェアを統合するのに役立ちます。

288092140294, 288092140294, 408202761791

- AWS Marketplace で製品が承認され、アドオンバージョンを公開するまで、アドオンのテストクラスターをアクティブのまま維持します。

Note

AWS Marketplace は、Amazon EKS クラスターでのコンテナ製品のテスト中に発生する AWS インフラストラクチャコストは負担しません。適切なサイジングメカニズムに従って、テスト結果を検証しながら、ノードを最小限の運用コストにトーンダウンできます。

アドオンを一般に公開する

Amazon EKS クラスターを介してソフトウェアをアドオンとして検証したら、[AWS Marketplace 管理ポータル](#) または AWS Marketplace Catalog API を使用して、Amazon EKS アドオンのバージョンをパブリックにリリースするリクエストを送信できます。

詳細については、「AWS Marketplace Catalog API リファレンス」の「[Update the visibility for an Amazon EKS add-on](#)」を参照してください。

AWS Marketplace 管理ポータルの [サーバー製品] ページの [リクエスト] タブからリクエストを追跡できます。取り込み時間は異なります。

のコンテナベースの製品要件 AWS Marketplace

AWS Marketplace は、のすべてのコンテナベースの製品とサービスについて、次の要件を維持します AWS Marketplace。これらの要件は、お客様に安全、安心、および信頼できるカタログを提供するのに役立ちます。また、販売者には、特定の商品のニーズを満たすために、必要に応じて追加の規制やプロトコルの導入を検討するよう奨励しています。

すべての製品とその関連メタデータは、送信時にレビューされ、現在の AWS Marketplace ポリシーを満たしているか超えているかが確認されます。これらのポリシーは、進化するセキュリティガイドラインに合わせて定期的に更新されます。AWS Marketplace は製品を継続的にスキャンして、既存の出品がこれらの要件に対する変更を引き続き満たしていることを確認します。製品がコンプライアンス違反になった場合、AWS Marketplace は販売者に連絡して、新しい基準を満たすように製品を更新します。場合によっては、問題が解決されるまで、新しいサブスクライバーが製品を一時的に使用できなくなることがあります。このプロセスは、すべてのユーザーの AWS Marketplace プラットフォームのセキュリティと信頼性を維持するのに役立ちます。

トピック

- [セキュリティポリシー](#)
- [顧客情報の要件](#)
- [製品の使用要件](#)
- [アーキテクチャの要件](#)
- [コンテナ製品の使用手順](#)
- [Amazon EKS アドオン製品の要件](#)

セキュリティポリシー

すべてのコンテナベースの製品は、以下の要件を満たしている必要があります。

- コンテナイメージには、既知の脆弱性、マルウェア、またはEnd-of-Life (EoL) ソフトウェアパッケージとオペレーティングシステムを含めることはできません。
- コンテナは、AWS サービスにアクセスするための AWS 認証情報をリクエストしてはいけません。製品が AWS サービスにアクセスする必要がある場合は、次のいずれかを使用する必要があります。
 - Amazon Elastic Kubernetes Service (Amazon EKS) ワークロード用のサービスアカウントの IAM ロール。
 - Amazon Elastic Container Service (Amazon ECS) ワークロード用のタスクの IAM ロール。
- コンテナベースの製品の実行には最小特権のみが必要です。詳細については、「[Amazon Elastic Container Service のセキュリティ](#)」および「[Amazon EKS のセキュリティ](#)」を参照してください。
- コンテナイメージは、デフォルトで非ルートの権限で実行するように設定する必要があります。
- コンテナには、システムユーザーとサービスのパスワード (ハッシュ化も含む)、プライベートキー、認証情報などのハードコードされたシークレットを含めることはできません。
- コンテナ内で実行されているサービスの認証では、起動時にユーザーがパスワードを生成、リセット、定義しても、パスワードベースの認証を使用しないでください。null および空白のパスワードも許可されません。
- コンテナイメージには、サポートされていないアーキテクチャ (In-to-Attestation Framework メタデータなど) を持つレイヤーを含めることはできません。

顧客情報の要件

すべてのコンテナベースの製品は、次の顧客情報の要件に従う必要があります。

- ソフトウェアは、所有しているライセンスの持ち込み (BYOL) で要求される場合を除き、顧客の知識と明示的な同意なしに顧客データを収集、またはエクスポートしてはなりません。顧客データを収集またはエクスポートするアプリケーションは、以下のガイドラインに従う必要があります。
- 顧客データの収集は、セルフサービスで、自動化され、安全である必要があります。購入者は、販売者がソフトウェアの導入を承認するのを待つ必要はありません。
- 顧客データの収集は、[AWS Marketplace の利用規約](#) AWS、[AWS サービス条件](#)、[AWS プライバシー通知](#)、[AWS カスタマーアグリーメント](#)など、との契約と一致する必要があります。
- 支払情報は収集してはなりません。

製品の使用要件

すべてのコンテナベースの製品は、次の製品使用の要件に従う必要があります。

- 販売者は完全に機能する商品のみを出品できます。トライアルまたは評価を目的としたベータ版またはプレリリース版の製品は許可されていません。商用ソフトウェアの開発者エディション、コミュニティエディション、BYOL エディションは、販売者が無料エディションの提供から 90 AWS Marketplace 日以内に同等の有料バージョンを に提供した場合、サポートされます。
- コンテナベースの製品のすべての使用説明書には、コンテナベースの製品をデプロイするためのすべての手順が記載されている必要があります。使用説明書には、上の対応するコンテナイメージを指すコマンドとデプロイリソースが記載されている必要があります。AWS Marketplace
- コンテナベースの製品には、サブスクライバーがソフトウェアを使用するために必要なすべてのコンテナイメージが含まれている必要があります。さらに、コンテナベースの製品では、ユーザーが外部からのイメージ AWS Marketplace (サードパーティーリポジトリからのコンテナイメージなど) を使用して製品を起動する必要はありません。
- コンテナとそのソフトウェアはセルフサービス方式でデプロイ可能である必要があります、追加の支払い方法や費用を必要としないものでなければなりません。デプロイ時に外部に依存する必要があるアプリケーションは、以下のガイドラインに従う必要があります。
- 要件は、リストの説明または使用説明書に明記する必要があります。例えば、この製品を正しくデプロイするにはインターネット接続が必要です。デプロイ時に以下のパッケージがダウンロードされます。<パッケージのリスト>。
- 販売者は、すべての外部依存関係を使用し、その可用性とセキュリティを確保する責任を負います。
- 外部依存関係が使用できなくなった場合は、製品 AWS Marketplace も から削除する必要があります。
- 外部依存関係によって追加の支払い方法や費用が必要になってはいけません。

- 購入者の直接の管理下でない外部リソース (外部 API、販売者または第三者が管理する AWS のサービス など) への継続的な接続を必要とするコンテナは、以下のガイドラインに従う必要があります。
- 要件は、リストの説明または使用説明書に明記する必要があります。例えば、この製品には継続的なインターネット接続が必要です。正しく機能するには、以下の継続的な外部サービスが必要です。<リソースのリスト>。
- 販売者は、すべての外部リソースを使用し、その可用性とセキュリティを確保する責任を負います。
- 外部リソースが使用できなくなった場合は、製品 AWS Marketplace も から削除する必要があります。
- 外部リソースは追加の支払い方法や費用を必要とせず、接続の設定を自動化する必要があります。
- 製品ソフトウェアとメタデータには、AWS Marketplaceでは利用できない他のクラウドプラットフォーム、追加の製品、またはアップセルサービスにユーザーをリダイレクトする言葉を含めてはいけません。
- 製品が別の製品または別の ISV 製品のアドオンである場合、製品の説明には、それが他の製品の機能を拡張するものであり、これがないと製品の有用性が非常に限られることを明記する必要があります。例えば、この製品は <製品名> の機能を拡張するものであり、それがなければ、この製品の有用性は非常に限られています。<製品名> は、このリストのすべての機能を利用するには、独自のライセンスが必要な場合がありますのでご注意ください。

アーキテクチャの要件

すべてのコンテナベースの製品は、次のアーキテクチャの要件に従う必要があります。

- のソースコンテナイメージは、 が所有する Amazon Elastic Container Registry (Amazon ECR) リポジトリにプッシュ AWS Marketplace する必要があります AWS Marketplace。これらのリポジトリは、コンテナ製品リストごとにサーバー製品の下に AWS Marketplace 管理ポータル に作成できます。
- コンテナイメージは Linux ベースである必要があります。
- 有料のコンテナベースの製品は [Amazon ECS](#)、[Amazon EKS](#)、または [AWS Fargate](#) にデプロイできる必要があります。
- 契約料金で と統合されている有料コンテナベースの製品は、Amazon EKS、Amazon ECS AWS Fargate、Amazon EKS Anywhere、Amazon ECS Anywhere、Red Hat OpenShift Service on AWS

(ROSA)、オンプレミスのセルフマネージド Kubernetes クラスター、または Amazon Elastic Compute Cloud にデプロイ AWS License Manager する必要があります。

コンテナ製品の使用手順

コンテナ製品の使用説明書を作成するときは、「[the section called “AMI とコンテナ製品の使用手順”](#)」の手順とガイダンスに従ってください。

Amazon EKS アドオン製品の要件

Amazon EKS アドオンは、Kubernetes アプリケーションをサポートする操作機能を提供するソフトウェアですが、アプリケーション固有のものではありません。例えば、Amazon EKS アドオンには、クラスターがネットワーク、コンピューティング、ストレージの基盤となる AWS リソースとやり取りできるようにするオブザーバビリティエージェントまたはKubernetesドライバーが含まれています。

コンテナ製品の販売者は、Amazon EKS を含むいくつかのデプロイオプションから選択できます。製品のバージョンをアドオンとして Amazon EKS AWS Marketplace アドオンカタログに公開できます。アドオンは、AWS や他のベンダーが管理するアドオンの横に Amazon EKS コンソールに表示されます。購入者は、他のアドオンと同様に簡単にソフトウェアをアドオンとしてデプロイできます。

詳細については、Amazon EKS ユーザーガイドの [Amazon EKS アドオン](#) を参照してください。

コンテナ製品を AWS Marketplace アドオンとして準備する

コンテナ製品を AWS Marketplace アドオンとして公開するには、次の要件を満たしている必要があります。

- コンテナ製品は で公開する必要があります AWS Marketplace。
- コンテナ製品は AMD64 と ARM64 の両方のアーキテクチャに対応するように構築されている必要があります。
- コンテナ製品で Bring Your Own License (BYOL) [料金モデル](#) を使用してはなりません。

Note

BYOL は Amazon EKS アドオン配信ではサポートされていません。

- すべての [コンテナイメージとチャートをマネージド Amazon ECR リポジトリにプッシュするなど、コンテナベースのすべての製品要件](#) に従う必要があります。Helm AWS Marketplace この要件には、nginx など、オープンソースイメージなどが含まれます。イメージとチャートは、[Amazon ECR Public Gallery](#)、Docker Hub、Quay などを含むがこれらに限定されない他の外部リポジトリではホストできません。
- Helm charts - ソフトウェアをHelmグラフとして準備してパッケージ化します。Amazon EKS アドオンフレームワークは、Helmチャートを Kubernetes マニフェストに変換します。一部の Helm 機能は Amazon EKS システムではサポートされていません。次のリストは、Amazon EKS アドオンとしてソフトウェアをオンボーディングする前に満たす必要がある要件を示しています。このリストでは、すべての Helm コマンドが Helm バージョン 3.8.1 を使用します。
- すべての Capabilities オブジェクトがサポートされていますが、.APIVersions は例外です。.APIVersions は、組み込まれていないカスタム Kubernetes API ではサポートされていません。
- Release.Name および Release.Namespace オブジェクトのみがサポートされています。
- Helm はフッキングしますが、lookup 関数はサポートされません。
- すべての依存チャートは、メインの Helm チャート (リポジトリパス file:// で指定) に配置する必要があります。
- Helm チャートは、エラーなしで Helm Lint と Helm テンプレートを正常に渡す必要があります。コマンドは次のとおりです。
 - Helm Lint - `helm lint helm-chart`

よくある問題には、親チャートのメタデータに宣言されていないグラフが含まれます。

例: chart metadata is missing these dependencies: chart-base Error: 1 chart(s) linted, 1 chart(s) failed

- Helm テンプレート - `helm template chart-name chart-location --set k8version=Kubernetes-version --kube-version Kubernetes-version --namespace addon-namespace --include-crds --no-hooks -f any-overridden-values`

-f フラグを使用してオーバーライドされた設定を渡します。

- すべてのコンテナバイナリを AWS Marketplace Amazon ECR リポジトリに保存します。マニフェストを作成するには、前述の Helm テンプレートコマンドを使用します。busybox や gcr イメージなどの外部イメージリファレンスのマニフェストを検索します。リクエストドロップダウンのリポジトリの追加オプションを使用して作成された AWS Marketplace Amazon ECR リポジトリに、すべてのコンテナイメージと依存関係をアップロードします。

- カスタム設定 — デプロイ中にカスタム変数を追加できます。エンドユーザーエクスペリエンスを識別し、ソフトウェア `aws_mp_configuration_schema.json` に名前を付けて、Helm チャートを使用してラッパーにパッケージ化する方法の詳細については、「[Amazon EKS アドオン: 詳細設定](#)」を参照してください。

「[\\$schema](#)」キーワードによると、`$schema` は、有効な `application/schema+json` リソースを指し URI である必要があります。

このファイルは、パスワード、ライセンスキー、証明書などの機密情報を承認する必要があります。

シークレットと証明書のインストールを処理するには、エンドユーザーにアドオン後のインストール手順またはアドオン前のインストール手順を提供します。製品は、外部ライセンスに依存してはなりません。製品は、AWS Marketplace エンタイトルメントに基づいて動作する必要があります。

`aws_mp_configuration_schema.json` の制限事項の詳細については、「[アドオンプロバイダーのアドオン設定要件とベストプラクティス](#)」を参照してください。

- ソフトウェアがデプロイされる名前空間を特定して作成する – 製品の最初のリリースでは、テンプレート化された名前空間を追加して、ソフトウェアがデプロイされる名前空間を特定する必要があります。
- カスタムリソース定義 (CRDs) – Amazon EKS アドオンフレームワークは、同じアドオンで適用された CRDs に基づく CRDs とカスタムリソース宣言のインストールをサポートしていません。アドオンにカスタムリソースがあり、CRDs に依存している場合は、次のいずれかを実行できます。
 - CRD 定義を別のアドオン (個別の helm チャート) に分割し、実際のカスタムリソースのインストールを別のアドオンに分割するという 2 つのアドオンを発行します。 <https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/>
 - 追加の手動手順を含む単一のアドオンを発行する: CRDs にインストールする単一のアドオンを発行します。エンドユーザーが CRDs に依存するカスタムリソースを設定できるように、kubernetes マニフェストファイルとともに使用手順を提供します。
- **serviceAccount** 該当する場合に を作成する – ソフトウェアが の有料ソフトウェアであるか、他のソフトウェアに接続 AWS Marketplace する必要がある場合は AWS のサービス、`serviceAccount` デフォルトで Helm グラフが作成されていることを確認してください。`serviceAccount` の作成が `values.yaml` ファイルのパラメータで処理された場合、パラメータ値を `true` に設定します。例えば、`serviceAccount.create = true`。これは、必要な権限をすでに取得している基本となるノードインスタンスから権限を継承してアド

オンをインストールすることを顧客が選択する可能性があるため、必須です。Helm チャートが、serviceAccount を作成しない場合は、serviceAccount に権限を付与できません。

- 追跡可能なデプロイまたはデーモンセット – Helm チャートにデーモンセットまたはデプロイがあることを確認します。Amazon EKS アドオンフレームワークは、それらを使用して Amazon EKS リソースのデプロイを追跡します。追跡可能なデプロイまたはデーモンセットがない場合、アドオンでデプロイエラーが発生します。アドオンにデプロイやデーモンセットがない場合、例えば、アドオンが追跡不可能なカスタムリソースや Kubernetes ジョブをたくさんデプロイする場合は、ダミーデプロイまたはデーモンセットオブジェクトを追加します。
- AMD および ARM アーキテクチャのサポート – 多くの Amazon EKS のお客様は、現在 ARM64 を使用して Graviton AWS インスタンスを使用しています。サードパーティーソフトウェアは両方のアーキテクチャをサポートしている必要があります。
- のライセンスまたは計測 APIs と統合 AWS Marketplace すると、複数の請求モデル AWS Marketplace がサポートされます。詳細については、「[コンテナ製品の請求、計測、ライセンスの統合](#)」を参照してください。PAYG メカニズムを使用して製品を販売する場合は、「[AWS Marketplace Metering Service を使用したコンテナ製品のカスタム計測の設定](#)」を参照してください。前払いモデルまたは契約モデルを使用して製品を販売する場合は、「[を使用したコンテナ製品の契約料金 AWS License Manager](#)」を参照してください。
- ソフトウェアとすべてのアーティファクトと依存関係をアップロードする – Helm チャートは自己完結型でなければならない、GitHub などの外部ソースからの依存関係を必要としません。ソフトウェアに外部依存関係が必要な場合は、依存関係を同じ AWS Marketplace リスト内の AWS Marketplace プライベート Amazon ECR リポジトリにプッシュする必要があります。
- ウェブサイトでデプロイ手順を提供する – 顧客が [create-addon](#) コマンドを使用してソフトウェアをデプロイする方法を特定するためのデプロイガイドをホストすることが推奨されます。
- アドオンのアクセス許可/IAM ロール – から公開されたアドオンが AWS サービスにアクセス AWS Marketplace する必要がある場合、ソフトウェアにはサービスにアクセスするための IAM ポリシーで注釈が付けられた Kubernetes AWS サービスアカウントが必要です。サービスアカウントに 2 つのオプションから選択して、AWS サービスに API リクエストを行うことができます。
- IRSA 経由の認証情報: このオプションを使用すると、ソフトウェアは Identity and Access Management (IAM) Role Service (IRSA) から継承認証情報を取得できます。詳細については、「[サービスアカウントの IAM ロール](#)」を参照してください。
- Amazon EKS ポッドアイデンティティ: このオプションを使用すると、ソフトウェアは Amazon EKS ポッドのポッドアイデンティティを使用して AWS サービスに API リクエストを行うことができます。詳細については、「[EKS Pod Identity がポッドに AWS サービスへのアクセスを許可する方法](#)」を参照してください。

アドオンには、現在のカスタム設定スキーマ () と同じディレクトリにある Helm チャートの最aws_mp_addon_parameters.json上位にある という名前の追加設定ファイルが必要ですaws_mp_configuration_schema.json。現在、このファイルはポッド ID 互換のアクセス許可のみを処理します。ファイル形式は次のとおりです。

```
{
  "permissions": {
    "isPodIdentityCompatible" : true,
    "permissionsList": [
      {
        "serviceAccount" : "String",
        "managedPolicies" : ["Policy Arn"],
      }
    ]
  }
}
```

ファイル名: **aws_mp_addon_parameters.json**

 Note

aws_mp_addon_parameters.json ファイルは、Amazon EKS コンソールのアドオン設定ページのアドオンアクセスセクションを有効にします。

フィールド名	タイプ	メモ	値の例
isPodIdentityCompatible	ブール値	現時点では、「true」のみがサポートされています。フィールドには、次の permissionsList リストで説明されているアクセス許可がポッド ID に適しているかどうかが表示されます。	正

フィールド名	タイプ	メモ	値の例
serviceAccount	String	アドオンがアクセス許可にアクセスするために使用するサービスアカウントの名前	kpow
managedPolicies	リスト < 文字列 >	EKS アドオンが引き受ける可能性のある、このサービスアカウントに使用するポリシー ARN のリスト	["arn:aws:iam::aws:policy/ReadOnlyAccess"]

Note

のPay-as-you-go (PAYG) アドオン製品は Amazon EKS Pod Identity AWS Marketplace を使用できないため、アクセスコントロールにはサービスアカウント (IRSA) の IAM ロールを使用する必要があります。

- バージョンを更新 – Amazon EKS は、アップストリームリリースから数週間後に新しい Kubernetes バージョンをリリースします。新しい Amazon EKS クラスターバージョンが一般公開されると、ベンダーは 45 日以内に新しい Amazon EKS クラスターバージョンリリースと互換性があるソフトウェアを認証または更新する必要があります。現在のアドオンバージョンが新しい Kubernetes バージョンをサポートしている場合は、検証して認証することで、バージョンの互換性マトリックスを更新できます。新しい Kubernetes バージョンリリースをサポートするために新しいアドオンバージョンが必要な場合は、オンボーディングのために新しいバージョンを送信してください。
- パートナーのソフトウェアは、次のいずれかのタイプに該当するか、Kubernetes または Amazon EKS を強化する運用ソフトウェアである必要があります: 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。
- ソフトウェアを [Container Network Interface \(CNI\)](#) にすることはできません。
- ソフトウェアは、を通じて販売 AWS Marketplace され、有料製品のライセンスおよび計測 APIs と統合されている必要があります。BYOL 製品は受け付けられません。

アドオンプロバイダーのアドオン設定要件とベストプラクティス

Amazon EKS では、アドオンプロバイダーからの [Helm JSON スキーマ](#) 文字列としての設定が必要です。必要な設定を必要とするアドオン、またはオプションの設定を許可するアドオンには、Helm チャートが送信された `aws_mp_configuration_schema.json` ファイルを含める必要があります AWS Marketplace。Amazon EKS はこのスキーマを使用して、顧客からの設定入力を検証し、スキーマに準拠しない入力値を持つ API コールを拒否します。アドオン設定は、通常 2 つのカテゴリに分類されます。

- ラベル、許容範囲、nodeSelector などの一般的な Kubernetes プロパティの設定。
- ライセンスキー、機能有効化、URL など、アドオン固有の設定。

このセクションでは、一般的な Kubernetes プロパティに関連する最初のカテゴリに焦点を当てます。

Amazon EKS では、Amazon EKS アドオンの設定に関するベストプラクティスに従うことが推奨されます。

- [スキーマ要件](#)
- [設定で許可される一般的なパラメータ](#)
- [設定に許可されていない一般的なパラメータ](#)

スキーマ要件

json スキーマを定義するときは、Amazon EKS アドオンでサポートされている jsonschema のバージョンを使用します。

サポートされているスキーマのリスト:

- <https://json-schema.org/draft-04/schema>
- <https://json-schema.org/draft-06/schema>
- <https://json-schema.org/draft-07/schema>
- <https://json-schema.org/draft/2019-09/schema>

他の json スキーマバージョンを使用すると、Amazon EKS アドオンと互換性がとれなくなり、修正されるまでアドオンをリリースできなくなります。

Helm スキーマファイルの例

```
{
  "$schema": "http://json-schema.org/schema#",
  "type": "object",
  "properties": {
    "podAnnotations": {
      "description": "Pod Annotations"
    },
    "podLabels": {
      "description": "Pod Labels"
    },
    "resources": {
      "description": "Resources"
    },
    "logLevel": {
      "description": "Logging Level"
    },
    "config": {
      "description": "Custom Configuration"
    }
  }
}
```

camelCase

設定パラメータは camelCase である必要があります。この形式に沿っていない場合は、拒否されます。

説明は必須

スキーマプロパティには常に意味のある説明を含めます。この説明は、設定パラメータごとに Amazon EKS コンソールでラベル名をレンダリングするために使用されます。

RBAC 定義

アドオンプロバイダーは、最小権限の原則を使用して、アドオンを正常にインストールするために必要な RBAC アクセス許可を定義して提供する必要があります。RBAC アクセス許可が、新しいバージョンのアドオンまたは CVE に対応するための修正に対して変更する必要がある場合、アドオンプロバイダーはその変更を Amazon EKS チームに通知する必要があります。各 Kubernetes リソースに必要なアクセス許可は、オブジェクトのリソース名に制限する必要があります。

```
apiGroups: ["apps"]
resources: ["daemonsets"]
resourceNames: ["ebs-csi-node"]
verbs: ["create", "delete", "get", "list", "patch", "update", "watch"]
```

シークレットの管理

このセクションは、顧客がアプリケーションキー、API キー、パスワードなどのシークレット情報を設定する必要があるアドオンにのみ適用されます。現在、Amazon EKS API では、セキュリティ上の影響により、機密情報をプレーンテキストで渡すことはサポートされていません。ただし、顧客は設定を使用して、アドオンに必要なキーを保持する Kubernetes シークレットの名前を渡すことができます。お客様は、前提条件のステップと同じ名前空間を持つキーを含む Kubernetes Secret オブジェクトを作成し、アドオンの作成時に設定 blob を使用して Secret の名前を渡す必要があります。アドオンプロバイダーは、顧客が誤って実際のキーと間違えないように、スキーマプロパティに名前を付けることをお勧めします。例: appSecretName、connectionSecretName など。

つまり、アドオンプロバイダーはスキーマを活用して、顧客がシークレットの名前を渡すことはできますが、実際にシークレット自体を保持するキーは渡すことはできません。

設定値の例

アドオンの設定で顧客を支援するために、スキーマに設定例を含めることができます。次の例は、AWS Distro for OpenTelemetry アドオンのスキーマからのものです。

```
"examples": [
  {
    "admissionWebhooks": {
      "namespaceSelector": {},
      "objectSelector": {}
    },
    "affinity": {},
```

```
"collector": {
  "amp": {
    "enabled": true,
    "remoteWriteEndpoint": "https://aps-workspaces.us-west-2.amazonaws.com/
workspaces/ws-xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx/api/v1/remote_write"
  },
  "cloudwatch": {
    "enabled": true
  },
  "mode": "deployment",
  "replicas": 1,
  "resources": {
    "limits": {
      "cpu": "256m",
      "memory": "512Mi"
    },
    "requests": {
      "cpu": "64m",
      "memory": "128Mi"
    }
  },
  "serviceAccount": {
    "annotations": {},
    "create": true,
    "name": "adot-collector"
  },
  "xray": {
    "enabled": true
  }
},
"kubeRBACProxy": {
  "enabled": true,
  "resources": {
    "limits": {
      "cpu": "500m",
      "memory": "128Mi"
    },
    "requests": {
      "cpu": "5m",
      "memory": "64Mi"
    }
  }
},
"manager": {
```

```
    "env": {},
    "resources": {
      "limits": {
        "cpu": "100m",
        "memory": "128Mi"
      },
      "requests": {
        "cpu": "100m",
        "memory": "64Mi"
      }
    },
    "nodeSelector": {},
    "replicaCount": 1,
    "tolerations": []
  }
]
```

設定で許可される一般的なパラメータ

以下は、顧客向け Helm スキーマファイルで推奨されるパラメータです。

パラメータ	説明	デフォルトの有無
additionalLabels	アドオンが管理するすべての Kubernetes オブジェクトに Kubernetes ラベルを追加します。	いいえ
additionalAnnotations	アドオンが管理するすべての Kubernetes オブジェクトに Kubernetes 注釈を追加します。	いいえ
podLabels	アドオンが管理する Pod に Kubernetes ラベルを追加します。	いいえ
podAnnotations	アドオンが管理する Pod に Kubernetes 注釈を追加します。	いいえ

パラメータ	説明	デフォルトの有無
logLevel	アドオンが管理するコンポーネントのログレベル。	はい
nodeSelector	ノード選択制約の最も簡単な推奨される形式。nodeSelector フィールドを Pod 仕様に追加すると、ターゲットノードに付けるノードラベルを指定できます。	例えば Linux ノードのみなど。
tolerations	許容値は Pod に適用されます。許容値により、スケジューラはテイントと一致する Pod をスケジューリングできます。許容値によってスケジューリングは可能ですが、スケジューリングを保証するものではありません。	おそらく、デモンセットでより一般的です。
affinity	アフィニティ機能には次の 2 種類があります: Node Affinity は nodeSelector フィールドのように機能しますが、より表現力が豊かで、ソフトウェアルールを指定できます。Inter-Pod Affinity/Anti-Affinity は、他の Pod のラベルに対して Pod を制約できます。	おそらく
topologySpreadConstraints	Topology Spread による制約を使用すると、リージョン、ゾーン、ノード、その他のユーザー定義トポロジードメインなどで障害を起こしたドメイン間で、Pod がクラスター全体に分散される方法を制御できます。これにより、高可用性と効率的なリソース使用率を実現できます。	おそらく

パラメータ	説明	デフォルトの有無
リソースリクエスト/制限	各コンテナに必要な CPU/メモリの量を指定します。リクエストを設定することが強く推奨されます。制限はオプションです。	はい
レプリカ	アドオンによって管理される Pod のレプリカの数。デーモンセットには適用されません。	はい

Note

ワークロードスケジューリング設定パラメータでは、必要に応じてスキーマの最上位のコンポーネントを分離する必要がある場合があります。例えば、Amazon EBS CSI ドライバーには、コントローラーとノードエージェントという 2 つの主要なコンポーネントが含まれています。顧客は、コンポーネントごとに異なるノードセクタ/許容値を設定する必要があります。

Note

JSON スキーマで定義されているデフォルト値は、ユーザードキュメントのみを目的としており、`values.yaml` ファイルで適切なデフォルトを使用する必要があります。デフォルトプロパティを使用する場合は、Helm チャートを変更するたびに、`values.yaml` のデフォルトがスキーマのデフォルトと一致し、`values.schema.json` と `values.yaml` の 2 つのアーティファクトが同期されていることを確認します。

```
"affinity": {
  "default": {
    "affinity": {
      "nodeAffinity": {
        "preferredDuringSchedulingIgnoredDuringExecution": [
          {
            "preference": {
```

```
        "matchExpressions": [
            {
                "key": "eks.amazonaws.com/compute-type",
                "operator": "NotIn",
                "values": [
                    "fargate"
                ]
            }
        ],
        "weight": 1
    }
],
},
"podAntiAffinity": {
    "preferredDuringSchedulingIgnoredDuringExecution": [
        {
            "podAffinityTerm": {
                "labelSelector": {
                    "matchExpressions": [
                        {
                            "key": "app",
                            "operator": "In",
                            "values": [
                                "ebs-csi-controller"
                            ]
                        }
                    ]
                },
                "topologyKey": "kubernetes.io/hostname"
            },
            "weight": 100
        }
    ]
}
},
"description": "Affinity of the controller pod",
"type": [
    "object",
    "null"
]
}
```

設定に許可されていない一般的なパラメータ

clusterName、region、vpcId、accountId などのクラスタメタデータパラメータは、Elastic Load Balancing Controller などのさまざまなアドオンで必要な場合があります。Amazon EKS サービスで知られているこれらに類似するパラメータは、Amazon EKS アドオンによって自動的に挿入されるため、ユーザーが設定オプションとして指定する責任はありません。パラメータには以下が含まれます。

- AWS リージョン
- Amazon EKS クラスター名
- クラスターの VPC ID
- コンテナレジストリ (特にネットワークアドオンで使用するビルドプロッドアカウント用)
- DNS クラスター IP (特に coredns アドオン用)
- Amazon EKS クラスター API エンドポイント
- クラスターで有効になっている IPv4
- クラスターで有効になっている IPv6
- クラスターで有効になっている IPv6 のプレフィックス委任

アドオンプロバイダーは、該当するパラメータにテンプレートが定義されていることを確認する必要があります。上記の各パラメータには、Amazon EKS で事前定義された parameterType 属性があります。リリースメタデータは、parameterType と テンプレート内のパラメータ名/パス間のマッピングを指定します。これにより、Amazon EKS は、顧客が設定でこれらを指定しなくても、動的に値を渡すことができるため、アドオンプロバイダーは、独自のテンプレート名/パスを柔軟に定義できます。Amazon EKS が動的に注入する必要がある上記のようなパラメータは、スキーマファイルから除外する必要があります。

リリースメタデータからのマッピングの例

```
"defaultConfiguration": [  
  {  
    "key": "image.containerRegistry",  
    "parameterType": "CONTAINER_REGISTRY"  
  }  
]
```

以下は、顧客向け Helm スキーマファイルでの設定が非推奨のパラメータです。パラメータは変更不可能のデフォルトにするか、アドオンテンプレートに一切含めるべきではありません。

パラメータ	説明	デフォルトの有無
画像	Kubernetes クラスターにデプロイされるコンテナイメージ。	いいえ、アドオン定義で管理されます
imagePullSecrets	シークレットを使用してプライベートレジストリからプルするように Pod を設定する。	該当なし
livenessProbe	Kubelet プロセスは、ライブネスプローブを使用して、コンテナを再起動するタイミングを確認します。例えば、ライブネスプローブは、アプリケーションが実行されているが進行できないデッドロックを取得する場合があります。このような状態でコンテナを再起動すると、バグが発生してもアプリケーションを使用できます。	はい
readinessProbe	コンテナの準備状況プローブを用意することが重要です。これにより、データプレーンで実行されている Kubelet プロセスは、コンテナがトラフィックを処理する準備ができたタイミングを知ることができます。Pod は、すべてのコンテナの準備が整った時点で準備完了と見なされます。このシグナルの 1 つの用途は、サービスのバックエンドとして使用される Pod を制御することです。Pod の準備が整っていない場合、Pod はサービスロードバランサーから削除されます。	はい

パラメータ	説明	デフォルトの有無
startupProbe	kubelet はスタートアッププローブを使用して、コンテナアプリケーションの開始時期を把握します。このようなプローブが設定されている場合、成功するまでライブネスと準備状況チェックを無効にし、それらのプローブがアプリケーションの起動に干渉しないことを確認します。これは、起動の遅いコンテナにライブネスチェックを採用するために使用され、起動して実行する前に kubelet による停止を回避できます。	オプションです。
podDisruptionBudget	Pod Disruption Budget (PDB) を定義して、自発的な中断時に最小限の数の PODS が実行され続けるようにします。PDB は、任意の中断によって同時にダウンするレプリケートされたアプリケーションの Pod の数を制限します。例えば、クォーラムベースのアプリケーションは、実行中のレプリカの数にクォーラムに必要な数を下回らないようにします。ウェブフロントエンドは、負荷に対応するレプリカの数に合計の一定割合を下回らないようにします。	はい。デフォルトで 3 つ以上のレプリカに設定されている場合
serviceAccount (名前)	実行されるサービスアカウント Pod の名前。	はい

パラメータ	説明	デフォルトの有無
serviceAccount (注釈)	サービスアカウントに適用される注釈。通常、サービスアカウント機能の IAM ロールに使用されます。	いいえ、IAM サービスアカウントロール ARN は最上位の Amazon EKS アドオン API に設定されています。このルールの例外として、アドオンに複数のデプロイ/コントローラー (Flux など) があり、個別の IRSA ロール ARN が必要な場合が挙げられます。
priorityClassName	Priority は、他の Pod と比較した Pod の重要性を示します。Pod をスケジュールできない場合、スケジューラは優先度の低い Pod を優先 (排除) して、保留中の Pod のスケジューリングを可能にします。	はい。ほとんどのアドオンはクラスター機能にとって重要であり、デフォルトで優先度クラスが設定されている必要があります。
podSecurityContext	セキュリティコンテキストは、Pod とコンテナの権限とアクセス制御設定を定義します。通常、fsGroup の設定に使用されます。これは v1.19 以下のクラスターの IRSA に必要でした。	Amazon EKS が Kubernetes v1.19 をサポートしなくなった場合、可能性は低いです。

パラメータ	説明	デフォルトの有無
SecurityContext	セキュリティコンテキストは、Pod とコンテナの権限とアクセス制御設定を定義します。	はい
updateStrategy	古い Pod を新しい Pod に置き換えるために使用される戦略を指定します。	はい
nameOverride	Pod の名前を上書きします。	いいえ
podSecurityPolicy	パラメータに制限を適用します。	いいえ - PSP は非推奨です
extraVolumeMounts /extraVolumes	Amazon EKS 以外のクラスターの IRSA に使用されます。	いいえ

のコンテナ製品の料金 AWS Marketplace

では AWS Marketplace、Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS)、およびの無料製品、Bring Your Own License Model (BYOL) 製品、有料製品を一覧表示できます AWS Fargate。1 つの製品につき 1 つの料金設定ができます。このトピックでは、コンテナ製品で利用できる料金モデルについて説明します。

Note

[AWS Marketplace Metering Service](#) を使用して、有料製品の使用権限管理および計測を強化します。タスク単位またはポッド単位の料金については、AWSによって自動的に使用量が計測されます。

コンテナ製品に設定した料金は、すべてのに適用されます AWS リージョン。コンテナ製品の料金を下げるたびに、新しい料金が購入者に即時に実装されます。料金を上げる場合、変更が請求に反映される 90 日前に既存の購入者はその変更について通知されます。新規購入者には新しい金額が請求されます。

Note

新規サブスクライバーの場合、料金変更はすぐに有効になります。既存のサブスクライバーの場合、料金変更は、料金変更通知が送信された日から始まる 90 日間の期間の翌月の初日に有効になります。例えば、3 月 16 日に料金変更通知を送信したとします。6 月 16 日は 3 月 16 日から約 90 日後です。料金変更は 90 日間の期間の翌月の初日に行われるため、変更の発効日は 7 月 1 日です。

トピック

- [コンテナ料金モデル](#)
- [コンテナ製品の契約料金](#)

コンテナ料金モデル

AWS Marketplace には、コンテナ製品の複数の料金モデルがあります。

次の表に、コンテナベースの製品の料金モデルに関する一般的な情報を示します。

コンテナ製品の料金モデル

料金モデル	説明
Bring-Your-Own-License (BYOL)	BYOL は、購入者と維持する外部請求関係 AWS Marketplace を通じて の外部で管理されます。
月別	[月額固定料金] ユーザーに翌月中の無制限の製品使用を提供する固定月額料金 例: 製品に 1 か月あたり 99 USD の価格を設定します。製品には、Amazon ECS タスク定義を使用してデプロイされた 3 つの異なるコンテナイメージが含まれます。 購入者が製品をサブスクライブした後、すぐに 99 USD が課金され、サブスクリプションをキャンセルするまで毎月繰り返されます。また購入者は、製品の無制限の使用を取得します。また、購入者はタスクが実行されているすべてのインフラストラクチャで個別に支払いを行います。サブスクライブ中は、コンテナイメージにアクセスできます。これらのイメージから、任意の

料金モデル	説明
	設定の Amazon ECS または Amazon EKS で、任意の数のコンテナを起動して実行できます。 購入者が月の途中でサブスクリプションをキャンセルすると、 がコンテナイメージ AWS Marketplace を保存する Amazon ECR リポジトリにアクセスできなくなります。購入者が元のイメージを取り出して保存した可能性があります。ただし、 を通じて利用可能にした新しいコンテナイメージバージョンをプルすることはできなくなりました AWS Marketplace。購入者には、最終月の未使用分が返金されます。購入者の使用量から合意した AWS Marketplace 料金を引いた金額が支払われます。
カスタムメトリクス料金ディメンション	定義されたディメンション (ユーザー、ノード、リポジトリ、または GB など) に基づくカスタム従量制料金 (製品あたり最大 24 ディメンション)。 例: ユーザーによる製品課金。管理者ユーザーと通常ユーザーがあり、管理ユーザーの料金を 2 USD、通常ユーザーの料金を 1 USD と定義します。商品を出品する際にそれぞれに異なるディメンションを設定できます。1 日にログインしたユーザーごとに課金され、一日あたりの使用量が計測されます。

料金モデル	説明
タスクあたりまたはポッドあたりの時間単位料金	[Amazon ECS タスクまたは Amazon EKS ポッド] Amazon ECS タスクごとあるいは Amazon EKS ポッドごとに 1 時間あたりの料金設定による秒単位の測定。 例: 製品には、コントローラーノード、ワーカーノード、分析ノードの 3 つの異なるコンテナイメージがあります。コントローラーノードがなければ製品は機能しなかったり、役に立たなかったりするため、使用量を課金するイメージを決めます。1 時間あたり 6 USD の料金を設定します。 コントローラーノードのコンテナイメージでソフトウェアを変更して、AWS Marketplace Metering Service RegisterUsage API オペレーションと統合します。これにより、アクティブなサブスクリプションがある購入者だけが、そのコンテナイメージを起動して実行することができます。その使用量は実行された時間に基づいて計測されます。 購入者は、実行中の各 Amazon EKS コントローラーポッドの使用量について、1 時間あたり 6 USD が課金されます。購入者がコントローラーノードコンテナを含む 5 つの Amazon EKS コントローラポッドを起動すると、1 時間あたり 30 USD (ポッドあたり 6 USD) が課金されます。また、購入者は、ポッドが実行するインフラストラクチャに対して個別に支払いを行います。 時間単位の課金の場合、課金は秒単位で、最小は 1 分です。登録者がこのコントローラーコンテナを 20 分 30 秒実行すると、$20 \times (\\$6/60) + 30 \times (\\$6/60/60) = \\$2 + \\$0.05 = \\$2.05$ の料金が課金されます。購入者の使用量から合意した AWS Marketplace 料金を引いた金額が支払われます。

料金モデル	説明
時間単位/長期契約による使用量	割引価格による長期契約、前払いまたは定期的な分割払い。長期契約は、カスタム従量制料金が設定されている既存の製品、またはタスクおよびポッドごとに追加できます。購入者が長期契約による購入を超えた消費を行うと、従量制料金を支払います。
	従量制料金モデルでは、購入者に対して長期契約価格を追加して、前払いのコミットの割引を受けることができます。例えば、通常、消費したユニット数につき 1 USD を請求するとします。1 時間あたり 1 ユニットを使用する購入者は、年間 8760 USD (365 days x 24 hours x \$1 per hour) を支払うことになります。購入者がその 365 日間、1 時間あたり 1 ユニットを半額 (4380 USD) で使用できるようにする契約を有効にできます。この場合、購入者は 1 年契約の前払いを約束し、料金は 1 ユニットあたり 1 ドルから 0.5 ドルに下がります。また、購入者がこれらの契約を複数購入できるようにすることもできます。計測された数量から、購入者が 1 時間に 10 ユニットを消費したことが示され、2 つの契約を結んでいた場合、2 つの契約には 2 ユニットが含まれます。追加の 8 ユニットには、通常の 1 時間あたり 1 USD、その 1 時間の合計で 8 USD が請求されます。
	タスク単位またはポッド単位の例では、購入者向けに長期契約料金を追加して、前払いのコミットの割引を受けることができます。通常、ポッドあたり 6 USD を請求する場合は、長期契約期間を 365 日に 13,140 USD (365 days x 24 hours x \$3 per pod per hour) の料金に設定できます。そうすれば、1 つの契約で、その 365 日間、1 時間あたり 1 ポッドをお客様に提供することになります。お客様は複数の契約を購入することを選択できます。例えば、1 人のお客様が 2 つの契約を購入すると、1 時間に 2 つのポッドを利用できるようになります。お客様が 1 時間あたりに実行するポッドの数が、対象となる契約数よりも多い場合、超過分のポッドには通常の時間単価で請求されます。
	いずれの場合も、長期契約を購入する購入者は、1 回限りの支払いまたは定期的に予定されるこの先の支払いのどちらに対しても、前払い請求されます。また、購入者は計測率時に契約範囲を超えた追加使用量に対して請求されます。

料金モデル	説明
コンテナ契約料金表	契約料金のコンテナ - 購入者が前払い料金を支払うコンテナベースの製品。

コンテナ製品の契約料金

契約料金のコンテナベースの製品の場合、は、お客様とお客様の間の契約に基づいて、お客様が定義した支払いスケジュールに基づいて、お客様に前払いまたは AWS Marketplace 請求します。その時点で、それらのリソースの使用権限が付与されます。

料金を設定するには、顧客に提供する 1 つ以上の契約期間を選択します。契約期間ごとに異なる価格を入力できます。契約期間は 1 か月、12 か月、24 か月、36 か月です。プライベートオファーでは、カスタム期間を月単位 (最大 60 か月) で指定できます。

製品の価格を最もよく表すカテゴリを選択します。料金カテゴリは、AWS Marketplace ウェブサイトで顧客に表示されます。[帯域幅] (GB/s、MB/s)、[データ] (GB、MB、TB)、[ホスト]、[リクエスト]、[階層]、[ユーザー] のいずれかを選択できます。事前定義カテゴリのどれもニーズに適合しない場合は、汎用的な [単位] カテゴリを選択できます。

このオファーには最大 24 のディメンションを追加できます。各ディメンションには次のデータが必要です。

- 契約カテゴリ - 製品が契約料金に加えて消費量ベースの計測をサポートしている場合、契約カテゴリは製品の測定または計測に使用されます。消費ベースの料金設定のない契約製品の場合は、契約に含まれるディメンションのカテゴリに最も近いカテゴリを選択できます。契約内のディメンションの単位に似た値がない場合は、Units を選択してください。
- 契約単位 - 製品が消費量ベースの計測をサポートしている場合、契約単位はカテゴリと共に計測に使用されます。選択したカテゴリに基づいて、ディメンションに最も近い単位の値を 1 つ選択します。
- 契約ディメンションは複数購入が可能 - このフィールドは、オファーが階層型の料金設定のオファーなのか、複数のディメンションの購入を許可する非階層型のオファーなのかを示すために使用されます。

階層型オファー - 購入者がオファー内の利用可能なディメンションのうちの 1 つだけをサブスクライブできるようにします。階層型オファーのディメンションには数量という概念はありません。特定のディメンションで契約を結ぶことは、基本的に、購入者がそのディメンションが示す特定の機能を選択したことを意味します。

非階層型オファー - 顧客は契約の一環として複数のディメンションを調達でき、各ディメンションの複数のユニットを調達できます。

このフィールドの値を true に設定すると、そのオファーは非階層型オファーであることを示します。このフィールドに false を設定すると、そのオファーは階層型オファーであることを示します。

ウィザードを使用してコンテナ製品の契約を作成する場合、料金ディメンションに次のフィールドを定義する必要があります。

- Contracts DimensionX API Name – 購入者の AWS License Manager アカウントで生成されたライセンスに表示される名前。この名前は Checkoutlicense API コールの Entitlement Name の値としても使用されます。
- [契約 DimensionX 表示名] - AWS Marketplace ウェブサイトの製品詳細ページと調達ページに表示される、顧客向けのディメンション名です。わかりやすい名前を付けてください。名前の最大長は 24 文字です。リストが公開されたら、Name の値を変更することはできません。
- [契約 DimensionX 説明] - 特定のディメンションが提供する機能など、製品のディメンションに関する追加情報を提供する、顧客向けのディメンションの説明です。説明の最大長は 70 文字です。
- [契約 DimensionX 数量] - 商品の契約が修正された場合の比例配分の計算に使用されます。このフィールドのこの値は、すべての契約オファーで 1 に設定する必要があります。この値は編集しないでください。
- [契約 DimensionX 1 か月料金] - このディメンションの 1 か月分の使用権限に対して請求される契約レート。非階層型オファーの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは、小数点以下 8 桁がサポートされています。
- [契約 DimensionX 12 か月料金] - ディメンションに対する 12 か月間の使用権限に対して請求される契約レート。非階層型オファーの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは、小数点以下 8 桁がサポートされています。
- [契約 DimensionX 24 か月料金] - ディメンションに対する 24 か月間の使用権限に対して請求される契約レート。非階層型オファーの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは、小数点以下 8 桁がサポートされています。
- [契約 DimensionX 36 か月料金] - ディメンションに対する 36 か月間の使用権限に対して請求される契約レート。非階層型オファーの場合、このレートは調達されたディメンションの単位ごとに請求されます。このフィールドでは、小数点以下 8 桁がサポートされています。

例: データストレージアプリケーション

	1 か月料金	12 か月料金	24 か月料金	P36-month price
暗号化されていないデータ (GB)	1.50 USD/GB	16.00 USD/GB	30.00 USD/GB	60.00 USD/GB
暗号化されているデータ (GB)	1.55 USD/GB	16.60 USD/GB	31.20 USD/GB	61.20 USD/GB

例: ログモニタリング製品

	1 か月料金	12 か月料金	24 か月料金	36 か月料金
Basic (10 ホストのモニタリング、5 コンテナのモニタリング)	100 USD	1,000 USD	2,000 USD	4000 USD
Standard (20 ホストのモニタリング、10 コンテナのモニタリング)	200 USD	2,000 USD	4000 USD	8,000 USD
Pro (40 ホストのモニタリング、20 コンテナのモニタリング)	400 USD	4000 USD	8,000 USD	16,000 USD
1 時間あたりのモニタリング対象追加ホスト	10 USD	100 USD	\$200	400 USD
1 時間あたりのモニタリング対象追加コンテナ	10 USD	100 USD	\$200	400 USD

Note

価格は、1 か月、12 か月、24 か月、36 か月の期間に対して定義できます。製品にこれらのオプションを 1 つ以上提供することができます。期間は各ディメンション間で同じである必要があります。

Example

たとえば、ReadOnlyUsers ディメンションと AdminUsers ディメンションがある場合、ReadOnlyUsers に年間料金を提供する場合は、AdminUsers の年間料金も提供する必要があります。

自動更新

顧客は、コンテナ契約 AWS Marketplace を使用して を通じて製品を購入すると、契約条件を自動的に更新することに同意できます。顧客は 1 年、2 年、または 3 年の期間中に、使用権限に対して毎月支払い続けます。

お客様はいつでも更新の設定を変更できます。詳細については、「AWS Marketplace 購入者ガイド」の「[既存の契約の変更](#)」を参照してください。

コンテナ契約が終了したとき

コンテナ契約製品には、契約の有効期限があります。契約が終了すると、次のイベントが発生します。

1. コンテナ製品は、購入者の使用権限が変更されたことを示す entitlement-updated 通知を受け取り、は空のレスポンス AWS Marketplace Entitlement Service を返します。
2. その顧客の残り使用量を計測するまでに 1 時間の猶予があります。このステップの後、この顧客の計測レコードを送信することはできません。

コンテナ製品の請求、計測、ライセンスの統合

AWS Marketplace は他の と統合 AWS のサービスとして、コンテナ製品の計測と契約ベースの料金の両方を提供します。使用料金を使用したコンテナベースの製品の場合、製品を使用する使用権限の確認および請求用使用計測の両方に対して、[AWS Marketplace Metering Service](#) を使用します。契約

料金のコンテナベースの製品の場合、を使用してライセンス AWS License Manager を製品に関連付けることができます。以下のセクションでは、AWS License Managerでの時間単位およびカスタム計測と AWS Marketplace Metering Service での契約料金について詳述します。

トピック

- [AWS Marketplace Metering Serviceによる時間単位計測およびカスタム計測](#)
- [との契約料金 AWS License Manager](#)
- [AWS Marketplace Metering Service による時間単位の計測を設定する](#)
- [AWS Marketplace Metering Service を使用したコンテナ製品のカスタム計測の設定](#)
- [を使用したコンテナ製品の契約料金 AWS License Manager](#)

AWS Marketplace Metering Serviceによる時間単位計測およびカスタム計測

製品を使用する使用権限および請求用使用計測の権限の両方を確認するには、[AWS Marketplace Metering Service](#) を使用します。各自で料金単位を決定し、請求用に使用量を計測するには、[MeterUsage](#) API オペレーションを使用して統合します。使用したタスクまたはポッドの数に基づいて製品の料金を設定し、その使用量を自動的に計測する場合は、[RegisterUsage](#) API AWS オペレーションを使用して統合します。どちらのタイプの料金設定でも、AWS Marketplace Metering Serviceとの統合を変更する必要なしで長期契約料金を追加できます。

で新しいコンテナ製品を作成すると AWS Marketplace 管理ポータル、製品を と統合するために使用される一連の製品識別子 (製品コードとパブリックキー) が提供されます AWS Marketplace Metering Service。

エンタイトルメント

と を統合する AWS Marketplace Metering Service ことで、有料ソフトウェアを実行している顧客が で製品をサブスクライブしていることを検証し AWS Marketplace、コンテナの起動時の不正使用を防ぐことができます。使用権限を確認するには、料金モデルに応じて、[MeterUsage](#) API オペレーションまたは [RegisterUsage](#) API オペレーションを使用します。時間単位および固定月単位の料金モデルの場合は、RegisterUsage API オペレーションを使用します。カスタム計量料金モデルの場合は、MeterUsage API オペレーションを使用します。

購入者に商品の購入資格がない場合、これらの API オペレーションは CustomerNotEntitledException 例外を返します。

Note

購入者が製品を実行中に登録を解除した場合、購入者にはその製品を実行し続ける権利が与えられます。ただし、製品の追加のコンテナを起動することはできません。

統合ガイドライン

コンテナ製品を作成して公開し、使用権限と計測に MeterUsage API オペレーションあるいは RegisterUsage API オペレーションを使用する際は、以下のガイドラインに留意してください。

- ソフトウェアまたは Docker コンテナイメージ内で AWS 認証情報を設定しないでください。購入者の AWS 認証情報は、コンテナイメージが Amazon ECS タスクまたは Amazon EKS ポッド内で実行されているときに、実行時に自動的に取得されます。
- Amazon EKS から MeterUsage または RegisterUsage API オペレーションを呼び出すには、[サポートされている AWS SDK を使用する必要があります](#)。Amazon EKS の MeterUsage あるいは RegisterUsage 統合をテストするには、Kubernetes 1.13.x 以降が搭載されている Amazon EKS クラスターを実行する必要があります。ポッドサポートの AWS Identity and Access Management (IAM) ロールには Kubernetes 1.13 が必要です。実行中のポッドが Amazon EKS でこれらのアクションを呼び出すために必要な AWS 認証情報を取得するには、IAM ロールが必要です。
- ローカルな開発を実行できますが、PlatformNotSupportedException 例外が発生します。この例外は、コンテナサービス (Amazon ECS、Amazon EKS、Fargate) で AWS コンテナを起動しても発生しません。

サポートされている AWS リージョン

AWS Marketplace サポートされているすべての のリストについては AWS リージョン、グローバルインフラストラクチャウェブサイトの [「リージョンテーブル」](#) を参照してください。

計測 AWS リージョン 用の の取得

計測用のコンテナを MeterUsage または RegisterUsage API オペレーションと統合する場合は、特定の を使用するよう AWS SDK を設定しないでください AWS リージョン。リージョンは実行時に動的に取得する必要があります。

Example

例えば、顧客が Amazon ECS タスクまたは Amazon EKS ポッドを起動したとします。RegisterUsage API オペレーションは、Amazon ECS タスクまたは Amazon EKS ポッドが起

動されたリージョンとは異なるリージョンで呼び出されます。そのため、RegisterUsage API オペレーションでは InvalidRegionException エラーが発生します。

AWS SDK 言語は、AWS_REGION一貫した方法で を決定しません。SDK が自動的に AWS_REGION に対応しない場合は、ソフトウェアを手動で作成して AWS_Region を判断する必要があります。例えば、環境変数や他の設定が存在しない場合、AWS SDK for Java は [Amazon EC2 インスタンスメタデータ](#) (特に ec2InstanceMetadata) を自動的に使用してリージョンを取得します。このインスタンスでは、AWS_REGION 環境変数が存在しない場合にのみ ec2InstanceMetadata を呼び出します。

ランタイム AWS リージョン に を動的に取得する方法については、プログラミング言語の [AWS SDK デベロッパーガイド](#)を参照してください。

計測変更の防止

RegisterUsage または MeterUsage への呼び出しを変更あるいは上書きする手段を購入者に提供すると、望ましくない請求および支払い問題が発生することがあります。計測および使用権限ロジックを統合することを強くお勧めします。

計測の変更を防ぐように製品を構築する場合は、次の点に留意してください。

- 購入者が CMD または ENTRYPOINT 説明を含む新しいイメージレイヤーを挿入できる場合、コンテナイメージを介して購入者が実行しているソフトウェアに RegisterUsage または MeterUsage を直接統合します。そうしないと、ベースイメージからの CMD または ENTRYPOINT を介した RegisterUsage または MeterUsage への呼び出しが購入者によって上書きされる可能性が高まります。
- ソフトウェアが入力として使用する AWS Marketplace 製品コード、RegisterUsageまたは購入者が変更できないMeterUsage方法で管理することをお勧めします。ただし、AWS CloudFormation Helm チャート、Kubernetes マニフェストなど、顧客が上書きできる方法で製品が製品コードを管理する場合は、信頼できる AWS Marketplace 製品コードのリストを維持する必要があります。これは、ソフトウェアが入力として RegisterUsage または MeterUsage に渡される製品コードが有効であることを確認するためです。
- 信頼される製品コードのいずれかが無料製品用の場合、その製品コードが有料製品のコードの代わりに使用できないように確認します。

との契約料金 AWS License Manager

契約料金のコンテナベースの製品の場合、を使用してライセンス AWS License Manager を製品に関連付けます。

AWS License Manager は、顧客が購入したライセンス (使用権限とも呼ばれます) をアプリケーションが追跡および更新できるようにするライセンス管理ツールです。このセクションでは、製品についての情報を提供します。このセクションでは、製品を AWS License Manager と統合する方法について説明します。統合が完了したら、AWS Marketplace に製品リストを公開できます。

詳細については AWS License Manager、[AWS License Manager 「ユーザーガイド」](#) および AWS CLI 「コマンドリファレンス」の[AWS License Manager 「」](#) セクションを参照してください。

Note

- 契約の有効期限を過ぎると、顧客はコンテナの新しいインスタンスを起動できなくなります。ただし、契約期間中は、インスタンスをいくつでも起動できます。これらのライセンスは特定のノードやインスタンスには適用されません。AWS 認証情報が割り当てられている限り、どのノードのどのコンテナで実行されているソフトウェアでもライセンスをチェックアウトできます。
- [プライベートオファーの作成] - 販売者は、AWS Marketplace 管理ポータルプライベートオファー作成ツールを使用して、製品のプライベートオファーを作成できます。
- [レポート] - AWS Marketplace 管理ポータルの [レポート] セクションで Amazon S3 バケットを設定して、データフィードを設定できます。詳細については、「[の販売者レポート、データフィード、ダッシュボード AWS Marketplace](#)」を参照してください。

ワークフロー統合

以下のステップは、コンテナ製品を AWS License Manager と統合するワークフローを示しています。

1. 販売者は、AWS License Manager 統合を使用して製品を作成します。
2. 販売者は製品を一覧表示します AWS Marketplace。
3. 購入者は製品を見つけ AWS Marketplace で購入します。
4. ライセンスは、AWS アカウントの購入者に送付されます。

- 購入者は Amazon EC2 インスタンス、Amazon ECS タスク、Amazon EKS ポッドソフトウェアを起動してソフトウェアを使用します。顧客は IAM ロールを使用してデプロイします。
- ソフトウェアは購入者の AWS License Manager アカウントでライセンスを読み取り、購入した使用権限を検出し、それに応じて機能をプロビジョニングします。

 Note

License Manager は追跡や更新を行いません。これは販売者のアプリケーションによって行われます。

AWS Marketplace Metering Service による時間単位の計測を設定する

 Note

Amazon EKS デプロイの場合、ソフトウェアは [サービスアカウント \(IRSA\) の IAM ロール](#) を使用して API オペレーションの [RegisterUsage](#) API コールに署名する必要があります。 [EKS Pod Identity](#)、ノードロール、または長期アクセスキーの使用はサポートされていません。

Amazon ECS デプロイの場合、ソフトウェアは [Amazon ECS タスク IAM](#) ロールを使用して API オペレーションの [RegisterUsage](#) API コールに署名する必要があります。ノードロールまたは長期アクセスキーの使用はサポートされていません。

コンテナ製品でカスタム従量制料金ディメンションの代わりに時間単位、タスク単位またはポッド単位の料金を使用する場合、カスタム計測ディメンションを定義する必要はありません。AWS Marketplace Metering Service を使用すると AWS Marketplace でコンテナ製品を時間単位で計測できます。以下のセクションでは、AWS Marketplace Metering Service を使用した時間単位の計測を設定する方法を説明します。

RegisterUsage API オペレーションは、Amazon Elastic Container Service (Amazon ECS) タスクまたは Amazon Elastic Kubernetes Service (Amazon EKS) ポッドごとに、1 時間あたりのソフトウェア使用量を測定します。使用量は、秒単位で比例配分されます。存続期間の短いタスクやポッドには、最低 1 分の使用量が適用されます。ソフトウェア使用のための継続的な計測は、によって自動的に処理されます AWS Marketplace Metering Control Plane。ソフトウェアでは、開始に使用するソフトウェアの計測のために RegisterUsage を 1 回呼び出すこと以外に、計測に固有のアクションを実行する必要はありません。

RegisterUsage はコンテナを起動したらずに呼び出す必要があります。コンテナの起動から最初の 6 時間以内にコンテナを登録しない場合、AWS Marketplace Metering Service は前月分の計測保証を提供しません。ただし、計測は当月以降、コンテナが終了するまで継続されます。

は AWS Marketplace Metering Control Plane、顧客のサブスクリプション状態に関係なく、Amazon ECS タスクと Amazon EKS ポッドの実行に対して引き続き顧客に請求します。これにより、タスクまたはポッドが最初に正常に起動した後に、ソフトウェアが使用権限チェックを行う必要がなくなります。

AWS Marketplace Metering Service API とコンテナ製品を時間単位の料金で統合する方法の詳細については、AWS Marketplace 販売者ワークショップの「[時間単位の計測ラボとの統合](#)」を参照してください。

トピック

- [時間単位の計測の前提条件](#)
- [の統合のテスト RegisterUsage](#)
- [RegisterUsage でのエラー処理](#)
- [を使用してコンテナ製品を AWS Marketplace Metering Service と統合する AWS SDK for Java](#)

時間単位の計測の前提条件

製品を公開する前に、次を行う必要があります。

1. で新しいコンテナ製品を作成し AWS Marketplace 管理ポータル、その製品コードを書き留めます。

詳細については、「[概要: コンテナ製品を作成する](#)」を参照してください。

2. (AWS Identity and Access Management IAM) ロールは、 を呼び出すために必要な IAM アクセス許可を持つアプリケーションを実行しているタスクまたはポッドに使用します RegisterUsage。IAM マネージドポリシー AWSMarketplaceMeteringRegisterUsage には、これらのアクセス許可があります。ポリシーの詳細については、「AWS マネージドポリシーリファレンス」の[AWSMarketplaceMeteringFullAccess](#)」を参照してください。
3. (オプション) ログ記録を表示する場合は、タスクまたはポッド定義で AWS CloudTrail ログ記録を有効にすることをお勧めします。
4. 定義したすべての料金ディメンションのレコードを使用して、RegisterUsage API オペレーションをテストコールします。

の統合のテスト RegisterUsage

RegisterUsage API オペレーションを使用して、イメージを に送信して AWS Marketplace 公開する前に統合をテストします。

Amazon ECS または Amazon EKS で製品を実行して、コンテナイメージ RegisterUsage から呼び出します。製品を一覧表示するために使用している AWS アカウントを使用します AWS Marketplace。計測統合では、ハードコーディングするのではなく AWS リージョン、 を動的に設定する必要があります。ただし、テストするときは、米国東部 (バージニア北部) リージョンで、有料コンテナを含む Amazon ECS タスクまたは Amazon EKS ポッドを少なくとも 1 つ起動してください。これにより、AWS Marketplace 運用チームは、そのリージョンのログでの作業を検証できます。

Note

製品が Amazon ECS と Amazon EKS の両方をサポートしている場合は、Amazon EKS を起動するだけで、統合が検証されます。

製品および必要なすべてのメタデータと料金情報が公開されるまでは、統合を完全にテストすることはできません。リクエストされた場合、AWS Marketplace カタログオペレーションチームは計測レコードの受信を確認できます。

RegisterUsage でのエラー処理

コンテナイメージが と統合 AWS Marketplace Metering Service され、コンテナの起動 ThrottlingException 時以外の例外が発生した場合は、不正使用を防ぐためにコンテナを終了する必要があります。

ThrottlingException 以外の例外は、RegisterUsage API オペレーションへの最初の呼び出しでのみスローされます。同じ Amazon ECS タスクまたは Amazon EKS ポッドの以降の呼び出しでは、タスクまたはポッドの実行中に顧客がサブスクライブを解除した場合でも、CustomerNotSubscribedException はスローされません。この顧客は、サブスクリプションを解除した後で使用状況が追跡されると、実行中のコンテナに対して課金されます。

次の表は、RegisterUsage API オペレーションがスローする可能性があるエラーを説明しています。各 AWS SDK プログラミング言語には、追加情報を参照できる一連のエラー処理ガイドラインがあります。

[エラー]	説明
<code>InternalServerErrorException</code>	<code>RegisterUsage</code> は使用できません。
<code>CustomerNotEntitledException</code>	製品の有効なサブスクリプションが顧客にありません。
<code>InvalidProductCodeException</code>	リクエストの一部として渡された <code>ProductCode</code> 値が存在しません。
<code>InvalidPublicKeyException</code>	リクエストの一部として渡された <code>PublicKeyVersion</code> 値が存在しません。
<code>PlatformNotSupportedException</code>	AWS Marketplace は、基盤となるプラットフォームからの計測使用量をサポートしていません。Amazon ECS、Amazon EKS、およびのみがサポート AWS Fargate されています。
<code>ThrottlingException</code>	<code>RegisterUsage</code> の呼び出しが抑制されます。
<code>InvalidRegionException</code>	<code>RegisterUsage</code> は、Amazon ECS タスクまたは Amazon EKS ポッドが起動された AWS リージョン のと同じ で呼び出される必要があります。これにより、 <code>RegisterUsage</code> を呼び出す際にコンテナがリージョン (<code>withRegion("us-east-1")</code> など) を選択することを防ぎます。

を使用してコンテナ製品を AWS Marketplace Metering Service と統合する AWS SDK for Java

を使用して AWS SDK for Java、AWS Marketplace Metering Service と統合できます。ソフトウェア使用のための継続的な計測は、によって自動的に処理されます AWS Marketplace Metering Control Plane。ソフトウェアでは、開始に使用するソフトウェアの計測のために `RegisterUsage` を 1 回呼び出すこと以外に、計測に固有のアクションを実行する必要はありません。このトピック

では、を使用して [AWS Marketplace Metering Service](#) の RegisterUsage アクションと統合 AWS SDK for Java する実装例を示します。

RegisterUsage はコンテナを起動したらすぐに呼び出す必要があります。コンテナの起動から最初の 6 時間以内にコンテナを登録しない場合、AWS Marketplace Metering Service は前月分の計測保証を提供しません。ただし、計測は当月以降、コンテナが終了するまで継続されます。

完全なソースコードについては、「[RegisterUsage Java の例](#)」を参照してください。これらのステップの多くは、AWS SDK 言語に関係なく適用されます。

AWS Marketplace Metering Service 統合の手順の例

1. [AWS Marketplace 管理ポータル](#) にサインインします。
2. [Assets (アセット)] から [コンテナ] を選択して、新しいコンテナ製品の作成を開始します。製品を作成すると、製品とコンテナイメージを統合するための製品コードが生成されます。IAM アクセス許可の設定については、「[the section called “AWS Marketplace 計測と使用権限 API アクセス許可”](#)」を参照してください。
3. 公開 [AWS Java SDK](#) をダウンロードします。

Important

Amazon EKS から計測 APIs を呼び出すには、[サポートされている AWS SDK を使用し](#)、Kubernetes 1.13 以降を実行している Amazon EKS クラスターでを実行する必要があります。

4. (オプション) RegisterUsage アクションを統合し、デジタル署名検証を実行する場合は、アプリケーションのクラスパスで [BouncyCastle](#) 署名検証ライブラリを設定する必要があります。

JSON ウェブトークン (JWT) を使用する場合は、アプリケーションのクラスパスに [JWT Java](#) ライブラリも含める必要があります。JWT を使用することで署名検証により簡単にアプローチできますが、必須ではありません。代わりにスタンドアロン BouncyCastle を使用できます。JWT を使用するか BouncyCastle を使用するにかかわらず、アプリケーションのクラスパスに BouncyCastle または JWT の推移従属性を含めるには、Maven などのビルドシステムを使用する必要があります。

```
// Required for signature verification using code sample
<dependency>
  <groupId>org.bouncycastle</groupId>
  <artifactId>bcpkix-jdk15on</artifactId>
```

```

    <version>1.60</version>
  </dependency>

  // This one is only required for JWT
  <dependency>
    <groupId>com.nimbusds</groupId>
    <artifactId>nimbus-jose-jwt</artifactId>
    <version>6.0</version>
  </dependency>

```

5. 製品提供の各有料コンテナイメージから RegisterUsage を呼び出します。ProductCode と PublicKeyVersion は必須パラメータとなり、その他の入力はすべてオプションです。RegisterUsage のペイロード例を次に示します。

```

{
  "ProductCode" : "string", // (required)
  "PublicKeyVersion": 1,    // (required)
  "Nonce": "string",        // (optional) to scope down the registration
                             //                to a specific running software
                             //                instance and guard against
                             //                replay attacks
}

```

Note

AWS Marketplace 計測サービスへの接続に一時的な問題が発生する可能性があります。AWS Marketplace では、短期的な停止やネットワークの問題を避けるため、指数関数的なバックオフを含めて最大 30 分間の再試行を実施することを強くお勧めします。

6. RegisterUsage は、リクエストの信頼性の検証に使用できる SHA-256 を使用して、RSA-PSS デジタル署名を生成します。署名には、ProductCode、PublicKeyVersion、Nonce のフィールドが含まれています。デジタル署名を検証するには、リクエストからこれらのフィールドを保持する必要があります。次のコードは、RegisterUsage 呼び出しに対するレスポンスの例です。

```

{
  "Signature": "<<JWT Token>>"
}

// Where the JWT Token is composed of 3 dot-separated,
// base-64 URL Encoded sections.

```

```
// e.g. eyJhbGcVCJ9.eyJzdWIMzkwMjJ9.rr09Qw0SXRWTe

// Section 1: Header/Algorithm
{
  "alg": "PS256",
  "typ": "JWT"
}

// Section 2: Payload
{
  "ProductCode" : "string",
  "PublicKeyVersion": 1,
  "Nonce": "string",
  "iat": date // JWT issued at claim
}

// Section 3: RSA-PSS SHA256 signature
"rr09Q4FEi3gweH3X4lrt2okf5zwIatUUwERlw016wTy_21Nv8S..."
```

7. RegisterUsage 呼び出しを含むコンテナイメージを再構築し、コンテナにタグ付けして、Amazon ECR または Amazon ECR Public など、Amazon ECS または Amazon EKS と互換性がある任意のコンテナレジストリにプッシュします。Amazon ECR を使用している場合は、Amazon ECS タスクまたは Amazon EKS ポッドを起動するアカウントに、Amazon ECR リポジトリへのアクセス許可があることを確認します。そうしないと、起動は失敗します。
8. 次のコードで定義されているように、コンテナが RegisterUsage を呼び出すアクセス許可を付与する [IAM](#) ロールを作成します。Amazon ECS タスクまたは Amazon EKS ポッド定義の [タスクロール](#) パラメータでこの IAM ロールを指定する必要があります。

JSON

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

}

9. と統合されたコンテナを参照 AWS Marketplace し、ステップ 7 で作成した IAM ロールを参照する Amazon ECS タスクまたは Amazon EKS ポッド定義を作成します。AWS CloudTrail ログ記録を表示するには、タスク定義でログ記録を有効にする必要があります。
10. Amazon ECS または Amazon EKS クラスターを作成して、タスクまたはポッドを実行します。Amazon ECS クラスターの作成の詳細については、「Amazon Elastic Container Service デベロッパーガイド」の「[クラスターの作成](#)」を参照してください。Amazon EKS クラスターの作成 (Kubernetes バージョン 1.1.3.x 以降を使用) の詳細については、「[Amazon EKS クラスターの作成](#)」を参照してください。
11. Amazon ECS または Amazon EKS クラスターを設定し、us-east-1 で作成した Amazon ECS タスク定義または Amazon EKS ポッドを起動します AWS リージョン。製品が本番稼働になる前のこのテストプロセス中にのみ、このリージョンを使用する必要があります。
12. RegisterUsage から有効なレスポンスを取得すると、コンテナ製品の作成を開始することができます。ご質問がある場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

RegisterUsage Java の例

次の例では、AWS SDK for Java および AWS Marketplace Metering Service を使用して RegisterUsage オペレーションを呼び出します。署名の検証はオプションですが、署名の検証を実行する場合は、必要なデジタル署名検証ライブラリを含める必要があります。この例は、例示のみを目的としています。

```
import com.amazonaws.auth.PEM;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMetering;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMeteringClientBuilder;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageRequest;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageResult;
import com.amazonaws.util.json.Jackson;
import com.fasterxml.jackson.databind.JsonNode;
import com.nimbusds.jose.JWSObject;
import com.nimbusds.jose.JWSVerifier;
import com.nimbusds.jose.crypto.RSASSAVerifier;
import java.io.ByteArrayInputStream;
import java.nio.charset.StandardCharsets;
import java.security.PublicKey;
import java.security.Security;
import java.security.Signature;
```



```
import java.security.interfaces.RSAPublicKey;
import java.util.Base64;
import java.util.Optional;
import java.util.UUID;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

/**
 * Class for making calls out to &MKT; Metering Service.
 */
class RegisterUsage {

    private static final String PRODUCT_CODE = ".....";

    private final AWSMarketplaceMetering registerUsageClient;
    private final SignatureVerifier signatureVerifier;
    private final int publicKeyVersion;

    public RegisterUsage(final SignatureVerifier signatureVerifier) {
        this.signatureVerifier = signatureVerifier;
        this.publicKeyVersion = PublicKeyProvider.PUBLIC_KEY_VERSION;
        this.registerUsageClient =
AWSMarketplaceMeteringClientBuilder.standard().build();
    }

    /**
     * Shows how to call RegisterUsage client and verify digital signature.
     */
    public void callRegisterUsage() {
        RegisterUsageRequest request = new RegisterUsageRequest()
            .withProductCode(PRODUCT_CODE)
            .withPublicKeyVersion(publicKeyVersion)
            .withNonce(UUID.randomUUID().toString());

        // Execute call to RegisterUsage (only need to call once at container startup)
        RegisterUsageResult result = this.registerUsageClient.registerUsage(request);

        // Verify Digital Signature w/o JWT
        boolean isSignatureValid = this.signatureVerifier.verify(request, result);
        if (!isSignatureValid) {
            throw new RuntimeException("Revoke entitlement, digital signature
invalid.");
        }
    }
}
```

```
/**
 * Signature verification class with both a JWT-library based verification
 * and a non-library based implementation.
 */
class SignatureVerifier {
    private static BouncyCastleProvider BC = new BouncyCastleProvider();

    private static final String SIGNATURE_ALGORITHM = "SHA256withRSA/PSS";

    private final PublicKey publicKey;

    public SignatureVerifier(PublicKeyProvider publicKeyProvider) {
        this.publicKey = publicKeyProvider.getPublicKey().orElse(null);
        Security.addProvider(BC);
    }

    /**
     * Example signature verification using the NimbusJOSEJWT library to verify the JWT
     * Token.
     *
     * @param request RegisterUsage Request.
     * @param result RegisterUsage Result.
     * @return true if the token matches.
     */
    public boolean verifyUsingNimbusJOSEJWT(final RegisterUsageRequest request, final
RegisterUsageResult result) {
        if (!getPublicKey().isPresent()) {
            return false;
        }

        try {
            JWSVerifier verifier = new RSASSAVerifier((RSAPublicKey)
getPublicKey().get());
            JWSObject jwsObject = JWSObject.parse(result.getSignature());
            return jwsObject.verify(verifier) &&
validatePayload(jwsObject.getPayload().toString(), request, result);
        } catch (Exception e) {
            // log error
            return false;
        }
    }
}

/**
```

```

    * Example signature verification without any JWT library support.
    *
    * @param request RegisterUsage Request.
    * @param result RegisterUsage Result.
    * @return true if the token matches.
    */
    public boolean verify(final RegisterUsageRequest request, final RegisterUsageResult
result) {
        if (!getPublicKey().isPresent()) {
            return false;
        }
        try {
            String[] jwtParts = result.getSignature().split("\\.");
            String header = jwtParts[0];
            String payload = jwtParts[1];
            String payloadSignature = jwtParts[2];

            Signature signature = Signature.getInstance(SIGNATURE_ALGORITHM, BC);
            signature.initVerify(getPublicKey().get());
            signature.update(String.format("%s.%s", header,
payload).getBytes(StandardCharsets.UTF_8));
            boolean verified = signature.verify(Base64.getUrlDecoder()
                .decode(payloadSignature.getBytes(StandardCharsets.UTF_8)));

            String decodedPayload = new String(Base64.getUrlDecoder().decode(payload));
            return verified && validatePayload(decodedPayload, request, result);
        } catch (Exception e) {
            // log error
            return false;
        }
    }

    /**
     * Validate each value in the returned payload matches values originally
     * supplied in the request to RegisterUsage. TimeToLiveInMillis and
     * PublicKeyExpirationTimestamp will have the values in the payload compared
     * to values in the signature
     */
    private boolean validatePayload(final String payload, final RegisterUsageRequest
request,
                                   final RegisterUsageResult result) {
        try {
            JsonNode payloadJson = Jackson.get.ObjectMapper().readTree(payload);
            boolean matches = payloadJson.get("productCode")

```

```

        .asText()
        .equals(request.getProductCode());
    matches = matches && payloadJson.get("nonce")
        .asText()
        .equals(request.getNonce());
    return matches = matches && payloadJson.get("publicKeyVersion")
        .asText()
        .equals(String.valueOf(request.getPublicKeyVersion()));

    } catch (Exception ex) {
        // log error
        return false;
    }
}

private Optional<PublicKey> getPublicKey() {
    return Optional.ofNullable(this.publicKey);
}
}

/**
 * Public key provider taking advantage of the &AWS; PEM Utility.
 */
class PublicKeyProvider {
    // Replace with your public key. Ensure there are new-lines ("\n") in the
    // string after "-----BEGIN PUBLIC KEY-----\n" and before "\n-----END PUBLIC
    KEY-----".
    private static final String PUBLIC_KEY =
        "-----BEGIN PUBLIC KEY-----\n"
            + "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDdlatRjRjogo3WojgGHFHYLugd
\n"
            + "UWAY9iR3fy4arWNA1KoS8kVw33cJibXr8bvWUAUparCwlvdBh6dvE0fou0/gCFQs
\n"
            + "HUFQrSDv+MuSUMAe8jzKE4qW+jK+xQU9a03GUnKHkkle+Q0pX/g6jXZ7r1/xAK5D
\n"
            + "o2kQ+X5xK9cipRgEKwIDAQAB\n"
            + "-----END PUBLIC KEY-----";

    public static final int PUBLIC_KEY_VERSION = 1;

    public Optional<PublicKey> getPublicKey() {
        try {
            return Optional.of(PEM.readPublicKey(new ByteArrayInputStream(
                PUBLIC_KEY.getBytes(StandardCharsets.UTF_8))));
        }
    }
}

```

```
    } catch (Exception e) {  
        // log error  
        return Optional.empty();  
    }  
}  
}
```

AWS Marketplace Metering Service を使用したコンテナ製品のカスタム計測の設定

Note

Amazon EKS デプロイの場合、ソフトウェアは[サービスアカウント \(IRSA\) の IAM ロール](#)を使用して API オペレーションの [MeterUsage](#) API コールに署名する必要があります。[EKS Pod Identity](#)、ノードロール、または長期アクセスキーの使用はサポートされていません。Amazon ECS デプロイの場合、ソフトウェアは [Amazon ECS タスク IAM](#) ロールを使用して API オペレーションの [MeterUsage](#) API コールに署名する必要があります。ノードロールまたは長期アクセスキーの使用はサポートされていません。

AWS Marketplace コンテナ製品では、製品ごとに最大 24 の異なる料金ディメンションでカスタム計測を行うことができます。各ディメンションには、長期契約料金を関連付けることができます。カスタム計測を有効にするには、コンテナ製品を AWS Marketplace Metering Service と統合してください。[MeterUsage](#) API オペレーションを使用して、請求 AWS のために への使用量の独自の料金単位とカスタム計測を定義できます。以下のセクションでは、コンテナ製品のカスタム計測を設定する方法を示します。

料金ディメンションは 2 つの場所で定義され、1 つは AWS Marketplace 管理ポータル（販売者ポータル）で製品を作成するとき、もう 1 つはソフトウェアで MeterUsage オペレーションを実行します。この 2 つの要素方法により、後続のオファーが一般公開される前に意図したとおりに作動することを確実にします。

カスタム計測を設定するには、使用カテゴリ、単位タイプ、料金ディメンションを選択する必要があります。

- 使用カテゴリ - 使用カテゴリは購入者が製品の内容とその使用方法を理解するために役立ちます。
- 単位タイプ - 単位タイプは請求の測定単位を定義します。例えば、GBps または MBps で測定された帯域幅、ホスト数、MB、GB、TB で測定されたデータなど。

- **料金ディメンション** - 料金ディメンションは、個数ごとの料金を設定した機能またはサービス (ユーザー、スキャン、vCPU、デプロイされたエージェントなど) を表します。料金設定ディメンションは公開されています。ただし、公開されている製品については、引き続きプライベートオファアと Bring Your Own License (BYOL) を定義できます。計測レコードに料金設定を送信しないでください。単位の数量を計測すると、製品を作成するときに定義した料金と共にこれを使用して、購入者の請求書を計算します。

製品の料金設定が事前定義されたカテゴリまたは単位タイプに適合しない場合は、汎用 [単位] カテゴリを選択できます。次に、ディメンションの説明を使用して単位が何であることを説明します。

オプションで、追跡するプロパティごとに使用量を配分することもできます。割り当ては購入者にはタグとして表示されます。これらのタグにより、購入者はコストをタグ値別に使用量に分けて表示できます。例えば、ユーザーごとに課金し、ユーザーに「Department」プロパティがある場合、「Department」のキーを持つタグを使用して使用量の割り当てを作成し、値ごとに 1 つの割り当てを行うことができます。これによってレポートする料金、サイズ、総使用量は変更されませんが、顧客は製品に適したカテゴリ別にコストを確認できます。

1 時間ごとに計測レコードを送信することをお勧めします。ただし、日単位または月単位の使用量を集計することもできます。停止が発生した場合は、購入者ソフトウェアの使用を集約し、次の時間帯に送信できます。1 時間に複数のレコードを送信することはできません。

コンテナ製品の AWS Marketplace Metering Service API をカスタム計測料金と統合する方法の詳細については、AWS Marketplace 販売者ワークショップの「カスタム[計測ラボとの統合](#)」を参照してください。

Important

無料トライアルと前払いの使用権限は、時間単位で追跡されます。その結果、これらのレコードを別々に送信すると、購入者に過大請求が発生する可能性があります。

トピック

- [カスタム計測の前提条件](#)
- [MeterUsage 統合のテスト](#)
- [MeterUsage でのエラー処理](#)
- [ベンダー計測タグ付け \(オプション\)](#)
- [コード例](#)

- [カスタム計測を使用したコンテナ製品の AWS Marketplace Metering Service および との統合](#)
[AWS SDK for Java](#)

カスタム計測の前提条件

製品を公開する前に、次を行う必要があります。

1. で新しいコンテナ製品を作成し AWS Marketplace 管理ポータル、その製品コードを書き留めます。
2. (AWS Identity and Access Management IAM) ロールは、 を呼び出すために必要な IAM アクセス許可を持つアプリケーションを実行しているタスクまたはポッドに使用します MeterUsage。IAM マネージドポリシー AWSMarketplaceMeteringRegisterUsage には、これらのアクセス許可があります。ポリシーの詳細については、「AWS マネージドポリシーリファレンス」の [AWSMarketplaceMeteringFullAccess](#) を参照してください。
3. (オプション) AWS CloudTrail ログ記録を表示するには、タスクまたはポッド定義でログ記録を有効にすることをお勧めします。
4. 定義したすべての料金ディメンションのレコードを使用して、MeterUsage API オペレーションをテストコールします。

MeterUsage 統合のテスト

にイメージを送信して公開する前に、AWS Marketplace MeterUsage オペレーションを使用して統合をテストします。

製品を一覧表示するために使用する AWS アカウント を使用して、Amazon Elastic Container Service (Amazon ECS) または Amazon Elastic Kubernetes Service (Amazon EKS) で製品を実行して、コンテナイメージ MeterUsage から を呼び出します AWS Marketplace。計測統合では、ハードコーディングではなく AWS リージョン、 を動的に設定する必要があります。ただし、テスト時に、米国東部 (バージニア北部) リージョンで有料コンテナを含む Amazon ECS タスクまたは Amazon EKS ポッドを少なくとも 1 つ起動して、AWS Marketplace オペレーションチームがそのリージョンのログで作業を検証できるようにします。

Note

- 製品が Amazon ECS と Amazon EKS の両方をサポートしている場合は、Amazon EKS を起動するだけで、統合が検証されます。

- 製品を一般公開する前に、および新しいディメンションを追加した後に、すべてのディメンションをテストします。コンテナ製品に関連付けられた各ディメンションの計測レコードを送信しない場合、リクエストが失敗するエラーが発生します。

製品および必要なすべてのメタデータと料金情報が公開されるまでは、統合を完全にテストすることはできません。リクエストされた場合、AWS Marketplace カタログオペレーションチームは計測レコードの受信を確認できます。

MeterUsage でのエラー処理

コンテナ起動時にコンテナイメージが MeterUsage オペレーションを統合し、ThrottlingException 以外の例外を受け取る場合、コンテナイメージを終了して不正使用を防止する必要があります。

ThrottlingException 以外の例外は、MeterUsage への最初の呼び出しでのみスローされます。同じ Amazon ECS タスクまたは Amazon EKS ポッド呼び出しからの以降の呼び出しでは、CustomerNotSubscribedException はスローされません。タスクまたはポッドが実行されている間に顧客がサブスクリプションを解除した場合でも同様です。この顧客は、サブスクリプションを解除したあとで使用状況が追跡されると、実行中のコンテナに対して課金されます。

MeterUsage の一般的なエラーの詳細については、「AWS Marketplace Metering Service API リファレンス」の「[MeterUsage](#)」を参照してください。各 AWS SDK プログラミング言語には、追加情報を参照できる一連のエラー処理ガイドラインがあります。

ベンダー計測タグ付け (オプション)

ベンダー計測タグ付けは、独立系ソフトウェアベンダー (ISV) がソフトウェアの使用状況についてより詳細な情報を購入者に提供し、コスト配分を行うのに役立ちます。

購入者が使用したソフトウェア使用状況をタグ付けするには、さまざまな方法があります。1つの方法は、コスト配分に何を期待するかを購入者にたずねます。次に、購入者のアカウントで追跡するプロパティ全体で使用量を分割できます。プロパティの例には、AccountId、Business Unit、Cost Centers、および商品に関連する他のメタデータが含まれます。これらのプロパティはタグとして購入者に公開されます。タグを使用すると、購入者は AWS 請求コンソール (<https://console.aws.amazon.com/costmanagement/>) のタグ値によってコストを使用量に分割して表示できます。ベンダー計測タグ付けでは、報告する価格、ディメンション、合計使用量は変わりません。お客様は商品に適したカテゴリ別にコストを確認できます。

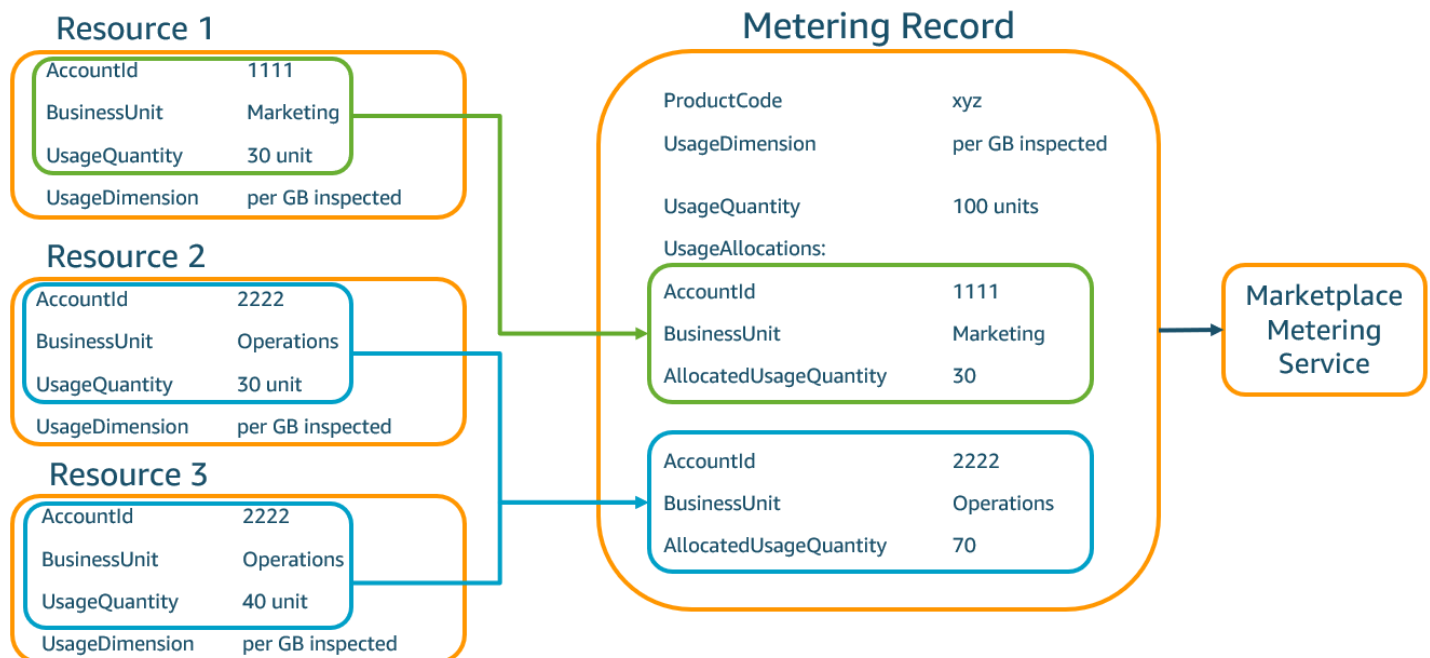
一般的なユースケースでは、購入者が 1 つの AWS アカウントを使って商品を購入します。また、購入者には同じ製品サブスクリプションに関連するユーザーが多数います。AccountId のキーを持つタグを使用して使用量の割り当てを作成し、各ユーザーに使用量を割り当てることができます。この場合、購入者は Billing and Cost Management コンソールで AccountId タグを有効にして、個々のユーザーの使用状況を分析できます。

販売者のエクスペリエンス

販売者は、すべてのリソースの使用量を集計する代わりに、同じタグセットを使用してリソースの計測レコードを集計できます。例えば、販売者は UsageAllocations の異なるバケットを含む計測レコードを作成できます。各バケットは、AccountId や BusinessUnit などの、タグのセットの UsageQuantity を表します。

以下の図では、[リソース 1] には固有の AccountId と BusinessUnit タグセットがあり、[計測レコード] に 1 つのエントリとして表示されます。

[リソース 2] と [リソース 3] にはどちらも同じ AccountId タグ、2222、および同じ BusinessUnit タグ Operations があります。その結果、これらは [計測レコード] の 1 つの UsageAllocations エントリにまとめられます。



また、販売者は、タグのないリソースを割り当てられた使用量の単一の UsageAllocation にまとめて、UsageAllocations のエントリの 1 つとして送信することもできます。

制限には以下が含まれます。

- タグの数 - 5
- UsageAllocations のサイズ (カーディナリティ) - 2,500

検証には次の項目が含まれます。

- タグのキーと値に使用できる文字 - a-zA-Z0-9+ - = . _ : \ / @
- UsageAllocation リスト全体の最大タグ数 - 5
- 2 つの UsageAllocations タグに同じタグ (つまり、同じタグキーと値の組み合わせ) を持つことはできません。その場合は、同じ UsageAllocation を使用する必要があります。
- UsageAllocation の AllocatedUsageQuantity の合計は、総使用量である UsageQuantity と等しくなる必要があります。

購入者エクスペリエンス

次の表は、購入者が AccountId および BusinessUnit ベンダータグを有効化した後の購入者エクスペリエンスの例を示しています。

この例では、購入者は [コスト使用状況レポート] で割り当てられた使用量を確認できます。ベンダー計測タグにはプレフィックス “aws:marketplace:isv” が使用されます。購入者は、請求情報とコスト管理の、[コスト配分タグ]、[AWS生成コスト配分タグ] でそれらを有効化できます。

[コスト使用状況レポート] の最初と最後の行は、出品者が Metering Service に送信する内容に関係します ([販売者のエクスペリエンス](#) 例を参照)。

コスト使用状況レポート (簡略版)

ProductCode	購入者	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	70	2222	オペレーション
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	30	3333	財務

ProductCode	購入者	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	20	4444	IT
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	20	5555	マーケティング
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	30	1111	マーケティング

コード例については、[使用量割り当てタグ付きの MeterUsage コード例 \(オプション\)](#) を参照してください。

コード例

次のコード例は、コンテナ製品を製品の公開と保守に必要な AWS Marketplace APIs と統合するのに役立ちます。

使用量割り当てタグ付きの **MeterUsage** コード例 (オプション)

次のコード例は、消費料金モデルを持つコンテナ製品に関連しています。Python の例では、従量制料金を顧客に請求するために、適切な使用量割り当てタグを含む計測レコードを AWS Marketplace に送信します。

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited
#
# Import AWS Python SDK
```

```
import boto3
import time

usageRecord = [
    {
        "AllocatedUsageQuantity": 2,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "IT" },
                { "Key": "AccountId", "Value": "123456789" },
            ]
    },
    {
        "AllocatedUsageQuantity": 1,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "Finance" },
                { "Key": "AccountId", "Value": "987654321" },
            ]
    }
]

marketplaceClient = boto3.client("meteringmarketplace")

response = marketplaceClient.meter_usage(
    ProductCode="testProduct",
    Timestamp=int(time.time()),
    UsageDimension="Dimension1",
    UsageQuantity=3,
    DryRun=False,
    UsageAllocations=usageRecord
)
```

MeterUsage の詳細については、「AWS Marketplace Metering Service API リファレンス」の「[MeterUsage](#)」を参照してください。

レスポンスの例

```
{ "MeteringRecordId": "string" }
```

カスタム計測を使用したコンテナ製品の AWS Marketplace Metering Service およびとの統合 AWS SDK for Java

AWS Marketplace コンテナ製品は、製品ごとに最大 24 の異なる料金ディメンションでカスタム計測を行うことができます。カスタム計測を有効にするには、コンテナ製品を AWS Marketplace Metering Service と統合してください。[MeterUsage](#) API オペレーションを使用して、AWS に請求するために独自の料金単位とその使用量に対するカスタム計測を定義できます。次の例では、を使用して [AWS Marketplace Metering Service](#) MeterUsage オペレーションと AWS SDK for Java 統合する実装の概要を示します。

詳細については、「[MeterUsage Java の例](#)」を参照してください。次の手順の多くは、言語にかかわらず適用されます。

例：AWS Marketplace Metering Service の統合

1. [AWS Marketplace 管理ポータル](#) にサインインします。
2. [アセット] から [コンテナ] を選択して、新しいコンテナ製品の作成を開始します。製品を作成すると、製品とコンテナイメージを統合するための製品コードが生成されます。AWS Identity and Access Management (IAM) アクセス許可の設定については、「」を参照してください [the section called “AWS Marketplace 計測と使用権限 API アクセス許可”](#)。
3. 公開 [AWS Java SDK](#) をダウンロードします。

Important

Amazon Elastic Kubernetes Service (Amazon EKS) から計測 API オペレーションを呼び出すには、[サポートされている AWS SDK を使用](#)し、Kubernetes 1.13 以降を実行している Amazon EKS クラスターで実行する必要があります。

4. 各ディメンションの使用状況について、1 時間に 1 回、タスクまたはポッドから MeterUsage オペレーションを呼び出します。API オペレーションは、Dimension、Resource、Hour の一意の組み合わせに対して 1 つの計測レコードを受け入れます。リソースは、Amazon Elastic Container Service (Amazon ECS) タスクまたは Amazon Elastic Container Service (Amazon ECS) ポッドのいずれかです。

```
{
  "ProductCode" : "string", // (required)
  "UsageDimension" : "string", // (required)
  "UsageQuantity": int, // (optional) Default is 0. Acceptable value from [0,
  2147483647 (INT_MAX)]
```

```
"Timestamp": Date, // (required) Timestamp in UTC. Value can be one hour in the
past.
"UsageAllocations": List<UsageAllocation> // (optional) UsageAllocations across
1 or more tags.
}
```

Note

への接続で一時的な問題が発生する可能性があります AWS Marketplace Metering Service。AWS Marketplace 短期的な停止やネットワークの問題を回避するために、では最大 30 分間の再試行をエクスポネンシャルバックオフで実装することを強くお勧めします。

5. MeterUsage 呼び出しを含むコンテナイメージを再構築し、コンテナにタグ付けして、Amazon Elastic Container Registry (Amazon ECR) など、Amazon ECS または Amazon EKS と互換性がある任意の Docker レジストリにプッシュします。Amazon ECR を使用している場合は、Amazon ECS タスクまたは Amazon EKS ポッドを起動するアカウントに、Amazon ECR リポジトリへのアクセス許可があることを確認します。それ以外の場合は、このオペレーションは失敗します。
6. 次のコード例で定義されているように、コンテナが MeterUsage を呼び出すアクセス許可を付与する [IAM](#) ロールを作成します。Amazon ECS タスクまたは Amazon EKS ポッド定義の [タスクロール](#) パラメータで、この AWS Identity and Access Management (IAM) ロールを指定する必要があります。

JSON

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


7. と統合されたコンテナを参照 AWS Marketplace し、ステップ 6 で作成した IAM ロールを参照する Amazon ECS タスクまたは Amazon EKS ポッド定義を作成します。ログ記録を表示するには、タスク定義で AWS CloudTrail ログ記録を有効にします。
8. Amazon ECS または Amazon EKS クラスターを作成して、タスクまたはポッドを実行します。Amazon ECS クラスターの作成の詳細については、「Amazon Elastic Container Service デベロッパーガイド」の「[クラスターの作成](#)」を参照してください。Amazon EKS クラスターの作成 (Kubernetes バージョン 1.13.x 以降を使用) の詳細については、「[Amazon EKS クラスターの作成](#)」を参照してください。
9. Amazon ECS または Amazon EKS クラスターを設定し、us-east-1 AWS Region のステップ 8 で作成した Amazon ECS タスク定義または Amazon EKS ポッドを起動します。製品が本番稼働になる前のこのテストプロセス中にのみ、このリージョンを使用する必要があります。
10. 製品に対して発行されるディメンションごとに MeterUsage から有効な応答が返されると、コンテナ製品の作成を開始できます。ご質問がある場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

MeterUsage Java の例

次のコード例では、AWS SDK for Java および AWS Marketplace Metering Service を使用して MeterUsage オペレーションを呼び出します。

次のコード例は、UsageAllocations なしで MeterUsage オペレーションを呼び出します。

```
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMetering;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMeteringClientBuilder;
import com.amazonaws.services.marketplacemetering.model.MeterUsageRequest;
import com.amazonaws.services.marketplacemetering.model.MeterUsageResult;

import java.util.Date;

public class MeterUsage {
    private static final String PRODUCT_CODE = ".....";
    private final AWSMarketplaceMetering awsMarketplaceMetering;

    public MeterUsage() {
        awsMarketplaceMetering =
        AWSMarketplaceMeteringClientBuilder.standard().build();
    }

    /**
```

```

    * Submits metering record for a FCP Dimension. The API accepts 1 metering record
    per dimension
    * for a given buyer's resource for a given timestamp hour. Ex. If a buyer is
    running 10 tasks,
    * the API will accepts 1 call to MeterUsage in an hour for a given dimension for
    each running task.
    *
    * @param dimension - FCP dimension name provided during the publishing of the
    product.
    * @param quantity - FCP dimension consumption value for the hour.
    * @param timestamp - Timestamp, in UTC, for which the usage is being reported.
    *
    *           Timestamp cant be more than 1 hour in the past.
    *
    *           Make sure the timestamp value is not before the start of the
    software usage.
    */
    public void callMeterUsage(String dimension, int quantity, Date timestamp) {
        MeterUsageRequest meterUsageRequest = new MeterUsageRequest()
            .withProductCode(PRODUCT_CODE)
            .withUsageDimension(dimension)
            .withUsageQuantity(quantity)
            .withTimestamp(timestamp);
        MeterUsageResult meterUsageResult =
            awsMarketplaceMetering.meterUsage(meterUsageRequest);
    }
}

```

次のコード例は、UsageAllocations で MeterUsage オペレーションを呼び出します。

```

private static String callMeterUsageWithAllocationsByTag(AWSMarketplaceMetering
marketplaceMetering) {
    // Tag Keys for the product
    String tagKey1 = "Key1";
    String tagKey2 = "Key2";
    String tagKey3 = "Key3";

    // 1st Usage Allocation bucket which has two Tags [{Key1, Key1Value1},{Key2,
    Key2Value1}]
    List<Tag> tagsForUsageAllocation1 = Arrays.asList(new
    Tag().withKey(tagKey1).withValue("Key1Value1"),
        new Tag().withKey(tagKey2).withValue("Key2Value1"));
    UsageAllocation usageAllocation1 = new UsageAllocation()
        .withTags(tagsForUsageAllocation1)
        .withAllocatedUsageQuantity(20);
}

```

```
// 2nd Usage Allocation bucket which has two Tags [{Key1, Key1Value2},{Key2,
Key2Value1}]
List<Tag> tagsForUsageAllocation2 = Arrays.asList(new
Tag().withKey(tagKey1).withValue("Key1Value2"),
    new Tag().withKey(tagKey2).withValue("Key2Value1"));
UsageAllocation usageAllocation2 = new UsageAllocation()
    .withTags(tagsForUsageAllocation2)
    .withAllocatedUsageQuantity(20);

// 3rd Usage Allocation bucket which has two Tags [{Key1, Key1Value2},{Key2,
Key2Value2},{Key3, Key3Value1}]
List<Tag> tagsForUsageAllocation3 = Arrays.asList(new
Tag().withKey(tagKey1).withValue("Key1Value2"),
    new Tag().withKey(tagKey2).withValue("Key2Value2"),
    new Tag().withKey(tagKey3).withValue("Key3Value1"));
UsageAllocation usageAllocation3 = new UsageAllocation()
    .withTags(tagsForUsageAllocation3)
    .withAllocatedUsageQuantity(15);

// 4th Usage Allocation bucket with no tags
UsageAllocation usageAllocation4 = new UsageAllocation()
    .withAllocatedUsageQuantity(15);

List<UsageAllocation> usageAllocationList = Arrays.asList(usageAllocation1,
    usageAllocation2,
    usageAllocation3,
    usageAllocation4);

MeterUsageRequest meterUsageRequest = new MeterUsageRequest()
    .withProductCode("TestProductCode")
    .withUsageDimension("Dimension1")
    .withTimestamp(new Date())
    //UsageQuantity value must match with sum of all AllocatedUsageQuantity
    .withUsageQuantity(70)
    .withUsageAllocations(usageAllocationList);

MeterUsageResult meterUsageResult;
try {
    meterUsageResult = marketplaceMetering.meterUsage(meterUsageRequest);
} catch (Exception e) {
    // Log Error
    throw e;
}
```

```
        return meterUsageResult.getMeteringRecordId();  
    }
```

を使用したコンテナ製品の契約料金 AWS License Manager

契約料金のコンテナベースの製品の場合、を使用してライセンス AWS License Manager を製品に関連付けることができます。AWS License Manager は、顧客が購入したライセンス (使用権限とも呼ばれます) をアプリケーションが追跡および更新できるようにするライセンス管理ツールです。このセクションでは、製品を と統合する方法について説明します AWS License Manager。統合が完了したら、AWS Marketplaceに製品リストを公開できます。

License Manager を Amazon EKS AnywhereAmazon ECS Anywhere、またはオンプレミスインフラストラクチャ用の Containers Anywhere 製品 AWS Marketplace 用の と統合する場合は、「」の手順に従います[AWS Marketplace for Containers Anywhere と License Manager の統合](#)。Amazon EC2

詳細については AWS License Manager、[AWS License Manager 「ユーザーガイド」](#) およびAWS CLI 「コマンドリファレンス」の[AWS License Manager 「」](#) セクションを参照してください。

コンテナ製品 AWS License Manager と契約料金との統合の詳細については、AWS Marketplace 販売者ワークショップの「前払い[支払いラボとの統合](#)」を参照してください。

トピック

- [ライセンスモデル](#)
- [AWS License Manager 統合の前提条件](#)
- [コンテナ製品と License Manager の統合](#)
- [License Manager API オペレーション](#)
- [ライセンスの更新とアップグレード](#)
- [AWS Marketplace for Containers Anywhere と License Manager の統合](#)

ライセンスモデル

AWS Marketplace と の統合では、次の 2 つのライセンスモデル AWS License Manager がサポートされています。

- [設定可能なライセンスモデル](#)
- [階層型ライセンスモデル](#)

設定可能なライセンスモデル

設定可能なライセンスモデル (定量化可能なライセンスモデルとも呼ばれます) では、購入者がライセンスを調達した後に、特定の量のリソースを購入者に付与できます。

料金設定ディメンションと単価を設定します。次に、購入者は購入するリソースの数量を選択できます。

Example 料金ディメンションと単価の

料金設定ディメンション (データバックアップなど) と単価 (1 ユニットあたり 30 USD など) を設定できます。

購入者は 5 単位、10 単位、または 20 単位の購入を選択できます。

製品は使用状況を追跡して測定し、消費されたリソースの量を測定します。

設定モデルでは、使用権限は次の 2 つのいずれかにカウントされます。

- [ドローダウンライセンス](#)
- [フローティングライセンス](#)

ドローダウンライセンス

ライセンスは、使用時に許可されたライセンス数のプールから引き出されます。その使用権限は永久にチェックアウトされ、ライセンスプールに戻すことはできません。

Example 限られた量のデータを処理する

ユーザーには 500 GB のデータを処理する権限があります。ユーザーがデータを処理し続けると、500 GB のライセンスがすべて消費されるまで 500 GB のプールからデータが引き出されます。

ドローダウンライセンスの場合は、CheckoutLicense API オペレーションを使用して、消費されたライセンスユニット (使用権限) を確認できます。

Example Amazon S3 へのバックアップの単位数/年

Amazon Simple Storage Service へのバックアップを 1 年間最大 1,024 ユニットのデータで許可するストレージ製品があります。アプリケーションは、複数の Amazon EC2 インスタンスを使用して起動できます。アプリケーションには、データを追跡して集約するメカニズムがあります。ソフトウェア

アは、バックアップのたびに、または一定の間隔で、製品 ID を使用して CheckoutLicense API オペレーションを呼び出し、消費量を更新します。

この例では、ソフトウェアが CheckoutLicense API オペレーションを呼び出して 10 単位のデータをチェックアウトします。合計容量が顧客が購入したバックアップ制限に達すると、API コールは失敗します。

リクエスト

```
linux-machine ~]$ aws license-manager checkout-license\
--product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \
--checkout-type "PERPETUAL" \
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements "Name=DataConsumption, Value=10, Unit=Count" \
--client-token "AKIAIOSFODNN7EXAMPLE"
```

レスポンス

```
{"CheckoutType": "PERPETUAL",
"EntitlementsAllowed": [{
  "Name": "IntermediateTier",
  "Units": "None"
}],
"Expiration": "2021-04-22T19:02:36",
"IssuedAt": "2021-04-22T18:02:36",
"LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
"LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

フローティングライセンス

ライセンスは、使用後に許可されたライセンス数のプールに返却されます。

フローティングライセンスの場合、アプリケーションはリソースが使用されているときに CheckoutLicense API オペレーションを使用して使用権限プールから使用権限をチェックアウトします。CheckoutLicense API オペレーションのレスポンスには、チェックアウト時の一意の識別子であるライセンス消費トークンが含まれます。ライセンス消費トークンを使用して、チェックアウトした使用権限をライセンスに戻したり、チェックアウトを延長したりするなど、追加のアクションを実行できます。

使用権限をプールに戻すには、リソースが使用されなくなった時に CheckInLicense API オペレーションを使用します。

```
aws license-manager check-in-license --license-consumption-token
"f1603b3c1f574b7284db84..."
```

使用権限のチェックインに失敗した場合 (アプリケーションがクラッシュした場合)、使用権限は 60 分後に自動的にプールにチェックインされます。リソースが 60 分以上使用されている場合は、リソースが使用されている限り ExtendLicenseConsumption API オペレーションを使用して使用権限をプールからチェックアウトしたままにしておくことがベストプラクティスです。

```
aws license-manager extend-license-consumption --license-consumption-token
"f1603b3c1f574b7284..."
```

Example 固定された上限を超えるユーザー数

1 人のユーザーには、アプリケーションで 500 人の同時ユーザーを割り当てることができます。ユーザーがログインしてログアウトすると、そのユーザーは引き出され、500 人のユーザープールに戻ります。ただし、500 人の同時ユーザーが固定の上限であるため、アプリケーションはプールから 500 人を超えるユーザーを引き出すことはできません。

フローティング使用権限の場合は、CheckInLicense API オペレーションを使用してライセンス単位を使用権限プールに戻すことができます。

Example 1 年間の同時ユーザー数

製品の料金は、同時接続ユーザー数に基づいて決定されます。顧客は 10 ユーザー分のライセンスを 1 年間購入します。顧客は AWS Identity and Access Management (IAM) アクセス許可を与えてソフトウェアを起動します。ユーザーがログインすると、アプリケーションが CheckoutLicense API オペレーションを呼び出して数量を 1 つ減らします。ユーザーがログアウトすると、アプリケーションは CheckInLicense API オペレーションを呼び出してそのライセンスをプールに戻します。CheckInLicense を呼び出さない場合、ライセンス単位は 1 時間後に自動的にチェックインされます。

Note

次のリクエストでは、key-fingerprint はプレースホルダー値ではなく、すべてのライセンスが公開されるフィンガープリントの実際の値です。

リクエスト


```
aws license-manager checkout-license\  
--product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \  
--checkout-type "PROVISIONAL" \  
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \  
--entitlements "Name=ReadOnlyUSers, Value=10, Unit=Count" \  
--client-token "AKIAIOSFODNN7EXAMPLE"
```

レスポンス

```
{  
  "CheckoutType": "PROVISIONAL",  
  "EntitlementsAllowed": [  
    {  
      "Name": "ReadOnlyUsers",  
      "Count": 10,  
      "Units": "Count",  
      "Value": "Enabled"  
    }  
  ],  
  "Expiration": "2021-04-22T19:02:36",  
  "IssuedAt": "2021-04-22T18:02:36",  
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",  
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"  
}
```

階層型ライセンスモデル

階層型ライセンスモデルでは、購入者がライセンスを調達した後に、特定のレベル (階層) のアプリケーション機能を利用できるようになります。

製品には、ベーシック、中級、プレミアムなどの階層を作成します。次に、購入者はあらかじめ定義されている階層の 1 つを選択します。

アプリケーションはアプリケーションの使用状況を追跡したり、測定する必要はありません。

階層型ライセンスモデルでは、使用権限はカウントされず、代わりに顧客が購入したサービスの階層が示されます。

バンドルされた機能を一緒に提供する場合は、階層化するのが望ましいです。

Example ベーシック、中級、プレミアムの各レベル

お客様は、ソフトウェアの3つのレベル (ベーシック、中級、プレミアム) のいずれかの契約に署名できます。これらの階層にはそれぞれ独自の料金設定があります。CheckoutLicense API オペレーションを呼び出し、リクエストで可能なすべての階層を指定して、ソフトウェアが顧客がサインアップした階層を特定できます。

リクエストのレスポンスには、顧客が調達した階層に対応する使用権限が含まれています。この情報に基づいて、ソフトウェアは適切なカスタマーエクスペリエンスを提供できます。

リクエスト

```
linux-machine ~]$ aws license-manager checkout-license\
--product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \
--checkout-type "PROVISIONAL" \
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements "Name=BasicTier, Unit=None" "Name=IntermediateTier, Unit=None" \
"Name=PremiumTier, Unit=None"
```

レスポンス

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "IntermediateTier",
      "Units": "None"
    }
  ],
  "Expiration": "2021-04-22T19:02:36",
  "IssuedAt": "2021-04-22T18:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

AWS License Manager 統合の前提条件

製品を公開する前に、次を行う必要があります。

1. で新しいコンテナ製品を作成し AWS Marketplace 管理ポータル、その製品コードを書き留めます。

詳細については、「[概要: コンテナ製品を作成する](#)」を参照してください。

2. CheckoutLicense、ExtendLicenseConsumption、CheckInLicense API オペレーションを呼び出すために必要な IAM アクセス許可を持つアプリケーションを実行するタスクまたはポッドの IAM ロールを使用します。

必要な IAM アクセス許可の詳細は、次の IAM ポリシーに記載されています。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "license-manager:CheckoutLicense",
        "license-manager:GetLicense",
        "license-manager:CheckInLicense",
        "license-manager:ExtendLicenseConsumption",
        "license-manager:ListReceivedLicenses"
      ],
      "Resource": "*"
    }
  ]
}
```

3. 定義したすべての料金ディメンションのレコードを使用して、RegisterUsage API オペレーションをテストコールします。

コンテナ製品と License Manager の統合

コンテナベースの製品を License Manager と統合するには

1. License Manager を呼び出すための IAM アクセス許可を設定します。詳細については、「[AWS License Manager 統合の前提条件](#)」を参照してください。
2. AWS SDK をダウンロードします。

Note

ソフトウェア内で AWS 認証情報を設定しないでください。購入者の AWS 認証情報は、コンテナが Amazon EC2 インスタンス、Amazon ECS タスク、または Amazon EKS ポッド内で実行されているときに、実行時に自動的に取得されます。

3. ライセンスチェックを製品に追加します。

ライセンスチェックを実行する必要がある場所ならどこでも、製品が CheckoutLicense API オペレーションを呼び出すことができます。ライセンスを確認するには、製品が以下を認識している必要があります。

1. ライセンスの信頼できる発行者 (AWS Marketplace)
2. アプリケーションの製品 SKU (製品 ID)
3. このアプリケーションを確認する使用権限。

API コールは、設定した料金ライセンスの種類によって異なります。

4. 製品リストを公開します AWS Marketplace。

License Manager API オペレーション

顧客のライセンスマネージャーアカウントに保存されているライセンスを管理するために、ソフトウェアは次の API オペレーションを使用できます。

- GetLicense - ソフトウェアがクエリできる API。購入したライセンスのステータス (有効期限が切れた、または間もなく期限切れになる) を取得し、ステータス通知を顧客に送信します。
- CheckoutLicense - ユーザーが購入したライセンスを検出します。ユーザーがいくらかのライセンスを消費した時に、CheckoutLicense API オペレーションを使用してライセンス数を更新することもできます。CheckoutLicense を使用すると、顧客が使用したライセンス数を常に確認することができます。顧客がすべてのライセンスを使い果たすと、この呼び出しはエラーを返します。CheckoutLicense を実行するための推奨ケイデンスについては、[「the section called “ライセンスの更新とアップグレード”」](#)を参照してください。
- ExtendLicenseConsumption - フローティングディメンションの場合、ソフトウェアがライセンスをチェックアウトすると、60 分後にライセンスは自動的にプールに戻ります。ライセンスがチェックアウトされたままの期間を延長する場合は、ExtendLicenseConsumption API オペレーションを使用してライセンスをさらに 60 分間延長します。

- CheckInLicense - フローティングディメンションの場合、ライセンスを使用権限プールに戻す場合は、CheckInLicense API オペレーションを使用してください。
- ListReceivedLicenses API - 購入者が購入したライセンスを一覧表示します。

ライセンスの更新とアップグレード

顧客は AWS Marketplace 管理ポータルでライセンスを更新またはアップグレードできます。追加購入を行うと、新しい使用権限を反映する新しいバージョンのライセンス AWS Marketplace を生成します。ソフトウェアは同じ API オペレーションを使用して新しい使用権限を読み取ります。License Manager の統合に関しては、更新やアップグレードを処理するために特別なことをする必要はありません。

ライセンスの更新、アップグレード、キャンセルなどが発生するため、製品の使用中は定期的に CheckoutLicense API オペレーションを呼び出すことをお勧めします。CheckoutLicense API オペレーションを定期的に使用して、製品はアップグレードや有効期限などの使用権限の変更を検出できます。

CheckoutLicense API コールは 15 分ごとに実行することをお勧めします。

AWS Marketplace for Containers Anywhere と License Manager の統合

AWS Marketplace 販売者は、Amazon EKS Anywhere、Amazon ECS Anywhere、Amazon EC2、またはオンプレミスインフラストラクチャ AWS Marketplace 用の Containers Anywhere 製品用の AWS License Manager と統合できます。Amazon ECS Anywhere 以下のセクションでは、この統合の手順について説明します。

使用可能なライセンスモデルなど AWS Marketplace、License Manager との統合に関する一般的な情報については、「」を参照してください [を使用したコンテナ製品の契約料金 AWS License Manager](#)。AWS License Manager の詳細については、「[AWS License Manager ユーザーガイド](#)」および「AWS CLI コマンドリファレンス」の「[AWS License Manager](#) セクション」を参照してください。

トピック

- [AWS Marketplace for Containers Anywhere 製品を License Manager と統合する](#)
- [License Manager 統合のローカルテスト](#)
- [Amazon EKS での License Manager 統合のテスト](#)
- [License Manager を使用したフローティングライセンス使用権限](#)
- [オンプレミスデプロイでの License Manager との統合に関するベストプラクティス](#)

- [LicenseManagerCredentialsProvider - Java 実装](#)
- [LicenseManagerCredentialsProvider - Golang 実装](#)

AWS Marketplace for Containers Anywhere 製品を License Manager と統合する

以下の手順を使用して、AWS Marketplace for Containers Anywhere 製品を と統合します AWS License Manager。

AWS Marketplace for Containers Anywhere 製品を License Manager と統合するには

1. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
2. 以下のステップを実行して、コンテナ製品の製品 ID を作成します。この ID をコンテナイメージで使用して、後のステップでライセンスチェックを行います。
 - a. メニューバーから [Assets (アセット)] を展開し、[コンテナ] を選択します。
 - b. 商品に購入者向けの名前を入力し、[作成] を選択します。この名前は後で変更できます。
 - c. [プロジェクト ID] を書き留めます。製品料金の詳細を作成または更新するときに使用します。

 Tip

製品 ID を紛失した場合は、アセットメニューからコンテナ AWS Marketplace 管理ポータル を選択して で確認できます。[コンテナ] ページには、製品のリストとそれに関連する製品 ID が表示されます。

3. 最新のパブリック AWS SDK をダウンロードし、コンテナアプリケーションにインストールします。任意の AWS SDK のインストール手順については、「AWS [で構築するツール](#)」を参照してください。

 Note

Amazon EKS Anywhere または によって提供されていない Kubernetes クラスターから License Manager API オペレーションを呼び出すには AWS、サポートされている AWS SDK を使用する必要があります。サポートされている AWS SDKs [「サポートされている AWS SDK の使用」](#) を参照してください。

4. カスタム認証情報プロバイダーを使用して AWS License Manager クライアントを作成し、AWS およびオンプレミスにデプロイされたコンテナアプリケーションに認証情報を提供でき

るようにします。カスタム認証情報プロバイダー (LicenseCredentialsProvider) の完全なソースコードについては、以下のセクションを参照してください。

- [LicenseManagerCredentialsProvider - Java 実装](#)
- [LicenseManagerCredentialsProvider - Golang 実装](#)

LicenseCredentialsProvider は、を追加して AWS、SDK のデフォルトの認証情報プロバイダーチェーンをオンプレミス用に拡張します LicenseManagerTokenCredentialsProvider。これにより、オンプレミス環境で License Manager OIDC が発行した ID トークンを使用して認証情報が提供されます。LicenseCredentialsProvider のソースコードをアプリケーションのクラスパスに含める必要があります。

 Note

を拡張DefaultCredentialsProviderすると、同じコンテナアプリケーションがで実行されている場合 AWS とオンプレミス環境で実行されている場合に認証情報を取得できます。コンテナアプリケーションが既にデフォルトではなくカスタム認証情報プロバイダーチェーンを使用している場合は、LicenseManagerTokenCredentialsProvider をカスタムチェーンに追加して拡張することもできます。

次のコードスニペットは、Java を使用して AWS License Manager クライアントを作成する例です。

```
LicenseManagerClientBuilder clientBuilder =  
    LicenseManagerClient.builder().credentialsProvider(LicenseCredentialsProvider.create());
```

5. 製品オファー内の各有料コンテナイメージから `aws license-manager checkout-license` コマンドを使用して、CheckoutLicense API オペレーションを呼び出します。これにより、購入者がアプリケーションのライセンスを使用する使用権限があるかどうかを確認されます。購入者にアプリケーションに対する使用権限がある場合は、CheckoutLicense は、申請を承諾し、リクエストされた使用権限とその値を返します。購入者にアプリケーションに対する使用権限がない場合は、CheckoutLicense は、例外をスローします。

CheckoutLicense API オペレーションを呼び出す際には、以下のパラメータが必要です。

- CheckoutType - 有効な値は PROVISIONAL または PERPETUAL です。
- チェックアウトされた使用権限の数がプールからなくなる場合に、PERPETUAL を使用します。

例: 購入者には 500 GB のデータを処理する権限があります。データの処理を続けると、500 GB のプールからその量が引き出され、使い果たされます。

- 使用権がプールからチェックアウトされ、使用後に返却されるフローティングライセンス使用権に PROVISIONAL を使用します。

例: ユーザーには、アプリケーション内で同時に 500 人のユーザーを使用する権利があります。ユーザーがログインまたはログアウトすると、そのユーザーは引き出されるか、500 人のユーザープールに戻ります。フローティングライセンス使用権限の詳細については、[「License Manager を使用したフローティングライセンス使用権限」](#)を参照してください。

- ClientToken - 大文字と小文字が区別される一意の識別子。固有のリクエストごとにランダムな UUID を使用することをお勧めします。
- Entitlements - チェックアウトする使用権限のリスト。
- 機能の使用権限の場合は、Name プロパティと Unit プロパティを次のように指定します。

```
{
  "Name": "<Entitlement_Name>",
  "Unit": "None"
}
```

- カウント対象使用権限については、以下のように、Name プロパティ、Unit プロパティ、および Count プロパティを指定します。

```
{
  "Name": "<Entitlement_Name>",
  "Unit": "<Entitlement_Unit>",
  "Value": <Desired_Count>
}
```

- KeyFingerprint - AWS Marketplace が発行するライセンスのキーフィンガープリントは aws:294406891311:AWS/Marketplace:issuer-fingerprint です。このキーフィンガープリントを使用すると、ライセンスは信頼できないエンティティによって発行 AWS Marketplace されるのではなく、によって発行されます。
- ProductSKU – 前のステップ AWS Marketplace 管理ポータル で生成された製品 ID。

次のスニペットは、AWS CLIを使用した CheckoutLicense API オペレーションによる呼び出しの例です。

```
aws license-manager checkout-license \  
--product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \  
--checkout-type "PROVISIONAL" \  
--client-token "79464194dca9429698cc774587a603a1" \  
--entitlements "Name=AWS::Marketplace::Usage/Drawdown/DataConsumption, Value=10, Unit=Gigabytes" \  
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint"
```

 Note

ライセンスを確認するには、コンテナアプリケーションが License Manager を使用するためのアウトバウンドネットワークアクセスが必要です。オンプレミスでデプロイされたアプリケーションでは、アウトバウンドネットワークアクセスの信頼性が低くなったり、速度が低下する可能性があります。これらのアプリケーションでは、License Manager を呼び出す際に適切に再試行する必要があります。詳細については、「[オンプレミスデプロイでの License Manager との統合に関するベストプラクティス](#)」を参照してください。

- CheckoutLicense API オペレーションを定期的に呼び出して、AWS Marketplaceで行われた更新、アップグレード、またはキャンセルによって顧客のライセンスが変更されたかどうかを確認してください。ケイデンスはアプリケーションによって異なります。購入者の介入なしに変更が自動的に反映されるように、1 日 1 回ライセンスを確認することをお勧めします。

オンプレミスにデプロイされたアプリケーションでは、定期的にライセンスをチェックするためのアウトバウンドネットワークアクセスが信頼できない場合があります。このような場合、アプリケーションは十分な耐障害性を保つためにキャッシュされたライセンスを使用する必要があります。詳細については、「[オンプレミスデプロイでの License Manager との統合に関するベストプラクティス](#)」を参照してください。

- CheckoutLicense 呼び出しをコンテナアプリケーションと統合したら、変更を加えた新しいバージョンの Docker コンテナイメージを構築します。
- アプリケーションの Helm チャートを更新して、License Manager API を使用してライセンスにアクセスするための設定を含む Kubernetes シークレットをオプション入力として受け入れます。設定シークレットには、License Manager によって発行された ID トークンと、コンテナア

アプリケーションがオンプレミスにデプロイされたときに License Manager APIs を呼び出すための AWS 認証情報を取得するために、前述のカスタム認証情報プロバイダーによって使用される AWS Identity and Access Management ロールが含まれます。また、デフォルト値 us-east-1 を使用して AWS リージョン を入力として追加します。

オンプレミスでコンテナアプリケーションをデプロイする購入者は、コンテナ製品の AWS Marketplace 購入者エクスペリエンスを通じて Kubernetes シークレットを作成できます。Kubernetes シークレット名を `helm install` コマンドへの入力として指定します。設定シークレットは次の形式で構成されます。

```
apiVersion: v1
kind: Secret
metadata:
  name: aws-marketplace-license-config
type: Opaque
stringData:
  license_token: <token_value> // License Manager issued JWT token
  iam_role: <role_arn> // AWS Identity and Access Management role to assume with
  license token
```

9. と統合されたコンテナイメージの Helm チャートのアプリケーションデプロイテンプレートを更新 AWS License Manager して、以下を含めます。

- ポッドのサービスアカウント - Amazon EKS に Helm をデプロイするにはサービスアカウントが必要です。コンテナイメージのサービスアカウントの IAM ロールを設定して、License Manager API オペレーションを呼び出す権限を取得するために使用されます。サービスアカウントの IAM ロールの詳細については、「[サービスアカウントの IAM ロール](#)」を参照してください。
- オンプレミスデプロイのライセンスアクセス - ライセンス設定シークレットは、オンプレミス環境での Helm デプロイの License Manager API オペレーションを呼び出すための認証情報と適切なアクセス許可を提供するために必要です。購入者は、AWS Marketplace 購入者エクスペリエンスからライセンスシークレットを生成して Helm に提供します。

以下のコードスニペットは、サービスアカウント、ライセンス設定、イメージプルシークレットを含むデプロイ仕様のサンプルです。

```
apiVersion: apps/v1
kind: Deployment
metadata:
```

```

  name: example-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: example-app
  template:
    metadata:
      labels:
        app: example-app
spec:
  // Service account for pod
  serviceAccountName: {{ .Values.serviceAccountName }}
  containers:
    - name: example-app
      image: example-app
      ports:
        - containerPort: 8001
// Add the following conditional attributes
{{ - if .Values.awsmpl.licenseConfigSecretName }}
  //Mount the license volume to the container image
  volumeMounts:
    - name: awsmpl-product-license
      mountPath: "/var/run/secrets/product-license"
  //Add following environment variable to container for credential
provider
  env:
    - name: AWS_WEB_IDENTITY_REFRESH_TOKEN_FILE
      value: "/var/run/secrets/product-license/license_token"
    - name: AWS_ROLE_ARN
      valueFrom:
        secretKeyRef:
          name: {{ .Values.aws.licenseConfigSecretName }}
          key: iam_role
  //Mount the license secret as a volume to the pod
  volumes:
    - name: awsmpl-product-license
      secret:
        secretName: {{ .Values.aws.licenseConfigSecretName }}
        optional: true
{{ - end }}

```

Note

ライセンス設定シークレットはオプションです。購入者はこの値をオンプレミスデプロイにのみ使用します。AWS デプロイの場合、デプロイ仕様には License Manager 統合イメージのサービスアカウントが含まれている必要があります。

10. 以下のセクションのステップを実行して、License Manager 統合をローカルと Amazon EKS でテストします。
 - a. [License Manager 統合のローカルテスト](#)
 - b. [Amazon EKS での License Manager 統合のテスト](#)
11. License Manager の統合をオンプレミス AWS とオンプレミスの両方で正常に検証したら、「」の手順に従ってコンテナ製品リストを作成できます [概要: コンテナ製品を作成する](#)。

License Manager 統合のローカルテスト

minikube やその他のセットアップを使用して、ローカルで任意の Kubernetes クラスターで License Manager の統合をテストできます。Kubernetes クラスターに、License Manager API オペレーションを呼び出すためのアウトバウンドインターネットアクセスがあることを確認します。

License Manager 統合をローカルでテストするには

1. テスト用販売者アカウントで、希望する使用権限を持つテストライセンスを作成します。テストライセンスを設定するには、「AWS License Manager API リファレンス」の「[CreateLicense](#)」を参照してください。または、次のスクリプトを使用してテストライセンスを作成し、テスト購入者アカウントにライセンスを付与してライセンスを使用します。次のスクリプトでは、テスト用販売者アカウントの認証情報を使用しています。

```
read -p 'AWS Account for test buyer: ' TEST_BUYER_ACCOUNT_ID
read -p 'License entitlements: ' ENTITLEMENTS

# TEST_SELLER_ACCOUNT_ID="109876543210"
# ENTITLEMENTS="{\"Name\": \"ByData\", \"MaxCount\": 1000, \"Overage\": true, \"Unit\": \"Gigabits\", \"AllowCheckIn\": true}"

# Create License

NOW=$(date +"%Y-%m-%dT00:00:00+00:00")
```

```

PRODUCT_NAME="My awesome product"
PRODUCT_SKU="c97b7825-44c4-4f42-b025-12baa4c171e0"

LICENSE_BENEFICIARY=" arn:aws:iam::$TEST_BUYER_ACCOUNT_ID:root "
LICENSE_ISSUER_NAME="test-seller"
LICENSE_NAME="test-seller-license"

CLIENT_TOKEN="b3920968-a94f-4547-af07-3dd232319367"
CONSUMPTION_TTL=180
CONSUMPTION_RENEW_TYPE="None"

HOME_REGION="us-east-1"

LICENSE_ARN=$(aws license-manager create-license --license-name
"$LICENSE_NAME" --product-name "$PRODUCT_NAME" --product-sku
"$PRODUCT_SKU" --issuer Name="$LICENSE_ISSUER_NAME" --home-region
"$HOME_REGION" --validity Begin="$NOW" --entitlements "$ENTITLEMENTS"
--beneficiary "$LICENSE_BENEFICIARY" --consumption-configuration
RenewType="$CONSUMPTION_RENEW_TYPE",ProvisionalConfiguration={MaxTimeToLiveInMinutes=$
CONSUMPTION_TTL} --client-token "$CLIENT_TOKEN" | jq -r ".LicenseArn" )

echo "License arn: $LICENSE_ARN"

# Create Grant

GRANT_TOKEN="e9a14140-4fca-4219-8230-57511a6ea6"
GRANT_NAME="test-grant"

GRANT_ARN=$(aws license-manager create-grant --grant-name "$GRANT_NAME"
--license-arn "$LICENSE_ARN" --principals "$LICENSE_BENEFICIARY" --home-
region "$HOME_REGION" --client-token "$GRANT_TOKEN" --allowed-operations
"CheckoutLicense" "CheckInLicense" "ExtendConsumptionLicense" "CreateToken" | jq -
r ".GrantArn")

echo "Grant arn: $GRANT_ARN"

```

2. 以前に定義したシークレット形式を使用して、ライセンストークンと IAM ロールを使用して Kubernetes シークレットを作成します。License Manager CreateToken API オペレーションを使用してライセンストークンを生成します。次に、IAM CreateRole API オペレーションを使用して、アクセス許可と信頼ポリシーを含む IAM ロールを作成します。このスクリプトの例を参照してください。次のスクリプトでは、テスト用購入者アカウントの認証情報を使用しています。

```

read -p 'AWS Account for test license: ' TEST_ACCOUNT_ID
read -p 'License Arn' LICENSE_ARN
# Create IAM Role
ROLE_NAME="AWSLicenseManagerConsumptionTestRole"
ROLE_DESCRIPTION="Role to test AWS License Manager integration on-prem"
ROLE_POLICY_ARN="arn:aws:iam::aws:policy/service-role/
AWSLicenseManagerConsumptionPolicy"
ROLE_TRUST_POLICY="{\"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect\":
\"Allow\", \"Principal\": { \"Federated\": \"openid-license-manager.amazonaws.com
\" }, \"Action\": \"sts:AssumeRoleWithWebIdentity\", \"Condition\":
{ \"ForAnyValue:StringLike\": { \"openid-license-manager.amazonaws.com:amr\":
\"aws:license-manager:token-issuer-account-id:${TEST_ACCOUNT_ID}\" } } ]}"
ROLE_SESSION_DURATION=3600

ROLE_ARN=$(aws iam create-role --role-name "$ROLE_NAME" --description
"$ROLE_DESCRIPTION" --assume-role-policy-document "$ROLE_TRUST_POLICY" --max-
session-duration $ROLE_SESSION_DURATION | jq ".Role" | jq -r ".Arn")

aws iam attach-role-policy --role-name "$ROLE_NAME" --policy-arn "$ROLE_POLICY_ARN"

echo "Role arn: $ROLE_ARN"

# Create Token
CLIENT_TOKEN="b3920968-a94f-4547-af07-3dd232319367"

TOKEN=$(aws license-manager create-token --license-arn $LICENSE_ARN --role-arns
$ROLE_ARN --client-token $CLIENT_TOKEN | jq '.Token')

echo "License access token: $TOKEN"

```

3. 外部でホストされている Kubernetes クラスターをセットアップします AWS。これを使用して、コンテナアプリケーションが 以外の環境から AWS License Manager API に接続できること AWS 、およびカスタム認証情報プロバイダーがアプリケーションに適切に統合されていることをテストします。
4. 以前に生成したライセンストークンと IAM ロールをローカル Kubernetes クラスターにデプロイします。

```

kubectl create secret generic "awsmp-license-access-config" \
--from-literal=license_token=${TOKEN} \
--from-literal=iam_role=${ROLE_ARN}

```


5. シークレット名を入力としてアプリケーションを Helm 経由でデプロイし、アプリケーションが License Manager API オペレーションを呼び出して使用権限チェックを実行できることを確認します。Helm とデプロイ仕様の変更については、[AWS Marketplace for Containers Anywhere 製品を License Manager と統合する](#) の「ステップ 9」を参照してください。

Amazon EKS での License Manager 統合のテスト

Amazon EKS で License Manager の統合をテストすることもできます。ライセンス設定シークレットがなくても、アプリケーションが License Manager API オペレーションを呼び出せることをテストして確認します。また、サービスアカウントを使用してサービスアカウント (IRSA) の IAM ロールを設定し、関連する認証情報をアプリケーションに提供することも確認してください。

Amazon EKS で License Manager 統合をテストするには

1. テスト用販売者アカウントで、希望する使用権限を持つテストライセンスを作成します。
「[CreateLicense API リファレンス](#)」を参照してテストライセンスを設定するか、次のスクリプトを使用してライセンスを作成し、テスト購入者アカウントにライセンスを付与してライセンスを消費します。次のスクリプトでは、テスト用販売者アカウントの認証情報を使用しています。

```
read -p 'AWS Account for test buyer: ' TEST_BUYER_ACCOUNT_ID
read -p 'License entitlements: ' ENTITLEMENTS

# TEST_SELLER_ACCOUNT_ID="109876543210"
# ENTITLEMENTS="{\"Name\": \"ByData\", \"MaxCount\": 1000, \"Overage\": true, \"Unit\": \"Gigabits\", \"AllowCheckIn\": true}"

# Create License

NOW=$(date +"%Y-%m-%dT00:00:00+00:00")

PRODUCT_NAME="My awesome product"
PRODUCT_SKU="c97b7825-44c4-4f42-b025-12baa4c171e0"

LICENSE_BENEFICIARY=" arn:aws:iam::$TEST_BUYER_ACCOUNT_ID:root "
LICENSE_ISSUER_NAME="test-seller"
LICENSE_NAME="test-seller-license"

CLIENT_TOKEN="b3920968-a94f-4547-af07-3dd232319367"
CONSUMPTION_TTL=180
CONSUMPTION_RENEW_TYPE="None"
```

```
HOME_REGION="us-east-1"

LICENSE_ARN=$(aws license-manager create-license --license-name
"$LICENSE_NAME" --product-name "$PRODUCT_NAME" --product-sku
"$PRODUCT_SKU" --issuer Name="$LICENSE_ISSUER_NAME" --home-region
"$HOME_REGION" --validity Begin="$NOW" --entitlements "$ENTITLEMENTS"
--beneficiary "$LICENSE_BENEFICIARY" --consumption-configuration
RenewType="$CONSUMPTION_RENEW_TYPE",ProvisionalConfiguration={MaxTimeToLiveInMinutes=$
CONSUMPTION_TTL} --client-token "$CLIENT_TOKEN" | jq -r ".LicenseArn" )

echo "License arn: $LICENSE_ARN"

# Create Grant

GRANT_TOKEN="e9a14140-4fca-4219-8230-57511a6ea6"
GRANT_NAME="test-grant"

GRANT_ARN=$(aws license-manager create-grant --grant-name "$GRANT_NAME"
--license-arn "$LICENSE_ARN" --principals "$LICENSE_BENEFICIARY" --home-
region "$HOME_REGION" --client-token "$GRANT_TOKEN" --allowed-operations
"CheckoutLicense" "CheckInLicense" "ExtendConsumptionLicense" "CreateToken" | jq -
r ".GrantArn")

echo "Grant arn: $GRANT_ARN"
```

2. 必要な設定のテスト Amazon EKS クラスターを作成するか、以下のコマンドを実行してデフォルト設定を使用します。

```
aws ec2 create-key-pair --region us-west-2 --key-name eks-key-pair
```

```
eksctl create cluster \
--name awssmp-eks-test-example \
--region us-west-2 \
--with-oidc \
--ssh-access \
--ssh-public-key eks-key-pair
```

3. 既存のクラスターのサービスアカウントを作成し、それを IAM ロールに関連付けます。次のコマンドは、AWSLicenseManagerConsumptionPolicy を使用して IAM ロールを作成します。次に、コマンドは、License Manager 統合イメージをデプロイする必要がある Amazon EKS クラスターの test_sa サービスアカウントにアタッチします。その結果、サービスアカ

ウントは、License Manager API オペレーションを呼び出すための適切な認証情報を取得できません。

```
eksctl create iamserviceaccount \  
--name test_sa \  
--namespace test_namespace \  
--cluster awsm-p-eks-test-example \  
--attach-policy-arn "arn:aws:iam::aws:policy/service-role/  
AWSLicenseManagerConsumptionPolicy" \  
--approve \  
--override-existing-serviceaccounts
```

4. Helm を通じて、前のコマンドで IAM ロールが関連付けられているサービスアカウントにアプリケーションをデプロイします。アプリケーションが License Manager API オペレーションを呼び出して使用権限チェックを実行できることを確認します。

License Manager を使用したフローティングライセンス使用権限

フローティングライセンスでは、ユーザーがアプリケーションにログインすると、使用可能なライセンスのプールからライセンスが引き出されます。ユーザーがログアウトすると、ライセンスは使用可能なライセンスのプールに再び追加されます。

フローティングライセンスの場合、アプリケーションは CheckoutLicense API オペレーションを使用して、リソースが使用されているときに使用権限プールから使用権限をチェックアウトします。CheckoutLicense API オペレーションのレスポンスには、チェックアウト時の一意の識別子であるライセンス消費トークンが含まれます。ライセンス消費トークンは、チェックアウトした使用権限をライセンスプールに戻したり、チェックアウトを延長したりするなど、追加のアクションを実行できます。

リソースが使用されなくなると、アプリケーションは CheckInLicense API オペレーションを使用して使用権限をプールにチェックインし直します。

```
aws license-manager check-in-license \  
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

オペレーション中にアプリケーションがクラッシュした場合など、ライセンスをプールにチェックインし直せなかった場合、60 分後に使用権限が自動的にプールにチェックインされます。このため、リソースが 60 分以上使用されている場合は、使用権限をプールからチェックアウトしたままにしておくことがベストプラクティスです。そのためには、リソースが使用されている限り ExtendLicenseConsumption API オペレーションを使用してください。

```
aws license-manager extend-license-consumption \  
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

オンプレミスデプロイでの License Manager との統合に関するベストプラクティス

オンプレミス環境にコンテナアプリケーションをデプロイすると、信頼性の低いアウトバウンドネットワークアクセスが発生する可能性があります。以下のベストプラクティスを活用して回復性を高め、インターネット接続の不備による潜在的な問題による購入者のサービスの中断を回避してください。

- 適切な再試行 — 一時的なネットワークの問題により、アプリケーションがに接続できなくなる可能性があります AWS License Manager。最大で 30 分間の再試行をし、指数関数的にバックオフを行います。これにより、短期的な停止やネットワークの問題を回避できます。
- ハードリミットの回避 - 接続されたクラスターにデプロイされたアプリケーションは、定期的にライセンスをチェックして、アップグレードや更新による変更を特定できます。アウトバウンドアクセスの信頼性が低いと、アプリケーションはそれらの変更を識別できない可能性があります。可能な限り、アプリケーションは、License Manager を通じてライセンスを確認できないことによる購入者へのサービスの中断を避ける必要があります。ライセンスの有効期限が切れてライセンスが有効かどうかを確認できなくなると、アプリケーションは無料試用エクスペリエンスまたはオープンソースエクスペリエンスに頼ることができます。
- 顧客への通知 - キャッシュ型ライセンスを使用する場合、ライセンスへの変更 (更新やアップグレードを含む) は、実行中のワークロードに自動的に反映されません。アプリケーションがキャッシュされたライセンスを更新できるように、アプリケーションへのアウトバウンドアクセスを一時的に再度許可する必要があることを顧客に通知します。例えば、アプリケーション自体またはドキュメントを通じて顧客に通知します。同様に、下位の機能に頼る場合は、その使用権限がなくなったか、ライセンスの有効期限が切れていることを顧客に通知します。その後、顧客はアップグレードまたは更新のどちらかを選択できます。

LicenseManagerCredentialsProvider - Java 実装

LicenseCredentialsProvider は、を追加して AWS、SDK のデフォルトの認証情報プロバイダーチェーンをオンプレミス用に拡張しますLicenseManagerTokenCredentialsProvider。

LicenseCredentialsProvider

```
package com.amazon.awsmp.license;  
  
import software.amazon.awssdk.auth.credentials.AwsCredentials;
```

```
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProviderChain;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.auth.credentials.internal.LazyAwsCredentialsProvider;
import software.amazon.awssdk.utils.SdkAutoCloseable;

public class LicenseCredentialsProvider implements AwsCredentialsProvider,
    SdkAutoCloseable {
    private static final LicenseCredentialsProvider CREDENTIALS_PROVIDER = new
    LicenseCredentialsProvider();
    private final LazyAwsCredentialsProvider providerChain;

    private LicenseCredentialsProvider() {
        this.providerChain = createChain();
    }

    public static LicenseCredentialsProvider create() {
        return CREDENTIALS_PROVIDER;
    }

    @Override
    public AwsCredentials resolveCredentials() {
        return this.providerChain.resolveCredentials();
    }

    @Override
    public void close() {
        this.providerChain.close();
    }

    private LazyAwsCredentialsProvider createChain() {
        return LazyAwsCredentialsProvider.create(() -> {
            AwsCredentialsProvider[] credentialsProviders = new
            AwsCredentialsProvider[]{
                DefaultCredentialsProvider.create(),
                LicenseManagerTokenCredentialsProvider.create()};

            return AwsCredentialsProviderChain.builder().reuseLastProviderEnabled(true)
                .credentialsProviders(credentialsProviders).build();
        });
    }
}
```

LicenseManagerTokenCredentialsProvider

LicenseManagerTokenCredentialsProvider により、オンプレミス環境で License Manager OIDC が発行した ID トークンを使用して認証情報が提供されます。LicenseCredentialsProvider のソースコードをアプリケーションのクラスパスに含める必要があります。

```
package com.amazon.awsmp.license;

import software.amazon.awssdk.auth.credentials.AnonymousCredentialsProvider;
import software.amazon.awssdk.auth.credentials.AwsCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.core.SdkSystemSetting;
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.core.retry.RetryPolicyContext;
import software.amazon.awssdk.core.retry.conditions.OrRetryCondition;
import software.amazon.awssdk.core.retry.conditions.RetryCondition;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.regions.providers.DefaultAwsRegionProviderChain;
import software.amazon.awssdk.services.licensemanager.LicenseManagerClient;
import software.amazon.awssdk.services.licensemanager.model.GetAccessTokenRequest;
import software.amazon.awssdk.services.licensemanager.model.GetAccessTokenResponse;
import software.amazon.awssdk.services.sts.StsClient;
import
    software.amazon.awssdk.services.sts.auth.StsAssumeRoleWithWebIdentityCredentialsProvider;
import software.amazon.awssdk.services.sts.model.AssumeRoleWithWebIdentityRequest;
import software.amazon.awssdk.services.sts.model.IdpCommunicationErrorException;
import software.amazon.awssdk.utils.IoUtils;
import software.amazon.awssdk.utils.SdkAutoCloseable;
import software.amazon.awssdk.utils.StringUtils;
import software.amazon.awssdk.utils.SystemSetting;

import java.io.IOException;
import java.io.InputStream;
import java.io.UncheckedIOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Duration;
import java.util.function.Supplier;

public class LicenseManagerTokenCredentialsProvider implements AwsCredentialsProvider,
    SdkAutoCloseable {
```

```

private final StsAssumeRoleWithWebIdentityCredentialsProvider credentialsProvider;
private final RuntimeException loadException;

private Path licenseAccessTokenFile;
private String roleArn;
private String roleSessionName;
private StsClient stsClient;
private LicenseManagerClient lmClient;

public static LicenseManagerTokenCredentialsProvider create() {
    return new Builder().build();
}

@Override
public AwsCredentials resolveCredentials() {
    if (this.loadException != null) {
        throw this.loadException;
    }
    return this.credentialsProvider.resolveCredentials();
}

@Override
public void close() {
    IoUtils.closeQuietly(this.credentialsProvider, null);
    IoUtils.closeQuietly(this.stsClient, null);
    IoUtils.closeIfCloseable(this.lmClient, null);
}

private LicenseManagerTokenCredentialsProvider(Builder builder) {
    StsAssumeRoleWithWebIdentityCredentialsProvider credentialsProvider = null;
    RuntimeException loadException = null;

    try {
        this.licenseAccessTokenFile =
Paths.get(StringUtils.trim(LicenseSystemSetting.AWS_WEB_IDENTITY_REFRESH_TOKEN_FILE.getStringValue()),
        this.roleArn = SdkSystemSetting.AWS_ROLE_ARN.getStringValueOrNull();
        this.roleSessionName =
SdkSystemSetting.AWS_ROLE_SESSION_NAME.getStringValue().orElse("aws-sdk-java-" +
System.currentTimeMillis());
        this.stsClient = builder.stsClient != null ? builder.stsClient :
StsClientFactory.create();
        this.lmClient = builder.lmClient != null ? builder.lmClient :
LicenseManagerClientFactory.create();
    } catch (Exception e) {
        loadException = new RuntimeException(e);
    }
    credentialsProvider = new StsAssumeRoleWithWebIdentityCredentialsProvider(
        this.stsClient, this.roleArn, this.roleSessionName, this.licenseAccessTokenFile);
    this.credentialsProvider = credentialsProvider;
    this.loadException = loadException;
}

```

```
        AssumeRoleWithWebIdentityRequest request =
AssumeRoleWithWebIdentityRequest.builder()

        .roleArn(this.roleArn).roleSessionName(this.roleSessionName).build();

        Supplier<AssumeRoleWithWebIdentityRequest> supplier = new
AssumeRoleRequestSupplier(request,
                this.licenseAccessTokenFile, this.lmClient);

        credentialsProvider =
StsAssumeRoleWithWebIdentityCredentialsProvider.builder()
                .stsClient(this.stsClient).refreshRequest(supplier).build();
    } catch (RuntimeException ex) {
        loadException = ex;
    }

    this.credentialsProvider = credentialsProvider;
    this.loadException = loadException;
}

public static final class Builder {
    private Path licenseAccessTokenFile;
    private String roleArn;
    private String roleSessionName;
    private StsClient stsClient;
    private LicenseManagerClient lmClient;

    public LicenseManagerTokenCredentialsProvider build() {
        return new LicenseManagerTokenCredentialsProvider(this);
    }

    public LicenseManagerTokenCredentialsProvider.Builder
licenseAccessTokenFile(Path licenseAccessTokenFile) {
        this.licenseAccessTokenFile = licenseAccessTokenFile;
        return this;
    }

    public LicenseManagerTokenCredentialsProvider.Builder roleArn(String roleArn) {
        this.roleArn = roleArn;
        return this;
    }
}
```



```
        public LicenseManagerTokenCredentialsProvider.Builder roleSessionName(String
roleSessionName) {
            this.roleSessionName = roleSessionName;
            return this;
        }

        public LicenseManagerTokenCredentialsProvider.Builder stsClient(StsClient
stsClient) {
            this.stsClient = stsClient;
            return this;
        }

        public LicenseManagerTokenCredentialsProvider.Builder
lmClient(LicenseManagerClient lmClient) {
            this.lmClient = lmClient;
            return this;
        }
    }

    private static final class AssumeRoleRequestSupplier implements Supplier {
        private final LicenseManagerClient lmClient;
        private final AssumeRoleWithWebIdentityRequest request;
        private final Path webIdentityRefreshTokenFile;

        AssumeRoleRequestSupplier(final AssumeRoleWithWebIdentityRequest request,
                                final Path
webIdentityRefreshTokenFile,
                                final LicenseManagerClient lmClient) {

            this.lmClient = lmClient;
            this.request = request;
            this.webIdentityRefreshTokenFile = webIdentityRefreshTokenFile;
        }

        public AssumeRoleWithWebIdentityRequest get() {
            return this.request.toBuilder()
                .webIdentityToken(getIdentityToken())
                .build();
        }

        private String getIdentityToken() {
            return refreshIdToken(readRefreshToken(this.webIdentityRefreshTokenFile));
        }

        private String readRefreshToken(Path file) {
```

```

        try (InputStream webIdentityRefreshTokenStream =
Files.newInputStream(file)) {
            return IoUtils.toUtf8String(webIdentityRefreshTokenStream);
        } catch (IOException e) {
            throw new UncheckedIOException(e);
        }
    }

    private String refreshIdToken(String licenseRefreshToken) {
        final GetAccessTokenRequest request = GetAccessTokenRequest.builder()
            .token(licenseRefreshToken)
            .build();

        GetAccessTokenResponse response = this.lmClient.getAccessToken(request);
        return response.accessToken();
    }

    private static final class LicenseManagerClientFactory {
        private static final Duration DEFAULT_API_TIMEOUT = Duration.ofSeconds(30);
        private static final Duration DEFAULT_API_ATTEMPT_TIMEOUT =
Duration.ofSeconds(10);

        public static LicenseManagerClient create() {
            return getLicenseManagerClient();
        }

        private static LicenseManagerClient getLicenseManagerClient() {
            ClientOverrideConfiguration configuration =
ClientOverrideConfiguration.builder()
                .apiCallTimeout(DEFAULT_API_TIMEOUT)
                .apiCallAttemptTimeout(DEFAULT_API_ATTEMPT_TIMEOUT)
                .build();

            LicenseManagerClient client = LicenseManagerClient.builder()
                .region(configureLicenseManagerRegion())
                .credentialsProvider(AnonymousCredentialsProvider.create())
                .overrideConfiguration(configuration).build();
            return client;
        }

        private static Region configureLicenseManagerRegion() {
            Region defaultRegion = Region.US_EAST_1;

```

```
        Region region;
        try {
            region = (new DefaultAwsRegionProviderChain()).getRegion();
        } catch (RuntimeException ex) {
            region = defaultRegion;
        }
        return region;
    }
}

private static final class StsClientFactory {
    private static final Duration DEFAULT_API_TIMEOUT = Duration.ofSeconds(30);
    private static final Duration DEFAULT_API_ATTEMPT_TIMEOUT =
Duration.ofSeconds(10);

    public static StsClient create() {
        return getStsClient();
    }

    private static StsClient getStsClient() {
        OrRetryCondition retryCondition = OrRetryCondition.create(new
StsRetryCondition(),
            RetryCondition.defaultRetryCondition());

        ClientOverrideConfiguration configuration =
ClientOverrideConfiguration.builder()
            .apiCallTimeout(DEFAULT_API_TIMEOUT)
            .apiCallAttemptTimeout(DEFAULT_API_ATTEMPT_TIMEOUT)
            .retryPolicy(r -> r.retryCondition(retryCondition))
            .build();

        return StsClient.builder()
            .region(configureStsRegion())
            .credentialsProvider(AnonymousCredentialsProvider.create())
            .overrideConfiguration(configuration).build();
    }

    private static Region configureStsRegion() {
        Region defaultRegion = Region.US_EAST_1;
        Region stsRegion;
        try {
            stsRegion = (new DefaultAwsRegionProviderChain()).getRegion();
        } catch (RuntimeException ex) {
            stsRegion = defaultRegion;
        }
    }
}
```

```

        }
        return stsRegion;
    }

    private static final class StsRetryCondition implements RetryCondition {
        public boolean shouldRetry(RetryPolicyContext context) {
            return context.exception() instanceof IdpCommunicationErrorException;
        }
    }
}

private enum LicenseSystemSetting implements SystemSetting {
    AWS_WEB_IDENTITY_REFRESH_TOKEN_FILE("aws.webIdentityRefreshTokenFile");

    private String systemProperty;
    private String defaultValue = null;

    LicenseSystemSetting(String systemProperty) {
        this.systemProperty = systemProperty;
    }

    @Override
    public String property() {
        return this.systemProperty;
    }

    @Override
    public String environmentVariable() {
        return this.name();
    }

    @Override
    public String defaultValue() {
        return this.defaultValue;
    }
}
}

```

LicenseManagerCredentialsProvider - Golang 実装

LicenseCredentialsProvider

LicenseCredentialsProvider は、を追加して AWS、SDK のデフォルトの認証情報プロバイダーチェーンをオンプレミス用に拡張しますLicenseManagerTokenCredentialsProvider。

```
package lib

import (
    "context"
    "fmt"
    "sync"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
)

// LicenseCredentialsProvider is the custom credential provider that can retrieve valid
// temporary aws credentials
type LicenseCredentialsProvider struct {
    fallbackProvider aws.CredentialsProvider
    mux              sync.RWMutex
    licenseCredentials aws.Credentials
    err              error
}

// NewLicenseCredentialsProvider method will create a LicenseCredentialProvider Object
// which contains valid temporary aws credentials
func NewLicenseCredentialsProvider() (*LicenseCredentialsProvider, error) {
    licenseCredentialProvider := &LicenseCredentialsProvider{}
    fallbackProvider, err := createCredentialProvider()
    if err != nil {
        return licenseCredentialProvider, fmt.Errorf("failed to create
LicenseCredentialsProvider, %w", err)
    }
    licenseCredentialProvider.fallbackProvider = fallbackProvider
    return licenseCredentialProvider, nil
}

// Retrieve method will retrieve temporary aws credentials from the credential provider
func (l *LicenseCredentialsProvider) Retrieve(ctx context.Context) (aws.Credentials,
    error) {
    l.mux.RLock()
    defer l.mux.RUnlock()
    l.licenseCredentials, l.err = l.fallbackProvider.Retrieve(ctx)
    return l.licenseCredentials, l.err
}

func createCredentialProvider() (aws.CredentialsProvider, error) {
```

```
// LoadDefaultConfig will examine all "default" credential providers
ctx := context.TODO()
cfg, err := config.LoadDefaultConfig(ctx)
if err != nil {
    return nil, fmt.Errorf("failed to create FallBackProvider, %w", err)
}

var useFallbackProvider bool
if cfg.Credentials != nil {
    if _, err := cfg.Credentials.Retrieve(ctx); err != nil {
        // If the "default" credentials provider cannot retrieve credentials, enable
        fallback to customCredentialsProvider.
        useFallbackProvider = true
    }
} else {
    useFallbackProvider = true
}

if useFallbackProvider {
    customProvider, err := newLicenseManagerTokenCredentialsProvider()
    if err != nil {
        return cfg.Credentials, fmt.Errorf("failed to create fallBackProvider, %w", err)
    }
    // wrap up customProvider with CredentialsCache to enable caching
    cfg.Credentials = aws.NewCredentialsCache(customProvider)
}
return cfg.Credentials, nil
}
```

LicenseManagerTokenCredentialsProvider

LicenseManagerTokenCredentialsProvider により、オンプレミス環境で License Manager OIDC が発行した ID トークンを使用して認証情報が提供されます。LicenseCredentialsProvider のソースコードをアプリケーションのクラスパスに含める必要があります。

```
package lib

import (
    "context"
    "fmt"
    "io/ioutil"
    "os"
```

```

"sync"
"time"

"github.com/aws/aws-sdk-go-v2/aws"
"github.com/aws/aws-sdk-go-v2/config"
"github.com/aws/aws-sdk-go-v2/service/sts"
)

const awsRefreshTokenFilePathEnvVar = "AWS_LICENSE_ACCESS_FILE"

// licenseManagerTokenCredentialsProvider defines and contains
  StsAssumeRoleWithWebIdentityProvider
type licenseManagerTokenCredentialsProvider struct {
  stsCredentialProvider *stsAssumeRoleWithWebIdentityProvider
  mux                   sync.RWMutex
  licenseCredentials    aws.Credentials
  err                   error
}

// Retrieve method will retrieve credentials from credential provider.
// Make this method public to make this provider satisfies CredentialProvider interface
func (a *licenseManagerTokenCredentialsProvider) Retrieve(ctx context.Context)
  (aws.Credentials, error) {
  a.mux.RLock()
  defer a.mux.RUnlock()
  a.licenseCredentials, a.err = a.stsCredentialProvider.Retrieve(ctx)
  return a.licenseCredentials, a.err
}

// newLicenseManagerTokenCredentialsProvider will create and return
  a LicenseManagerTokenCredentialsProvider Object which wraps up
  stsAssumeRoleWithWebIdentityProvider
func newLicenseManagerTokenCredentialsProvider()
  (*licenseManagerTokenCredentialsProvider, error) {
  // 1. Retrieve variables From yaml environment
  envConfig, err := config.NewEnvConfig()
  if err != nil {
    return &licenseManagerTokenCredentialsProvider{}, fmt.Errorf("failed to create
LicenseManagerTokenCredentialsProvider, %w", err)
  }
  roleArn := envConfig.RoleARN
  var roleSessionName string
  if envConfig.RoleSessionName == "" {
    roleSessionName = fmt.Sprintf("aws-sdk-go-v2-%v", time.Now().UnixNano())
  }
}

```

```
} else {
    roleSessionName = envConfig.RoleSessionName
}
tokenFilePath := os.Getenv(awsRefreshTokenFilePathEnvVar)
b, err := ioutil.ReadFile(tokenFilePath)
if err != nil {
    return &licenseManagerTokenCredentialsProvider{}, fmt.Errorf("failed to create
LicenseManagerTokenCredentialsProvider, %w", err)
}
refreshToken := aws.String(string(b))

// 2. Create stsClient
cfg, err := config.LoadDefaultConfig(context.TODO())
if err != nil {
    return &licenseManagerTokenCredentialsProvider{}, fmt.Errorf("failed to create
LicenseManagerTokenCredentialsProvider, %w", err)
}
stsClient := sts.NewFromConfig(cfg, func(o *sts.Options) {
    o.Region = configureStsClientRegion(cfg.Region)
    o.Credentials = aws.AnonymousCredentials{}
})

// 3. Configure StsAssumeRoleWithWebIdentityProvider
stsCredentialProvider := newStsAssumeRoleWithWebIdentityProvider(stsClient, roleArn,
roleSessionName, refreshToken)

// 4. Build and return
return &licenseManagerTokenCredentialsProvider{
    stsCredentialProvider: stsCredentialProvider,
}, nil
}

func configureStsClientRegion(configRegion string) string {
    defaultRegion := "us-east-1"
    if configRegion == "" {
        return defaultRegion
    } else {
        return configRegion
    }
}
```


コンテナ製品の Amazon SNS 通知

通知を受け取るには、製品の作成時に AWS Marketplace 提供される の Amazon Simple Notification Service (Amazon SNS) トピックをサブスクライブできます。このトピックでは、製品の顧客のサブスクリプションの変更に関する通知が提供されます。例えば、これらの通知を使用すると、顧客がプライベートオファーを承諾した時期を知ることができます。

Note

製品作成プロセス中に、製品の Amazon SNS トピックが作成されます。通知をサブスクライブするには、Amazon SNS トピックの Amazon リソースネーム (ARN) が必要です (例: `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`)。ARN は、サーバー製品の販売者ポータルでは使用できません。[AWS Marketplace オペレーションチーム](#)に連絡して ARN をリクエストします。

次の Amazon SNS トピックはコンテナ製品で使用できます。

- [Amazon SNS トピック: `aws-mp-subscription-notification`](#) - このトピックは、購入者が製品をサブスクライブまたはサブスクライブを解除した時に通知します。これは、時間単位や長期の時間単位を含む時間単位の料金モデルで利用できます。

Amazon SNS トピック: `aws-mp-subscription-notification`

`aws-mp-subscription-notification` トピック内の各メッセージの形式は次のとおりです。

```
{
  "action": "<action-name>",
  "customer-identifier": " X01EXAMPLEX",
  "product-code": "n0123EXAMPLEXXXXXXXXXXXXX",
  "offer-identifier": "offer-abcexample123"
}
```

`<action-name>` は、通知によって異なります。指定できるアクションは以下のとおりです。

- `subscribe-success`
- `subscribe-fail`
- `unsubscribe-pending`

- `unsubscribe-success`

アクション `offer-identifier` が `subscribe-success` または の場合にのみ、 が通知に含まれます `subscribe-fail`。アクションが `unsubscribe-pending` または の場合、通知には含まれません `unsubscribe-success`。2024 年 1 月より前に作成されたオファーの場合、この識別子はプライベートオファーの通知にのみ含まれます。2024 年 1 月以降に作成されたオファーの場合、この識別子はプライベートオファーとパブリックオファーの両方を含むすべてのオファーの通知に含まれます。

オファータイプの詳細については、[DescribeEntity API](#) からのレスポンスまたは契約 [更新ダッシュボード](#) の契約のオファーの可視性を参照してください。

 Note

[DescribeEntity API](#) の場合、そのオファーのターゲットルールのファセットをターゲットとする がアカウント AWS アカウント で見つかった場合、これはプライベートオファーです。そのオファーのターゲットルールのファセットをターゲットとする が AWS アカウント アカウント にない場合は、パブリックオファーです。

Amazon SNS トピックへ Amazon SQS キューをサブスクライブする

Amazon SQS キューを指定された SNS トピックにサブスクライブすることを推奨します。SQS キューを作成し、そのキューをトピックにサブスクライブする方法の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[Amazon SNS トピックへ Amazon SQS キューをサブスクライブする](#)」を参照してください。

 Note

製品を販売する AWS アカウント ために使用される AWS Marketplace からのみ SNS トピックをサブスクライブできます。ただし、メッセージを別のアカウントに転送することはできません。詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[別のアカウントの Amazon SQS キューへ Amazon SNS メッセージを送信する](#)」を参照してください。

通知の SQS キューのポーリング

SQS キューを SNS トピックにサブスクライブすると、メッセージは SQS に保存されます。継続的にキューをポーリングし、メッセージを探して必要に応じて処理するサービスを定義する必要があります。

の機械学習製品 AWS Marketplace

AWS Marketplace 販売者は、購入者がデプロイできる機械学習 (ML) アルゴリズムとモデルを作成できます AWS。このトピックでは、 にリストされている Amazon SageMaker AI 製品タイプについて説明します AWS Marketplace。

SageMaker AI 製品には 2 種類あります AWS Marketplace。

モデルパッケージ

購入者による追加のトレーニングを必要としない、予測を行うための事前トレーニング済みモデル。

アルゴリズム

予測を行う前に、購入者がトレーニングデータを提供する必要があるモデル。トレーニングアルゴリズムは含まれています。

これらの製品は、Amazon SageMaker AI コンソールまたは を通じて購入者が利用できます AWS Marketplace。購入者は、製品説明、ドキュメント、カスタマーレビュー、料金表、サポート情報を確認できます。モデルパッケージ製品またはアルゴリズム製品をサブスクライブすると、SageMaker AI コンソールの製品リストに追加されます。購入者は、AWS SDKs、AWS Command Line Interface (AWS CLI)、または SageMaker AI コンソールを使用して、フルマネージド REST 推論エンドポイントを作成したり、データのバッチに対して推論を実行したりすることもできます。

Amazon SageMaker AI を使用した機械学習製品の作成のサポートについては、 [AWS Marketplace Seller Operations](#) チームにお問い合わせください。

機械学習製品について

AWS Marketplace は、Amazon SageMaker AI を使用する 2 つの機械学習製品タイプをサポートしています。モデルパッケージ製品とアルゴリズム製品のどちらのタイプでも、予測を行うためのデプロイ可能な推論モデルが生成されます。

SageMaker AI モデルパッケージ

[Amazon SageMaker AI モデルパッケージ](#)製品には、事前トレーニング済みのモデルが含まれています。事前トレーニング済みのモデルを SageMaker AI にデプロイして、リアルタイムまたはバッチで

推論または予測を行うことができます。本製品には、モデルアーティファクト (存在する場合) を含むトレーニング済みの推論コンポーネントが含まれています。販売者は、SageMaker AI を使用してモデルをトレーニングすることも、独自のモデルを導入することもできます。

SageMaker AI アルゴリズム

購入者は [SageMaker AI アルゴリズム](#) 製品を使用して、完全な機械学習ワークロードを実行できます。アルゴリズム製品には、トレーニングと推論という 2 つの論理コンポーネントがあります。SageMaker AI では、購入者は独自のデータセットを使用して、トレーニングコンポーネントでトレーニングジョブを作成します。アルゴリズムがコンポーネントのトレーニングを完了すると、機械学習モデルのモデルアーティファクトが生成されます。SageMaker AI は、モデルアーティファクトを購入者の Amazon Simple Storage Service (Amazon S3) バケットに保存します。SageMaker AI では、購入者は、生成されたモデルアーティファクトとともに推論コンポーネントをデプロイして、リアルタイムまたはバッチで推論 (または予測) を実行できます。

推論モデルのデプロイ

推論モデルがモデルパッケージから作成されたものであれ、アルゴリズムから作成されたものであれ、それらをデプロイするには次の 2 つの方法があります。

- エンドポイント – このメソッドは SageMaker AI を使用してモデルをデプロイし、API エンドポイントを作成します。購入者はこのエンドポイントをバックエンドサービスの一部として使用してアプリケーションを強化できます。データがエンドポイントに送信されると、SageMaker AI はそれをモデルコンテナに渡して API レスポンスで結果を返します。エンドポイントとコンテナは、購入者が停止するまで実行され続けます。

Note

AWS Marketplace エンドポイントメソッドはリアルタイム推論と呼ばれ、SageMaker AI ドキュメントではホスティングサービスと呼ばれます。詳細については、[Amazon SageMaker でモデルをデプロイする](#)を参照してください。

- バッチ変換ジョブ – この方法では、購入者は推論用データセットを Amazon S3 に保存します。バッチ変換ジョブが開始されると、SageMaker AI はモデルをデプロイし、S3 バケットからモデルのコンテナにデータを渡し、その結果を Amazon S3 バケットに返します。ジョブが完了すると、SageMaker AI はジョブを停止します。詳細については、「[Use Batch Transform](#)」を参照してください。

Note

SageMaker AI はデータをモデルに渡し、結果を購入者に返すため、どちらの方法もモデルに対して透過的です。

機械学習製品のライフサイクル

の機械学習製品は、1 つ以上のソフトウェアバージョンと関連するメタデータ AWS Marketplace で構成されます。製品設定には、名前、説明、使用手順、料金、分類、検索キーワードなどの重要なプロパティが含まれます。

機械学習製品の作成プロセス

で機械学習製品を一覧表示するには AWS Marketplace、以下を完了する必要があります。

1. [the section called “SageMaker AI で製品を準備する”](#)
2. [the section called “での製品の一覧表示 AWS Marketplace”](#)

機械学習製品を作成したら、製品を編集および管理できます。詳細については、「[the section called “製品の管理”](#)」を参照してください。

機械学習製品のステータス

新製品の可視性は当初は限られており、許可リストに登録されたアカウントと製品作成者のみがアクセスできます。テストと検証の後、製品を公開して、すべての購入者が AWS Marketplace カタログで利用できるようにします。の製品は、次のステータス値を持つ AWS Marketplace ことができます。

ステータス**定義****Staging**

このステータスは、まだ情報を追加している不完全な製品を示します。セルフサービスエクスペリエンスを最初に保存して終了すると、は、完了したステップの情報を含む未公開の製品 AWS Marketplace を作成します。このス

ステータス

定義

ステータスから、引き続き情報を追加したり、送信された詳細を変更したりできます。

Limited

A product attains this status after it's submitted to AWS Marketplace and passes all validation checks. At this point, the product has a detail page accessible only to your account and allowlisted entities. You can conduct product testing through this detail page.

Public

When you're prepared to make your product visible to buyers for subscription, update the product visibility in the console. Once processed, the product transitions from Limited to Public status. For information about AWS guidelines, see [the section called “要件とベストプラクティス”](#).

Restricted

新しいユーザーが製品をサブスクライブできないようにするには、可視性設定を更新することで製品を制限できます。制限付きステータスでは、既存の許可リストに登録されたユーザーが製品を引き続き使用できますが、一般公開されたり、新しいユーザーが使用したりすることはできなくなります。

詳細またはサポートについては、[AWS Marketplace Seller Operations チーム](#)にお問い合わせください。

の機械学習製品の料金 AWS Marketplace

では、Amazon SageMaker AI 製品の利用可能な料金モデルから選択できます AWS Marketplace。製品をサブスクライブする購入者は、独自の SageMaker AI で製品を実行します AWS アカウント。購入者に提示される料金は、購入者の AWS アカウント で稼働するリソースのインフラストラクチャコストと、設定した製品価格の組み合わせです。以下のセクションでは、の SageMaker AI 製品の料金モデルについて説明します。 AWS Marketplace

トピック

- [インフラストラクチャの料金](#)
- [ソフトウェアの料金](#)

インフラストラクチャの料金

購入者は、製品の使用中に SageMaker AI のすべてのインフラストラクチャコストを負担する責任があります。これらのコストは によって設定 AWS され、[Amazon SageMaker AI の料金](#)ページで確認できます。

ソフトウェアの料金

購入者が製品の使用に対して AWS Marketplace 課金するソフトウェア料金を決定します。機械学習製品を AWS Marketplaceに追加する際に、料金と条件を設定します。

インスタンスタイプごとのすべてのインフラストラクチャとソフトウェアの料金は、購入者がサブスクライブ AWS Marketplace する前に、 の製品リストページに表示されます。

トピック

- [料金無料](#)
- [時間単位の料金](#)
- [推論の料金](#)
- [無料トライアル](#)

料金無料

製品を無料で提供することもできます。この場合、購入者はインフラストラクチャの料金のみを支払います。

時間単位の料金

SageMaker AI で実行されているソフトウェアのインスタンスごとに 1 時間あたりの料金で製品を提供できます。ソフトウェアが実行するインスタンスタイプごとに異なる時間単位の料金を請求できます。購入者がソフトウェアを実行している間、 は使用状況 AWS Marketplace を追跡し、それに応じて購入者に請求します。使用量は分単位で計算されます。

モデルパッケージ製品の場合、購入者はソフトウェアを 2 つの方法で実行できます。エンドポイントを継続的にホストしてリアルタイムの推論を実行する方法と、データセットに対してバッチ変換

ジョブを実行する方法です。購入者がソフトウェアを実行する 2 つの方法にそれぞれ異なる料金を設定できます。

アルゴリズム製品では、前述のように推論の実行に対して料金を決定するだけでなく、トレーニングジョブに対しても時間単位の料金を決定します。トレーニングイメージがサポートするインスタンスタイプごとに異なる時間単位の料金を請求できます。

推論の料金

購入者がエンドポイントをホストしてソフトウェアを実行し、リアルタイムの推論を継続的に実行する場合、推論ごとに料金を設定することができます。

Note

バッチ変換プロセスでは、常に時間単位の料金が適用されます。アルゴリズム製品のトレーニングジョブでも、常に時間単位の料金が適用されます。これらの料金は、推論の料金と無関係に、また相互にも無関係に設定できます。

デフォルトでは、推論料金では、はエンドポイントの呼び出しごとに購入者に AWS Marketplace 課金します。ただし、ソフトウェアが 1 回の呼び出しで推論のバッチを処理する場合もあります (ミニバッチとも呼ばれます)。エンドポイントのデプロイでは、1 回の呼び出しに対して AWS Marketplace が購入者に請求する推論の数をカスタムで指定できます。これを行うには、次の例のように、呼び出しの HTTP レスponsヘッダーにカスタムメタリングヘッダーを含めます。この例では、3 件の推論を購入者に請求する呼び出しを示しています。

```
X-Amzn-Inference-Metering: {"Dimension": "inference.count", "ConsumedUnits": 3}
```

Note

推論料金では、HTTP レスponsコードが 2XX であるリクエストに対して AWS Marketplace のみ購入者に課金されます。

無料トライアル

オプションで、製品の無料トライアルを作成し、無料トライアルの日数を定義できます。無料トライアルは 5~31 日です。無料トライアル期間中、購入者はソフトウェアを好きなだけ実行でき、ソフ

トウェア料金は発生しません。無料トライアル中、インフラストラクチャの料金は発生します。トライアル期間が終了すると、通常のソフトウェア料金とインフラストラクチャの料金が請求されます。

購入者が無料トライアルのある製品をサブスクライブすると、ウェルカムメールメッセージが届きます。メッセージには、無料トライアルの期間、計算された有効期限、サブスクリプションの解除に関する詳細が含まれます。有効期限が切れる 3 日前に通知メールが送信されます。

で製品の無料トライアルを提供する場合 AWS Marketplace、無料トライアルの特定の[返金ポリシー](#)に同意したことになります。

 Note

機械学習のプライベートオファーについて詳しくは、「[プライベートオファー](#)」をご覧ください。

での機械学習製品のサービス制限とクォータ AWS Marketplace

このセクションでは、AWS Marketplaceの機械学習 (ML) 製品の制限とクォータについて説明します。

トピック

- [ネットワークの隔離](#)
- [イメージのサイズ](#)
- [ストレージサイズ](#)
- [インスタンスサイズ](#)
- [推論のペイロードサイズ](#)
- [推論の処理時間](#)
- [Service Quotas](#)
- [非同期推論](#)
- [サーバーレス推論](#)
- [マネージドスポットトレーニング](#)
- [Docker イメージと AWS アカウント](#)
- [組み込みアルゴリズムまたは からのモデルパッケージの公開 AWS Marketplace](#)
- [公開 AWS リージョン でサポート](#)

ネットワークの隔離

セキュリティ上の理由から、コンテナ化された製品を購入者がサブスクライブすると、Docker コンテナはネットワークアクセスのない隔離された環境で実行されます。コンテナの作成時には、インターネット経由での呼び出しに依存しないでください。その呼び出しは失敗します。への呼び出しも失敗 AWS のサービス します。

イメージのサイズ

Docker イメージのサイズは、Amazon Elastic Container Registry (Amazon ECR) の [Service Quotas](#) によって管理されます。Docker イメージのサイズは、トレーニングジョブ、バッチ変換ジョブ、エンドポイント作成時の起動時間に影響します。パフォーマンスを向上させるため、最適な Docker イメージのサイズを維持してください。

ストレージサイズ

エンドポイントを作成すると、Amazon SageMaker AI はエンドポイントをホストする各 ML コンピューティングインスタンスに Amazon Elastic Block Store (Amazon EBS) ストレージボリュームをアタッチします。(エンドポイントは、リアルタイム推論または Amazon SageMaker AI ホスティングサービスとも呼ばれます)。ストレージボリュームのサイズは、インスタンスタイプによって異なります。詳細については、[「Amazon SageMaker AI デベロッパーガイド」の「インスタンスストレージボリュームのホスト」](#)を参照してください。Amazon SageMaker

バッチ変換については、Amazon SageMakerデベロッパーガイド」の [「バッチ変換でのストレージ」](#)を参照してください。

インスタンスサイズ

SageMaker AI は、さまざまな ML ユースケースに合わせて最適化されたインスタンスタイプの選択を提供します。インスタンスタイプは、CPU、GPU、メモリ、およびネットワーク容量のさまざまな組み合わせで構成されます。インスタンスタイプでは、ML モデルの構築、トレーニング、デプロイに適したリソースの組み合わせを柔軟に選択できます。詳細については、[Amazon SageMaker AI ML インスタンスタイプ](#)」を参照してください。

推論のペイロードサイズ

エンドポイントの場合、1 回の呼び出しあたりの入力データの最大サイズが 6 MB に制限されます。この値を調整することはできません。

バッチ変換の場合、1 回の呼び出しあたりの入力データの最大サイズは 100 MB です。この値を調整することはできません。

推論の処理時間

エンドポイントの場合、1回の呼び出しあたりの最大処理時間は 60 秒です。この値を調整することはできません。

バッチ変換の場合、1回の呼び出しあたりの最大処理時間は 60 分です。この値を調整することはできません。

Service Quotas

トレーニングと推論に関連するクォータの詳細については、[Amazon SageMaker AI Service Quotas](#)」を参照してください。

非同期推論

で公開されたモデルパッケージとアルゴリズムは、[Amazon SageMaker AI 非同期推論](#)用に設定されたエンドポイントにデプロイ AWS Marketplace できません。非同期推論用に設定されたエンドポイントの場合、モデルにネットワーク接続が必要です。すべての AWS Marketplace モデルはネットワーク分離で動作します。詳細については、「[No network access](#)」を参照してください。

サーバーレス推論

で発行されたモデルパッケージとアルゴリズムは、Amazon AWS Marketplace SageMaker AI Serverless Inference 用に設定されたエンドポイントにデプロイできません。[Amazon SageMaker](#) サーバーレス推論用に設定されたエンドポイントの場合、モデルにネットワーク接続が必要です。すべての AWS Marketplace モデルはネットワーク分離で動作します。詳細については、「[No network access](#)」を参照してください。

マネージドスポットトレーニング

からのすべてのアルゴリズムについて AWS Marketplace、[マネージドスポットトレーニング](#)のチェックポイントが実装されている場合でも、の値は 3,600 秒 (60 分) にMaxWaitTimeInSeconds設定されます。この値を調整することはできません。

Docker イメージと AWS アカウント

公開するには、イメージを AWS アカウント 販売者の が所有する Amazon ECR リポジトリに保存する必要があります。別の が所有するリポジトリに保存されているイメージを公開することはできません AWS アカウント。

組み込みアルゴリズムまたは からのモデルパッケージの公開 AWS Marketplace

[Amazon SageMaker AI 組み込みアルゴリズム](#) または AWS Marketplace サブスクリプションのアルゴリズムを使用してトレーニングジョブから作成されたモデルパッケージは公開できません。

トレーニングジョブのモデルアーティファクトは引き続き使用できますが、モデルパッケージを公開するには独自の推論イメージが必要です。

公開 AWS リージョン でサポート

AWS Marketplace は、以下の両方 AWS リージョン が当てはまる からのモデルパッケージとアルゴリズムリソースの発行をサポートします。

- [Amazon SageMaker AI がサポート](#) するリージョン
- デフォルトでオプトインされている [利用可能なリージョン](#) (例えば、[describe-regions](#) が返す "OptInStatus": "opt-in-not-required" など)

モデルパッケージまたはアルゴリズム製品の公開に必要なすべてのアセットは、公開元と同じリージョンに保存する必要があります。これには以下が含まれます。

- Amazon SageMaker AI で作成されるモデルパッケージとアルゴリズムリソース
- Amazon ECR リポジトリにアップロードされる推論イメージとトレーニングイメージ
- Amazon Simple Storage Service に保存され、モデルパッケージリソースのモデルデプロイ中に動的にロードされるモデルアーティファクト (存在する場合)
- Amazon S3 に保存されている推論およびトレーニング検証用のテストデータ

SageMaker AI でサポートされている任意のリージョンで製品を開発およびトレーニングできます。ただし、公開する前に、AWS Marketplace が発行元をサポートしているリージョンにすべてのアセットをコピーして、そのリージョンでリソースを再作成する必要があります。

Amazon SageMaker AI のセキュリティと知的財産

Amazon SageMaker AI は、 から取得したモデルとアルゴリズムの知的財産と購入者データの両方を保護します AWS Marketplace。以下のセクションでは、SageMaker AI が知的財産を保護する方法と顧客データのセキュリティについて詳しく説明します。

トピック

- [知的財産の保護](#)

- [ネットワークアクセスなし](#)
- [顧客データのセキュリティ](#)

知的財産の保護

製品を作成すると、コードは Docker コンテナイメージにパッケージ化されます。詳細については、このガイドで後述する「[SageMaker AI で製品を準備する](#)」を参照してください。コンテナイメージをアップロードすると、イメージとアーティファクトは転送中も保存時も暗号化されます。また、イメージは公開前にスキャンされ、脆弱性がないことを確認されます。

知的財産を保護するために、SageMaker AI では購入者のみが AWS のサービス エンドポイントを通じて製品にアクセスできます。購入者は、コンテナイメージやモデルアーティファクトに直接アクセスしたり、コンテナイメージやモデルアーティファクトを取得したりすることはできません。また、基盤となるインフラストラクチャにアクセスすることはできません。

ネットワークアクセスなし

購入者が作成する SageMaker AI モデルとアルゴリズムとは異なり、購入者が製品を起動すると AWS Marketplace、モデルとアルゴリズムはネットワークアクセスなしでデプロイされます。SageMaker AI は、ネットワークまたは AWS のサービス エンドポイントにアクセスできない環境にイメージをデプロイします。例えば、コンテナイメージは、インターネット、[VPC エンドポイント](#)、またはその他の AWS のサービスへのアウトバウンド API 呼び出しを行うことができません。

顧客データのセキュリティ

製品は購入者の内で SageMaker AI で実行されます AWS アカウント。そのため、購入者が製品を使用してデータ推論を行っても、販売者はデータにアクセスできません。

アルゴリズム製品の場合、トレーニングジョブのたびにトレーニングイメージからモデルアーティファクトが出力されます。モデルアーティファクトは購入者のアカウントに保存されます。トレーニングジョブのモデルアーティファクトは、購入者が顧客の推論イメージを使用してモデルをデプロイするときに使用されます。モデルアーティファクトに含まれる可能性のある知的財産を保護するには、公開前に暗号化してください。

Important

このセキュリティモデルにより、実行時にコードがインターネットにアクセスするのを防ぎます。したがって、コードはインターネット上のリソースやライブラリを使用できない

め、依存関係は Docker コンテナイメージにパッケージ化してください。トレーニングジョブから出力されたアーティファクトを暗号化する場合、これは特に重要です。ランタイムには、アーティファクトを暗号化および復号化するキーにインターネット経由でアクセスすることはできません。これらは、イメージとパッケージ化する必要があります。

詳細については、[Amazon SageMakerのセキュリティ](#)」を参照してください。

の機械学習レポート AWS Marketplace

AWS Marketplace は、購入者、財務、使用状況、税金に関するデータを含む Amazon SageMaker AI 製品のレポートを生成します。すべてのレポートは、[レポートページの](#) AWS Marketplace 管理ポータル で利用できます。詳細については、「[販売者レポート](#)」を参照してください。以下のセクションでは、機械学習製品レポートの概要について説明します。

トピック

- [日別ビジネスレポート](#)
- [月別収益レポート](#)
- [支払いレポート](#)
- [その他のレポートと分析](#)

日別ビジネスレポート

日次ビジネスレポートには、各購入者と製品のインスタンスタイプ、使用時間、ソフトウェア料金による収益、その他の詳細が記載されています。購入者は AWS アカウント ID で識別できます。詳細については、「[日次ビジネスレポート](#)」を参照してください。

月別収益レポート

月次収益レポートには、ソフトウェアの使用に対して購入者に請求された月次収益が表示されます。詳細については、「[月別請求済み収益レポート](#)」を参照してください。

支払いレポート

月次支払いレポートには、ソフトウェア料金の決済期間中にお客様に代わって収集された総額の内訳が表示されます。レポートに反映される決済総額は、お客様の銀行口座に入金される金額と一致する必要があります。詳細については、「[支払いレポート](#)」を参照してください。

その他のレポートと分析

利用可能なその他のレポートについては、「[販売者レポート](#)」を参照してください。

AWS Marketplaceから利用可能な [の販売者配信データフィード AWS Marketplace](#) を使用してカスタムレポートを作成することもできます。

SageMaker AI で製品を準備する

で製品を公開する前に AWS Marketplace、Amazon SageMaker AI で準備する必要があります。にリストされている SageMaker AI 製品には、モデルパッケージとアルゴリズムの AWS Marketplace 2 種類があります。詳細については、「[の機械学習製品 AWS Marketplace](#)」を参照してください。このトピックでは、製品を準備するために必要な 3 つのステップの概要を説明します。

1. [での機械学習製品のイメージへのコードのパッケージ化 AWS Marketplace](#) - モデルパッケージまたはアルゴリズム製品を準備するには、製品用の Docker コンテナイメージを作成する必要があります。
2. [Amazon Elastic Container Registry へのイメージのアップロード](#) - コードをコンテナイメージにパッケージ化してローカルでテストしたら、イメージをアップロードしてスキャンし、既知の脆弱性を持っていないかを確認します。続行する前に、すべての脆弱性を修正してください。
3. [Amazon SageMaker AI リソースの作成](#) - イメージが正常にスキャンされたら、それらを使用して SageMaker AI でモデルパッケージまたはアルゴリズムリソースを作成できます。

での機械学習製品のイメージへのコードのパッケージ化 AWS Marketplace

の機械学習製品はAmazon SageMaker AI AWS Marketplace を使用して、購入者に提供する機械学習ロジックを作成して実行します。SageMaker AI は、ロジックを含む Docker コンテナイメージを実行します。SageMaker AI は、安全でスケーラブルなインフラストラクチャでこれらのコンテナを実行します。詳細については、「[Amazon SageMaker AI のセキュリティと知的財産](#)」を参照してください。以下のセクションでは、SageMaker AI の Docker コンテナイメージにコードをパッケージ化する方法について説明します。

トピック

- [どのタイプのコンテナイメージを作成すればよいですか？](#)
- [モデルパッケージイメージの作成](#)
- [アルゴリズムイメージの作成](#)

どのタイプのコンテナイメージを作成すればよいですか？

コンテナイメージには、推論イメージとトレーニングイメージの2つのタイプがあります。

モデルパッケージ製品を作成するには、推論イメージのみが必要です。詳細な手順については、「[モデルパッケージイメージの作成](#)」を参照してください。

アルゴリズム製品を作成するには、トレーニングイメージと推論イメージの両方が必要です。詳細な手順については、「[アルゴリズムイメージの作成](#)」を参照してください。

コードをコンテナイメージに適切にパッケージ化するには、コンテナが SageMaker AI ファイル構造に準拠している必要があります。サービスがコンテナとデータをやり取りできるよう、コンテナは正しいエンドポイントを公開する必要があります。以下のセクションでは、このプロセスを詳しく説明します。

Important

セキュリティ上の理由から、コンテナ化された製品を購入者がサブスクライブすると、Docker コンテナはインターネット接続のない隔離された環境で実行されます。コンテナの作成時には、インターネット経由での呼び出しに依存しないでください。その呼び出しは失敗します。への呼び出しも失敗 AWS のサービス します。詳細については「[Amazon SageMaker AI のセキュリティと知的財産](#)」セクションを参照してください。

必要に応じて、推論イメージとトレーニングイメージを作成するときに、開始点として、[使用可能な深層学習コンテナイメージ](#)のコンテナを使用します。イメージは既にさまざまな機械学習フレームワークで適切にパッケージ化されています。

モデルパッケージイメージの作成

Amazon SageMaker AI モデルパッケージは、予測を行う事前トレーニング済みのモデルであり、購入者による追加のトレーニングは必要ありません。SageMaker AI でモデルパッケージを作成し、機械学習製品を公開できます AWS Marketplace。以下のセクションでは、のモデルパッケージを作成する方法について説明します AWS Marketplace。これには、コンテナイメージの作成、イメージのローカルでのビルドとテストが含まれます。

トピック

- [概要](#)

• [モデルパッケージ用の推論イメージの作成](#)

概要

モデルパッケージには、次のコンポーネントが含まれています。

- [Amazon Elastic Container Registry](#) (Amazon ECR) に保存された推論イメージ
- (オプション) [Amazon S3](#) に個別に保存されているモデルアーティファクト

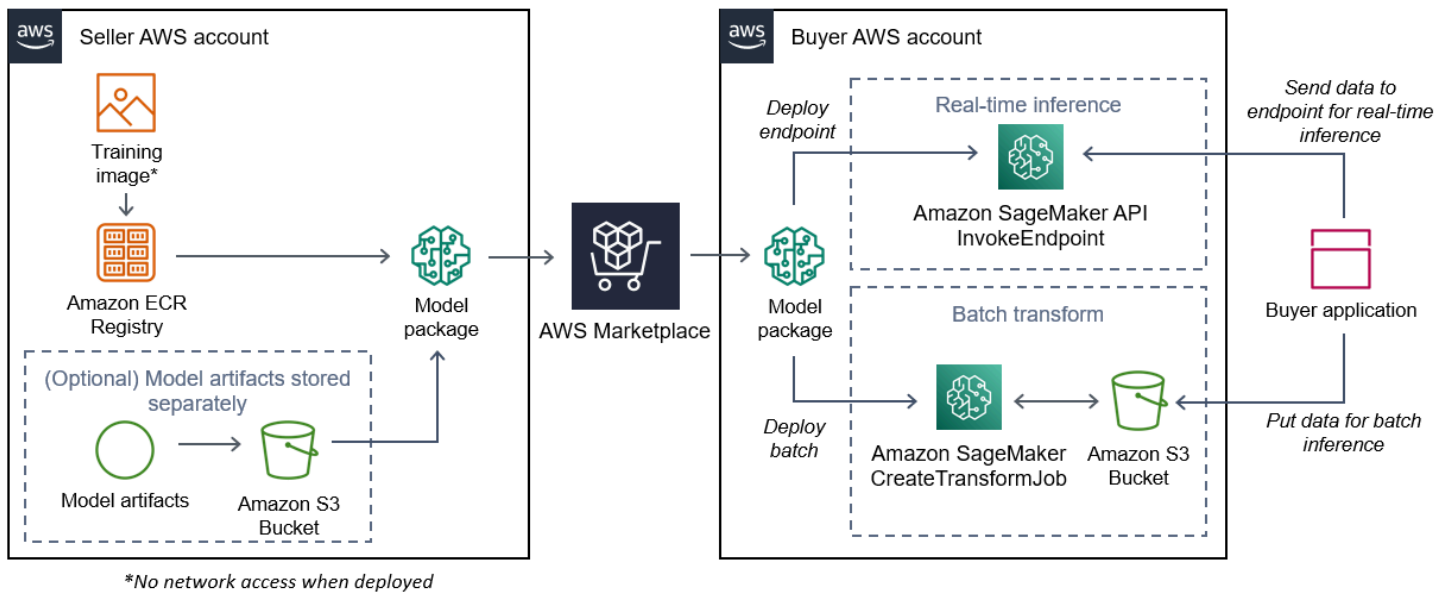
Note

モデルアーティファクトは、モデルが予測を行うために使用するファイルであり、通常は独自のトレーニングプロセスの結果です。アーティファクトは、モデルが必要とするどのようなファイルタイプでも構いませんが、.tar.gz 圧縮を使用する必要があります。モデルパッケージの場合、推論イメージ内にバンドルすることも、Amazon SageMaker AI に個別に保存することもできます。Amazon S3 に保存されているモデルアーティファクトは、実行時に推論コンテナにロードされます。モデルパッケージを公開すると、これらのアーティファクトは購入者が直接アクセスできない AWS Marketplace 所有の Amazon S3 バケットに公開され、保存されます。

Tip

推論モデルが Gluon、Keras、MXNet、PyTorch、TensorFlow、TensorFlow-Lite、ONNX などの深層学習フレームワークで構築されている場合は、Amazon SageMaker AI Neo の使用を検討してください。Neo は、ml.c4、ml.p2 などの、クラウドインスタンスタイプに特定のファミリにデプロイする推論モデルを自動的に最適化できます。詳細については、Amazon SageMaker AI デベロッパーガイド」の「[Neo を使用してモデルのパフォーマンスを最適化](#)する」を参照してください。

次の図は、モデルパッケージ製品を公開して使用するワークフローを示しています。



の SageMaker AI モデルパッケージを作成するためのワークフロー AWS Marketplace には、次のステップが含まれます。

1. 販売者は推論イメージ (デプロイ時のネットワークアクセス不可) を作成し、Amazon ECR レジストリにプッシュします。

モデルアーティファクトは推論イメージにバンドルすることも S3 に個別に保存することもできます。

2. 次に、販売者は Amazon SageMaker AI でモデルパッケージリソースを作成し、ML 製品を公開します AWS Marketplace。
3. 購入者は ML 製品をサブスクライブしてモデルをデプロイします。

Note

このモデルは、リアルタイムの推論を行うエンドポイントとしてデプロイすることも、データセット全体の予測を一度に取得するバッチジョブとしてデプロイすることもできます。詳細については、「[推論のためのモデルをデプロイする](#)」を参照してください。

4. SageMaker AI は推論イメージを実行します。推論イメージにバンドルされていない販売者提供のモデルアーティファクトは、ランタイムに動的に読み込まれます。
5. SageMaker AI は、コンテナの HTTP エンドポイントを使用して購入者の推論データをコンテナに渡し、予測結果を返します。

モデルパッケージ用の推論イメージの作成

このセクションでは、推論コードをモデルパッケージ製品の推論イメージにパッケージ化する手順を説明します。そのプロセスは、以下のステップで構成されています。

ステップ

- [ステップ 1: コンテナイメージを作成する](#)
- [ステップ 2: イメージをローカルでビルドしてテストする](#)

推論イメージは、推論ロジックを含む Docker イメージです。実行時にコンテナは HTTP エンドポイントを公開し、SageMaker AI がコンテナとの間でデータを送受信できるようにします。

Note

以下は、推論イメージのパッケージコードの一例です。詳細については、GitHub の [SageMaker AI での Docker コンテナの使用](#) および [AWS Marketplace SageMaker AI の例](#) を参照してください。

以下の例ではわかりやすくするために [Flask](#) というウェブサービスを使用していますが、本番環境に対応しているとは見なされていません。

ステップ 1: コンテナイメージを作成する

推論イメージを SageMaker AI と互換性を持たせるには、Docker イメージが HTTP エンドポイントを公開する必要があります。コンテナの実行中、SageMaker AI は推論のために購入者入力をコンテナの HTTP エンドポイントに渡します。推論結果は HTTP レスポンスの本文で返されます。

次のチュートリアルでは、Linux Ubuntu ディストリビューションを使用する開発環境で Docker CLI を使用します。

- [ウェブサーバスクリプトを作成する](#)
- [コンテナ実行用のスクリプトを作成する](#)
- [Dockerfile の作成](#)
- [モデルアーティファクトをパッケージ化またはアップロードする](#)

ウェブサーバースクリプトを作成する

この例では [Flask](#) という Python サーバーを使用していますが、フレームワークに適した任意のウェブサーバーを使用できます。

Note

ここではわかりやすくするために [Flask](#) を使用しています。本番環境に対応するウェブサーバーとは見なされません。

SageMaker AI が使用する TCP ポート 8080 で 2 つの HTTP エンドポイントを提供する Flask ウェブサーバースクリプトを作成します。想定されるエンドポイントは次の 2 つです。

- /ping – SageMaker AI はこのエンドポイントに HTTP GET リクエストを行い、コンテナの準備が整っているかどうかを確認します。コンテナの準備が完了すると、コンテナはこのエンドポイントでの HTTP GET リクエストに HTTP 200 レスポンスコードで応答します。
- /invocations – SageMaker AI は、推論のためにこのエンドポイントに HTTP POST リクエストを行います。推論用の入力データはリクエストの本文で送信されます。ユーザー指定のコンテンツタイプは HTTP ヘッダーで渡されます。レスポンスの本文は推論出力です。タイムアウトの詳細については、「[機械学習製品を作成するための要件とベストプラクティス](#)」を参照してください。

./web_app_serve.py

```
# Import modules
import json
import re
from flask import Flask
from flask import request
app = Flask(__name__)

# Create a path for health checks
@app.route("/ping")
def endpoint_ping():
    return ""

# Create a path for inference
@app.route("/invocations", methods=["POST"])
def endpoint_invocations():
```

```
# Read the input
input_str = request.get_data().decode("utf8")

# Add your inference code between these comments.
#
#
#
#
# Add your inference code above this comment.

# Return a response with a prediction
response = {"prediction":"a","text":input_str}
return json.dumps(response)
```

前の例には、実際の推論ロジックはありません。実際の推論イメージについては、ウェブアプリに推論ロジックを追加し、入力进行处理して実際の予測を返します。

推論イメージには、インターネットアクセスも、への呼び出しもできないため、必要な依存関係がすべて含まれている必要があります AWS のサービス。

Note

これと同じコードがリアルタイム推論とバッチ推論の両方で呼び出されます。

コンテナ実行用のスクリプトを作成する

Docker コンテナイメージの実行時に SageMaker AI `serve` が実行する という名前のスクリプトを作成します。次のスクリプトは HTTP ウェブサーバーを起動します。

./serve

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker
flask run --host 0.0.0.0 --port 8080
```

Dockerfile の作成

ビルドコンテキストに Dockerfile を作成します。この例では Ubuntu 18.04 を使用していますが、フレームワークに適していれば、どのベースイメージからでも開始できます。

./Dockerfile

```
FROM ubuntu:18.04

# Specify encoding
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8

# Install python-pip
RUN apt-get update \
&& apt-get install -y python3.6 python3-pip \
&& ln -s /usr/bin/python3.6 /usr/bin/python \
&& ln -s /usr/bin/pip3 /usr/bin/pip;

# Install flask server
RUN pip install -U Flask;

# Add a web server script to the image
# Set an environment to tell flask the script to run
COPY /web_app_serve.py /web_app_serve.py
ENV FLASK_APP=/web_app_serve.py

# Add a script that Amazon SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

Dockerfile は、以前に作成した 2 つのスクリプトをイメージに追加します。serve スクリプトのディレクトリが PATH に追加されると、コンテナの実行時にそれを実行できるようになります。

モデルアーティファクトをパッケージ化またはアップロードする

モデルのトレーニングから推論イメージまで、モデルアーティファクトを提供する方法には、以下の 2 つがあります。

- 推論イメージと共に静的にパッケージ化します。

- ランタイムに動的にロードします。動的に読み込まれるため、同じイメージをさまざまな機械学習モデルのパッケージ化に使用できます。

モデルのアーティファクトを推論イメージと一緒にパッケージ化する場合は、アーティファクトを Dockerfile に含めてください。

モデルアーティファクトを動的にロードする場合は、それらのアーティファクトを Amazon S3 の圧縮ファイル (.tar.gz) に個別に保存します。モデルパッケージを作成するときは、圧縮ファイルの場所を指定します。SageMaker AI は、コンテナの実行 /opt/ml/model/ 時にコンテンツを抽出してコンテナディレクトリにコピーします。モデルパッケージを公開すると、それらのアーティファクトは、購入者が直接アクセスできない、AWS Marketplace 所有の Amazon S3 バケットに公開および保存されます。

ステップ 2: イメージをローカルでビルドしてテストする

ビルドコンテキストには、現在、以下のファイルが存在します。

- ./Dockerfile
- ./web_app_serve.py
- ./serve
- 推論ロジックと (オプションの) 依存関係

次に、コンテナイメージをビルド、実行、テストします。

イメージを構築する

ビルドコンテキストで Docker コマンドを実行し、イメージをビルドしてタグ付けします。この例ではタグ my-inference-image を使用します。

```
sudo docker build --tag my-inference-image ./
```

この Docker コマンドを実行してイメージをビルドすると、Dockerfile の各行に基づいて Docker がイメージをビルドするときの出力が表示されます。終了すると、次のようなものが表示されます。

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```


をローカルで実行する

ビルドが完了したら、イメージをローカルでテストできます。

```
sudo docker run \  
  --rm \  
  --publish 8080:8080/tcp \  
  --detach \  
  --name my-inference-container \  
  my-inference-image \  
  serve
```

コマンドの詳細は次のとおりです。

- `--rm` - コンテナが停止したら自動的に削除します。
- `--publish 8080:8080/tcp` - ポート 8080 を公開して、SageMaker AI が HTTP リクエストを送信するポートをシミュレートします。
- `--detach` - コンテナをバックグラウンドで実行します。
- `--name my-inference-container` - 実行中のこのコンテナに名前を付けます。
- `my-inference-image` - ビルドされたイメージを実行します。
- `serve` - コンテナの実行時に SageMaker AI が実行するのと同じスクリプトを実行します。

このコマンドを実行すると、Docker は、ビルドした推論イメージからコンテナを作成してバックグラウンドで実行します。コンテナは `serve` スクリプトを実行し、テスト目的でウェブサーバーを起動します。

HTTP エンドポイントへの ping をテストします。

SageMaker AI がコンテナを実行すると、定期的にエンドポイントに ping を送信します。エンドポイントがステータスコード 200 の HTTP レスポンスを返すと、コンテナが推論の準備ができていることを SageMaker AI に通知します。これをテストするには、次のコマンドを実行します。このコマンドはエンドポイントをテストしてレスポンスヘッダーを含めます。

```
curl --include http://127.0.0.1:8080/ping
```

出力例は次のとおりです。

```
HTTP/1.0 200 OK  
Content-Type: text/html; charset=utf-8
```

```
Content-Length: 0
Server: MyServer/0.16.0 Python/3.6.8
Date: Mon, 21 Oct 2019 06:58:54 GMT
```

推論 HTTP エンドポイントをテストします。

コンテナが ping に 200 のステータスコードを返すことで準備が整っていることを示すと、SageMaker AI は POST リクエストを介して推論データを HTTP /invocations エンドポイントに渡します。以下のコマンドを実行して、推論ポイントをテストします。

```
curl \
  --request POST \
  --data "hello world" \
  http://127.0.0.1:8080/invocations
```

出力例は次のとおりです。

```
{"prediction": "a", "text": "hello world"}
```

これら 2 つの HTTP エンドポイントが機能するので、推論イメージは SageMaker AI と互換性があります。

Note

モデルパッケージ製品のモデルは、リアルタイムとバッチの 2 つの方法でデプロイできます。どちらのデプロイでも、SageMaker AI は Docker コンテナの実行中に同じ HTTP エンドポイントを使用します。

コンテナを停止するには、次のコマンドを実行します。

```
sudo docker container stop my-inference-container
```

推論イメージの準備とテストが完了したら、次の「[Amazon Elastic Container Registry へのイメージのアップロード](#)」に進むことができます。

アルゴリズムイメージの作成

Amazon SageMaker AI アルゴリズムでは、予測を行う前に購入者が独自のデータを持ち込んでトレーニングする必要があります。AWS Marketplace 販売者は、SageMaker AI を使用して、購入者

がデプロイできる機械学習 (ML) アルゴリズムとモデルを作成できます AWS。以下のセクションでは、アルゴリズムイメージを作成する方法について説明します AWS Marketplace。これには、アルゴリズムをトレーニングするための Docker トレーニングイメージと、推論ロジックを含む推論イメージの作成が含まれます。アルゴリズム製品を公開するには、トレーニングイメージと推論イメージの両方が必要です。

トピック

- [概要](#)
- [アルゴリズム用のトレーニングイメージの作成](#)
- [アルゴリズムの推論イメージの作成](#)

概要

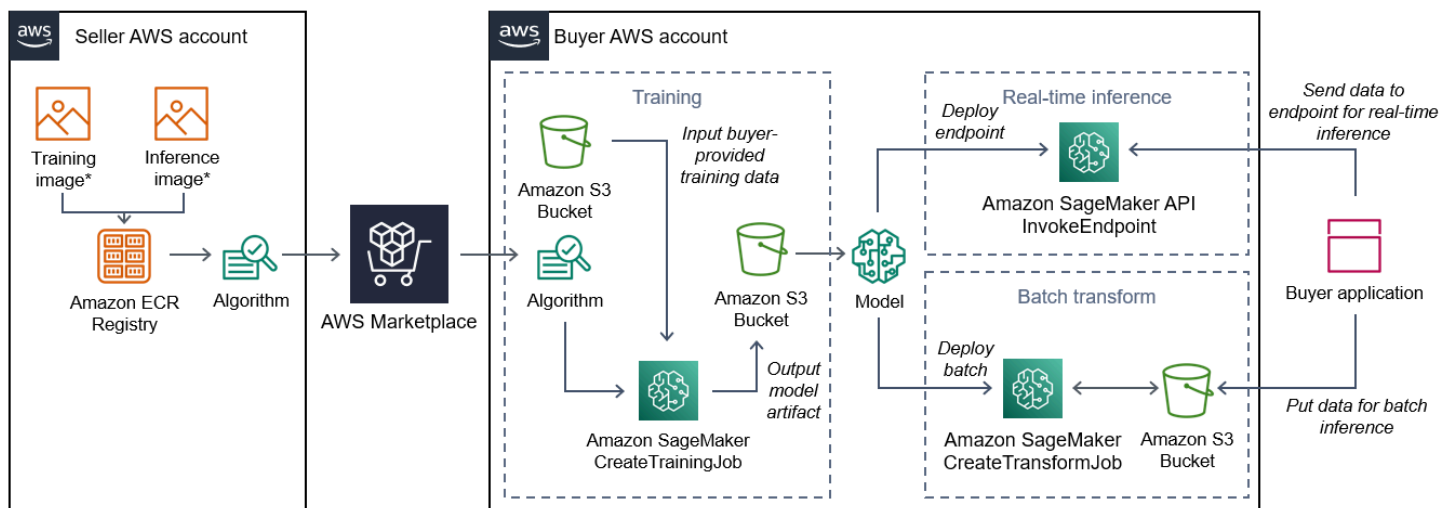
アルゴリズムには以下のコンポーネントが含まれています。

- [Amazon ECR](#) に保存されたトレーニングイメージ
- Amazon Elastic Container Registry (Amazon ECR) に保存された推論イメージ

Note

アルゴリズム製品では、トレーニングコンテナによってモデルアーティファクトが生成され、モデルのデプロイ時に推論コンテナに読み込まれます。

次の図は、アルゴリズム製品を公開して使用するワークフローを示しています。



の SageMaker AI アルゴリズムを作成するためのワークフロー AWS Marketplace には、次のステップが含まれます。

1. 販売者はトレーニングイメージと推論イメージ (デプロイ時のネットワークアクセス不可) を作成し、Amazon ECR レジストリにアップロードします。
2. 次に、販売者は Amazon SageMaker AI でアルゴリズムリソースを作成し、ML 製品を公開します AWS Marketplace。
3. 購入者は ML 製品をサブスクライブします。
4. 購入者は、互換性のあるデータセットと適切なハイパーパラメータ値を使用してトレーニングジョブを作成します。SageMaker AI はトレーニングイメージを実行し、トレーニングデータとハイパーパラメータをトレーニングコンテナにロードします。トレーニングジョブが完了すると、`/opt/ml/model/` にあるモデルアーティファクトが圧縮され、購入者の [Amazon S3](#) バケットにコピーされます。
5. 購入者は Amazon S3 に保存されているトレーニングのモデルアーティファクトを含むモデルパッケージを作成し、モデルをデプロイします。
6. SageMaker AI は推論イメージを実行し、圧縮されたモデルアーティファクトを抽出して、ファイルを推論コンテナディレクトリパスにロードします。`/opt/ml/model/` このパスでは、推論を提供するコードによってファイルが消費されます。
7. モデルがエンドポイントとしてデプロイされるかバッチ変換ジョブとしてデプロイされるかにかかわらず、SageMaker AI は購入者に代わってデータをコンテナの HTTP エンドポイント経由でコンテナに渡し、予測結果を返します。

 Note

詳細については、「[モデルのトレーニング](#)」を参照してください。

アルゴリズム用のトレーニングイメージの作成

このセクションでは、トレーニングコードをトレーニングイメージにパッケージ化する手順を説明します。アルゴリズム製品を作成するには、トレーニングイメージが必要です。

トレーニングイメージは、トレーニングアルゴリズムを含む Docker イメージです。コンテナは特定のファイル構造に準拠しているため、SageMaker AI はコンテナとの間でデータをコピーできます。

アルゴリズム製品を公開する際は、トレーニングイメージと推論イメージの両方が必要です。トレーニングイメージを作成したら、推論イメージを作成する必要があります。2 つのイメージは 1 つのイ

イメージに結合することも、個別のイメージのままにしておくこともできます。イメージを結合するか個別のイメージのままにするかは、ユーザーしだいです。通常、推論は学習よりも単純です。イメージを結合せずに個別のイメージのままにすると、推論のパフォーマンスが向上することがあります。

Note

以下は、トレーニングイメージのパッケージコードの一例です。詳細については、GitHub の「[で独自のアルゴリズムとモデル AWS Marketplace](#)を使用する」および[AWS Marketplace SageMaker AI の例](#)」を参照してください。

ステップ

- [ステップ 1: コンテナイメージを作成する](#)
- [ステップ 2: イメージをローカルでビルドしてテストする](#)

ステップ 1: コンテナイメージを作成する

トレーニングイメージを Amazon SageMaker AI と互換性を持たせるには、SageMaker AI がトレーニングデータと設定入力をコンテナ内の特定のパスにコピーできるように、特定のファイル構造に従う必要があります。トレーニングが完了すると、生成されたモデルアーティファクトは、SageMaker AI がコピーするコンテナ内の特定のディレクトリパスに保存されます。

以下では、Linux の Ubuntu ディストリビューションの開発環境にインストールされた Docker CLI が使用されています。

- [設定入力を読み取れるようプログラムを準備します。](#)
- [データ入力を読み取れるようプログラムを準備します。](#)
- [トレーニング出力を書き込めるようプログラムを準備します。](#)
- [コンテナ実行用のスクリプトを作成する](#)
- [Dockerfile の作成](#)

設定入力を読み取れるようプログラムを準備します。

トレーニングプログラムで購入者提供の設定入力が必要な場合は、実行時に以下がコンテナ内にコピーされます。必要に応じて、プログラムはこれらの特定のファイルパスから読み取りを行う必要があります。

- `/opt/ml/input/config` は、プログラムの実行方法を制御する情報が保存されているディレクトリです。
- `hyperparameters.json` は、ハイパーパラメータ名とハイパーパラメータ値の、JSON 形式の辞書です。値は文字列のため、変換が必要になる場合があります。
- `resourceConfig.json` は JSON 形式のファイルで、[分散型トレーニング](#)に使用されるネットワークレイアウトを記述しています。トレーニングイメージが分散型トレーニングをサポートしていない場合、このファイルは無視してかまいません。

Note

設定入力の詳細については、[Amazon SageMakerがトレーニング情報を提供する方法](#)を参照してください。

データ入力を読み取れるようプログラムを準備します。

トレーニングデータは、次の 2 つのモードのいずれかでコンテナに渡すことができます。コンテナ内で実行されるトレーニングプログラムは、これら 2 つのモードのいずれかでトレーニングデータのダイジェストを作成します。

ファイルモード

- `/opt/ml/input/data/<channel_name>/` にそのチャネルの入力データが含まれます。チャネルは `CreateTrainingJob` オペレーションの呼び出しに基づいて作成されますが、一般的にはアルゴリズムが期待するものとチャネルが一致することが重要です。各チャネルのファイルは [Amazon S3](#) からこのディレクトリにコピーされ、Amazon S3 のキー構造によって示されるツリー構造が保持されます。

パイプモード

- `/opt/ml/input/data/<channel_name>_<epoch_number>` は特定のエポックのパイプです。エポックは 0 から始まり、読み込まれるたびに 1 ずつ増えていきます。実行できるエポックの数に制限はありませんが、次のエポックを読み込む前に各パイプを閉じる必要があります。

トレーニング出力を書き込めるようプログラムを準備します。

トレーニングの出力は次のコンテナディレクトリに書き込まれます。

- `/opt/ml/model/` はトレーニングアルゴリズムが生成するモデルまたはモデルアーティファクトが書き込まれるディレクトリです。モデルはどのような形式でもかまいません。1つのファイルでもディレクトリツリー全体でもかまいません。SageMaker AI は、このディレクトリ内のすべてのファイルを圧縮ファイル (.tar.gz) にパッケージ化します。このファイルは、DescribeTrainingJob API オペレーションによって返される、Amazon S3 の場所にあります。
- `/opt/ml/output/` は、ジョブが失敗した理由が記述された failure ファイルをアルゴリズムが書き込むことができるディレクトリです。このファイルの内容は DescribeTrainingJob の結果の FailureReason フィールドに返されます。ジョブが成功してもこのファイルは無視されるため、このファイルを書き込む理由はありません。

コンテナ実行用のスクリプトを作成する

Docker コンテナイメージの実行時に SageMaker AI が実行する train シェルスクリプトを作成します。トレーニングが完了し、モデルアーティファクトがそれぞれのディレクトリに書き込まれたら、スクリプトを終了します。

`./train`

```
#!/bin/bash

# Run your training program here
#
#
#
#
```

Dockerfile の作成

ビルドコンテキストに Dockerfile を作成します。この例ではベースイメージとして Ubuntu 18.04 を使用していますが、フレームワークに適していれば、どのベースイメージからでも開始できます。

`./Dockerfile`

```
FROM ubuntu:18.04

# Add training dependencies and programs
#
#
#
```

```
#  
#  
# Add a script that SageMaker AI will run  
# Set run permissions  
# Prepend program directory to $PATH  
COPY /train /opt/program/train  
RUN chmod 755 /opt/program/train  
ENV PATH=/opt/program:${PATH}
```

Dockerfile は以前に作成した train スクリプトをイメージに追加します。スクリプトのディレクトリが PATH に追加され、コンテナの実行時にスクリプトを実行できるようになります。

前の例には、実際のトレーニングロジックはありません。実際のトレーニングイメージでは、トレーニングの依存関係を Dockerfile に追加し、トレーニング入力を読み取るロジックを追加してトレーニングを行い、モデルアーティファクトを生成します。

トレーニングイメージはインターネットにアクセスできないため、必要な依存関係がすべて含まれている必要があります。

詳細については、GitHub の [「で独自のアルゴリズムとモデル AWS Marketplace」](#) および [AWS Marketplace SageMaker AI の例](#)」を参照してください。

ステップ 2: イメージをローカルでビルドしてテストする

ビルドコンテキストには、現在、以下のファイルが存在します。

- ./Dockerfile
- ./train
- トレーニングの依存関係とロジック

次に、このコンテナイメージをビルド、実行、テストできます。

イメージを構築する

ビルドコンテキストで Docker コマンドを実行し、イメージをビルドしてタグ付けします。この例ではタグ my-training-image を使用します。

```
sudo docker build --tag my-training-image ./
```

この Docker コマンドを実行してイメージをビルドすると、Dockerfile の各行に基づいて Docker がイメージをビルドするときの出力が表示されます。終了すると、次のようなものが表示されます。


```
Successfully built abcdef123456  
Successfully tagged my-training-image:latest
```

をローカルで実行する

完了したら、次の例に示すようにイメージをローカルでテストします。

```
sudo docker run \  
  --rm \  
  --volume '<path_to_input>:/opt/ml/input:ro' \  
  --volume '<path_to_model>:/opt/ml/model' \  
  --volume '<path_to_output>:/opt/ml/output' \  
  --name my-training-container \  
  my-training-image \  
  train
```

コマンドの詳細は次のとおりです。

- `--rm` - コンテナが停止したら自動的に削除します。
- `--volume '<path_to_input>:/opt/ml/input:ro'` - テスト入力ディレクトリをコンテナが読み取り専用でできるようにします。
- `--volume '<path_to_model>:/opt/ml/model'` - トレーニングテストが完了したら、モデルアーティファクトが保存されているパスをホストマシンでバインドマウントします。
- `--volume '<path_to_output>:/opt/ml/output'` - 障害理由が書き込まれる failure ファイル内のパスをホストマシンでバインドマウントします。
- `--name my-training-container` - 実行中のこのコンテナに名前を付けます。
- `my-training-image` - ビルドされたイメージを実行します。
- `train` - コンテナの実行時に SageMaker AI が実行するのと同じスクリプトを実行します。

このコマンドを実行すると、Docker は、ビルドされたトレーニングイメージからコンテナを作成して実行します。コンテナは `train` スクリプトを実行します。これで、トレーニングプログラムが起動します。

トレーニングプログラムが終了し、コンテナが終了したら、出力モデルのアーティファクトが正しいことを確認します。さらに、ログ出力をチェックして、トレーニングジョブに関する十分な情報が提供されていることを確認し、併せて、不要なログが生成されていないことを確認します。

これで、アルゴリズム製品用のトレーニングコードのパッケージ化が完了しました。アルゴリズム製品には推論イメージも含まれるため、次のセクション、「[アルゴリズムの推論イメージの作成](#)」に進んでください。

アルゴリズムの推論イメージの作成

このセクションでは、推論コードをアルゴリズム製品の推論イメージにパッケージ化する手順を説明します。

推論イメージは、推論ロジックを含む Docker イメージです。実行時にコンテナは HTTP エンドポイントを公開し、SageMaker AI がコンテナとの間でデータをやり取りできるようにします。

アルゴリズム製品を公開する際は、トレーニングイメージと推論イメージの両方が必要です。これをまだ確認していない場合は、「[アルゴリズム用のトレーニングイメージの作成](#)」に関する前のセクションを参照してください。2つのイメージは1つのイメージに結合することも、個別のイメージのままにしておくこともできます。イメージを結合するか個別のイメージのままにするかは、ユーザー次第です。通常、推論は学習よりも単純です。イメージを結合せずに個別のイメージのままにすると、推論のパフォーマンスが向上することがあります。

Note

以下は、推論イメージのパッケージコードの一例です。詳細については、GitHub の「[で独自のアルゴリズムとモデル AWS Marketplace を使用する](#)」および [AWS Marketplace SageMaker AI の例](#)」を参照してください。

以下の例ではわかりやすくするために [Flask](#) というウェブサービスを使用していますが、本番環境に対応しているとは見なされていません。

ステップ

- [ステップ 1: 推論イメージを作成する](#)
- [ステップ 2: イメージをローカルでビルドしてテストする](#)

ステップ 1: 推論イメージを作成する

推論イメージを SageMaker AI と互換性を持たせるには、Docker イメージが HTTP エンドポイントを公開する必要があります。コンテナの実行中、SageMaker AI は購入者から提供された推論の入力をコンテナの HTTP エンドポイントに渡します。推論の結果は HTTP レスポンスの本文で返されます。

以下では、Linux の Ubuntu ディストリビューションの開発環境にインストールされた Docker CLI が使用されています。

- [ウェブサーバースクリプトを作成する](#)
- [コンテナ実行用のスクリプトを作成する](#)
- [Dockerfile の作成](#)
- [モデルアーティファクトを動的に読み込むためのプログラムを準備する](#)

ウェブサーバースクリプトを作成する

この例では [Flask](#) という Python サーバーを使用していますが、フレームワークに適した任意のウェブサーバーを使用できます。

Note

ここではわかりやすくするために [Flask](#) を使用しています。本番環境に対応するウェブサーバーとは見なされません。

SageMaker AI が使用する TCP ポート 8080 で 2 つの HTTP エンドポイントを提供する Flask ウェブサーバースクリプトを作成します。想定されるエンドポイントは次の 2 つです。

- `/ping` – SageMaker AI はこのエンドポイントに HTTP GET リクエストを行い、コンテナの準備が整っているかどうかを確認します。コンテナの準備が完了すると、コンテナはこのエンドポイントでの HTTP GET リクエストに HTTP 200 レスポンスコードで応答します。
- `/invocations` – SageMaker AI は、推論のためにこのエンドポイントに HTTP POST リクエストを行います。推論用の入力データはリクエストの本文で送信されます。ユーザー指定のコンテンツタイプは HTTP ヘッダーで渡されます。レスポンスの本文は推論出力です。

`./web_app_serve.py`

```
# Import modules
import json
import re
from flask import Flask
from flask import request
app = Flask(__name__)
```

```
# Create a path for health checks
@app.route("/ping")
def endpoint_ping():
    return ""

# Create a path for inference
@app.route("/invocations", methods=["POST"])
def endpoint_invocations():

    # Read the input
    input_str = request.get_data().decode("utf8")

    # Add your inference code here.
    #
    #
    #
    #
    # Add your inference code here.

    # Return a response with a prediction
    response = {"prediction":"a","text":input_str}
    return json.dumps(response)
```

前の例には、実際の推論ロジックはありません。実際の推論イメージについては、ウェブアプリに推論ロジックを追加し、入力进行处理して予測を返します。

推論イメージはインターネットにアクセスできないため、必要な依存関係がすべて含まれている必要があります。

コンテナ実行用のスクリプトを作成する

Docker コンテナイメージの実行時に SageMaker AI `serve`が実行する という名前のスクリプトを作成します。このスクリプトでは、HTTP ウェブサーバーを起動します。

./serve

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker AI
flask run --host 0.0.0.0 --port 8080
```

Dockerfile の作成

ビルドコンテキストに Dockerfile を作成します。この例では Ubuntu 18.04 を使用していますが、フレームワークに適していれば、どのベースイメージからでも開始できます。

./Dockerfile

```
FROM ubuntu:18.04

# Specify encoding
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8

# Install python-pip
RUN apt-get update \
&& apt-get install -y python3.6 python3-pip \
&& ln -s /usr/bin/python3.6 /usr/bin/python \
&& ln -s /usr/bin/pip3 /usr/bin/pip;

# Install flask server
RUN pip install -U Flask;

# Add a web server script to the image
# Set an environment to tell flask the script to run
COPY /web_app_serve.py /web_app_serve.py
ENV FLASK_APP=/web_app_serve.py

# Add a script that Amazon SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

Dockerfile は以前に作成した 2 つのスクリプトをイメージに追加します。serve スクリプトのディレクトリが PATH に追加されると、コンテナの実行時にそれを実行できるようになります。

モデルアーティファクトを動的に読み込むためのプログラムを準備する

アルゴリズム製品の場合、購入者は独自のデータセットとトレーニングイメージを使用して、独自のモデルアーティファクトを生成します。トレーニングプロセスが完了すると、トレーニングコンテナはモデルアーティファクトをコンテナディレクトリ /opt/ml/model/ に出力します。SageMaker

AI は、そのディレクトリ内のコンテンツを .tar.gz ファイルに圧縮し、Amazon S3 の購入者の AWS アカウント に保存します。

モデルがデプロイされると、SageMaker AI は推論イメージを実行し、Amazon S3 の購入者のアカウントに保存されている .tar.gz ファイルからモデルアーティファクトを抽出し、 /opt/ml/model/ ディレクトリの推論コンテナにロードします。実行時、推論コンテナコードはモデルデータを使用します。

Note

モデルアーティファクトファイルに含まれる可能性のある知的財産を保護するために、ファイル出力前の暗号化を選択できます。詳細については、「[Amazon SageMaker AI のセキュリティと知的財産](#)」を参照してください。

ステップ 2: イメージをローカルでビルドしてテストする

ビルドコンテキストには、現在、以下のファイルが存在します。

- ./Dockerfile
- ./web_app_serve.py
- ./serve

次に、このコンテナイメージをビルド、実行、テストできます。

イメージを構築する

Docker コマンドを実行し、イメージをビルドしてタグ付けします。この例ではタグ my-inference-image を使用します。

```
sudo docker build --tag my-inference-image ./
```

この Docker コマンドを実行してイメージをビルドすると、Dockerfile の各行に基づいて Docker がイメージをビルドするときの出力が表示されます。終了すると、次のようなものが表示されます。

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```

をローカルで実行する

ビルドが完了したら、イメージをローカルでテストできます。

```
sudo docker run \  
  --rm \  
  --publish 8080:8080/tcp \  
  --volume '<path_to_model>:/opt/ml/model:ro' \  
  --detach \  
  --name my-inference-container \  
  my-inference-image \  
  serve
```

コマンドの詳細は次のとおりです。

- `--rm` - コンテナが停止したら自動的に削除します。
- `--publish 8080:8080/tcp` - ポート 8080 を公開して、SageMaker AI が HTTP リクエストを送信するポートをシミュレートします。
- `--volume '<path_to_model>:/opt/ml/model:ro'` - テストモデルアーティファクトが保存されているホストマシン上のパスを、コンテナ内の推論コードで使えるよう読み取り専用としてバインドマウントします。
- `--detach` - コンテナをバックグラウンドで実行します。
- `--name my-inference-container` - 実行中のこのコンテナに名前を付けます。
- `my-inference-image` - ビルドされたイメージを実行します。
- `serve` - コンテナの実行時に SageMaker AI が実行するのと同じスクリプトを実行します。

このコマンドを実行すると、Docker は、推論イメージからコンテナを作成してバックグラウンドで実行します。コンテナは `serve` スクリプトを実行し、テスト目的でウェブサーバーを起動します。

HTTP エンドポイントへの ping をテストします。

SageMaker AI がコンテナを実行すると、定期的にエンドポイントに ping を送信します。エンドポイントがステータスコード 200 の HTTP レスポンスを返すと、コンテナが推論の準備ができていることを SageMaker AI に通知します。

次のコマンドを実行してエンドポイントをテストし、レスポンスヘッダーを含めます。

```
curl --include http://127.0.0.1:8080/ping
```

以下の例に、出力例を示します。

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 0
Server: MyServer/0.16.0 Python/3.6.8
Date: Mon, 21 Oct 2019 06:58:54 GMT
```

推論 HTTP エンドポイントをテストします。

コンテナが 200 ステータスコードを返すことで準備が整っていることを示すと、SageMaker AI は POST リクエストを介して推論データを /invocations HTTP エンドポイントに渡します。

以下のコマンドを実行して、推論エンドポイントをテストします。

```
curl \
  --request POST \
  --data "hello world" \
  http://127.0.0.1:8080/invocations
```

以下の例に、出力例を示します。

```
{"prediction": "a", "text": "hello world"}
```

これら 2 つの HTTP エンドポイントが機能するので、推論イメージは SageMaker AI と互換性があります。

Note

アルゴリズム製品のモデルは、リアルタイムとバッチの 2 つの方法でデプロイできます。どちらのデプロイでも、SageMaker AI は Docker コンテナの実行中に同じ HTTP エンドポイントを使用します。

コンテナを停止するには、次のコマンドを実行します。

```
sudo docker container stop my-inference-container
```

アルゴリズム製品のトレーニングイメージと推論イメージの両方の準備とテストが完了したら、「[Amazon Elastic Container Registry へのイメージのアップロード](#)」に進みます。

Amazon Elastic Container Registry へのイメージのアップロード

推論イメージとトレーニングイメージを作成したら、Amazon Elastic Container Registry にアップロードできます。[Amazon ECR](#) はフルマネージド型の Docker レジストリです。Amazon SageMaker AI は、Amazon ECR からイメージをプルして推論用のモデルパッケージまたはトレーニングジョブ用のアルゴリズムを作成します。AWS Marketplace または Amazon ECR からこれらのイメージを取得して、モデルパッケージとアルゴリズム製品を公開します。このトピックでは、推論イメージとトレーニングイメージを Amazon ECR にアップロードする手順を説明します。

トピック

- [アップロードする必要があるイメージ](#)
- [必要となる IAM アクセス許可](#)
- [Docker クライアントを にログインする AWS](#)
- [リポジトリを作成してイメージをアップロードします。](#)
- [アップロードしたイメージをスキャンします。](#)

アップロードする必要があるイメージ

モデルパッケージを公開する場合は、推論イメージのみをアップロードします。アルゴリズムを公開する場合は、推論イメージとトレーニングイメージの両方をアップロードします。推論イメージとトレーニングイメージを結合する場合は、結合されたイメージを 1 回だけアップロードしてください。

必要となる IAM アクセス許可

次の手順では、ローカルマシンに販売者の AWS Identity and Access Management (IAM) ロールまたはユーザー用の正しい AWS 認証情報があることを前提としています AWS アカウント。ロールまたはユーザーは、AWS Marketplace と Amazon ECR の両方に正しいポリシーを設定する必要があります。たとえば、次の AWS 管理ポリシーを使用できます。

- [AWSMarketplaceSellerProductsFullAccess](#) – へのアクセス用 AWS Marketplace
- [AmazonEC2ContainerRegistryFullAccess](#) - Amazon ECR へのアクセス用

Note

リンクをクリックすると、AWS マネージドポリシーリファレンスに移動します。

Docker クライアントを にログインする AWS

発行元の の変数を設定します (AWS リージョン「」を参照[公開 AWS リージョン でサポート](#))。この例では、米国東部 (オハイオ) リージョンを使用します。

```
region=us-east-2
```

次のコマンドを実行して、AWS アカウント ID で変数を設定します。この例では、current AWS Command Line Interface (AWS CLI) 認証情報が販売者の に属していることを前提としています AWS アカウント。

```
account=$(aws sts get-caller-identity --query Account --output text)
```

リージョンの AWS アカウント Amazon ECR Docker レジストリを使用して Docker CLI クライアントを認証するには、次のコマンドを実行します。

```
aws ecr get-login-password \  
--region ${region} \  
| sudo docker login \  
--username AWS \  
--password-stdin \  
${account}.dkr.ecr.${region}.amazonaws.com
```

リポジトリを作成してイメージをアップロードします。

アップロードしたイメージのタグに変数を設定し、アップロードしたイメージリポジトリの名前に別の変数を設定します。

```
image=my-inference-image  
repo=my-inference-image
```

Note

推論イメージとトレーニングイメージを作成した、このガイドの前のセクションで、これらのイメージは my-inference-image および my-training-image としてそれぞれタグ付けされていました。この例では、推論イメージを作成して同じ名前のリポジトリにアップロードします。

次のコマンドを実行して、Amazon ECR にイメージリポジトリを作成します。

```
aws ecr --region ${region} create-repository --repository-name "${repo}"
```

Amazon ECR リポジトリの場所のフルネームは、次のパーツで構成されています。 <account-id>.dkr.ecr.<region>.amazonaws.com/<image-repository-name>

イメージをリポジトリにプッシュするには、リポジトリの場所のフルネームをイメージにタグ付けする必要があります。

イメージリポジトリの場所のフルネームとして変数を latest タグと共に設定します。

```
fullname="${account}.dkr.ecr.${region}.amazonaws.com/${repo}:latest"
```

イメージにフルネームをタグ付けします。

```
sudo docker tag ${image} ${fullname}
```

最後に、Amazon ECR のリポジトリに推論イメージをプッシュします。

```
sudo docker push ${fullname}
```

アップロードが完了すると、公開元のリージョンの [Amazon ECR コンソールのリポジトリリスト](#) にイメージが表示されます。前の例では、イメージは米国東部 (オハイオ) リージョンにプッシュされました。

アップロードしたイメージをスキャンします。

[Amazon ECR コンソール](#)で、公開元の AWS リージョン を選択し、イメージがアップロードされたリポジトリを開きます。アップロードしたイメージを選択し、スキャンを開始して既知の脆弱性を確認します。は、Amazon SageMaker AI リソースで使用するコンテナイメージの Amazon ECR スキャン結果を公開前に AWS Marketplace チェックします。製品を作成するには、重要度が「重大」または「高」の脆弱性があるコンテナイメージを修正する必要があります。

イメージが正常にスキャンされたら、そのイメージを使用してモデルパッケージまたはアルゴリズムリソースを作成できます。

商品のスキャン中に誤検出のエラーがあったと思われる場合は、[AWS Marketplace Seller Operations](#) チームに連絡し、エラーのに関する情報を伝えてください。

次のステップ

- [機械学習製品を作成するための要件とベストプラクティス](#) におけるファイルサイズ制限を確認する
- 「[Amazon SageMaker AI リソースの作成](#)」に進む

Amazon SageMaker AI リソースの作成

モデルパッケージまたはアルゴリズム製品を公開するには、Amazon SageMaker AI でそれぞれの[モデルパッケージリソース](#)または[アルゴリズムリソース](#)を作成する必要があります。AWS Marketplace 製品用のリソースを作成するときは、検証ステップを通じて認定を受ける必要があります。検証ステップでは、公開前にモデルパッケージまたはアルゴリズムリソースをテストするためのデータを提供する必要があります。以下のセクションでは、モデルパッケージリソースまたはアルゴリズムリソースのいずれかで SageMaker AI リソースを作成する方法を示します。これには、SageMaker AI に検証の実行方法を指示する検証仕様の設定が含まれます。

Note

製品のイメージをまだ作成しておらず、Amazon Elastic Container Registry (Amazon ECR) にアップロードしていない場合は、その方法の情報について、「[での機械学習製品のイメージへのコードのパッケージ化 AWS Marketplace](#)」および「[Amazon Elastic Container Registry へのイメージのアップロード](#)」を参照してください。

トピック

- [モデルパッケージの作成](#)
- [アルゴリズムの作成](#)

モデルパッケージの作成

AWS Marketplaceのモデルパッケージを作成するための要件は次のとおりです。

- [Amazon ECR](#) に保存されている推論イメージ
- (オプション) [Amazon S3](#) に個別に保存されているモデルアーティファクト
- Amazon Simple Storage Service に保存されている推論に使用されるテストデータ

Note

以下は、モデルパッケージ製品の作成に関するものです。SageMaker AI のモデルパッケージの詳細については、[「モデルパッケージリソースの作成」](#)を参照してください。

モデルパッケージリソースの作成

以下の手順では、モデルパッケージリソースを作成する方法を順を追って説明します。

ステップ 1: モデルパッケージリソースを作成する

1. [Amazon SageMaker AI コンソール](#)を開きます。
2. ページの右上を見て、発行元の AWS リージョンにいることを確認します。公開については、[「公開 AWS リージョン でサポート」](#)セクションを参照してください。前のステップで Amazon ECR にアップロードした推論イメージが同じリージョンに存在する必要があります。
3. 左側のナビゲーションメニューから [モデルパッケージ] を選択します。
4. [Create model package (モデルパッケージの作成)] を選択します。

パッケージを作成したら、推論パッケージの仕様を設定する必要があります。

ステップ 2: 推論の仕様を設定する

1. モデルパッケージの [名前] (例えば、*my-model-package*) を指定します。
2. [推論イメージの場所] には、Amazon ECR にアップロードされた推論イメージの URI を入力します。URI は [Amazon ECR コンソール](#)でイメージを検索すると取得できます。
3. トレーニングで得たモデルアーティファクトが推論イメージのロジックにバンドルされている場合、モデルデータアーティファクトの場所は空欄のままにしておきます。それ以外の場合は、モデルアーティファクトの圧縮ファイル (.tar.gz) の、Amazon S3 の完全な場所を指定します。
4. ドロップダウンボックスを使用して、リアルタイム推論 (エンドポイントとも呼ばれる) ジョブとバッチ変換ジョブの両方でサポートされる推論イメージのインスタンスタイプを選択します。
5. [次へ] を選択します。

モデルパッケージを作成して公開するには、期待どおりに機能するかを確認するための検証が必要です。そのためには、提供した推論用テストデータを使用してバッチ変換ジョブを実行する必要があります。検証仕様は、検証の実行方法を SageMaker AI に指示します。

ステップ 3: 検証仕様を設定する

1. [AWS Marketplaceでこのモデルパッケージを公開する] を [はい] に設定します。これを [いいえ] に設定すると、このモデルパッケージを後で公開することはできません。Yes を選択すると、[のモデルパッケージが認証](#) AWS Marketplace され、検証ステップが必要です。
2. このプロセスを初めて完了する場合は、IAM ロールの [新規ロールの作成] を選択します。Amazon SageMaker AI は、モデルパッケージをデプロイするときにこのロールを使用します。これには、Amazon ECR からのイメージの取得や Amazon S3 からのアーティファクトの取得などのアクションが含まれます。設定を確認し、[ロールを作成] を選択します。ここでロールを作成すると、作成したロールに、[AmazonSageMakerFullAccess](#) IAM ポリシーに記述されているアクセス許可が付与されます。
3. 検証プロファイルの JSON を編集します。許可される値の詳細については、「[TransformJobDefinition](#)」を参照してください。
 1. TransformInput.DataSource.S3Uri: 推論用のテストデータを保存する場所を設定します。
 2. TransformInput.ContentType: テストデータのコンテンツタイプ (application/json、text/plain、image/png 、またはその他の値など) を指定します。SageMaker AI は実際の入力データを検証しません。この値は、Content-type ヘッダー値としてコンテナの HTTP エンドポイントに渡されます。
 3. TransformInput.CompressionType: Amazon S3 の推論用テストデータが圧縮されていない場合は None に設定します。
 4. TransformInput.SplitType: None に設定して、Amazon S3 内の各オブジェクトを推論用にまとめて渡します。
 5. TransformOutput.S3OutputPath: 推論出力が保存される場所に設定します。
 6. TransformOutput.AssembleWith: None に設定して、各推論を Amazon S3 の個別のオブジェクトとして出力します。
4. [Create model package (モデルパッケージの作成)] を選択します。

SageMaker AI は Amazon ECR から推論イメージをプルし、アーティファクトを推論コンテナにコピーし、推論用のテストデータを使用してバッチ変換ジョブを実行します。検証が成功すると、ステータスは [完了済み] に変わります。

Note

検証ステップでは、テストデータによるモデルの精度は評価されません。検証ステップでは、コンテナが実行されて期待どおりに応答するかどうかを確認されます。

モデル製品リソースの作成が完了しました。「[での製品の一覧表示 AWS Marketplace](#)」に進みます。

アルゴリズムの作成

AWS Marketplaceのアルゴリズムを作成するための要件は次のとおりです。

- Amazon ECR に保存された推論イメージ
- Amazon ECR に保存されたトレーニングイメージ
- Amazon S3 に保存されたトレーニング用テストデータ
- Amazon S3 に保存された推論用テストデータ

Note

次のウォークスルーでは、アルゴリズム製品を作成します。詳細については、「[アルゴリズムリソースを作成する](#)」を参照してください。

アルゴリズムリソースの作成

以下の手順では、アルゴリズムパッケージのリソースを作成する方法を順を追って説明します。

ステップ 1: アルゴリズムリソースを作成する

1. [Amazon SageMaker AI コンソール](#)を開きます。
2. ページの右上を見て、発行元の AWS リージョンにいることを確認します (「」を参照[公開 AWS リージョン でサポート](#))。前のステップで Amazon ECR にアップロードしたトレーニングイメージと推論イメージが同じリージョンに存在する必要があります。
3. 左のナビゲーションペインで [アルゴリズム] をクリックします。
4. [Create algorithm (アルゴリズムの作成)] を選択します。

アルゴリズムパッケージを作成したら、モデルのトレーニングとチューニングの仕様を設定する必要があります。

ステップ 2: トレーニングとチューニングの仕様を設定する

1. アルゴリズムの[名前] (例えば *my-algorithm*) を入力します。
2. [トレーニングイメージ] には、Amazon ECR にアップロードされたトレーニングイメージの、完全な URI の場所を貼り付けます。URI は [Amazon ECR コンソール](#) でイメージを検索すると取得できます。
3. ドロップダウンボックスを使用して、トレーニングイメージがサポートするトレーニング用のインスタンスタイプを選択します。
4. [チャネルの仕様] セクションで、アルゴリズムがサポートする各入力データセットに 1 つのチャネルを追加します (入力ソースの最大数は 20 チャネル)。詳細については、「[入力データ設定](#)」を参照してください。
5. [次へ] を選択します。
6. アルゴリズムがハイパーパラメータとハイパーパラメータ調整をサポートしている場合は、調整パラメータを指定する必要があります。
7. [次へ] を選択します。

Note

アルゴリズムがハイパーパラメータ調整をサポートして適切なパラメータを調整できるようになっていることが、強く推奨されます。これにより、データサイエンティストはモデルを調整して最良の結果を得ることができます。

調整パラメータがある場合、そのパラメータを設定したら、推論イメージの仕様を設定する必要があります。

ステップ 3: 推論イメージ仕様を設定する

1. [推論イメージの場所] には、Amazon ECR にアップロードされた推論イメージの URI を貼り付けます。URI は [Amazon ECR コンソール](#) でイメージを検索すると取得できます。
2. ドロップダウンボックスを使用して、リアルタイム推論 (エンドポイントとも呼ばれる) ジョブとバッチ変換ジョブの両方でサポートされる推論イメージのインスタンスタイプを選択します。
3. [次へ] を選択します。

アルゴリズムを作成して公開するには、期待どおりに機能するかを確認するための検証が必要です。そのためには、トレーニング用のテストデータを使用するトレーニングジョブと、提供した推論用テストデータを使用してバッチ変換ジョブの両方を実行する必要があります。検証仕様は、検証の実行方法を SageMaker AI に指示します。

ステップ 4: 検証仕様を設定する

1. [AWS Marketplaceでこのアルゴリズムを公開する]を [はい] に設定します。これを [いいえ] に設定すると、このアルゴリズムを後で公開することができません。Yes を選択すると、[のアルゴリズムが認定](#) AWS Marketplace され、検証仕様が必要です。
2. の機械学習パッケージを初めて作成する場合は AWS Marketplace、IAM ロールの新しいロールを作成するを選択します。Amazon SageMaker AI は、アルゴリズムをトレーニングし、後続のモデルパッケージをデプロイするときにこのロールを使用します。これには、Amazon ECR からのイメージの取得、Amazon S3 へのアーティファクトの保存、Amazon S3 からのトレーニングデータのコピーなどのアクションが含まれます。設定を確認し、[ロールを作成] を選択します。ここでロールを作成すると、作成したロールに、[AmazonSageMakerFullAccess](#) IAM ポリシーに記述されているアクセス許可が付与されます。
3. [トレーニングジョブ定義] の検証プロファイルの [JSON] ファイルを編集します。許可される値の詳細については、「[TrainingJobDefinition](#)」を参照してください。
 1. InputDataConfig: この JSON 配列に、トレーニング仕様のステップで指定した各チャンネルの [チャンネルオブジェクト](#) を追加します。チャンネルごとに、トレーニング用のテストデータを保存する場所を指定します。
 2. OutputDataConfig: トレーニングが完了すると、トレーニングコンテナディレクトリパス /opt/ml/model/ 内のモデルアーティファクトが圧縮され、Amazon S3 にコピーされます。圧縮ファイル (.tar.gz) が保存されている Amazon S3 の場所を指定します。
4. [変換ジョブの定義] の検証プロファイルの JSON ファイルを編集します。許可される値の詳細については、「[TransformJobDefinition](#)」を参照してください。
 1. TransformInput.DataSource.S3Uri: 推論用のテストデータを保存する場所を設定します。
 2. TransformInput.ContentType: テストデータのコンテンツタイプを指定します。例えば、application/json、text/plain、image/png、またはその他の値です。Amazon SageMaker AI は実際の入力データを検証しません。この値は、Content-type ヘッダー値としてコンテナの HTTP エンドポイントに渡されます。
 3. TransformInput.CompressionType: Amazon S3 の推論用テストデータが圧縮されていない場合は None に設定します。

4. `TransformInput.SplitType`: S3 内のオブジェクトをどのように分割するかを選択します。例えば、`None` では、Amazon S3 内の各オブジェクトが推論用にまとめて渡されます。詳細については、Amazon SageMaker AI API リファレンスの[SplitType](#)」を参照してください。
 5. `TransformOutput.S3OutputPath`: 推論出力が保存される場所に設定します。
 6. `TransformOutput.AssembleWith`: `None` に設定して、各推論を Amazon S3 の個別のオブジェクトとして出力します。
5. [アルゴリズムパッケージの作成] を選択します。

SageMaker AI は Amazon ECR からトレーニングイメージをプルし、データを使用してテストトレーニングジョブを実行し、モデルアーティファクトを Amazon S3 に保存します。次に、Amazon ECR から推論イメージを取得して、アーティファクトを Amazon S3 から推論コンテナにコピーし、推論用のテストデータを使用してバッチ変換ジョブを実行します。検証が成功すると、ステータスは [完了済み] に変わります。

Note

検証ステップでは、テストデータによるトレーニングまたはモデルの精度は評価されません。検証ステップでは、コンテナが実行されて期待どおりに応答するかどうかを確認されます。

検証ステップではバッチ処理のみが検証されます。リアルタイム処理が製品で機能することを検証するのはユーザーしだいです。

アルゴリズム製品リソースの作成が完了しました。「[での製品の一覧表示 AWS Marketplace](#)」に進みます。

での製品の一覧表示 AWS Marketplace

コードをモデルパッケージイメージまたはアルゴリズムイメージにパッケージ化し、イメージをアップロードして Amazon SageMaker AI リソースを作成したら、で機械学習製品を公開できます AWS Marketplace。以下のセクションでは、製品リストの作成、製品のテスト、公開へのサインオフなど、公開プロセスについて説明します。製品が公開されたら、リストを更新するための変更をリクエストします。詳細については、「[the section called “製品の管理”](#)」を参照してください。

トピック

- [前提条件](#)
- [ステップ 1: 新しいリストを作成する](#)
- [ステップ 2: 製品情報を提供する](#)
- [ステップ 3: 初期製品バージョンを追加する](#)
- [ステップ 4: 料金モデルを設定する](#)
- [ステップ 5: 返金ポリシーを設定する](#)
- [ステップ 6: EULA を設定する](#)
- [ステップ 7: 許可リストを設定する](#)

前提条件

モデルパッケージまたはアルゴリズムを で公開する前に AWS Marketplace、次のものがが必要です。

- AWS Marketplace 販売者として登録 AWS アカウント されている。これは、[AWS Marketplace 管理ポータル](#)で行えます。
- AWS Marketplace 管理ポータルの [\[設定\]](#) ページの完了済みセラープロフィール。
- 有料製品を公開するには、Tax Interview と銀行フォームに必要事項を記入する必要があります。これは無料製品の公開には必要ありません。詳細については、「[Seller registration process](#)」を参照してください。
- AWS Marketplace 管理ポータル および Amazon SageMaker AI にアクセスするには、アクセス許可が必要です。詳細については、「[必要な アクセス許可](#)」を参照してください。

必要な アクセス許可

Amazon SageMaker AI 製品を公開するには、AWS Marketplace サービスプリンシパルとの信頼関係を持つ有効な IAM ロール ARN を指定する必要があります。さらに、としてサインインしている IAM ユーザーまたはロールには、必要なアクセス許可が必要です。

サインイン許可の設定

- IAM ロールに次のアクセス許可を追加します。
 1. sagemaker:DescribeModelPackage — モデルパッケージを一覧表示する場合
 2. sagemaker:DescribeAlgorithm — アルゴリズムの一覧表示用

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:DescribeModelPackage",
        "sagemaker:DescribeAlgorithm"
      ],
      "Resource": "*"
    }
  ]
}
```

IAM ロール AddVersion/Create 製品の設定

1. カスタム信頼ポリシーを使用してロールを作成するには、次の手順に従います。詳細については、[「カスタム信頼ポリシーを使用した IAM ロールの作成 \(コンソール\)」](#)を参照してください。
2. カスタム信頼ポリシーステートメントに以下を入力します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "Service": "assets.marketplace.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

3. 次のアクセス許可ポリシーを入力します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:DescribeModelPackage",
        "sagemaker:DescribeAlgorithm"
      ],
      "Resource": "*"
    }
  ]
}
```

4. リクエストに応じてロール ARN を指定します。ロールは の形式に従う必要があります `arn:aws:iam::<account-id>:role/<role-name>`。

必要な AWS Marketplace アクセス許可、または販売者アカウントの管理については、[AWS Marketplace 「販売者のポリシーとアクセス許可」](#)を参照してください。

必須アセット

機械学習製品リストを作成する前に、次の必須アセットがあることを確認してください。

- Amazon リソースネーム (ARN) — 発行元の でモデルパッケージまたはアルゴリズムリソースの ARN を指定します (AWS リージョン「」を参照[公開 AWS リージョン でサポート](#))。
- モデルパッケージの ARN の形式は `arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>` です。

モデルパッケージ ARN を確認するには、[「マイマーケットプレイスモデルパッケージ」](#)を参照してください。

- アルゴリズムの ARN の形式は `arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>` です。

アルゴリズムリソース ARN をを見つけるには、[「マイアルゴリズム」](#)を参照してください。

- [the section called “使用状況情報の要件”](#) — 入力、出力、コード例に関する詳細を提供します。

- [the section called “入力と出力の要件”](#) — ファイルまたはテキストを指定します。
- [the section called “Jupyter Notebook の要件”](#) — 製品の完全な使用状況を示します。

ステップ 1: 新しいリストを作成する

機械学習製品の使用を開始するには、製品名を設定し、組織のオプションのリソースタグを追加し、製品 ID を生成して、出品プロセスを開始します。製品 ID は、製品のライフサイクル全体を通じて製品を追跡するために使用されます。

Note

リストを作成する前に、「[機械学習製品を作成するための要件とベストプラクティス](#)」で指定されている必要なリソースがあることを確認してください。

1. 販売者にサインイン AWS アカウント し、 に移動します [AWS Marketplace 管理ポータル](#)。
2. 上部のメニューで、製品に移動し、機械学習を選択します。
3. 機械学習製品の作成 を選択します。
4. 製品名 に、製品出品ページの上部の購入者と検索結果に表示される一意の製品名を入力します。
5. (オプション) タグ に、製品に関連付けるタグを入力します。詳細については、「[AWS リソースのタグ付け](#)」を参照してください。
6. 製品 ID とコード で、製品 ID とコードの生成 を選択します。
7. ウィザードに進む を選択します。ウィザードで詳細な製品情報を追加するプロセスを開始します。

ステップ 2: 製品情報を提供する

機械学習製品を に一覧表示するときは AWS Marketplace、包括的で正確な製品情報を提供することが重要です。ウィザードの「製品情報の提供」ステップを使用して、製品カテゴリやサポート情報など、製品に関する重要な詳細をキャプチャします。

1. 製品に関する情報を入力します。
2. Next を選択して、ウィザードの次のステップに進みます。

ステップ 3: 初期製品バージョンを追加する

このページでは、製品の初期バージョンを追加する方法について説明します。製品にはライフサイクルを通じて複数のバージョンがあり、各バージョンは一意の SageMaker AI ARN によって識別されます。

1. Amazon リソースネーム (ARNsの下 :

a. モデルまたはアルゴリズム Amazon SageMaker AI ARN を入力します。

- モデルパッケージ ARN の例: `arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>`

モデルパッケージ ARN を確認するには、[「マイマーケットプレイスモデルパッケージ」](#)を参照してください。

- アルゴリズム ARN の例: `arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>`

アルゴリズムリソース ARN を見つけるには、[「マイアルゴリズム」](#)を参照してください。

b. IAM アクセスロール ARN を入力します。

IAM ARN の例: `arn:aws:iam::<account-id>:role/<role-name>`

- バージョン情報に、バージョン名とリリースノートを入力します。
- モデル入力の詳細で、モデル入力の概要を入力し、リアルタイムおよびバッチジョブ入力のサンプル入力データを指定します。必要に応じて、任意の入力制限を指定できます。
- (オプション) 入力パラメータで、製品でサポートされている各入力パラメータに関する詳細情報を入力します。パラメータ名、説明、制約を指定し、パラメータが必須かオプションかを指定できます。最大 24 個の入力パラメータを指定できます。
- (オプション) カスタム属性で、製品でサポートされているカスタム呼び出しパラメータを指定します。属性ごとに、名前、説明、制約を指定し、属性が必要かオプションかを指定できます。
- モデル出力の詳細で、モデル出力の概要を入力し、リアルタイムおよびバッチジョブ出力のサンプル出力データを提供します。必要に応じて、任意の出力制限を指定できます。
- (オプション) 出力パラメータで、製品でサポートされている各出力パラメータに関する詳細情報を指定します。パラメータ名、説明、制約を指定し、パラメータが必須かオプションかを指定できます。最大 24 個の出力パラメータを指定できます。

8. 「使用手順」で、ベストプラクティス、一般的なエッジケースの処理方法、パフォーマンス最適化の提案など、モデルを効果的に使用するための明確な手順を提供します。
9. Git リポジトリとノートブックリンクの下に、サンプルノートブックと Git リポジトリへのリンクを提供します。サンプルノートブックには、モデルを呼び出す方法を含める必要があります。Git リポジトリには、ノートブック、データファイル、およびその他の開発者ツールが含まれている必要があります。
10. 推奨インスタンスタイプで、製品の推奨インスタンスタイプを選択します。

モデルパッケージでは、バッチ変換とリアルタイム推論の両方に推奨されるインスタンスタイプを選択します。

アルゴリズムパッケージでは、トレーニングジョブに推奨されるインスタンスタイプを選択します。

Note

選択できるインスタンスタイプは、モデルまたはアルゴリズムパッケージでサポートされているインスタンスタイプに制限されます。これらのサポートされているインスタンスタイプは、Amazon SageMaker AI で最初にリソースを作成したときに決定されました。これにより、製品は機械学習ソリューションを効果的に実行できるハードウェア設定にのみ関連付けられます。

11. Next を選択して、ウィザードの次のステップに進みます。

ステップ 4: 料金モデルを設定する

製品の料金モデルを設定するときは、製品を無料で提供したり、使用量ベースの料金を実装したりできます。製品の公開後に料金モデルを変更することはできません。

1. 料金モデルを選択します。バッチ変換とアルゴリズムトレーニング製品は、時間単位の使用に対してのみ無料または課金されます。
 - 製品を無料で提供することを選択した場合は、次へを選択してウィザードを続行します。
 - 使用料金を選択した場合は、次のステップを続行します。
2. 使用量に基づいて課金することを選択した場合は、使用コストを入力できます。すべてのインスタンスタイプに適用される料金を入力するか、インスタンスタイプごとに料金を入力するかを選択して、より詳細な料金を設定できます。
3. 製品の無料トライアルを提供する場合は、「はい」を選択します。

4. Next を選択して、ウィザードの次のステップに進みます。

ステップ 5: 返金ポリシーを設定する

返金を提供する必要はありませんが、公式の返金ポリシーを提出する必要があります AWS Marketplace。

1. 返金ポリシーを入力します。
2. 次へ を選択して、ウィザードの次のステップに進みます。

ステップ 6: EULA を設定する

このステップでは、顧客が製品を使用する方法を規定する法的契約を選択します。AWS標準の契約条件を選択するか、独自のカスタムエンドユーザーライセンス契約 (EULA) をアップロードできます。

1. 標準契約を選択するか、カスタムエンドユーザーライセンス契約を提供します。
2. Next を選択して、ウィザードの次のステップに進みます。

ステップ 7: 許可リストを設定する

製品を送信する前に、アクセスできるユーザーを指定する必要があります AWS アカウント。このオプションのステップは、製品の初期可視性を制御し、独自のアカウントへのアクセスと、許可リスト AWS アカウント に追加する特別に承認されたすべての へのアクセスを制限します。

1. 製品にアクセスする AWS アカウント IDs を入力します。
2. 送信を選択して製品を送信します。

製品には制限付き可視性ステータスがあり、製品および他の許可リストを作成した AWS アカウント にのみ表示されます AWS アカウント。

ステータスの詳細については、「」を参照してください [the section called “製品ステータス”](#)。

製品リストは、可視性が制限されている間に表示およびテストできます。製品の可視性を変更する準備ができたなら、「」を参照してください [the section called “”](#)。

機械学習製品の管理

AWS Marketplace 管理ポータルで、変更のリクエストを選択して、の製品またはバージョンを変更します AWS Marketplace。変更を送信すると、システムは変更を処理します。処理時間は、変更のタイプに応じて数分から数日で異なります。AWS Marketplace 管理ポータルで変更のステータスをモニタリングできます。

セルフサービス AWS Marketplace を使用して の ML 製品を変更できます。

- [製品情報の更新](#)
- [製品の可視化の更新](#)
- [許可リストの更新](#)
- [製品バージョンの管理](#)
- [製品料金の更新](#)
- [返金ポリシーの更新](#)
- [EULA を更新する](#)
- [製品を削除する](#)

Note

AWS Marketplace 管理ポータルを通じて変更を行うだけでなく、 を使用して変更を行うこともできます [AWS Marketplace Catalog API](#)。

製品情報の更新

機械学習 (ML) 製品を作成したら、説明、ハイライト AWS Marketplace、タイトル、SKU、カテゴリ、キーワードなど、特定の製品情報を で変更できます。

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、ターゲット製品を選択します。
3. 「変更のリクエスト」を選択し、「製品情報の更新」を選択します。
4. 必要に応じてフィールドを更新します。

Note

ロゴの仕様については、「」を参照してください[the section called “会社と製品のロゴ要件”](#)。

5. [Submit] を選択してください。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

製品の可視化の更新

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. 「変更のリクエスト」を選択し、「製品の可視性の更新」を選択し、「パブリック」または「制限付き」を選択します。
4. 変更を確認し、送信を選択します。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

許可リストの更新

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. 変更のリクエストを選択し、許可リストの更新を選択します。
4. 変更する必要がある情報を変更し、送信を選択します。詳細については、「[the section called “ステップ 7: 許可リストを設定する”](#)」を参照してください。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

製品バージョンの管理

販売者は、既存のバージョン情報の更新、新しいバージョンの追加、またはサポートされなくなったバージョンの削除 AWS Marketplace を行うことで、製品バージョンを管理できます。各バージョンには、購入者が製品の評価とデプロイに使用する一意の SageMaker AI ARN と関連情報があります。

Note

バージョンを追加する前に、製品 ID を作成し、料金を設定します。詳細については、「[the section called “ステップ 1: 新しいリストを作成する”](#)」を参照してください。

バージョン情報の更新

バージョンを作成したら、リリースノート、使用手順、インスタンスのレコメンデーションなどの関連情報を変更できます。

Note

バージョン名と ARNs は変更できません。これらの変更では、新しいバージョンを作成する必要があります。

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. 「変更のリクエスト」を選択し、「バージョン情報の更新」を選択します。
4. 更新するバージョンを選択します。
5. バージョンの編集 を選択します。
6. 必要なフィールドを変更し、次へを選択します。
7. 料金情報を入力し、送信を選択します。詳細については、「[the section called “ステップ 4: 料金モデルを設定する”](#)」を参照してください。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

新しいバージョンを追加する

製品の新しいバージョンを追加して、以前のバージョンへのアクセスを維持しながら、機能、更新、または改善を導入できます。

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. バージョン を選択し、新しいバージョンの追加 を選択します。
4. 「」の手順に従って、新しいバージョンの情報を入力します [the section called “ステップ 3: 初期製品バージョンを追加する”](#)。
5. 料金情報を入力し、送信を選択します。詳細については、「[the section called “ステップ 4: 料金モデルを設定する”](#)」を参照してください。

新しいバージョンを正常に追加すると、購入者は新しいバージョンが利用可能であるという E メール通知を受け取ります。

バージョンを制限する

バージョンが古くなったり、可用性を停止したりする場合は、他のバージョンへのアクセスを維持しながら、そのバージョンへの購入者のアクセスを制限できます。

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. バージョン を選択し、バージョンの制限 を選択します。

Note

少なくとも 1 つのバージョンが常に使用可能である必要があります。

4. [Submit] を選択してください。

バージョンを正常に制限すると、購入者はバージョンが制限されたという E メール通知を受け取ります。

製品料金の更新

料金モデル自体は変更できませんが AWS Marketplace、で機械学習製品の料金と無料トライアル期間を変更できます。有料モデルの場合、料金引き上げは翌月の初日の 90 日間の通知期間後に有効になることに注意してください。この通知期間中は、追加の料金変更を行うことはできません。

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. 「変更のリクエスト」を選択し、「パブリックオファ어의更新」を選択し、「オファー情報の編集」を選択します。
4. 変更する必要がある情報を変更し、送信を選択します。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

返金ポリシーの更新

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. 「変更のリクエスト」を選択し、「パブリックオファ어의更新」を選択し、「返金ポリシーの更新」を選択します。
4. 変更する必要がある情報を変更し、送信を選択します。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

EULA を更新する

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. 「変更のリクエスト」を選択し、「パブリックオファ어의更新」を選択し、「EULA の更新」を選択します。
4. 変更する必要がある情報を変更し、送信を選択します。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

製品を削除する

公開済み製品を削除 (サンセット) できます AWS Marketplace。削除されると、新規顧客はサブスクライブできませんが、少なくとも 90 日間は既存の顧客をサポートする必要があります。

以下は、 から製品を削除する条件です AWS Marketplace。

- 製品は AWS Marketplace 検索および検出ツールから削除されます。
- サブスクライブ機能は無効になります。
- 製品の詳細ページには、直接 URL からアクセスできます。
- 現在のサブスクライバーは、サブスクリプションをキャンセルするまでアクセス権を保持します。
- AWS Marketplace 削除について現在の購入者に通知します。

機械学習製品を削除するには：

1. [AWS Marketplace 管理ポータル](#)で販売者アカウントにサインインします。
2. 機械学習製品ページに移動し、製品を選択します。
3. 「変更のリクエスト」を選択し、「製品の可視性の更新」を選択し、「制限付き」を選択します。
4. (オプション) 代替製品 ID を入力します。
5. 変更を確認し、送信を選択します。

機械学習製品ページのリクエストタブからリクエストをモニタリングできます。ステータスの詳細については、「」を参照してください[the section called “製品ステータス”](#)。

削除されると、現在の製品リストに製品が表示され、製品スプレッドシートのみダウンロードできます。製品の削除についてご質問がある場合は、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。

機械学習製品のプライベートオファ어의作成

機械学習製品のプライベートオファ어를交渉して顧客に直接提供できます。プライベートオファ어의詳細については、「」を参照してください[プライベートオファ어의準備](#)。

前提条件:

- 有料出品が必要です AWS Marketplace。
- AWS Marketplace 管理ポータル (AMMP) にアクセスできる必要があります。

機械学習製品のプライベートオファーを作成するには：

1. AWS Marketplace 管理ポータルにサインインします。
2. オファーを選択し、プライベートオファーの作成を選択します。
3. [プライベートオファーの作成] ページで、プライベートオファーを作成する製品を選択します。作成できるのは、利用可能な製品のオファーのみです。
4. オファーの詳細ページで、次の操作を行います。
 - a. オファーの名前と説明を入力します。
 - b. 更新オプションを選択します。
 - c. オファーの有効期限を設定します。オファーは、設定した日付の 23:59:59 UTC に期限切れになります。
5. [次へ] を 2 回選択します。
6. オファーの料金と期間の設定ページで、以下を指定します。

- 料金オプション

(詳細については「[the section called “機械学習製品の料金”](#)」を参照してください)。

- 契約期間
- オファー通貨
- 製品ディメンション

Note

分割プランの詳細については、「」を参照してください [プライベートオファーの分割プラン](#)。

7. [次へ] を選択します。
8. 購入者の追加ページで、購入者の AWS アカウント IDs を入力します。次いで、[次へ] を選択します。

⚠ Important

リンクされたアカウントがプライベートオファーの恩恵を受けるには：

- 支払者 AWS アカウント ID を含めます。
- 支払いアカウントは、まずプライベートオファーの時間単位の条件に同意する必要があります。
- 支払いアカウントが承諾すると、リンクされたアカウントはプライベートオファーを受け入れることができます。

9. 法的条件とオファードキュメントの設定ページで、カスタム条件を追加し、次へを選択します。

i Note

最大 5 つのファイル (法的条件、作業明細書、部品表、料金表、または付録) を追加できます。システムはこれらを 1 つのドキュメントに結合します。

10. 確認と作成ページで、オファーの詳細を確認し、オファーの作成を選択します。

11. オファーがプライベートオファーの管理ページに表示されたら、アクションメニューを開き、オファー URL をコピーを選択して購入者に E メールで送信します。

i Note

オファーの公開には時間がかかる場合があります。購入者が承諾するまで、プライベートオファーの管理ページでオファーを編集できます。

機械学習製品を作成するための要件とベストプラクティス

モデルパッケージとアルゴリズム製品を購入者が簡単にテストできるかを確認することが重要です。以下のセクションでは、ML 製品のベストプラクティスについて説明します。要件と推奨事項の完全な要約については、「[ML 製品リストの要件と推奨事項の要約](#)」を参照してください。

 Note

公開済み製品が要件を満たさない場合、これらの要件を満たすために AWS Marketplace 担当者から連絡されることがあります。

トピック

- [ML 製品の一般的なベストプラクティス](#)
- [使用状況情報の要件](#)
- [入力と出力の要件](#)
- [Jupyter Notebook の要件](#)
- [ML 製品リストの要件と推奨事項の要約](#)

ML 製品の一般的なベストプラクティス

使用している機械学習製品について、以下の情報を提供します。

- 製品の説明には、以下を含めます。
 - モデルの機能
 - ターゲットとなる顧客
 - 最も重要なユースケース
 - モデルのトレーニング方法、または使用されたデータの量
 - パフォーマンスメトリクスと、使用された検証データ
 - 医療機器の場合、モデルが診断用かどうか
- デフォルトでは、機械学習製品は一般に公開されるよう設定されています。ただし、可視性が制限された製品を作成できます。詳細については、「[the section called “ステップ 7: 許可リストを設定する”](#)」を参照してください。
- (オプション) 有料製品の場合は、顧客に対して 14～30 日間の無料試用期間を提供し、製品を試してもらいます。詳細については、「[の機械学習製品の料金 AWS Marketplace](#)」を参照してください。

使用状況情報の要件

購入者の満足度を高めるには、製品の期待される入力と出力を説明した明確な使用情報 (例を含む) が不可欠です。

新しいバージョンのリソースを製品リストに追加するたびに、使用状況情報を提供する必要があります。

特定のバージョンの既存の使用状況情報を編集するには、「」を参照してください[the section called “バージョン情報の更新”](#)。

入力と出力の要件

サポートされている入力パラメータと返された出力パラメータを例とともに明確に説明することは、購入者が製品を理解して使用できるようにするために重要です。製品を理解すると、購入者は、入力データに対して必要な変換を行って最適な推論結果を得られるようになります。

Amazon SageMaker AI リソースを製品リストに追加すると、次のプロンプトが表示されます。

推論の入力と出力

推論入力の場合、リアルタイムエンドポイントジョブとバッチ変換ジョブの両方で製品が期待する入力データの説明を入力します。データの事前処理に必要なコードスニペットを含めてください。該当する場合は制限を含めます。[GitHub](#) でホストされている入力サンプルを提供します。

推論出力の場合、リアルタイムエンドポイントジョブとバッチ変換ジョブの両方で製品が返す出力データの説明を入力します。該当する場合は制限を含めます。[GitHub](#) でホストされている出力サンプルを提供します。

サンプルには、お使いの製品に対応する入力ファイルを提供してください。モデルが多クラス分類を実行する場合は、クラスごとに少なくとも 1 つのサンプル入力ファイルを提供してください。

トレーニング入力

「モデルをトレーニングするための情報」セクションに、データの事前処理に必要な入力データ形式とコードスニペットを指定します。該当する場合は、値と制限の説明を含めます。[GitHub](#) でホストされている入力サンプルを提供します。

購入者が提供できるオプション機能と必須機能の両方を説明し、PIPE 入力モードがサポートされているかどうかを明記します。[分散型トレーニング](#) (複数の CPU/GPU インスタンスによるトレーニング) がサポートされている場合は、その旨を明記してください。チューニングについては、推奨されるハイパーパラメータを一覧で表示してください。

Jupyter Notebook の要件

SageMaker AI リソースを製品リストに追加するときは、[GitHub](#) でホストされているサンプル Jupyter ノートブックへのリンクを提供します。このノートブックは、購入者にデータのアップロードや検索を依頼することなく、完全なワークフローを示しています。

AWS SDK for Python (Boto)を使用します。優れたサンプルノートブックを開発して提供できれば、購入者はリストを簡単に試して使用することができます。

モデルパッケージ製品の場合、サンプルノートブックには、入力データの準備、リアルタイム推論用エンドポイントの作成、バッチ変換ジョブの実行などが示されています。詳細については、GitHub の「[Model Package listing and Sample notebook](#)」を参照してください。サンプルノートブックについては、「[auto_insurance](#)」を参照してください。ノートブックは、パラメータを入力することなく AWS リージョン、購入者がサンプルデータを見つけることなく、すべて機能します。

Note

未開発のサンプル Jupyter Notebook ノートブックで、考えられる入力やデータの前処理手順が複数表示されていないと、購入者が製品の価値提案を完全に理解するのが難しい場合があります。

アルゴリズム製品の場合、サンプルノートブックには、トレーニング、調整、モデル作成、リアルタイム推論用エンドポイントの作成、バッチ変換ジョブのパフォーマンスがすべて示されています。詳細については、GitHub の「[Algorithm listing and Sample notebook](#)」を参照してください。サンプルノートブックについては、GitHub の「[amazon_demo_product](#)」と「[automl](#)」を参照してください。これらのサンプルノートブックはすべてのリージョンで機能し、パラメータを入力したり、購入者がサンプルデータを探したりする必要はありません。

Note

サンプルのトレーニングデータが不足していると、購入者が Jupyter Notebook を正常に実行できなくなります。サンプルノートブックが開発されていないと、購入者が製品を使用できなくなり、採用が見合わされるおそれがあります。

ML 製品リストの要件と推奨事項の要約

次の表に、機械学習製品の出品ページの要件と推奨事項の概要を示します。

詳細	モデルパッケージリスト用	アルゴリズムリスト用
Product descriptions		
Explain in detail what the product does for supported content types (for example, "detects X in images").	Required	Required
Provide compelling and differentiating information about the product (avoid adjectives like "best" or unsubstantiated claims).	Recommended	Recommended
List most important use case(s) for this product.	Required	Required
Describe the data (source and size) it was trained on and list any known limitations.	Required	Not applicable
Describe the core framework that the model was built on.	Recommended	Recommended
Summarize model performance metric on validation data (for example, "XX.YY percent accuracy benchmarked using the Z dataset").	Required	Not applicable
Summarize model latency and/or throughput metrics on recommended instance type.	Required	Not applicable
Describe the algorithm category. For example, "This decision forest regression algorithm is based on an	Not applicable	Required

詳細	モデルパッケージリスト用	アルゴリズムリスト用
ensemble of tree-structured classifiers that are built using the general technique of bootstrap aggregation and a random choice of features.”		
Usage information		
For inference, provide a description of the expected input format for both the real-time endpoint and batch transform job. Include limitations, if applicable. See 入力と出力の要件 .	Required	Required
For inference, provide input samples for both the real-time endpoint and batch transform job. Samples must be hosted on GitHub. See 入力と出力の要件 .	Required	Required
For inference, provide the name and description of each input parameter. Provide details about the its limitations and specify if it is required or optional.	Recommended	Recommended

詳細	モデルパッケージリスト用	アルゴリズムリスト用
For inference, provide details about the output data your product returns for both the real-time endpoint and batch transform job. Include any limitations, if applicable. See 入力と出力の要件 .	Required	Required
For inference, provide output samples for both the real-time endpoint and batch transform job. Samples must be hosted on GitHub. See 入力と出力の要件 .	Required	Required
For inference, provide an example of using an endpoint or batch transform job. Include a code example using the AWS Command Line Interface (AWS CLI) commands or using an AWS SDK.	Required	Required
For inference, provide the name and description of each output parameter. Specify if it is always returned.	Recommended	Recommended
For training, provide details about necessary information to train the model such as minimum rows of data required. See 入力と出力の要件 .	Not applicable	Required

詳細	モデルパッケージリスト用	アルゴリズムリスト用
For training, provide input samples hosted on GitHub. See 入力と出力の要件 .	Not applicable	Required
For training, provide an example of performing training jobs. Describe the supported hyperparameters, their ranges, and their overall impact. Specify if the algorithm supports hyperparameter tuning, distributed training, or GPU instances. Include code example such as AWS CLI commands or using an AWS SDK, for example.	Not applicable	Required
Provide a Jupyter notebook hosted on GitHub demonstrating complete use of your product. See Jupyter Notebook の要件 .	Required	Required
Provide technical information related to the usage of the product, including user manuals and sample data.	Recommended	Recommended

機械学習製品の問題のトラブルシューティング

このセクションでは、機械学習製品の公開プロセス中に発生する可能性のある一般的なエラーについて説明します。該当する問題がリストにない場合は、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。

全般: モデルパッケージまたはアルゴリズムの Amazon リソースネーム (ARN) を に追加すると、400 エラーが発生します。 AWS Marketplace 管理ポータル

一般的な原因

SageMaker AI で機械学習製品を作成するときに、製品を公開することを選択していませんでした AWS Marketplace。

解決方法

Amazon SageMaker AI コンソールを使用してリソースを作成した場合は、「このモデルパッケージを で公開 AWS Marketplaceする」のプロセスの最終ページで「はい」または「このアルゴリズムを で公開 AWS Marketplaceする」を選択する必要があります。[いいえ] を選択すると、後で公開することはできません。[はい] を選択しても、モデルパッケージやアルゴリズムは公開されません。ただし、モデルパッケージまたはアルゴリズムリソースは作成時に検証されるため、AWS Marketplace での使用に必要です。

AWS SDK を使用して [モデルパッケージを作成する](#) か、 [アルゴリズムを作成する](#) 場合は、パラメータ `CertifyForMarketplace` が に設定されていることを確認します `true`。

認定および検証済みのモデルパッケージまたはアルゴリズムリソースを再作成したら、新しい ARN を AWS Marketplace 管理ポータルに追加します。

全般: モデルパッケージまたはアルゴリズムの ARN を に追加すると、404 エラーが発生します。 AWS Marketplace 管理ポータル

一般的な原因

このエラーは、さまざまな理由で発生します。

- ARN が無効の可能性があります。
- モデルパッケージまたはアルゴリズムリソースが、販売者アカウント AWS アカウント と同じ に作成されていません。
- 公開に使用するユーザーまたはロールに、モデルパッケージまたはアルゴリズムリソースにアクセスするための適切な IAM 権限がありません。

解決方法

1. ARN をチェックして、ARN が正しいこと、および想定される形式であることを確認します。

モデルパッケージの場合、ARN は `arn:aws:sagemaker:us-east-2:000123456789:model-package/my-model-package-name` に似ています。

アルゴリズムの場合、ARN は `arn:aws:sagemaker:us-east-2:000123456789:algorithm/my-algorithm` に似ています。

2. 公開するリソースとアセットがすべて、公開元の販売者アカウントにあることを確認してください。
3. ユーザーまたはロールに、次のアクセス許可があることを確認してください。

モデルパッケージの場合、モデルパッケージリソースに対するアクション `sagemaker:DescribeModelPackage` が許可されている必要があります。

アルゴリズムの場合は、アルゴリズムリソースに対するアクション `sagemaker:DescribeAlgorithm` が許可されている必要があります。

Amazon SageMaker AI: モデルパッケージまたはアルゴリズムリソースを作成すると、「クライアントエラー: レジストリへのアクセスが拒否されました」という失敗メッセージが表示されます

一般的な原因

このエラーは、モデルパッケージまたはアルゴリズムの作成に使用されているイメージが、別の AWS アカウントに属する [Amazon ECR](#) リポジトリに保存されている場合に発生することがあります。モデルパッケージまたはアルゴリズムの検証は、クロスアカウントイメージをサポートしていません。

解決方法

公開に AWS アカウント 使用している が所有する Amazon ECR リポジトリにイメージをコピーします。次に、新しいイメージロケーションを使用してリソースを作成します。

Amazon SageMaker AI: 「開始されていません」と「クライアントエラー: スキャンがスケジュールされていません...」が表示されます。モデルパッケージまたはアルゴリズムリソースを作成すると、失敗するメッセージが表示される

一般的な原因

このエラーは、SageMaker AI が Amazon ECR に保存されている Docker コンテナイメージのスキャンを開始できない場合に発生する可能性があります。

解決方法

この場合は、[Amazon ECR コンソール](#)を開き、イメージがアップロードされた先のリポジトリを見つけてイメージを選択し、[スキャン] を選択してください。

の SaaS ベースの製品 AWS Marketplace

Software as a Service (SaaS) 製品を使用すると、インフラストラクチャで AWS ホストされているソフトウェアをデプロイし、AWS Marketplace 購入者に AWS 環境内のソフトウェアへのアクセスを許可できます。販売者は、ソフトウェア内での顧客のアクセスやアカウント作成、リソースのプロビジョニング、アカウント管理まで行います。最初の SaaS 製品を作成したら、変更リクエストを送信し、製品の説明 AWS Marketplace、国別の可用性、料金など、でそのプロパティを設定できます。また、SaaS サブスクリプションと契約の統合も設定します。以下のトピックでは、開始するために必要な情報について説明します。

API を Amazon API Gateway と統合する方法については、「Amazon API Gateway デベロッパーガイド」の「[AWS Marketplace で API Gateway API を販売する](#)」を参照してください。

SaaS 製品についてサポートが必要な場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

トピック

- [での SaaS 製品の開始方法 AWS Marketplace](#)
- [SaaS 製品の計画](#)
- [の SaaS 製品ガイドライン AWS Marketplace](#)
- [での SaaS 製品の料金 AWS Marketplace](#)
- [での SaaS 無料トライアルオファーの作成 AWS Marketplace](#)
- [を通じて SaaS 製品に顧客をオンボーディングする AWS Marketplace](#)
- [SaaS 製品に関する Amazon SNS 通知](#)
- [AWS Marketplace Metering Service API および使用権限管理サービス API へのアクセス](#)
- [での SaaS 製品のレポート AWS Marketplace](#)
- [SaaS 製品統合のコード例](#)
- [を使用して Amazon VPC 経由で製品を配信する AWS PrivateLink](#)

での SaaS 製品の開始方法 AWS Marketplace

AWS Marketplace 販売者は、Software as a Service (SaaS) 製品を に追加できます AWS Marketplace。これには、SaaS 製品の請求モデルに基づいて、SaaS 製品を適切な AWS

Marketplace API オペレーションと統合することが含まれます。このトピックでは、開始するために必要な前提条件や、SaaS 製品を作成および設定するプロセスの概要を説明します。

前提条件

開始するには、以下の前提条件を満たす必要があります。

1. [AWS Marketplace 管理ポータル](#) にアクセスして使用する。これは、販売者として登録し、販売する製品を管理するために使用するツールです AWS Marketplace。詳細については、「[AWS Marketplace 管理ポータル](#)」を参照してください。
2. 販売者として登録し、税金および銀行の情報を提出します。詳細については、「[AWS Marketplace 販売者としての登録](#)」を参照してください。
3. AWS Marketplaceで SaaS 製品を作成および統合する方法を計画する。詳細については、「[SaaS 製品の計画](#)」を参照してください。

次のステップ

必要な前提条件を完了したら、SaaS 製品を作成して設定できます。

トピック

- [での SaaS 製品のライフサイクル AWS Marketplace](#)
- [での SaaS 製品の作成 AWS Marketplace](#)
- [での最初の SaaS 製品ページの作成 AWS Marketplace](#)
- [での SaaS 製品設定の構成 AWS Marketplace](#)
- [SaaS サブスクリプション製品を と統合する AWS Marketplace](#)
- [SaaS 契約製品と の統合 AWS Marketplace](#)
- [SaaS 契約をpay-as-you-go製品と統合する AWS Marketplace](#)
- [サーバーレス SaaS 統合のデプロイ](#)

での SaaS 製品のライフサイクル AWS Marketplace

で Software as a Service (SaaS) 製品を作成すると AWS Marketplace、最初は可視性が制限されて公開され、アカウントのみがアクセスできます。準備ができたら、AWS Marketplace カタログに公開して、購入者が製品をサブスクライブして購入できるようにします。次のトピックでは、SaaS 製品

のライフサイクルについて説明します。SaaS 製品の作成に関する詳細については、「[での SaaS 製品の作成 AWS Marketplace](#)」を参照してください。

SaaS 製品ページでは、製品のリストを表示できます。製品ライフサイクルの段階に応じて、製品のステータスは以下のいずれかになります。

- ステージング - 情報を追加中の、不完全な製品。セルフサービスエクスペリエンスの最初の保存して終了で、変更リクエストが成功すると、送信した完了したステップの情報を含む未公開の製品が作成されます。このステータスから、引き続き商品に情報を追加したり、変更リクエストを通じて送信済みの情報を変更したりできます。
- 制限あり - 製品がシステムに送信され、システム内のすべての検証に合格すると完成します。その後、製品は制限あり状態にリリースされます。この時点で、製品にはアカウントと許可リストに登録したユーザーだけがアクセスできる詳細ページがあります。商品は詳細ページからテストできます。詳細またはヘルプについては、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。
- パブリック - 購入者が商品を開覧してサブスクライブできるように商品を公開する準備ができたら、可視性を更新する変更リクエストを使用します。このリクエストは、AWS Marketplace Seller Operations チームが AWS ポリシーに照らして製品を確認および監査するためのワークフローを開始します。製品が承認され、変更リクエストが処理されると、製品のステータスが制限ありからパブリックに移行します。AWS ガイドラインの詳細については、[SaaS 製品のガイドライン](#)」を参照してください。
- 制限付き - 新規ユーザーによる製品のサブスクライブを停止する場合は、可視性の更新変更リクエストを使用して製品を制限できます。制限付き ステータスは、既存のユーザーが引き続き製品を使用できることを意味します。ただし、製品は一般には公開されなくなり、新規ユーザーも利用できなくなります。

ステージング、制限あり、パブリックの各ステータスで製品を更新できます。詳細については、「[製品情報の更新](#)」を参照してください。

での SaaS 製品の作成 AWS Marketplace

AWS Marketplace 販売者は、Software as a Service (SaaS) 製品を に追加できます AWS Marketplace。これには、SaaS 製品を作成し、請求モデルに基づいて適切な AWS Marketplace API オペレーションと統合することが含まれます。でソフトウェアを SaaS 製品として販売するには AWS Marketplace、以下の広範なステップに従います。

- で SaaS 製品を作成します AWS Marketplace。

- SaaS [サブスクリプション](#)、[契約](#)、または[従量制料金の契約](#)の製品を AWS Marketplaceと統合します。
- [サブスクリプション](#)、[契約](#)、または[従量制料金の契約](#)の製品の AWS Marketplaceとの統合をテストします。
- 製品を発売用に送信します。

次の手順では、で SaaS 製品を作成する方法を示します AWS Marketplace。

SaaS 製品の作成

SaaS 製品を作成するには

1. SaaS 製品の出品を決定する

販売したい SaaS 製品がある AWS Marketplace。 [SaaS 製品の計画](#) する方法を確認し、理解します。

2. 価格とオファータイプを決定する

SaaS 製品には、サブスクリプション、契約、従量制料金の契約の 3 つのオファータイプがあります。オファータイプの選択は、SaaS 製品の統合方法に影響します AWS Marketplace。詳細については、「[料金の計画](#)」を参照してください。

3. アセットを収集する

製品の提出に必要なアセットを収集します。アセットには以下が含まれます。

- 製品ロゴ URL - 提供する製品の鮮明なロゴイメージが含まれている公開されている Amazon S3 URL です。
- エンドユーザー使用許諾契約 (EULA) URL - 製品には PDF ファイル形式で入手できる EULA が必要です。顧客が製品の AWS Marketplace ページで EULA を確認できる Amazon S3 バケットへのリンクを提供する必要があります。
- 製品登録 URL - これは、購入者が製品を AWS Marketplaceで正常にサブスクライブした後にリダイレクトされる URL です。
- 製品に関するメタデータ - AWS Marketplace 管理ポータルで製品作成ウィザードでメタデータを入力します。
- 製品のサポート情報 - この情報には、製品のサポートチャネルの E メールアドレスと URL が含まれます。

4. 製品を統合に送信する

販売者アカウントと AWS Marketplace 管理ポータルを使用して、製品を限定製品として AWS Marketplace 公開します [での最初の SaaS 製品ページの作成 AWS Marketplace](#)。つまり、統合とテストにのみ使用できます。製品コードと、Amazon Simple Notification Service (SNS) トピックは、製品概要ページに表示されます。

 Note

販売者と AWS Marketplace Seller Operations チームが高額なコストをかけずに製品をテストできるように、製品は割引価格のままにする必要があります。製品の公開をリクエストする際に、製品の実際の料金を確認します。

5. との統合 AWS Marketplace

製品は、アクセスを許可する前のサブスクリプションの検証や、場合によっては使用量の測定を含め、顧客が製品をオンボードして使用できるようにする必要があります。との統合方法は、製品に使用しているオファータイプ AWS Marketplace によって異なります。オファータイプに基づく統合の詳細については、以下のトピックを参照してください。

- [サブスクリプション統合](#)
- [契約統合](#)
- [従量制料金による契約統合](#)

製品を と統合する最後のステップ AWS Marketplace は、統合が正しく機能することを確認するためにテストすることです。

6. 製品のテストと契約のキャンセル

統合プロセスが完了したら、独自の製品をサブスクライブしてカスタマーエクスペリエンスを評価し、確認することをお勧めします。テストサブスクリプションを作成すると、次のことが可能になります。

- 製品情報を確認する
- 利用可能な購入オプションを確認する
- テスト支払いの処理
- 独自の製品ウェブサイトへのリンクを確認する

⚠ Important

製品の可視性を限定から公開に変更する前に、テストサブスクリプションをキャンセルします。サポートが必要な場合は、サポートチケットを送信して [Marketplace Seller Operations チーム](#) にお問い合わせください。

7. 製品を発売用に送信する

統合を確認し、製品を公開する準備ができたなら、[可視性を更新] を選択します。Seller AWS Marketplace Operations チームは、可視性を公開に更新する前に、製品を確認し、価格を更新します。

i Note

AWS Marketplace Seller Operations は、手動プロセスを使用して SaaS 製品を検証および更新します。このプロセスでは、可視性をパブリックに更新するのに 7~10 営業日かかり、チームがエラーを見つけた場合はさらに時間がかかります。タイミングの詳細については、本ガイドの「[Timing and expectations](#)」を参照してください。

での最初の SaaS 製品ページの作成 AWS Marketplace

Software as a Service (SaaS) アプリケーションメタデータを使用して、を使用して AWS Marketplace カタログに最初の SaaS 製品ページを作成できます AWS Marketplace 管理ポータル。また、製品情報、製品デプロイの詳細、およびパブリックオファーの詳細を追加することもできます。オプションで、許可リストにアカウントを追加して製品をテストすることもできます。詳細については、以下の手順を参照してください。

初期 SaaS 製品ページを作成するには

1. [AWS Marketplace 管理ポータル](#) にサインインします。
2. [SaaS 製品の作成] を選択し、[SaaS 製品] を選択します。
3. SaaS 製品 ID とコードを生成します。タグベースの承認をサポートするオプションのタグを追加することもできます。

Note

タグベースの認可の詳細については、「[AWS Identity and Access Management ユーザーガイド](#)」の「[タグを使用した AWS リソースへのアクセスの制御](#)」を参照してください。

4. セルフサービスエクスペリエンスを使用して AWS Marketplace リストを作成します。製品情報、製品デプロイの詳細、および公開オファ어의詳細を追加します。オプションで、許可リストにアカウントを追加して製品をテストすることもできます。

Note

ステップを完了する前にセッションを終了する必要がある場合は、[保存して終了] オプションを選択して、現在の選択内容をステージング領域に保存します。このオプションにより、入力した情報の検証を求めるリクエストが作成されます。リクエストが検証されている間は、製品を編集することはできません。リクエストが成功すると、[製品の作成を再開する] を選択して製品の作成を続行できます。

リクエストが成功しなかった場合は、製品リクエストログに表示される検証エラーが原因です。リクエストを選択してエラーを表示し、[アクション] の [新規にコピー] を選択してエラーを修正し、リクエストを再送信できます。前の手順を更新するには、製品詳細ページを開いて変更リクエストを送信します。

Note

テスト中の料金は、ディメンションごとにデフォルトで 0.001 USD になります。この料金設定により、高額な請求を発生させることなく、制限あり状態で製品をテストできます。実際の料金は、製品を公開する際に提示します。

5. [Submit] を選択してください。次に、は情報 AWS Marketplace を検証します。検証が成功すると、は制限付きステータスで製品を AWS Marketplace リリースします。検証が成功すると、製品をプレビュー、統合、テストできます。

Note

検証が処理されている間は、製品を編集することはできません。製品が最初に公開されると、製品の作成 AWS アカウント に使用した と AWS Marketplace Seller Operations

チームのテストアカウントでのみアクセスできます。SaaS 製品ページから製品を表示する場合は、[で表示](#) AWS Marketplaceを選択して、購入者 AWS Marketplace に表示される製品の詳細を表示できます。この詳細リストは他の AWS Marketplace ユーザーには表示されません。

での SaaS 製品設定の構成 AWS Marketplace

で [Software as a Service \(SaaS\) 製品を作成](#)したら AWS Marketplace、多くの製品設定を変更できます。以下のセクションでは、変更リクエストを送信し、料金の詳細、製品の可視性、その他の設定の更新など、製品設定を変更する方法について説明します。

トピック

- [変更リクエストを管理する](#)
- [製品情報を更新する](#)
- [アーキテクチャの詳細を更新する](#)
- [AWS アカウント IDsの許可リストを更新する](#)
- [製品の可視性を更新する](#)
- [料金条件を更新する](#)
- [料金設定ディメンションを追加する](#)
- [料金設定ディメンションを更新する](#)
- [料金ディメンションを制限する](#)
- [購入者が製品にアクセスする方法を決定する](#)
- [クイック起動を設定する](#)
- [国別の可用性を更新する](#)
- [製品の返金ポリシーを更新する](#)
- [エンドユーザー使用許諾契約書 \(EULA\) を更新する](#)

変更リクエストを管理する

[セルフサービスリスティング](#)では、変更リクエストを使用して製品に変更を加えます。現在のリクエストは、[リクエストタブ](#) AWS Marketplace 管理ポータル の にあります。ナビゲーションバーの下にある [変更をリクエスト] ドロップダウンリストから新しいリクエストを行うことができます。

SaaS 製品の変更リクエストを作成するには

1. を開き [AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [製品] タブのドロップダウンリストから [SaaS] を選択します。
3. リクエストが送信されると、処理が開始されます。変更リクエストは、[レビュー中]、[変更を準備中]、[変更を適用中] のステータスになります。
4. リクエストの処理が完了すると、そのステータスは以下のいずれかの値に変わります。
 - 成功 - このステータスは、リクエストされた変更が処理され、変更がシステムに反映されたことを示します。
 - 失敗 - このステータスは、リクエストに何か問題が生じたため、変更が処理されなかったことを示します。ステータスが、[失敗] の場合は、リクエストを選択して、問題の修正方法に関する推奨事項を示すエラーコードを検索できます。エラーのトラブルシューティングを行い、変更の新しいリクエストを作成できます。処理を速めるには、[新しいリクエストにコピー] 機能を使用して失敗したリクエストの詳細をコピーできます。必要な変更を加えてリクエストを再送信できます。

更新で始まる変更リクエストには、プロジェクトの現在の詳細が読み込まれます。その後、更新を行うと、既存の詳細が上書きされます。リクエストの追加と制限のペアは、リクエストが成功するたびに (セルフサービスエクスペリエンスで [保存して終了] と [送信] アクションを選択した後) プロビジョニングされる更新専用です。つまり、既存のサブスクライバーは、サブスクリプションまたは契約が終了するまで製品を使い続けることができます。ただし、[制限付き] ステータスの製品には新しいサブスクライバーを追加することはできません。

製品情報を更新する

製品を作成した後に、AWS Marketplaceでその製品に関連する情報を変更する必要がある場合があります。

1. を開き [AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) ページの [SaaS 製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンリストから [製品情報を更新] を選択します。
4. 変更する次のフィールドのいずれかを更新します。
 - [製品のタイトル]
 - [SKU]

- 短い説明
- [詳しい説明]
- 製品ロゴの URL
- [ハイライト]
- [製品カテゴリ]
- キーワード
- [製品動画 URL]
- リソース
- [サポート情報]

 Note

ロゴの形式について詳しくは、「[会社と製品のロゴ要件](#)」を参照してください。

5. 製品情報を更新するには、[送信] を選択します。
6. リクエストが [リクエスト] タブに [レビュー中] ステータスで表示されていることを確認します。新しいリクエストが表示されるようにするには、ページの更新が必要になる場合があります。

アーキテクチャの詳細を更新する

製品がデプロイされる特別な指定を受け取るには AWS、ホスティングパターンを選択し、アーキテクチャ図をアップロードして、AWS Marketplace 管理ポータル (AMMP) で製品のアーキテクチャの詳細を更新します。がにデプロイされている AWS と見なすホスティングパターンのリストについては AWS、「」を参照してください[ガイドライン](#)。

アーキテクチャの詳細を更新するには

1. [AWS Marketplace 管理ポータル](#) にサインインします。
2. 製品メニューで、SaaS を選択します。
3. SaaS 製品テーブルで、製品を選択します。
4. 「変更のリクエスト」で、「アーキテクチャの詳細の更新」を選択します。
5. ホスティングパターンを選択します。がデプロイされた AWS Marketplace と見なすホスティングパターンを選択した場合は AWS、次の 2 つの追加ステップを実行します。デプロイされたと

見なされるパターンの詳細については AWS、このガイドの前半の[ガイドライン](#)「」を参照してください。

- a. プロンプトが表示されたら、「アーキテクチャ図」セクションで「ファイルの選択」を選択して、アーキテクチャ図を PNG または JPG 形式でアップロードします。図の詳細については、「」を参照してください[アーキテクチャ図の作成](#)。
 - b. プロンプトが表示されたら、アプリケーションプレーンで、アプリケーションが実行される場所を選択します。詳細については、[「コントロールプレーンとアプリケーションプレーン」](#)を参照してください。
6. アーキテクチャの詳細の更新を選択します。

リクエストが完了すると、リクエストのステータスは成功に変わります。リクエストのステータスを確認するには、SaaS 製品テーブルで製品を選択し、リクエストログタブを選択します。

評価結果を表示するには、SaaS 製品テーブルで製品を選択し、アーキテクチャの詳細タブを選択します。

AWS アカウント IDsの許可リストを更新する

制限された状態で製品を表示できる AWS アカウント IDs のリストを変更できます。

1. を開き[AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) ページの [\[SaaS 製品\]](#) タブで、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストから [\[許可リストを更新\]](#) を選択します。リストには、現在許可リストに登録されている AWS アカウント IDsが表示されます。
4. [\[許可リストに登録された AWS アカウント\]](#) フィールドに AWS アカウント ID を入力し、カンマで区切ります。
5. AWS アカウント IDs の許可リストを更新するには、送信を選択します。

製品の可視性を更新する

クイック起動エクスペリエンスを表示できる購入者を変更するには AWS Marketplace、Update visibility を使用できます。

1. を開き[AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) ページから、変更する製品を選択します。

3. [変更をリクエスト] ドロップダウンから [可視性を更新] を選択します。

 Note

この変更リクエストを使用して、製品を [制限あり] ステータスから [パブリック] ステータスに移行するようリクエストできます。ただし、変更リクエストをパブリックに移行するには、AWS Marketplace Seller Operations チームの承認プロセスを経る必要があります。

4. 公開するときは、商品の実際の価格を入力します。この価格は、リストが一般に公開されることが承認された後に適用されます。
5. リクエストをレビューに送信するには、[送信] を選択します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

料金条件を更新する

SaaS 製品のディメンションあたりの料金を変更するには、Update 料金条件を使用します。

 Note

任意のディメンションの料金を引き上げると、料金更新オプションは少なくとも次の 90 日間は使用できなくなります。値下げと値上げの両方を更新する場合は、まず値下げを更新してください。

1. を開き [AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) ページの [SaaS 製品] タブで、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンリストから [パブリックオファーを更新] を選択し、[料金条件を更新] を選択します。
4. フィールドには現在の価格があらかじめ入力されています。現在の料金を削除して、新しい料金を追加できます。
5. リクエストをレビューに送信するには、[送信] を選択します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] または [失敗] に更新されます。

料金設定ディメンションを追加する

製品の請求に使用するディメンションを追加できます。ディメンションは、製品を使用する際に購入者に請求される基本的な測定単位です。

Note

既存の料金設定ディメンションの名前または説明を更新するには、「[the section called “料金設定ディメンションを更新する”](#)」を参照してください。

1. を開き [AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) タブから、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンから [\[料金設定ディメンションを更新\]](#) を選択し、[\[料金設定ディメンションを追加\]](#) を選択します。
4. ディメンション API ID、表示名、説明を入力して製品に新しいディメンションを追加し、[\[次へ\]](#) を選択します。

Note

API ID と名前は、すべてのディメンションにわたって一意である必要があります。ディメンションの作成後に API ID と単位を変更することはできません。

5. 追加した各ディメンションの料金を定義し、[\[次へ\]](#) を選択して変更を確認します。

Note

ディメンションを追加できるのは、製品用に選択した料金モデル (例えば、契約、使用量、SaaS 契約 (追加使用量あり) など) のみです。限定製品の場合、新しく追加されたディメンションの価格は 0.01 USD に設定されます。商品が一般に公開されるようになったら、料金を更新できます。

6. [\[送信\]](#) を選択して、リクエストを送信します。
7. [\[リクエスト\]](#) タブで、リクエストステータスが [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスは成功に変わります。

料金設定ディメンションを更新する

製品の請求に使用するディメンションを更新できます。ディメンションは、製品を使用する際に購入者に請求される基本的な測定単位です。

1. を開き [AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) タブから、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンから [\[料金設定ディメンションを更新\]](#) を選択し、[\[ディメンション情報を更新\]](#) を選択します。
4. 更新するディメンションを検索し、名前または説明を選択します。
5. 新しい名前または説明を入力し、チェックマークを選択して更新を確定します。ディメンション名は一意である必要があります。
6. [\[送信\]](#) を選択して、リクエストを送信します。
7. [\[リクエスト\]](#) タブで、リクエストステータスが [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスが [\[成功\]](#) に変わります。

料金ディメンションを制限する

製品に現在掲載されているディメンションを制限できます。このリクエストにより、選択したディメンションが製品から削除されます。

1. を開き [AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) タブから、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンから [\[料金設定ディメンションを更新\]](#) を選択し、[\[料金設定ディメンションを制限\]](#) を選択します。
4. 限定商品や公開商品については、[\[お問い合わせ\]](#) ボタンを使用して AWS Marketplace Seller Operations チームに連絡するよう求められます。フォームを使用して、製品リストから削除するディメンションの詳細を入力します。

Note

オペレーションでは、常に料金ディメンションを制限できるとは限りません。

購入者が製品にアクセスする方法を決定する

顧客が製品にアクセスする方法として、以下のいずれかのオプションを選択できます。

- [the section called “SaaS URL フルフィルメントオプションを更新する”](#) — 顧客は、AWS Marketplaceで製品をサブスクライブした後にリダイレクトされるサイトの URL を使用します。
- [the section called “クイック起動を設定する”](#) — 顧客は、簡略化されたプロセスを利用して製品を設定して起動できます。この設定は、限定的な可視性または公開された可視性の、既存の製品に対して行うことができます。

SaaS URL フルフィルメントオプションを更新する

SaaS 製品のフルフィルメントに使用される URL を更新するには、[フルフィルメントオプションを更新] を使用します。

1. を開き[AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [SaaS 製品](#) ページから、SaaS 製品タブで、変更する製品を選択します。
3. リクエストの変更ドロップダウンリストから、フルフィルメントオプションの更新を選択し、デフォルトのフルフィルメント URL の編集を選択します。
4. [フルフィルメント URL] フィールドに、SaaS 製品のフルフィルメントオプションの新しい URL を入力します。
5. リクエストをレビューに送信するには、[送信] を選択します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] または [失敗] に更新されます。

クイック起動を設定する

にリストされている SaaS 製品 AWS Marketplace では、多くの場合、サブスクライブしている購入者のアカウント (IAM ロールなど) に AWS リソースをデプロイする必要があります。クイック起動を使用すると、AWS CloudFormation テンプレートを使用したガイド付きのstep-by-stepの手順とリソースデプロイを購入者に提供できます。購入者は CloudFormation テンプレートを使用して製品を設定、起動します。クイック起動設定プロセスの詳細については、[Enable SaaS Quick Launch](#) ラボを参照してください。

顧客が SaaS 製品を起動するために使用できるクイック起動エクスペリエンスを設定するには、[フルフィルメントオプション] タブを使用します。

1. [\[SaaS 製品\]](#) ページの [\[SaaS 製品\]](#) タブで、変更する製品を選択します。

 Note

クイック起動エクスペリエンスを設定するには、製品が、限定的な可視性または公開された可視性である必要があります。

2. 商品詳細ページで、[\[フルフィルメントオプション\]](#) タブを選択します。
3. [\[クイック起動\]](#) で、[\[有効化と設定\]](#) ボタンを選択します。
4. [\[アカウントログイン情報\]](#) で、購入者がログインしたりアカウントを作成したりできるサイトの URL を入力します。この URL から、購入者エクスペリエンスの新しいタブが開きます。その後、購入者はログインまたはアカウントを作成し、AWS Marketplace に戻ってテンプレートを起動します。
5. AWS CloudFormation テンプレートを作成します。

 Tip

AWS Well-Architected フレームワークに従って、購入者の AWS アカウントにリソースをデプロイする CloudFormation テンプレートを作成します AWS。詳細とリソースについては、[AWS Well-Architected](#) フレームワークをお読み [ください](#)。

[\[AWS CloudFormation テンプレート\]](#) で、[\[AWS CloudFormation テンプレートを追加\]](#) ボタンを選択し、次の情報を入力します。

- タイトル – CloudFormation デプロイの名前を指定します。
- 説明 – テンプレートの説明を入力します。
- スタック名 – スタックの名前を指定します。この名前は CloudFormation の購入者のスタック名です。
- CloudFormation テンプレート URL – Amazon Simple Storage Service がこのテンプレートを確認します。

 Note

顧客の起動プロセスを簡略化するために、設定プロセスに関連するテンプレートの数を最小限に抑えることをお勧めします。理想的には、製品の使用に必要なリソースをデプロイするテンプレートを 1 つ用意するとよいでしょう。CloudFormation テンプ

レートに関する質問については、AWS Marketplace ビジネス開発パートナーまたは [AWS Marketplace Seller Operations](#) チームにお問い合わせください。

- 必要な IAM アクセス許可 – CloudFormation テンプレートのデプロイに必要なアクセス許可を指定します。購入者との間で、[AWS Secrets Manager](#) にシークレットとして保存されているデプロイパラメータを共有する場合は、ポリシーに以下のアクションを含める必要があります。
 - `secretsManager:ListSecrets`
 - `secretsManager:DescribeSecret`
 - `secretsManager:ReplicateSecretToRegions`
 - `secretsManager:GetSecretValue`

 Note

製品に販売者が提供する CloudFormation デプロイパラメータ (API キーや[外部 ID](#) など) が必要な場合は、`PutDeploymentParameter` オペレーションを使用してパラメータを顧客と共有します。詳細については、「AWS Marketplace デプロイサービス API リファレンス」の「[PutDeploymentParameter](#)」を参照してください。

6. (オプション) [手動設定手順] では、製品を手動で設定する購入者向けの説明を入力します。製品のオンボーディングガイドとドキュメントへのリンクを含めることを検討してください。
7. [起動の詳細] で、CloudFormation スタックがデプロイされた後に購入者が製品にアクセスする URL を入力します。
8. (オプション) クイック起動の許可リストに登録されたアカウントについては、可視性が制限され AWS アカウント がクイック起動エクスペリエンスを表示できる のカンマ区切りリストを指定します。
9. [送信] ボタンを選択します。クイック起動エクスペリエンスは、限定的な可視性になります。つまり、自らのアカウントと、許可されているアカウントにのみが表示されます。限定的な可視性になっている場合は、製品をサブスクライブして [アカウントの設定] ボタンを選択した後に、[設定と起動] ページを使用して設定をテストできます。
 - 購入者の視点からページをプレビューするには、プレビューを選択します。両方のページが開いている間は、設定の変更がプレビューに反映されます。
10. 準備ができたら、クイック起動エクスペリエンスを AWS Marketplace カタログに公開できます。製品詳細ページの [フルフィルメントオプション] タブにある [クイック起動の可視性を更新] ボタンを使用します。

可視性をパブリックに変更すると、AWS Marketplace Seller Operations チームが設定を確認し、購入者テストを実施して、エクスペリエンスを公開します。

 Note

クイック起動エクスペリエンスを有効にする際にサポートが必要な場合は、[AWS Marketplace 販売者オペレーション](#) チームにお問い合わせください。

国別の可用性を更新する

製品を提供できる国を定義することができます。

1. を開き[AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) タブから、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンから [\[パブリックオファーを更新\]](#) を選択し、[\[国別の可用性を更新\]](#) を選択します。
4. 以下のオプションのいずれかを選択してください：
 - すべての国 - サポートされているすべての国で利用できます。
 - 除外対象のすべての国 - 選択した国を除くすべてのサポート対象国で利用できます。
 - カスタムリスト - 製品が販売されている特定の国のリスト。
5. [\[送信\]](#) を選択して、リクエストを送信します。
6. [\[リクエスト\]](#) タブで、リクエストステータスが [\[レビュー中\]](#) と表示されていることを確認します。リクエストが完了すると、ステータスが [\[成功\]](#) に変わります。

製品の返金ポリシーを更新する

[\[返金ポリシーを更新\]](#) を使用して、製品の返金ポリシーを更新できます。

1. を開き[AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) ページの [\[SaaS 製品\]](#) タブで、変更する製品を選択します。
3. [\[変更をリクエスト\]](#) ドロップダウンリストから [\[パブリックオファーを更新\]](#) を選択し、[\[返金ポリシーを更新\]](#) を選択します。

4. 現在の返金ポリシーの詳細がテキストボックスに表示されます。必要に応じて、これらの詳細を確認して変更します。リクエストを送信すると、現在の返金ポリシーが上書きされます。
5. リクエストをレビューに送信するには、[送信] を選択します。
6. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] または [失敗] に更新されます。

エンドユーザー使用許諾契約書 (EULA) を更新する

製品をサブスクライブする新規ユーザー向けの EULA を更新できます。

1. を開き [AWS Marketplace 管理ポータル](#)、販売者アカウントにサインインします。
2. [\[SaaS 製品\]](#) タブから、変更する製品を選択します。
3. [変更をリクエスト] ドロップダウンから [パブリックオファーを更新] を選択し、[EULA を更新] を選択します。
4. [AWS Marketplace \(SCMP\) の標準契約](#) を選択するか、カスタム EULA を送信できます。カスタム EULA の場合は、契約の Amazon Simple Storage Service (Amazon S3) URL を指定する必要があります。Amazon S3 バケットへのパブリックアクセスが許可されている必要があります。
5. [送信] を選択して、リクエストを送信します。
6. [リクエスト] タブで、リクエストステータスが [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] に変わります。

SaaS サブスクリプション製品を と統合する AWS Marketplace

製品を と統合することは、 の 1 つのステップ AWS Marketplace です [での SaaS 製品の作成 AWS Marketplace](#)。Software as a Service (SaaS) サブスクリプション製品を と統合するには AWS Marketplace、コードを記述し、いくつかの顧客シナリオに正常に対応できることを証明する必要があります。以下のセクションでは、SaaS サブスクリプションモデルを と統合する方法を示します AWS Marketplace。

Note

開始する前に、AWS Marketplaceで対象の Software as a Service (SaaS) 製品に適した料金モデルを選択していることを確認してください。詳細については、「[SaaS 製品の計画](#)」を参照してください。

トピック

- [シナリオ: サービスが新規顧客を検証する](#)
- [シナリオ: 使用量を計測する](#)
- [シナリオ: ユーザーサブスクリプションの変更を監視する](#)
- [シナリオ: 顧客のサブスクリプションを確認する](#)
- [SaaS サブスクリプション製品統合のテスト](#)

シナリオ: サービスが新規顧客を検証する

顧客が製品をサブスクライブすると、その顧客は登録 URL (一時的な `x-amzn-marketplace-token` トークンを含む HTTP POST リクエスト) にリダイレクトされます。このリクエストには、次の方法で応答します。

1. AWS Marketplace Metering Service で [ResolveCustomer](#) API オペレーションを呼び出して、`CustomerIdentifier`、`CustomerAWSAccountId`、`ProductCode` のトークンを交換します。
2. `CustomerIdentifier`、`CustomerAWSAccountId`、`ProductCode` は、今後の呼び出しに備えてシステム内に残しておきます。顧客が有効なサブスクリプションを持っているかどうかと、顧客について必要な情報をすべて保存する必要があります。
3. リクエストへの対応として、ユーザーが初めて使用するときの操作を (サービスに応じて) 表示する必要があります。

シナリオ: 使用量を計測する

顧客がサービスを利用し始めたら、計測レコードを 1 時間ごとに送信する必要があります。計測の詳細については、[SaaS サブスクリプションでの使用量の計測の設定](#) を参照してください。

AWS CloudTrail を使用してアクティビティをモニタリングし、請求情報が に送信されていることを確認することをお勧めします AWS。計測レコードを送信する場合は、次の点に注意してください。

- 計測リクエストは、毎時 0 分に重複排除されます。
- 送信したレコードは 1 時間ごとに累積されます。
- 過去 1 時間以内にレコードがない場合でも、ベストプラクティスとして、使用量 0 の計測レコードを 1 時間ごとに送信することを強くお勧めします。

シナリオ: ユーザーサブスクリプションの変更を監視する

Amazon Simple Queue Service (Amazon SQS) キューを設定し、製品の Amazon SNS トピックをサブスクライブします。SNS トピック情報は、製品の作成時に AWS Marketplace Seller Operations チームから受け取った E メールメッセージに含まれていました。詳細については、「[での SaaS 製品の作成 AWS Marketplace](#)」を参照してください。SNS トピックをサブスクライブすると、特定の顧客へのアクセス権の付与や取り消しなど、顧客のサブスクリプションの変更に関する通知が届きます。

Note

Amazon SNS トピックの Amazon リソースネーム (ARN) は `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>` のようになります。

応答する必要がある通知は以下のとおりです。

- `subscribe-success` - 顧客がサブスクライブされ、顧客 ID に照合して正常に計測できる。
- `unsubscribe-pending` - 顧客がサブスクリプション解除中である。最新の計測レコードはすべて送信する必要があります。
- `unsubscribe-success` - 顧客が購読を解除した。顧客の計測レコードは受け付けられなくなります。顧客リソースを閉鎖する手順に従い、保存ポリシーを遵守してください。
- `subscribe-fail` - 顧客のサブスクリプションが失敗した。顧客 ID に照合して計測したり、顧客に代わってリソースを作成したりしないでください。

シナリオ: 顧客のサブスクリプションを確認する

顧客に代わってリソースを作成する前に、顧客が製品にアクセスできることを確認してください。Amazon SQS 経由で受け取った通知から顧客の最新のステータスを保存し、顧客にアクセス許可があるかどうかを確認できます。

SaaS サブスクリプション製品統合のテスト

SaaS サブスクリプション製品を と統合したら AWS Marketplace、詳細なテストを実施して、統合が成功していることを確認する必要があります。次の手順では、製品統合を確認するステップを示します。

Note

自分のアカウントを使って製品をサブスクライブし、正常に統合できたことを確認します。料金を一時的に下げて、それらのアカウントで高額な料金を発生させずに購入フローをテストすることができます。価格を一時的に引き下げたり、追加のテストアカウントに製品へのアクセスを許可したりする方法については、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。

製品が発売された後も、サービスは新規顧客向けに引き続きこれらのシナリオに対応する必要があります。

1. 許可したアカウントを使用して、製品をサブスクライブしてカスタマーエクスペリエンスをテストします。
2. 許可したアカウントでサブスクライブした後、アカウントが登録 URL にリダイレクトされることと、そのリダイレクトが一時的トークンを含む POST リクエストであることを確認します。将来の呼び出しに備えて、アプリケーションに顧客 ID が保持されていることを確認します。これは [シナリオ: サービスが新規顧客を検証する](#) の一部をテストします。
3. 上記の手順でテストアカウントを検証したら、そのアカウントをアプリケーションにオンボードします。例えば、テスト用の顧客でフォームに入力して、新しいユーザーを作成することができます。または、SaaS アプリケーションにアクセスするための次の手順を別途用意します。これは [シナリオ: サービスが新規顧客を検証する](#) の一部をテストします。
4. テスト顧客がオンボーディングされたら、の BatchMeterUsage API オペレーションを使用して、請求 AWS 目的で計測レコードを に送信するリクエストを行います AWS Marketplace Metering Service。これは [シナリオ: 使用量を計測する](#) をテストします。
5. サブスクリプションの変更をテストします。考えられるシナリオには、サブスクリプションの解除、サブスクリプションの成功、サブスクリプションの失敗などがあります。これは [シナリオ: ユーザーサブスクリプションの変更を監視する](#) をテストします。
6. サブスクリプションの成功を確認します。テストアカウントでサブスクリプションの成功のメッセージの Amazon SNS 通知を受け取ると、計測を開始できます。Amazon SNS 通知を受信する AWS Marketplace Metering Service 前に に送信されたレコードは計測されません。これは [シナリオ: 顧客のサブスクリプションを確認する](#) をテストします。

Note

請求に関する問題を防ぐため、プログラムでこの通知を待ってから顧客のためのリソースを起動することを強くお勧めします。

- すべての統合要件を満たし、ソリューションをテストしたら、AWS Marketplace Seller Operations チームに通知します。BatchMeterUsage API オペレーションで計測レコードが正常に送信されたことを確認して、ソリューションに関する一連の最終テストを実行します。

統合とテストが完了したら、最終レビューを実行し、製品を一般公開できます AWS Marketplace。詳細については、「[での SaaS 製品の作成 AWS Marketplace](#)」を参照してください。

SaaS 契約製品と の統合 AWS Marketplace

Software as a Service (SaaS) 契約製品を と統合することは、 の 1 つのステップ AWS Marketplace です [での SaaS 製品の作成 AWS Marketplace](#)。Software as a Service (SaaS) 契約製品を と統合するには AWS Marketplace、コードを記述し、いくつかの顧客シナリオに正常に対応できることを証明する必要があります。以下のセクションでは、これらのシナリオとその対応方法について説明し、統合テストの概要を説明します。

Note

開始する前に、AWS Marketplaceで対象の Software as a Service (SaaS) 製品に適した料金モデルを選択していることを確認してください。詳細については、「[SaaS 製品の計画](#)」を参照してください。

トピック

- [シナリオ: サービスが新規顧客を検証する](#)
- [シナリオ: サービスが顧客のリクエストを処理する](#)
- [シナリオ: ユーザーサブスクリプションの変更を監視する](#)
- [SaaS 契約製品統合のテスト](#)

シナリオ: サービスが新規顧客を検証する

顧客が製品をサブスクライブすると、登録 URL にリダイレクトされます。これは一時的な `x-amzn-marketplace-token` トークンを含む HTTP POST リクエストです。このリクエストには、次の方法で応答します。

1. AWS Marketplace Metering Service で [ResolveCustomer](#) API オペレーションを呼び出して、`CustomerIdentifier`、`CustomerAWSAccountId`、`ProductCode` のトークンを交換します。
2. AWS Marketplace Entitlement Service で [GetEntitlements](#) API オペレーションを呼び出して、顧客がアクセスできるサブスクリプションと数量 (該当する場合) を確認します。
3. `CustomerIdentifier`、`CustomerAWSAccountId`、`ProductCode` は、今後の呼び出しに備えてシステム内に残しておきます。顧客が有効なサブスクリプションを持っているかどうかと、顧客について必要な情報をすべて保存しておきます。
4. リクエストへの対応として、ユーザーが初めて使用するときの操作を (サービスに応じて) 表示する必要があります。

シナリオ: サービスが顧客のリクエストを処理する

顧客がサービスにリクエストをしたら、次のシナリオに適切なアクションまたはメッセージで対応する必要があります。

- 顧客のシステムには顧客 ID がありません。つまり、まだサブスクライブしていないということです。サブスクライブ方法をユーザーに伝える必要があります。
- 顧客 ID があり、`GetEntitlements` API オペレーションによって適切なエンタイトルメントが返されます。このシナリオでは、リクエストに応答する必要があります。
- 顧客 ID はあるが、`GetEntitlements` API オペレーションから返されるエンタイトルメントがないか、リクエストを満たすのに十分な数量がありません。このシナリオでは、アクセスの処理方法とユーザーエクスペリエンスの管理方法を決定する必要があります。

シナリオ: ユーザーサブスクリプションの変更を監視する

Amazon Simple Queue Service (Amazon SQS) キューを設定し、製品の Amazon SNS トピックをサブスクライブします。SNS トピック情報は、製品を作成したときに AWS Marketplace オペレーションチームから受信した E メールに記載されていたものです。詳細については、「[での SaaS 製品の](#)

[作成 AWS Marketplace](#)」を参照してください。SNS トピックをサブスクライブすると、特定の顧客へのアクセス権の付与や取り消しなど、顧客エンタイトルメントの変更に関する通知が届きます。

Note

SNS トピックの Amazon リソースネーム (ARN) は `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>` のようになります。

返答が必要な通知は、以下のものだけです。

- entitlement-updated - 顧客エンタイトルメントが変更されたため、GetEntitlements API オペレーションを呼び出して新しいステータスを確認する必要があります。カスタマーストアを更新し、該当する場合は (例えば、顧客の契約が失効している場合)、カスタマーリソース閉鎖のプラクティスに従い、リテンションポリシーを遵守します。

Note

詳細については、「[を使用した使用権限の確認 AWS Marketplace Entitlement Service](#)」を参照してください。

SaaS 契約製品統合のテスト

SaaS 契約製品を と統合したら AWS Marketplace、詳細なテストを実施して、統合が成功していることを確認する必要があります。次の手順では、製品統合を確認するステップを示します。

Note

自分のアカウントを使って製品をサブスクライブし、正常に統合できたことを確認します。料金を一時的に下げて、それらのアカウントで高額な料金を発生させずに購入フローをテストすることができます。価格を一時的に引き下げたり、追加のテストアカウントに製品へのアクセスを許可したりする方法については、[お問い合わせください](#)。
製品が発売された後も、サービスは新規顧客向けに引き続きこれらのシナリオに対応する必要があります。

1. 許可したアカウントを使用して、製品を契約してカスタマーエクスペリエンスをテストします。

2. アカウントで契約した後、アカウントが登録 URL にリダイレクトされることと、そのリダイレクトが一時的トークンを含む POST リクエストであることを確認します。アプリケーションが今後の通話に備えて顧客 ID を保持し、顧客が持つエンタイトルメントを正しく処理するようにしてください。これは [シナリオ: サービスが新規顧客を検証する](#) の一部をテストします。
3. 上記の手順でテストアカウントを検証したら、そのアカウントをアプリケーションにオンボードします。例えば、テスト用の顧客でフォームに入力して、新しいユーザーを作成することができます。または、SaaS アプリケーションにアクセスするための次の手順を別途用意します。これは [シナリオ: サービスが新規顧客を検証する](#) の一部をテストします。
4. GetEntitlements API オペレーションからエンタイトルメントが返されない場合は、オンボーディング中または検証の進行中に、エンタイトルメントのないユーザーのアクセスとエクスペリエンスを管理する方法を決定します。これは [シナリオ: サービスが顧客のリクエストを処理する](#) をテストします。
5. サブスクリプションの変更をテストします。サブスクリプション解除、サブスクリプション成功、サブスクリプション失敗のシナリオをアプリケーションが正しく処理することを確認します。これは [シナリオ: ユーザーサブスクリプションの変更を監視する](#) をテストします。
6. すべての統合要件を満たし、ソリューションをテストしたら、AWS Marketplace 運用チームに通知します。次に、GetEntitlements API オペレーションが正常に呼び出され、新規顧客が十分にオンボーディングされたことを確認して、ソリューションをテストします。

統合とテストが完了したら、最終レビューを実行し、製品を一般公開できます AWS Marketplace。詳細については、「[での SaaS 製品の作成 AWS Marketplace](#)」を参照してください。返金リクエストフォームに記入して、テストサブスクリプションをキャンセルすることもできます。サブスクリプションのキャンセルの詳細については、「[the section called “製品の返金のリクエスト”](#)」を参照してください。

SaaS 契約をpay-as-you-go製品と統合する AWS Marketplace

製品を と統合 AWS Marketplace することは、 の 1 つのステップです [での SaaS 製品の作成 AWS Marketplace](#)。Software as a Service (SaaS) 契約製品を と統合するには AWS Marketplace、コードを記述し、いくつかの顧客シナリオに正常に対応できることを証明する必要があります。以下のセクションでは、これらのシナリオとその対応方法について説明し、統合テストの概要を説明します。

Note

開始する前に、AWS Marketplaceで対象の Software as a Service (SaaS) 製品に適した料金モデルを選択していることを確認してください。詳細については、「[SaaS 製品の計画](#)」を参照してください。

トピック

- [シナリオ: サービスが新規顧客を検証する](#)
- [シナリオ: サービスが顧客のリクエストを処理する](#)
- [シナリオ: 使用量を計測する](#)
- [シナリオ: ユーザーエンタイトルメントの変更を監視する](#)
- [pay-as-you-goの統合による SaaS 契約のテスト](#)

シナリオ: サービスが新規顧客を検証する

顧客が製品をサブスクライブすると、登録 URL にリダイレクトされます。これは一時的な x-amzn-marketplace-token トークンを含む HTTP POST リクエストです。このリクエストには、次の方法で応答します。

1. AWS Marketplace Metering Serviceで [ResolveCustomer](#) API オペレーションを呼び出して、CustomerIdentifier、CustomerAWSAccountId、ProductCode のトークンを交換します。
2. AWS Marketplace Entitlement Serviceで [GetEntitlements](#) アクションを呼び出して、顧客がアクセスできるサブスクリプションと数量 (該当する場合) を確認します。
3. CustomerIdentifier、CustomerAWSAccountId、ProductCode は、今後の呼び出しに備えてシステム内に残しておきます。顧客が有効なサブスクリプションを持っているかどうかと、顧客について必要な情報をすべて保存しておきます。
4. リクエストへの対応として、ユーザーが初めて使用するときの操作を (サービスに応じて) 表示する必要があります。

シナリオ: サービスが顧客のリクエストを処理する

顧客がサービスにリクエストをしたら、次のシナリオに適切なアクションまたはメッセージで対応する必要があります。

- 顧客のシステムには顧客 ID がありません。つまり、まだサブスクライブしていないということです。サブスクライブ方法を説明するメッセージを送る必要があります。
- 顧客 ID があり、GetEntitlements API オペレーションによって適切なエンタイトルメントが返されます。このシナリオでは、リクエストに応答する必要があります。
- 顧客 ID はあるが、GetEntitlements API オペレーションから返されるエンタイトルメントがないか、リクエストを満たすのに十分な数量がありません。このシナリオでは、アクセスの処理方法とユーザーエクスペリエンスの管理方法を決定する必要があります。

シナリオ: 使用量を計測する

顧客がサービスを利用し始めたら、計測レコードを 1 時間ごとに送信する必要があります。計測の詳細については、[SaaS サブスクリプションでの使用量の計測の設定](#) を参照してください。

AWS CloudTrail を使用してアクティビティをモニタリングし、請求情報が に送信されていることを確認することをお勧めします AWS。計測レコードを送信する場合は、次の点に注意してください。

- 計測リクエストは、毎時 0 分に重複排除されます。
- 送信したレコードは 1 時間ごとに累積されます。
- 過去 1 時間以内にレコードがない場合でも、ベストプラクティスとして、使用量 0 の計測レコードを 1 時間ごとに送信することを強くお勧めします。

シナリオ: ユーザーエンタイトルメントの変更を監視する

Amazon Simple Queue Service (Amazon SQS) キューを設定し、製品の Amazon SNS トピックをサブスクライブします。SNS トピックは 2 つあり、1 つはエンタイトルメントの変更に、もう 1 つはサブスクリプションの変更に使われます。トピック情報は、製品の作成時に AWS Marketplace Seller Operations チームから受け取った E メールメッセージに含まれていました。詳細については、「[での SaaS 製品の作成 AWS Marketplace](#)」を参照してください。SNS トピックをサブスクライブすると、特定の顧客へのアクセス権の付与や取り消しなど、顧客サブスクリプションの変更に通知が届きます。

Note

サブスクリプション変更の SNS トピックの Amazon リソースネーム (ARN) は、arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code> のようになります。エンタイトルメント変更の SNS ト

ピックの ARN は `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>` のようになります。

返答する必要がある通知は次のとおりです。

- entitlement-updated (エンタイトルメント SNS トピック内) - 顧客エンタイトルメントが変更されたため、GetEntitlements API オペレーションを呼び出して新しいステータスを確認する必要があります。カスタマーストアを更新し、該当する場合は (例えば、顧客の契約が失効している場合)、カスタマーリソース閉鎖のプラクティスに従い、リテンションポリシーを遵守します。
- subscribe-success (サブスクリプション SNS トピック内) - 顧客はサブスクライブしており、顧客 ID と正常に照合できます。
- unsubscribe-pending (サブスクリプション SNS トピック内) - 顧客はサブスクリプション解除中です。最新の計測レコードはすべて送信する必要があります。
- unsubscribe-success (サブスクリプション SNS トピック内) - 顧客がサブスクリプションを解除しました。顧客の計測レコードは受け付けられなくなります。顧客リソースを閉鎖する手順に従い、保存ポリシーを遵守してください。
- subscribe-fail (サブスクリプション SNS トピック内) - 顧客のサブスクリプションが失敗しました。顧客の顧客 ID と照合して計測したり、顧客に代わってリソースを有効にしたりしないでください。

Note

詳細については、「[を使用した使用権限の確認 AWS Marketplace Entitlement Service](#)」を参照してください。

pay-as-you-goの統合による SaaS 契約のテスト

契約をpay-as-you-go制製品と統合したら AWS Marketplace、統合が成功するように詳細なテストを実施する必要があります。次の手順では、製品統合を確認するステップを示します。

Note

自分のアカウントを使って製品をサブスクライブし、正常に統合できたことを確認します。料金を一時的に下げて、それらのアカウントで高額な料金を発生させずに購入フローをテス

トすることができます。価格を一時的に引き下げたり、追加のテストアカウントに製品へのアクセスを許可したりする方法については、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。

製品が発売された後も、サービスは新規顧客向けに引き続きこれらのシナリオに対応する必要があります。

1. 許可したアカウントを使用して、製品を契約してカスタマーエクスペリエンスをテストします。
2. アカウントで契約した後、アカウントが登録 URL にリダイレクトされることと、そのリダイレクトが一時的トークンを含む POST リクエストであることを確認します。アプリケーションが今後の通話に備えて顧客 ID を保持し、顧客が持つエンタイトルメントを正しく処理するようにしてください。これは [シナリオ: サービスが新規顧客を検証する](#) の一部をテストします。
3. 上記の手順でテストアカウントを検証したら、そのアカウントをアプリケーションにオンボードします。例えば、テスト用の顧客でフォームに入力して、新しいユーザーを作成することができます。または、SaaS アプリケーションにアクセスするための次の手順を別途用意します。これは [シナリオ: サービスが新規顧客を検証する](#) の一部をテストします。
4. GetEntitlements API オペレーションからエンタイトルメントが返されない場合は、オンボーディング中または検証の進行中に、エンタイトルメントのないユーザーのアクセスとエクスペリエンスを管理する方法を決定します。これは [シナリオ: サービスが顧客のリクエストを処理する](#) をテストします。
5. テスト顧客がオンボーディングされたら、の BatchMeterUsage API オペレーションを使用して、AWS 請求目的で計測レコードを送信するリクエストを行います AWS Marketplace Metering Service。これは [シナリオ: 使用量を計測する](#) をテストします。
6. サブスクリプションの変更をテストします。サブスクリプション解除、サブスクリプション成功、サブスクリプション失敗のシナリオをアプリケーションが正しく処理することを確認します。これは [シナリオ: ユーザーエンタイトルメントの変更を監視する](#) をテストします。
7. すべての統合要件を完了し、ソリューションをテストしたら、AWS Marketplace Seller Operations チームに通知します。次に、GetEntitlements API オペレーションが正常に呼び出され、新規顧客が十分にオンボーディングされたことを確認して、ソリューションをテストします。また、BatchMeterUsage API オペレーションを使用して計測レコードが正常に送信されていることも確認します。

統合とテストが完了したら、最終レビューを実行し、製品を一般公開できます AWS Marketplace。詳細については、「[での SaaS 製品の作成 AWS Marketplace](#)」を参照してください。

サーバーレス SaaS 統合のデプロイ

SaaS ソリューションを作成したら、それを の対応するリストと統合します AWS Marketplace。そのためには、AWS CloudFormation テンプレートを使用してサーバーレス統合を作成できます。統合により、顧客の登録、アクセスの許可と取り消し、顧客エンタイトルメントの更新、計測された使用状況のレポートを行うことができます。

サーバーレス SaaS 統合の作成の詳細については、「[Integrate your Saas with the Serverless SaaS Integration reference](#)」を参照してください。これは、AWS CloudFormation テンプレートをデプロイし、関連するタスクを完了する方法を説明するラボです。

SaaS 製品の計画

Software as a Service (SaaS) 製品を に追加する前に AWS Marketplace、まずいくつかの計画を行う必要があります。このステップは、製品が成功を収めるために不可欠です。計画が不十分だと、請求に問題が生じたり、AWS Marketplaceで製品の再作成が必要になったりする場合があります。以下のセクションでは、SaaS 製品の計画方法を示します。

Important

製品の設定のほとんどは、設定後に変更することはできません。で製品を作成した後に変更する必要がある場合は AWS Marketplace、おそらく正しい設定で新しい製品を作成する必要があります。

トピック

- [料金の計画](#)
- [請求の統合の計画](#)
- [Amazon SNS インテグレーションを計画する](#)
- [顧客が製品にアクセスする方法の計画](#)

料金の計画

での SaaS 製品には 3 つの料金モデルがあります AWS Marketplace。製品に適した料金モデルを選択することは、製品を計画するにあたり最も重要な決定です。間違った価格モデルを選択すると、数週間の遅れが生じる可能性があります。料金モデルによって、顧客の支払いオプションと、作成、テ

スト、デプロイする必要がある請求統合コードが決まります。さまざまなタイプの価格モデルについて詳しくは、「[SaaS 製品の価格設定](#)」を参照してください。

Note

すべての SaaS 価格モデルは無料トライアルをサポートしています。詳細については、「[SaaS 無料トライアル](#)」を参照してください。

請求の統合の計画

で SaaS 製品を使用する利点の 1 つは、請求の統合 AWS Marketplace です。この利点を活用するには、選択した料金モデルに応じて AWS Marketplace Metering Service または AWS Marketplace Entitlement Service と統合する必要があります。これらの 2 つのサービスにより、請求と使用量のレポートが正確に行われるようになります。

統合を計画したら、有効にする前に製品との統合をテストする必要があります。統合とテストの詳細については、「[AWS Marketplace Metering Service API および使用権限管理サービス API へのアクセス](#)」を参照してください。

Amazon SNS インテグレーションを計画する

SaaS 製品についてサブスクライブできる Amazon Simple Notification Service (Amazon SNS) トピックは 2 つあります。詳細については、「[SaaS 通知](#)」を参照してください。これらのメッセージは、AWS または顧客によって開始されたサブスクリプションや契約の変更をプログラムで処理するのに役立ちます。Amazon SNS 通知は、顧客が製品登録ウェブサイトで新しいアカウントを登録できるようにするプログラムによるトリガーとして使用できます。また、サブスクリプションの有効期限が切れた顧客による製品へのアクセスを拒否することもできます。通知の処理をどのようにプログラムしているかに応じて、顧客が通知を受け取る方法を選択できます。

顧客が製品にアクセスする方法の計画

このセクションでは、購入者が製品にアクセスできるようにする方法について説明します。

SaaS 製品の登録ウェブサイトの計画

SaaS 製品を購入する顧客は、その製品にアクセスする必要があります。顧客が製品にアクセスする方法を計画および実装することが必要です。SaaS 製品は、以下のアクセスオプションをサポートしています。

- クイック起動
- AWS PrivateLink
- 販売者の製品ウェブサイト

登録ウェブサイトを使用して AWS Marketplace 顧客を検証するには、[SaaS 顧客オンボーディング](#)」を参照してください。

顧客が製品にアクセスするためのクイック起動の使用

クイック起動デプロイオプションを使用すると、購入者が製品の設定、デプロイ、発売に必要な時間とリソースを削減できます。クイック起動を使用すると、購入者がプロセス中にアクセスしなければならないサイトの数が減ります。詳細については、「[クイック起動の設定](#)」を参照してください。

顧客 AWS PrivateLink による SaaS 製品へのアクセスに を使用する

[を使用して Amazon VPC 経由で製品を配信する AWS PrivateLink](#) を使用して Amazon Virtual Private Cloud (Amazon VPC) エンドポイントサービスとして、サービスを設定できます。顧客は、VPC エンドポイントを作成して、AWS クラウド クラウド仮想ネットワークを通じて販売者のソフトウェアにアクセスできます。または、販売者が所有および管理するウェブサイトを通じてソフトウェア製品へのアクセスを提供することもできます。その場合、顧客がインターネット間の接続を作成します。

独自の Web サイトの使用

販売者の SaaS 製品は、販売者の環境でホストされており、販売者が管理および保守を行うウェブサイトなどのパブリックエンドポイントを通じてインターネット経由でアクセスする必要があります。一般的に、販売者には、顧客が製品の登録、製品の使用のためのサインイン、製品のサポートへのアクセスのために使用するウェブサイトがあります。

の SaaS 製品ガイドライン AWS Marketplace

AWS Marketplace では、お客様の安全、安全、信頼できるプラットフォームを推進 AWS Marketplace するために、のすべての Software as a Service (SaaS) 製品およびサービスについて以下のガイドラインを維持しています。以下のセクションでは、での SaaS 製品のガイドラインについて説明します AWS Marketplace。

すべての製品とそれに関連するメタデータは、送信時にレビューされ、現在の AWS Marketplace ガイドラインを満たしているか上回っていることを確認します。これらのガイドラインは、進化するセキュリティ要件を満たすために、見直され、調整されます。さらに、は AWS Marketplace 製品を

継続的にレビューして、これらのガイドラインに対する変更を満たしていることを確認します。製品がコンプライアンスに準拠していない場合、製品の更新が必要となる場合があります。場合によっては、問題が解決するまで新規サブスクライバーが製品を一時的に利用できなくなることがあります。

トピック

- [製品設定ガイドライン](#)
- [顧客情報の要件](#)
- [製品使用ガイドライン](#)
- [アーキテクチャガイドライン](#)

製品設定ガイドライン

すべての SaaS 製品は、次の製品設定ガイドラインに従う必要があります。

- 料金ディメンションはプライベートオファーのみに制限することはできません。購入者は、パブリック製品の料金ディメンションのいずれかをサブスクライブできる必要があります。
- 少なくとも 1 つの価格ディメンションの料金が 0.00 USD を超える必要があります。
- すべての価格ディメンションは、実際のソフトウェアに関連するものでなければならず、ソフトウェアに関係のない他の製品またはサービスを含むことはできません。
- AWS GovCloud (US) リージョンでのみ提供される SaaS 製品には、製品タイトルの GovCloud どこかを含める必要があります。

顧客情報の要件

すべての SaaS 製品は、次の顧客情報の要件に従う必要があります。

- SaaS 製品は、AWS Marketplace に記載されているディメンションに基づいて完全に課金される必要があります。
- クレジットカードや銀行口座情報など、SaaS 製品に対する顧客支払い情報はいかなる時点でも収集できません。
- SaaS 登録ページには、購入者の E メールアドレスの入力フィールドが含まれている必要があります。次のような追加のフィールドを含めることができます。
 - 名前
 - ZIP コード
 - 電話番号

- 会社情報
- 製品セットアップの設定

複数の言語を使用する場合は、登録ページの英語ビューを提供する必要があります。

製品使用ガイドライン

すべての SaaS 製品は、次の製品使用ガイドラインに従う必要があります。

- で製品をサブスクライブすると AWS Marketplace、お客様は SaaS アプリケーション内にアカウントを作成し、ウェブコンソールにアクセスできるようになります。顧客がすぐにアプリケーションにアクセスできない場合は、アクセスするタイミングに関する具体的な指示を記載したメッセージを提供する必要があります。アカウントが作成された後に、アカウントが作成されたことを確認する通知を明確な次のステップとともに顧客に送信する必要があります。
- 顧客がすでに SaaS アプリケーションにアカウントを持っている場合、顧客はフルフィルメントランディングページからログインできる必要があります。
- 顧客は、関連する契約またはサブスクリプション使用の情報など、SaaS アプリケーション内でサブスクリプションのステータスを確認できる必要があります。
- 顧客は、アプリケーションの使用、トラブルシューティング、返金のリクエスト (該当する場合) などの問題について、簡単にサポートを受けることができる必要があります。サポートの問い合わせオプションは、フルフィルメントランディングページで指定する必要があります。
- 製品ソフトウェアとメタデータには、AWS Marketplaceでは利用できない他のクラウドプラットフォーム、追加の製品、アップセルサービス、または無料トライアルオファーにユーザーをリダイレクトする言葉を含めてはいけません。

SaaS 製品の無料トライアルの詳細については、「[での SaaS 無料トライアルオファーの作成 AWS Marketplace](#)」を参照してください。

- 製品が別の製品または別の ISV 製品のアドオンである場合、製品の説明には、それが他の製品の機能を拡張するものであり、これがないと製品の有用性が非常に限られることを明記する必要があります。例えば、この製品は <製品名> の機能を拡張するものであり、それがないと、この製品の有用性は非常に限られています。<製品名> は、このリストのすべての機能を利用するには、独自のライセンスが必要な場合がありますのでご注意ください。

アーキテクチャガイドライン

以下のトピックでは、SaaS 製品のアーキテクチャガイドラインを一覧表示して説明します。

トピック

- [ガイドライン](#)
- [アーキテクチャ図の作成](#)

ガイドライン

Note

次のガイドラインは、2025 年 5 月 1 日をもって有効になります。

- すべての SaaS アーキテクチャを公開できます。
- にデプロイされた製品は、検索結果とその製品詳細ページで特別な指定 AWS AWS Marketplace を受けます。AWS Marketplace が製品を「デプロイ済み」と見なすには AWS、製品が完全に実行される必要があります AWS。これには、アプリケーションプレーンとコントロールプレーンが含まれます。

アプリケーションプレーンは、販売者の AWS アカウント、購入者の AWS アカウント、またはその両方で実行できます。詳細については、[コントロールプレーンとアプリケーションプレーン](#)のホワイトペーパーを参照してください。

コンテンツ配信ネットワーク (CDNs)、ドメインネームシステム (DNSs)、企業 ID プロバイダー (IdPs) を除く、アプリケーションデータを送信、保存、または処理するために製品で使用するサードパーティーのサービスも、完全に実行する必要があります AWS。

Note

アプリケーションデータは、購入者に属しているか、購入者用に生成されるデータです。

セキュリティ、モニタリング、データレプリケーション、または移行のために製品で使用するエージェントまたはゲートウェイは AWS、オンプレミスを含む外部の購入者所有環境で実行できますが、ストレージと分析 AWS のためにのみ にデータを送信する必要があります。

レビューにはアーキテクチャ図を含める必要があります。図を公開することはできません。詳細については、次のセクション [アーキテクチャ図の作成](#) の「」を参照してください。

- 販売者は、完全には実行されない製品を公開できます AWS。

- 購入者のインフラストラクチャでリソースを必要とするアプリケーションは、次のガイドラインに従う必要があります。
- マネージドサービスではなく SaaS 製品と見なされるには、ホワイトペーパーの[SaaS アーキテクチャの基礎 AWS](#)」で定義されているコントロールプレーンが、管理するインフラストラクチャに存在する必要があります。詳細については、[SaaS とマネージドサービスプロバイダー](#)のホワイトペーパーを参照してください。
- 製品説明では、AWS Marketplace 取引とは別にインフラストラクチャ料金が発生した場合 AWS は、それらの料金を支払う必要があることを顧客に通知する必要があります。
- AWS セキュリティトークンサービス (AWS STS) や AWS Identity and Access Management (IAM) を使用するなど、安全な方法でリソースをプロビジョニングする必要があります。
- アプリケーションにアクセス許可を付与する使用手順またはデプロイテンプレートを作成するときは、[最小特権の原則](#)に従う必要があります。
- プロビジョニングされた AWS すべてのサービス、IAM ポリシーステートメント、IAM ロールまたはユーザーが顧客のアカウントでどのようにデプロイおよび使用されるかを説明する追加のドキュメントを提供する必要があります。
- 購入者が必要なリソースを AWS アカウントにデプロイできるようにする手順またはデプロイテンプレートを提供する必要があります。
- 購入者の AWS アカウントにリソースをデプロイするための AWS CloudFormation テンプレート (CFTs) を提供する場合は、[AWS Marketplace CFTs のポリシー](#)に準拠する必要があります。これらの CFTs、購入者の SaaS [クイック起動デプロイオプションを有効にするときに提供される方法に従って、SaaS](#) 出品の一部として公開する必要があります。SaaS クイック起動を使用すると、購入者は SaaS ソリューションを簡単に設定できます。
- Amazon マシンイメージ (AMIs) が購入者の AWS アカウントにデプロイされている場合は、[AWS Marketplace AMIs のポリシー](#)に準拠する必要があります。AMIs、AWS Marketplace 管理ポータル (販売者ポータル) の [AMI スキャナー](#)を通過する必要があります。[製品を公開することをリクエスト](#)する場合は、[オペレーションに連絡し AWS Marketplace](#)、スキャン結果の証明を提供する必要があります。
- コンテナイメージが購入者の AWS アカウントにデプロイされる場合、イメージは[AWS Marketplace コンテナのポリシー](#)に準拠している必要があります。コンテナイメージはの外部でホストできますが AWS、[Amazon Elastic Container Registry \(Amazon ECR\)](#) で[スキャン](#)し、重大な脆弱性がないことが必要です。[製品を公開するようリクエスト](#)する場合は、[オペレーションに連絡し AWS Marketplace](#)、コンテナがスキャンに合格した証拠を提供する必要があります。

- プロバイダーとして登録し、SaaS 発行リクエストを送信 AWS アカウント した から AWS Marketplace APIs を正常に呼び出します。SaaS 価格モデルによって、呼び出される API が決まります。
- SaaS 契約 – の [GetEntitlements](#) AWS Marketplace Entitlement Service。
- 消費を伴う SaaS 契約 – の [GetEntitlements](#) AWS Marketplace Entitlement Service と AWS Marketplace Metering Service の [BatchMeterUsage](#)。
- SaaS サブスクリプション - AWS Marketplace Metering Service での [BatchMeterUsage](#)。
- AWS GovCloud (US) リージョンでのみ提供される SaaS 製品は、他の AWS リージョン と AWS GovCloud (US) リージョン間のアーキテクチャの境界、製品のユースケース、製品に推奨されないワークロードについて説明する必要があります。

SaaS アーキテクチャの詳細については、[SaaS アーキテクチャの基礎](#) AWS 」ホワイトペーパーを参照してください。

アーキテクチャ図の作成

製品がデプロイされる特別な指定を受け取るには AWS、AWS Marketplace 管理ポータルで[製品のアーキテクチャの詳細を更新します](#)。にデプロイされるホスティングパターンを選択し AWS、AWS レビューするアーキテクチャ図をアップロードします。が にデプロイされていると見なす AWS Marketplace ホスティングパターンについては AWS、前のセクション[ガイドライン](#)の「」を参照してください。ホスティングパターンが変更された場合は、製品のアーキテクチャの詳細を更新する必要があります。

Note

- 図を公開することはできません。
- 完全には実行されない製品を公開できます AWS。

図を作成するときは、次の基準を使用します。

- [アプリケーションプレーンまたはコントロールプレーン](#)の一部としてコンポーネントをグループ化してラベル付けします。
- 製品のコアビジネスロジックの一部 AWS である 以外のコンポーネントについては、アプリケーションプレーンでグループ化します。

- コンポーネントは、低レベルの詳細 (コンピューティングインスタンスやネットワークサブネットなど) または高レベルサービス (データ分析プラットフォームなど) を表すことができます。
- コンポーネントは、使用される AWS サービスまたは非AWS サービスの名前を識別する必要はありません。
- コンポーネントを論理的に実行する場所に配置します。たとえば、販売者の AWS アカウント、購入者の AWS アカウント、販売者の非AWS 環境、またはその他の環境などです。
- データレプリケーションまたはワークロード移行製品の場合は、サポートされているすべてのソース環境とターゲット環境を含めます。

Note

SaaS 製品のアーキテクチャの詳細を更新するために使用するアーキテクチャ図は公開されておらず、購入者は公開されていません。

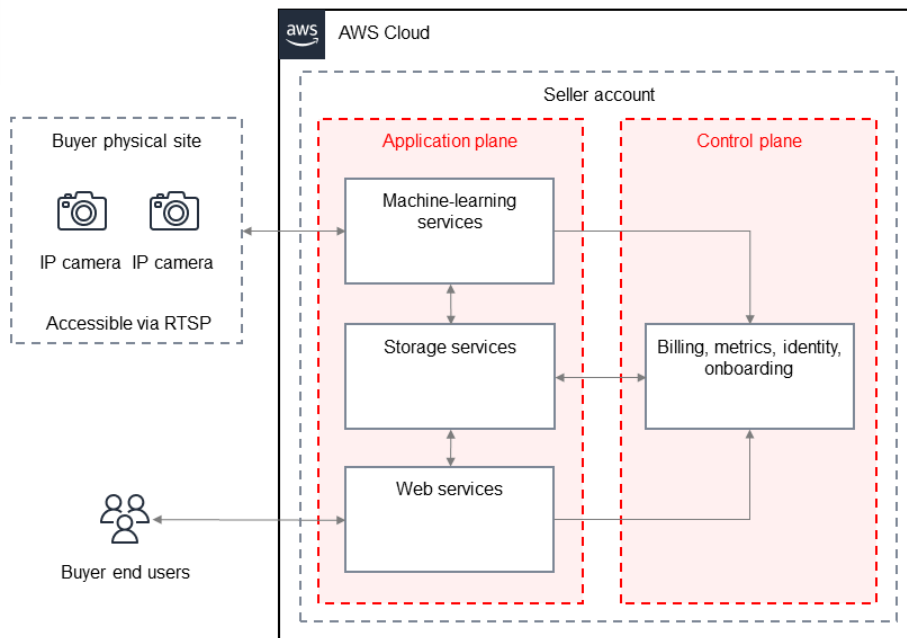
詳細レベル

主要なシステムコンポーネント、基本的なデータフロー、アプリケーションプレーンとコントロールプレーンサービスに焦点を当てた概要図を作成できます。または、各コンポーネントを分類し、特定の接続を表示し、さまざまな詳細レベルの技術仕様を含む、低レベルの詳細な図を作成できます。

次の図は、架空のビデオ分析 SaaS アプリケーションのアーキテクチャを示しています。それぞれに異なる詳細レベルが表示されます。どちらも許容されます。これらを、独自の図に含める詳細レベルの例として使用します。

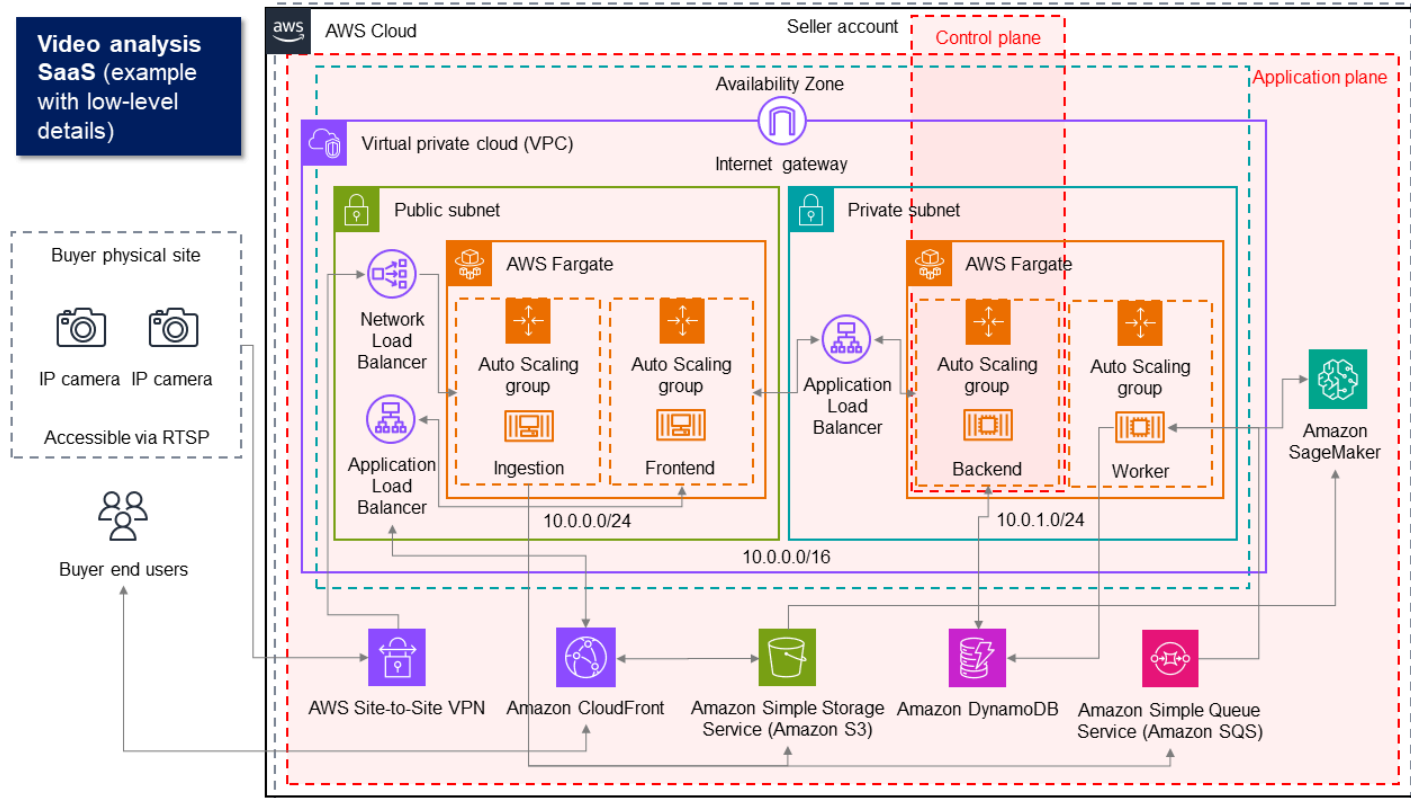
以下は、大まかな図の例です。

**Video analysis
SaaS (example
with high-level
services)**



以下は、低レベルの詳細な図の例です。

**Video analysis
SaaS (example
with low-level
details)**



詳細については、「[アーキテクチャダイアグラムとは](#)」を参照してください。図を作成したら、AWS Marketplace 管理ポータル (AMMP) でアーキテクチャの詳細を更新します。詳細については、「[アーキテクチャの詳細を更新する](#)」を参照してください。

での SaaS 製品の料金 AWS Marketplace

購入者が Software as a Service (SaaS) 製品を購入すると AWS Marketplace、AWS Marketplace は請求識別子を提供します。販売者はその請求識別子を使用して、AWS Marketplace Entitlement Service および AWS Marketplace Metering Service を呼び出します。その後、顧客は AWS 環境内または作成した Virtual Private Cloud (VPC) エンドポイント接続を介して製品にアクセスします。このトピックでは、の SaaS 料金モデルのリストを示します AWS Marketplace。

Note

すべての SaaS 価格モデルは無料トライアルをサポートしています。詳細については、「[SaaS 無料トライアル](#)」を参照してください。

SaaS 料金モデル

料金モデル	説明
SaaS サブスクリプション	SaaS 製品の時間単位の使用に基づいて購入者に請求する従量制料金モデル。詳細については、「 SaaS サブスクリプションの料金 」を参照してください。
SaaS 契約	購入者にソフトウェアの使用について前払いで請求するか、販売者が柔軟な支払いスケジュールを提示することができます。顧客は、契約を超えた追加使用量に対して支払うこともできます。詳細については、「 SaaS 契約の料金 」を参照してください。
従量制料金による SaaS 契約	購入者にソフトウェアの使用について前払いで請求するか、販売者が柔軟な支払いスケジュールを提示することができます。購入者には、契約価格に加えて、従量制の追加料金も請求され

料金モデル	説明
	ます。詳細については、「 SaaS 契約の料金 」を参照してください。

Note

出品を作成して制限付きに公開すると、料金モデルを変更することはできません。

SaaS 製品を利用可能にするには AWS Marketplace、[SaaS サブスクリプション料金モデル](#)と [SaaS 契約料金モデル](#)のどちらを提供するかを決定します。

SaaS サブスクリプションの料金

Software as a Service (SaaS) サブスクリプションの場合、は、お客様が当社に送信する計測レコードに基づいて顧客に AWS Marketplace 請求します。料金は必ず、顧客のアカウントにデプロイされたソフトウェアから 1 時間ごとに計測し、報告する必要があります。すべての使用量は毎月計算され、AMI ベースの AWS Marketplace サービスと同じメカニズムを使用して毎月請求されます。当社は、お客様からの計測レコードの受領に基づき、製品の使用料金を請求することができます。製品の計測レコードの送受信を正常に行う必要があります。

サブスクリプション価格で SaaS 製品を公開するには、まず次の手順を実行します。

1. で新しい SaaS 製品を作成し AWS Marketplace 管理ポータル、新しい SaaS サブスクリプションを選択します。
2. [全般] タブのフィールドに必要な情報を入力します。製品コードを書き留めます。
3. [料金] タブの [料金設定] で、製品の料金を最も正確に表すカテゴリを選択します。料金カテゴリは、AWS Marketplace ウェブサイトで顧客に表示されます。帯域幅 (GBps、MBps)、データ (GB、MB、TB)、ホスト (時間)、リクエスト、ユーザー (時間) のいずれかを選択できます。事前定義カテゴリのどれもニーズに適合しない場合は、汎用的な [単位] カテゴリを選択できます。

次に、料金ディメンションを定義します。各料金ディメンションは、単位あたりの価格を設定できる機能またはサービスを表します。ディメンションの例として、ユーザー数、スキャンされたホスト数、取り込まれたログの GB 数があります。最大で 24 のディメンションを定義できます。定義するディメンションごとに、以下の情報を追加する必要があります。

- デイメンション API 名 - 計測レコードを [AWS Marketplace Metering Service](#) に送信する際に使用する API 名。この名前は顧客がどのデイメンションを使用したかを表します。この名前は請求レポートに記載されます。そのレポートにアクセスできるのは販売者だけであるため、わかりやすい名前にする必要はありません。この名前を設定した後に変更することはできません。
- デイメンションの説明 - 製品のデイメンションを記述した顧客向けステートメント。説明は 70 文字以内でなければならず、ユーザーフレンドリである必要があります。説明の例としては、「1 時間あたりの管理者数」や「プロビジョニングされた Mbps あたりの帯域幅」などがあります。製品が発行された後に、この説明を変更することはできません。
- デイメンションレート - この製品の FCP 単位あたりのソフトウェア料金 (USD)。このフィールドでは、小数点以下 8 桁がサポートされています。

SaaS サブスクリプションの終了時

顧客は、AWS Management Consoleを通じて SaaS サブスクリプション製品のサブスクリプションを解除できます。SaaS サブスクリプション終了プロセスの主なポイントは次のとおりです。

1. その顧客の Amazon SNS トピックを通じて、SaaS 製品に unsubscribe-pending 通知が送信されます。
2. その顧客の残り使用量を計測するまでに 1 時間の猶予があります。
3. その時間が経過すると、販売者は unsubscribe-success 通知を受信します。この時点で、この顧客の計測レコードを送信することはできません。

サブスクリプションを解除した顧客の SaaS アプリケーションで機能をどのように無効化するかは、販売者に任されています。例えば、製品でその顧客の既存の仕事を完了するが、その顧客が新しい仕事を作成できないようにすることが考えられます。使用が無効になったことを示すメッセージを顧客に表示するとよいでしょう。お客様は を通じて製品を再サブスクライブできます AWS Marketplace。

SaaS サブスクリプションがキャンセルされたとき

SaaS サブスクリプションキャンセルプロセスの主なポイントは次のとおりです。

1. 顧客は、AWS Marketplace ウェブサイトの Your Marketplace Software ページで、SaaS サブスクリプション製品のサブスクリプションをキャンセルできます。

その顧客の Amazon SNS トピックを通じて、SaaS 製品に通知が送信されます。

2. その顧客の残り使用量を計測するまでに 1 時間の猶予があります。

3. 販売者は、キャンセルが進行中の製品から顧客に通知します。顧客が製品を介してキャンセルを指示している場合は、顧客を AWS Marketplace にリダイレクトします。今後料金が発生しないようにするには、お客様はキャンセルを確認する必要があります AWS Marketplace。

SaaS 契約の料金

Software as a Service (SaaS) 契約の場合、お客様は、対象ソフトウェアの購入をもって、契約が成立したものとみなします。契約のもとで、顧客は SaaS 製品を特定の数量、使用する権利を持ちます。AWS Marketplace はこれらの使用権限を SaaS アプリケーションに伝えます。これは、AWS Marketplace Entitlement Service を介して行います。SaaS 契約の料金モデルを使用するとき、アプリケーションは計測レコードを送信しません。その代わりに、AWS Marketplace Entitlement Service を呼び出して使用権限を検証します。使用カテゴリ、ディメンション、契約期間を定義します。

AWS Marketplace は、販売者と顧客の間の契約に基づいて、前払いまたは販売者が定義した支払いスケジュールで顧客に請求します。その時点で、それらのリソースの使用権限が付与されます。契約を超える追加の使用量に対しては、ソフトウェアから顧客に使用量レポートを送信する必要があります。AWS Marketplace は AWS Marketplace Metering Service を介して当社が受信した計測レコードに基づいて顧客に請求します。

契約価格で SaaS 製品を公開するには、まず次の手順を実行します。

1. AWS Marketplace 管理ポータル で新しい SaaS 製品を作成し、[新規 SaaS 契約] を選択します。
2. [全般] タブのフィールドに必要な情報を入力します。製品コードを書き留めます。
3. [価格設定] タブ:
 - a. [価格設定] では、顧客に提供する [契約期間] を選択します。契約期間ごとに異なる価格を入力できます。毎月、1 年、2 年、3 年のオプションから 1 つ以上を選択できます。プライベート オファーを作成する場合は、カスタム期間を月単位 (最大 60 か月) で選択できます。
 - b. [提供する契約タイプを選択] では、次のオプションから顧客が製品をどのように購入できるかを選択します。
 - 購入者は提供されているオプションを 1 つ以上選択可能 - 顧客は販売者が提供している各価格ディメンションの数量を選択できます。
 - 購入者は提供されている階層を 1 つ選択可能 - 顧客は、機能、サービス、および使用量のさまざまな組み合わせが含まれている選択肢から 1 つの階層を選択します。
 - c. 製品の料金を最もよく表す使用単位カテゴリを選択します。価格カテゴリは、AWS Marketplace ウェブサイトでの顧客に表示されます。帯域幅 (GBps、MBps)、データ (GB、MB、TB)、ホスト (時間)、リクエスト、ユーザー (時間) のいずれかを選択できます。事

前定義カテゴリのどれもニーズに適合しない場合は、汎用的な [単位] カテゴリを選択できます。

4. カテゴリを選択したら、料金ディメンションを定義します。各料金ディメンションは、単位あたりの料金を設定できる機能またはサービスを表します。ディメンションの例として、ユーザー数、スキャンされたホスト数、取り込まれたログの GB 数があります。定義したディメンションごとに、名前、説明、価格、および API 名を追加します。名前、価格、および説明は顧客に表示されます。API 名は、AWS Marketplace による追跡および報告に使用されます。

- [AWS Marketplace Entitlement Service](#) を呼び出して、顧客が購入したディメンションを取得する。
- [AWS Marketplace Metering Service](#) を呼び出して、顧客が使用したディメンションを示す。

契約の料金ディメンションごとに、そのディメンションで契約を超えた追加使用量に応じて顧客が支払うことを選択できます。使用した量に対してのみ顧客が支払う、契約料金なしのディメンションを追加することもできます。

ウィザードを使用して SaaS 製品の契約を作成する場合、料金ディメンションに次のフィールドを定義する必要があります。

- ディメンション API 名 - 使用権限 API を呼び出すときに使用される名前です。この名前は請求レポートに表示され、レポートは外部向けではありません。API 名は最大 15 文字です。この名前を設定した後に変更することはできません。
- ディメンション表示名 - 顧客に表示されるディメンションの名前。この名前は、顧客が製品のディメンションを理解できる名前にします。名前は、最大長が 24 文字で、わかりやすいものである必要があります。この値は変更できます。
- ディメンション説明 - 製品のディメンションに関する追加の情報を提供する、顧客向けのディメンションの説明。説明の最大長は 70 文字です。
- ディメンション - 月別料金 - このディメンションの 1 か月間オプションの、単位あたりのソフトウェア料金。このフィールドでは小数点以下 3 桁がサポートされています。
- ディメンション - 1 年料金 - このディメンションの 12 か月間オプションの、単位あたりのソフトウェア料金。このフィールドでは小数点以下 3 桁がサポートされています。月額料金ではありません。この価格は、12 か月間の 1 回限りの料金を表している必要があります。
- ディメンション - 2 年料金 - このディメンションの 24 か月間オプションの、単位あたりのソフトウェア料金。このフィールドでは小数点以下 3 桁がサポートされています。
- ディメンション - 3 年料金 - このディメンションの 36 か月間オプションの、単位あたりのソフトウェア料金。このフィールドでは小数点以下 3 桁がサポートされています。

例: データストレージアプリケーション

	月額料金	12 か月料金	24 か月料金	追加使用量に対する従量課金制料金
暗号化されていないデータ (GB)	1.50 USD/GB	16.00 USD/GB	30.00 USD/GB	0.1 USD/GB
暗号化されているデータ (GB)	1.55 USD/GB	16.60 USD/GB	31.20 USD/GB	0.11 USD/GB

例: ログモニタリング製品

	月額料金	12 か月料金	追加使用量に対する従量課金制料金
Basic (10 ホストのモニタリング、5 コンテナのモニタリング)	100 USD	1,000 USD	
Standard (20 ホストのモニタリング、10 コンテナのモニタリング)	200 USD	2,000 USD	
Pro (40 ホストのモニタリング、20 コンテナのモニタリング)	400 USD	4,000 USD	
1 時間あたりのモニタリング対象追加ホスト			0.1 USD
1 時間あたりのモニタリング対象追加コンテナ			0.2 USD

 Note

価格は、1 か月、12 か月、24 か月、36 か月の期間に対して定義できます。製品にこれらのオプションを 1 つ以上提供することができます。期間は各ディメンション間で同じである必要があります。例えば、ReadOnlyUsers と AdminUsers のディメンションがあると仮定します。ReadOnlyUsers の年間料金を提示する場合、AdminUsers の年間料金も提示する必要があります。

SaaS 契約アップグレード

顧客は、期間が長いこと以外でより高い価値の契約にアップグレードできます。たとえば、数量が多い契約や上位の使用権限にアップグレードできます。顧客には既存の契約に案分計算されたクレジットが付与されます。顧客は既存の契約のサイズを縮小することはできません。サイズを縮小できるのは更新時のみであり、そうしない場合は更新をキャンセルします。

使用権限は、SaaS 製品で AWS Marketplace Entitlement Service を呼び出すことによって確認されます。

自動更新

顧客は AWS Marketplace を通じて SaaS 契約で製品を購入する際に、契約期間の自動更新に同意することができます。顧客は 1 年、2 年、または 3 年の期間中に、使用権限に対して毎月支払い続けます。顧客は常に更新設定を変更することができます。顧客は更新をキャンセルすることも、異なる数量や期間で契約を更新することもできます。

SaaS 契約が終了したとき

SaaS 契約製品には、契約の有効期限があります。契約が終了すると、次のイベントが発生します。

1. SaaS 製品は、購入者の資格が変更されたことを示す entitlement-updated 通知を受け取ります。AWS Marketplace Entitlement Service は空のレスポンスを返します。
2. その顧客の残り使用量を計測するまでに 1 時間の猶予があります。この時間が経過した後は、この顧客の計測レコードを送信することはできません。

SaaS 契約がキャンセルされたとき

SaaS 契約のキャンセルプロセスの主なポイントは次のとおりです。

1. 顧客は、AWS サポート で SaaS 契約製品のキャンセルと返金をリクエストできます。

顧客は、AWS サポート を通じて 48 時間以内に返金をリクエストする必要があります。

通常、全額または日割り計算による払い戻しは、3~5 営業日以内に行われます。

2. その顧客の Amazon SNS トピックを通じて、SaaS 製品に通知が送信されます。

3. 追加料金が発生した場合は、1 時間以内に最終的な計測レコードを顧客に送信してください。

4. 販売者は、キャンセルが進行中の製品から顧客に通知します。顧客が製品を介してキャンセルを指示している場合は、顧客を AWS Marketplace にリダイレクトします。今後は請求されないようにするために、顧客は AWS Marketplace を通じてキャンセルを確定する必要があります。

での SaaS 無料トライアルオファターの作成 AWS Marketplace

AWS Marketplace 販売者は、AWS Marketplace 管理ポータル (AMMP) で Software as a Service (SaaS) 無料トライアルオファターを作成できます。顧客は、SaaS 無料トライアルオプションを使用して、大規模な購入決定を下す前にソフトウェア製品を評価できます。顧客が製品をサブスクライブすると、その製品は有料ユーザーに行うのと同じ方法でエンタイトルメントチェックを行います。

各 AWS アカウント は、SaaS 製品の無料トライアルを 1 回のみ使用できます。無料トライアル中に付与された無料利用額は、AWS 組織内のリンクされたアカウント間で共有されません。1 つのメイン支払いアカウント内の複数の連結アカウントが、それぞれ個別の無料トライアルを作成できます。

Note

- Seller Data Delivery Service (SDDS) を使用する場合、Amazon Simple Storage Service バケットに[契約の詳細トライアルレポート](#)が送信されます。レポートには、サブスクライバーの名前と ID、オファター ID、契約の開始日と終了日などの契約の詳細が含まれます。
- 販売者は、新しいサブスクリプションの作成時に [Amazon Simple Notification Service \(Amazon SNS\) 通知](#) も受け取ります。Amazon SNS 通知には、isFreeTrialTermPresent 無料トライアル契約を識別するフラグが含まれています。
- また、無料トライアルをサブスクライブしているお客様は、追加のトークン を使用して登録 URL にリダイレクトされます x-amzn-marketplace-offer-type=free-trial。トークンを使用して、無料トライアルを使用するお客様向けに一意の登録エクスペリエンスを作成できます。

SaaS 無料トライアルオファーの作成

販売者は、AWS Marketplace 管理ポータル (AMMP) で SaaS 無料トライアルオファーを作成できます。

SaaS 無料トライアルオファーを作成するには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. で AWS Marketplace 管理ポータル、次のいずれかを選択します。
 - オファーを作成または管理
 - [オファー] タブ
3. [オファー] ページで [公開無料トライアル] タブを選択し、すべての SaaS 無料トライアルを確認します。
4. [無料トライアルオファーを作成] を選択します。販売者は、公開 SaaS 製品ごとに 1 つ、SaaS 無料トライアルオファーを作成できます。
5. [オファーの基本] で、[製品] を選択し、[次へ] を選択します。
6. [無料トライアル設定] で以下を行います。
 - a. [無料トライアルの期間 (日数)] に日数を入力します。

無料トライアル期間の範囲は 7～90 日間です。
 - b. 既存の公開オファーの [製品ディメンション] を表示します。

SaaS サブスクリプションの無料トライアルの製品ディメンションを変更することはできません。

SaaS 契約の無料トライアルでは、ディメンションごとに数量制限を設定したり、ディメンションを削除または追加したりできます。
7. サービス契約をご覧ください。

EULA バージョンで、[AWS Marketplaceの標準契約] または [カスタム EULA] のいずれかを選択し、[オファーを確認] を選択します。
8. オファーに関するすべての情報を検証して確認し、[オファーを作成] を選択します。

SaaS 無料トライアルオファーのキャンセル

販売者は、AWS Marketplace 管理ポータルからいつでも無料トライアルのオファーをキャンセルできます。

SaaS 無料トライアルオファーをキャンセルするには

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. で AWS Marketplace 管理ポータル、次のいずれかを選択します。
 - オファーを作成または管理
 - [オファー] タブ
3. [オファー] ページで、オファーを選択します。
4. [オファーを表示] を選択します。
5. [オファーをキャンセル] を選択します。

オファーがキャンセルされると、そのオファーの有効な契約は有効期限が切れるまで有効です。キャンセルされたオファーの新しい契約は作成できません。

を通じて SaaS 製品に顧客をオンボーディングする AWS Marketplace

Software as a Service (SaaS) サブスクリプションと SaaS 契約により、顧客は を通じて製品をサブスクライブします AWS Marketplace が、AWS 環境内の製品にアクセスします。製品をサブスクライブした後、顧客は販売者が SaaS 製品の一部として作成および管理するウェブサイトのリダイレクトされ、自分のアカウントを登録して製品を設定します。

SaaS 製品のリストを作成するときに、登録ランディングページの URL を指定します。当社はその URL を使用して、顧客がサブスクライブすると、顧客を販売者の登録ランディングページにリダイレクトします。ソフトウェアの登録ランディングページで、顧客のアカウントを作成するために必要な情報を収集します。使用量の通知について顧客にメールで連絡する予定がある場合は、顧客のメールアドレスを収集することをお勧めします。

登録ランディングページは、請求用の顧客の識別子 AWS Marketplace を持つ のフォームデータで x-amzn-marketplace-token トークンを識別して受け入れることができる必要があります。次に、そのトークン値を に渡 AWS Marketplace Metering Service して、一意の顧客 AWS アカウント

ID、顧客識別子 (2025 年 12 月 31 日以降は廃止)、および対応する製品コードについて解決する必要があります。コード例については、[ResolveCustomer コード例](#) を参照してください。

Note

登録トークンは特定のサブスクリプション顧客向けに解決され、生成された各トークンの有効期限は 4 時間です。呼び出し元が同じトークンを使用して API を呼び出している限り、トークンの有効期限が切れるまで同じ応答値を返し続けます。

新しい購入者を受け入れるための SaaS 製品の設定

新しい顧客を受け入れ、適切に計測するために SaaS ソフトウェアを正しく設定する責任を負います。次のプロセスでは、新規顧客のソフトウェアへのアクセスを特定、実装、計測するための推奨方法の 1 つについて説明します。

1. 顧客が AWS Marketplace ウェブサイトの製品ページにアクセスすると、顧客は製品をサブスクライブすることを選択します。
2. お客様の AWS アカウント がお客様の製品にサブスクライブされている。つまり、製品から送信されたサブスクリプションレコードと計測レコードは、顧客の AWS 請求書の一部になります。
3. ID、顧客識別子、製品コードを含む AWS アカウント 登録トークンが顧客に対して生成されます。
4. 顧客は、ソフトウェアの登録ランディングページにリダイレクトされます。そのページでは、顧客 ID が含まれている登録トークンを受け入れる必要があります。
5. 顧客のブラウザは、ソフトウェアの登録ランディングページ URL に POST リクエストを送信します。そのリクエストには、顧客の登録トークンが入っている POST パラメータ、x-amzn-marketplace-token が 1 つ含まれています。登録ウェブサイトの観点からは、顧客がこのパラメータを持つフォームを送信しました。登録トークンは不透明な文字列です。オファertype が無料トライアルの場合は、2 つ目のパラメータ、free-trial 値を含む x-amzn-marketplace-offer-type がリクエストに追加されます。
6. この登録トークンを顧客 AWS アカウント ID、顧客識別子、製品コードと引き換えるには、ウェブサイトが AWS Marketplace Metering Service で [ResolveCustomer](#) を呼び出す必要があります。ResolveCustomer 呼び出しの例については、「」を参照してください [ResolveCustomer コード例](#)。顧客識別子は顧客の AWS アカウント ID ではありませんが、製品間で共通しているため、顧客レコードの一部として内部ソースに保存する必要があります。

製品コードは、AWS が提供する SaaS 製品の一意の文字列です。各 AWS 製品には一意の製品コードが 1 つあり、登録時に割り当てられます。

7. 顧客は、製品でアカウントを作成するか、または既存のアカウントにサインインするように指示されます。

 Note

製品内の既存の顧客アカウントを設定またはリンクするためにチームが手動で行う必要がある場合は、お問い合わせフォームを使用して顧客の連絡先情報を収集できます。連絡先情報を収集し、AWS アカウント ID と一意の顧客識別子 (ステップ 6 で取得) を解決したら、顧客への通知メッセージを表示します。通知で、アカウントがセットアップされていることを示し、連絡を待つようリクエストします。予想される時期と連絡先情報を顧客に伝えます。また、同じ詳細を記載した E メールメッセージを顧客に送信します。

8. 顧客は、その SaaS 製品に固有の認証情報を使用して、販売者のウェブサイトにサインインしています。アカウントのデータベースでは、顧客ごとのエントリを含むことができます。アカウントデータベースには AWS アカウント ID の列が必要です。システム内の他のアカウントが AWS アカウント ID を共有していないことを確認します。
9. 販売者登録プロセス中に、顧客が製品をサブスクライブした、またはサブスクリプションを解除したときに通知する、Amazon SNS トピックをサブスクライブします。これらは、顧客のアクションを通知する JSON 形式の Amazon SNS 通知です。
 - 資格通知 - 契約を含む価格モデルの製品では、購入者が新しい契約を作成、アップグレード、更新したとき、または契約が期限切れになったときに通知されます。販売者のアカウントデータベースには、サブスクリプション状態用に列が追加されている必要があります。詳細については、「[Amazon SNS トピック: aws-mp-entitlement-notification](#)」を参照してください。
 - サブスクリプション通知 - 契約やサブスクリプションなど、あらゆる価格モデルの製品について、購入者が製品をサブスクライブしたときまたはサブスクリプションを解除したときに通知が届きます。詳細については、「[Amazon SNS トピック: aws-mp-subscription-notification](#)」を参照してください。

Amazon Simple Queue Service (Amazon SQS) を使用してこれらのメッセージをキャプチャすることを推奨します。販売者が `subscribe-success` でサブスクリプション通知を受信すると、その顧客アカウントは計測する準備ができています。この通知を受信する前に販売者が送信

したレコードは計測されません。その方法については、「Amazon Simple Queue Service デベロッパーガイド」の「[ステップ 2: Amazon SQS キューにメッセージを送信する許可を Amazon SNS トピックに付与する](#)」を参照してください。

 Note

subscribe-success 通知を受信しない限り、製品サブスクリプションをアクティブ化しないでください。

10. データベースに保存されている AWS アカウント ID を使用して、AWS Marketplace Metering Service を通じて使用状況を計測するか、 を通じて使用権限を確認します AWS Marketplace Entitlement Service。

セキュリティと注文

販売者は、 からすぐに返された ID AWS またはシステムが署名した AWS アカウント IDs のみを信頼する責任があります。登録トークンは約 1 時間後に期限切れになる可能性があるため、すぐに解決することをお勧めします。登録トークンを解決したら、登録が完了するまで AWS アカウント、ID を署名付き属性として顧客のブラウザセッションに保存します。

SaaS 製品に関する Amazon SNS 通知

通知を受け取るには、製品の作成時に AWS Marketplace 提供される の Amazon Simple Notification Service (Amazon SNS) トピックをサブスクライブします。このトピックでは、製品の顧客のサブスクリプションおよび契約の使用権限の変更に関する通知が提供されます。これにより、特定の顧客に対するアクセス許可の付与および取り消しをいつ行うかを把握できます。

 Note

製品の作成プロセス中に SNS トピックの実際の Amazon リソースネーム (ARN) に発行されます。例: `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`

Software as a Service (SaaS) 製品では、次の Amazon SNS トピックを使用できます。

- [Amazon SNS トピック: aws-mp-entitlement-notification](#) - このトピックでは、購入者が新しい契約を作成したり、アップグレードしたり、更新したり、有効期限が切れたりしたときに通

知されます。これは、契約 (SaaS 契約および SaaS 契約 (追加使用量あり) (超過) と呼ばれる) を含む価格モデルの製品でのみ利用できます。

- [Amazon SNS トピック: aws-mp-subscription-notification](#) - このトピックでは、購入者が製品をサブスクライブまたはサブスクライブ解除したときに通知します。これには、プライベートオファーの場合は offer-identifier が、SaaS 無料トライアルの場合は無料トライアルフラグが含まれます。これは、契約やサブスクリプション (SaaS サブスクリプション、SaaS 契約、SaaS 契約 (追加使用量あり) (超過) と呼ばれる) を含むすべての価格モデルで利用できます。

これらの通知に応答するシナリオの詳細は、次のトピックを参照してください。

- [SaaS サブスクリプション製品を と統合する AWS Marketplace](#)
- [SaaS 契約製品と の統合 AWS Marketplace](#)
- [SaaS 契約を pay-as-you-go 製品と統合する AWS Marketplace](#)

Amazon SNS トピック: aws-mp-entitlement-notification

aws-mp-entitlement-notification トピック内の各メッセージの形式は次のとおりです。

```
{
  "action": "<action-name>",
  "customer-identifier": " X01EXAMPLEX",
  "product-code": "n0123EXAMPLEXXXXXXXXXXXXX",
}
```

<action-name> は常に entitlement-updated になります。

Note

- 使用権限メッセージの場合、アクション (新規、アップグレード、更新、期限切れ) に関係なく、メッセージは同じです。更新の内容を確認するには、以降に GetEntitlement を呼び出す必要があります。
- SaaS 契約 (追加使用量あり) (超過) の場合、販売者には [aws-mp-subscription-notification SNS トピック](#) が提供されます。これは、超過料金が追加された場合に出品者が受け取る追加の通知です。販売者が新規顧客を獲得すると、販売者に

entitlement-updated (何らかのアクションを指す場合があります) が届くだけでなく、新規顧客であることを示すサブスクライブメッセージが届きます。

- 将来の日付の契約 (FDA) の場合、このトピックは (契約署名日ではなく) 契約開始日に開始されます。また、契約のキャンセル、置き換え、更新、有効期限切れなど、権限に追加の変更が発生したときにも開始されます。

契約料金がある製品 (従量制料金の契約を含む) は、これらのメッセージに対応する必要があります。応答する方法の詳細については、[シナリオ: ユーザーサブスクリプションの変更を監視する](#) を参照してください。

Amazon SNS トピック: **aws-mp-subscription-notification**

aws-mp-subscription-notification トピック内の各メッセージの形式は次のとおりです。

```
{
  "action": "<action-name>",
  "customer-identifier": " X01EXAMPLEX",
  "product-code": "n0123EXAMPLEXXXXXXXXXXXXX",
  "offer-identifier": "offer-abcexample123",
  "isFreeTrialTermPresent": "true"
}
```

アクション offer-identifier が subscribe-success または の場合にのみ、 が通知に含まれます subscribe-fail。アクションが unsubscribe-pending または の場合、通知には含まれません unsubscribe-success。2024 年 1 月より前に作成されたオファーの場合、この識別子はプライベートオファーの通知にのみ含まれます。2024 年 1 月以降に作成されたオファーの場合、この識別子はプライベートオファーとパブリックオファーの両方を含むすべてのオファーの通知に含まれます。

オファータイプの詳細については、[DescribeEntity API](#) からのレスポンスまたは契約[更新ダッシュボード](#)の契約のオファーの可視性を参照してください。

Note

[DescribeEntity API](#) の場合、そのオファーのターゲットルールのファセットをターゲットとする がアカウント AWS アカウント で見つかった場合、これはプライベートオファーです。

そのオファ어의ターゲットルールのファセットをターゲットとする が AWS アカウント アカウントにない場合は、パブリックオファーです。

購入者のサブスクリプションが無料トライアルかどうかは、isFreeTrialTermPresent プロパティによって示されます。このプロパティの JSON 値は boolean データ型ではありません。代わりに、値は文字列データ型に変換されます。詳細については、「[SaaS 無料トライアル](#)」を参照してください。

<action-name> は、通知によって異なります。指定できるアクションは以下のとおりです。

- **subscribe-success** - この subscribe-success メッセージは、出品者が計測レコードの送信を開始できる時期を通知します。購入者が[契約ベースのオファー](#)を受け入れると、このメッセージは新しい offer-identifier で再度送信されます。
- **subscribe-fail** - メッセージが生成された場合、購入者subscribe-failがすでに から販売者の SaaS ランディングページに移行しているにもかかわらず AWS Marketplace、支払いが失敗した可能性があります。販売者は、商品の使用を許可する前に subscribe-success メッセージを待つ必要があります。
- **unsubscribe-pending** - 購入者がサブスクリプションを解除すると、最初に unsubscribe-pending メッセージが送信されます。これは、購入者が完全にキャンセルされる前に、販売者が最終的な計測レコードを送信するまでの時間が限られている (約 1 時間) ことを示しています。
- **unsubscribe-success** - unsubscribe-success このメッセージはキャンセルの完了を知らせるもので、それ以降の計測レコードは受け付けられません。

Note

- 購入者がサブスクリプションを解除し、最後の unsubscribe-success メッセージが送信される前にすぐに再サブスクライブに成功した場合、最後の unsubscribe-success メッセージは送信されず、代わりに subscribe-success メッセージが送信されます。
- 将来の日付の契約 (FDA) の場合、subscribe-success アクション (契約署名日ではなく) 契約開始日に開始されます。

サブスクリプション料金がある製品 (従量制料金の契約を含む) は、これらのメッセージに対応する必要があります。応答の管理方法については、次のトピックを参照してください。

- [SaaS サブスクリプション製品を と統合する AWS Marketplace](#)
- [SaaS 契約をpay-as-you-go製品と統合する AWS Marketplace](#)

SNS トピックへの SQS キューのサブスクライブ

Amazon SQS キューを指定された SNS トピックにサブスクライブすることを推奨します。SQS キューを作成し、そのキューをトピックにサブスクライブする方法の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[Amazon SNS トピックへ Amazon SQS キューをサブスクライブする](#)」を参照してください。

Note

製品を販売する AWS アカウント ために使用される AWS Marketplace からのみ SNS トピックをサブスクライブできます。ただし、メッセージを別のアカウントに転送することはできません。詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[別のアカウントの Amazon SQS キューへ Amazon SNS メッセージを送信する](#)」を参照してください。

通知の SQS キューのポーリング

SQS キューを SNS トピックにサブスクライブすると、メッセージは SQS に保存されます。継続的にキューをポーリングし、メッセージを探して必要に応じて処理するサービスを定義する必要があります。

AWS Marketplace Metering Service API および使用権限管理サービス API へのアクセス

このセクションでは、AWS Marketplace Metering Service または AWS Marketplace Entitlement Serviceとの統合プロセスについての概要を示します。これは、Software as a Service (SaaS) 製品の顧客使用状況の請求とレポートが正確であることを確認するために使用されます。限定された状態で公開済みの SaaS サブスクリプション製品または SaaS 契約製品を送信済みであることを前提としています。限定された状態では、テストアカウントを使用して、製品が適切に構成されていて機能していることを確認できますが、製品は一般公開されていません。

Note

SaaS 製品が別の方法で計測を処理する別の AWS マネージドサービスと統合されている場合 (Amazon SageMaker Ground Truth や など AWS WAF)、AWS Marketplace 計測サービスと統合する必要はありません。顧客への二重請求を避けるため、製品の計測は 1 つのシステムでのみ行う必要があります。

トピック

- [SaaS サブスクリプションでの使用量の計測の設定](#)
- [を使用した使用権限の確認 AWS Marketplace Entitlement Service](#)
- [SaaS 製品統合チェックリスト](#)

と認証情報の設定については AWS CLI、AWS Command Line Interface ユーザーガイドの「[AWS CLI の設定](#)」を参照してください。Python SDK AWS を初めて使用する場合は、Boto 3 [クイックスタート](#)を参照してください。

SaaS サブスクリプションでの使用量の計測の設定

Software as a Service (SaaS) サブスクリプションでは、すべての使用量を計測し、指定した計測レコード AWS に基づいて から顧客に請求されます。SaaS 契約では、販売者は、顧客の契約による使用権限を超えた使用量のみを計測します。以下のセクションでは、SaaS 製品の使用量の計測の設定方法について説明します。

アプリケーションが顧客の使用量を計測すると、アプリケーションは蓄積された AWS 使用量を提供します。アプリケーションは、製品を作成するときに定義した価格ディメンション (所定の 1 時間に、転送された GB 数やスキャンされたホスト数など) を計測します。例えば、アプリケーションに送信されたデータ量に基づいて課金する場合、データ量を測定し、対応する計測レコードを 1 時間に 1 回送信できます。は、計測データを使用して、製品の作成時に指定した料金とともに顧客の請求書 AWS を計算します。

Note

オプションとして、追跡するプロパティ全体で使用量を分割できます。これらのプロパティはタグとして購入者に公開されます。これらのタグにより、購入者はコストをタグ値別の使用量に分けて表示できます。例えば、ユーザーごとに課金し、ユーザーに Department プロパティがある場合、Department のキーを持つタグを使用して使用量の割り当てを作成

し、値ごとに 1 つの割り当てを行うことができます。これによってレポートする料金、サイズ、総使用量は変更されませんが、顧客は製品に適したカテゴリ別にコストを確認できます。詳細については、「[ベンダー計測タグ付け \(オプション\)](#)」を参照してください。

トピック

- [時間単位での計測](#)
- [使用状況を計測するように製品を設定します。](#)
- [ベンダー計測タグ付け \(オプション\)](#)

時間単位での計測

すべてのお客様の使用状況は AWS、一度に最大 25 個のバッチで 1 時間ごとに報告することをお勧めします。これにより、お客様は使用量とコストを可能な限り細かく把握できます。1 時間を超える期間 (1 日など) で使用量を集計する場合は、以下の考慮事項に注意してください。

- AWS は、お客様から計測レコードを受け取った場合にのみ、製品の使用に対して顧客に請求できます。製品の計測レコードが正常に送受信されるようにしておくことは販売者の責任です。を使用して AWS CloudTrail、送信するレコードが正しいことを確認できます。その情報を使用すると、継続的な監査を実行することもできます。詳細については、「[を使用した AWS Marketplace Metering API コールのログ記録 AWS CloudTrail](#)」を参照してください。
- 「サブスクリプション」料金モデルの SaaS の場合 (「契約」料金モデルまたは「消費量での契約」ではない)、購入者はいつでもサブスクリプションを解除できます。購入者がこのサブスクリプション解約アクションを開始すると、販売者は unsubscribe-pending [通知](#)を受け取り、最終 unsubscribe-success 通知の前に報告されていないすべての使用量を送信を 1 時間以内に行う必要があります。2 回目の通知以降は、すべて無視されます。他の 2 つの料金モデルには、サブスクリプションの時間に基づいて設定された期間があり、購入者はその期間中にサブスクリプションを解約することはできません。自動更新をオフにすることのみが可能です。自動更新でない場合は、その期間の終了時に同じ通知が送信されます。
- 計測レコードを 1 時間ごとに送信せず、アプリケーションまたはネットワークの停止が発生した場合、レコードはさらに遅れます。これにより、サブスクリプションの有効期限が切れた後にアプリケーションまたはネットワークの停止が復元された場合、未報告の使用が発生する可能性があります。
- 報告する使用量がない場合でも、1 時間ごとに計測レコードを送信し続け、その時間について報告する使用量がない場合は数量 0 と記録できます。購入者のディメンションの使用量を 0 以上報

告した後は、レコードを変更できないことに注意してください。したがって、過去 1 時間の使用量を報告するのがベストプラクティスです。報告された使用状況は、請求サイクル全体で購入者の [AWS 請求とコスト管理](#) コンソールに表示されます。コンソールには、製品の各ディメンションが個別の明細項目に表示され、ディメンションの価格が単位あたり 0 USD であっても、消費されたユニットの合計数と、その請求サイクルまたはこれまでに消費された明細項目の合計コストが表示されます。有効にすると、[AWS Cost and Usage Report \(CUR\)](#) には、単位あたりの料金を含む詳細が表示されます。これらのレポートはリアルタイムでは更新されません。

- 公開中、AWS Marketplace 運用チームは、SaaS SaaS アプリケーションが製品を公開する前に計測レコードを正常に送信することをテストします。通常、チームは SaaS のモックサインアップを実行し、計測レコードが受信されたことを確認します。

Note

SaaS 製品が別の方法で計測を処理する別の AWS マネージドサービスと統合されている場合 (Amazon SageMaker Ground Truth や など AWS WAF)、AWS Marketplace 計測サービスと統合する必要はありません。顧客への二重請求を避けるため、製品の計測は 1 つのシステムでのみ行う必要があります。現時点では、AWS Marketplace は新しい AWS WAF 製品を公開していないことに注意してください。

使用状況を計測するように製品を設定します。

AWS Marketplace Metering Service の BatchMeterUsage オペレーションを使用して、計測レコードを AWS に配信します。以下に留意してください。

- 当社では、BatchMeterUsage オペレーションによるバッチ処理を使用することを販売者に求めています。
- 当社では、毎時 0 分に計測リクエストを重複排除します。
 - リクエストは、製品/顧客/時間/ディメンションの組み合わせごとに重複排除されます。
 - リクエストはいつでも再試行できますが、元の数量と異なる数量で計測した場合は、元の数量で課金されます。
 - 同じ顧客、ディメンション、時間に対して複数のリクエストを送信する場合、レコードは集計されません。
- 顧客が商品を購入している場合、販売者は最大で過去 6 時間のタイムスタンプ付きの計測レコードを送信できます。顧客が登録を解除した場合、出品者は顧客が登録を解除してから 1 時間以内に計測レコードを送信する必要があります。

- BatchMeterUsage ペイロードは 1 MB を超えてはなりません。ペイロードのサイズを超えないように、BatchMeterUsage リクエストで送信する使用状況レコードの数を選択します。
- AWS Marketplace Metering Service は、「AWS 全般のリファレンス」の[AWS Marketplace 「エンドポイントとクォータ AWS リージョン」](#)に記載されている で利用できます。SaaS 計測製品では、製品のリクエスト時に、デフォルトで米国東部 (バージニア北部) リージョンが有効になっています。他のリージョンを使用する場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。詳細については、「[BatchMeterUsage](#)」を参照してください。

コード例については、[SaaS 製品統合のコード例](#) を参照してください。

例: ホストスキャン

販売者は、コンピューティングハードウェアで既知のセキュリティ脆弱性を分析する製品を提供しているとします。顧客は、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスのスキャンを手動で開始またはスケジュールします。製品は、スキャンを実行すると、スキャンした固有のホストの数を 1 時間ごとに集計します。この例の製品では [Hosts (ホスト)] カテゴリを使用しています。スキャンされたホストのタイプに複数のディメンションを宣言できます。たとえば、小型、中型、そして大型のホストに異なる料金を請求できます。

例: ログ分析

顧客の製品、トレンドの報告、および異常によって生成されたログを取り込む SaaS 製品を提供しているとします。顧客がログをその製品にアップロードすると、販売者は受信したデータの量をメガバイト、ギガバイト、またはテラバイト単位で計測します。毎時 10 分ごとの cron ジョブが、直前の 1 時間の顧客ごとの使用量を読み取ります。そのジョブは、バッチレポートを作成し、BatchMeterUsage オペレーションを使用してそのレポートを AWS に送信します。この例の製品では [Data (データ)] カテゴリを使用しています。この製品は、特定の 1 時間に保存されたログデータの量を計測することもできます。その場合、この製品は、1 時間に受信したデータと 1 時間に保存された総データの 2 つのディメンションで計測できます。顧客がこのデータを削除するか、期限が切れるまで、保存されたデータの計測を続行できます。

ベンダー計測タグ付け (オプション)

ベンダー計測タグ付けは、独立系ソフトウェアベンダー (ISV) がソフトウェアの使用状況についてより詳細な情報を購入者に提供し、コスト配分を行うのに役立ちます。

購入者が使用したソフトウェア使用状況をタグ付けするには、さまざまな方法があります。1 つの方法は、コスト配分に何を期待するかを購入者にたずねます。次に、購入者のアカウントで追

跡するプロパティ全体で使用量を分割できます。プロパティの例には、Account ID、Business Unit、Cost Centers、および商品に関連する他のメタデータが含まれます。これらのプロパティはタグとして購入者に公開されます。タグを使用すると、購入者は AWS 請求コンソール (<https://console.aws.amazon.com/costmanagement/>) のタグ値によってコストを使用量に分割して表示できます。ベンダー計測タグ付けでは、報告する価格、ディメンション、合計使用量は変わりません。お客様は商品に適したカテゴリ別にコストを確認できます。

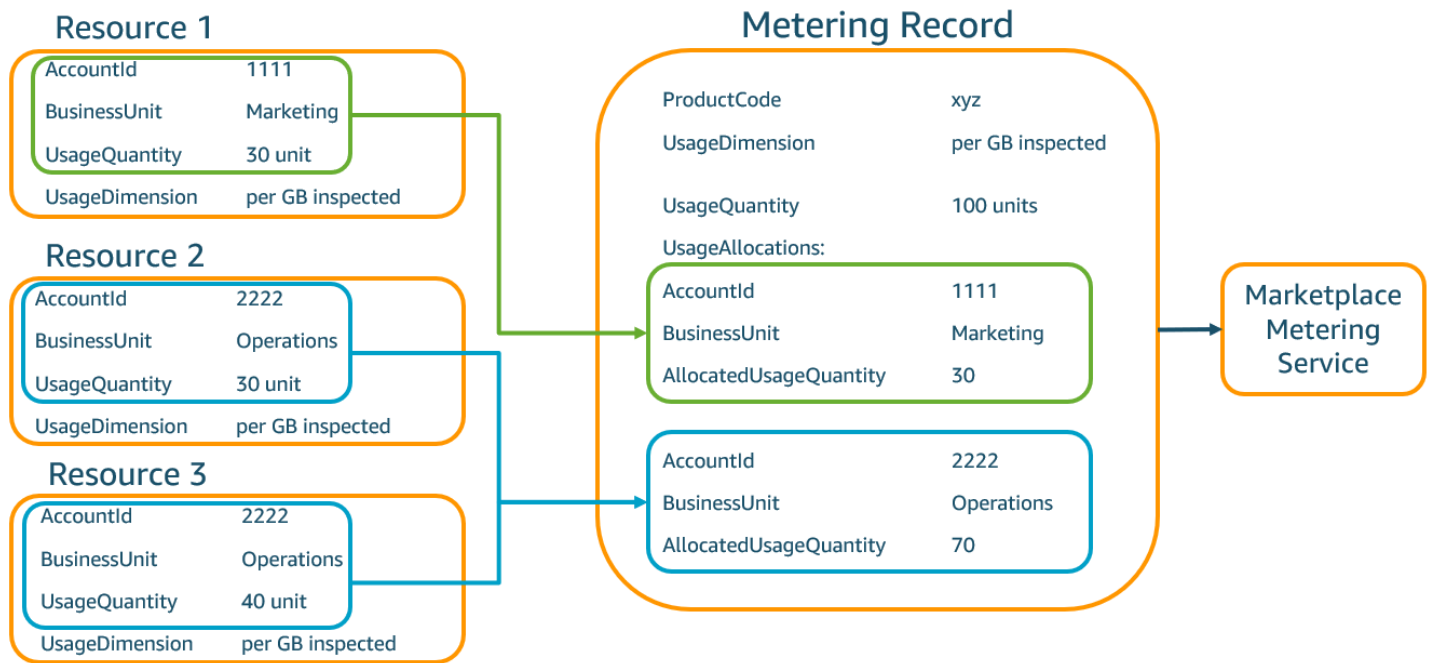
一般的なユースケースでは、購入者が 1 つの AWS アカウントを使って商品を購入します。また、購入者には同じ製品サブスクリプションに関連するユーザーが多数います。Account ID のキーを持つタグを使用して使用量の割り当てを作成し、各ユーザーに使用量を割り当てることができます。この場合、購入者は Billing and Cost Management コンソールで Account ID タグを有効にして、個々のユーザーの使用状況を分析できます。

販売者のエクスペリエンス

販売者は、すべてのリソースの使用量を集計する代わりに、同じタグセットを使用してリソースの計測レコードを集計できます。例えば、販売者は UsageAllocations の異なるバケットを含む計測レコードを作成できます。各バケットは、AccountId や BusinessUnit などの、タグのセットの UsageQuantity を表します。

以下の図では、[リソース 1] には固有の AccountId と BusinessUnit タグセットがあり、[計測レコード] に 1 つのエントリとして表示されます。

[リソース 2] と [リソース 3] にはどちらも同じ AccountId タグ、2222、および同じ BusinessUnit タグ Operations があります。その結果、これらは [計測レコード] の 1 つの UsageAllocations エントリにまとめられます。



また、販売者は、タグのないリソースを単一の UsageAllocation にまとめ、UsageAllocations のエントリの 1 つとして送信することもできます。

制限には以下が含まれます。

- タグの数 - 5
- UsageAllocations のサイズ (カーディナリティ) - 2,500
- リクエストの最大サイズ - 1 MB

検証には次の項目が含まれます。

- タグのキーと値に使用できる文字 - a-zA-Z0-9+ - = . _ \ V @
- UsageAllocation リスト全体の最大タグ数 - 5
- 2 つの UsageAllocations タグに同じタグ (つまり、同じタグキーと値の組み合わせ) を持つことはできません。その場合は、同じ UsageAllocation を使用する必要があります。
- UsageAllocation の AllocatedUsageQuantity の合計は、総使用量である UsageQuantity と等しくなる必要があります。
- ペイロードの最大サイズは 1 MB 以下でなければなりません。これには、入力属性キー (例: UsageRecords、AllocatedUsageQuantity、タグ) が含まれます。

Note

ペイロードの制限を超えないようにするには、ビジネス要件に基づいて最大サイズを設定したサンプルリクエストオブジェクトを作成し、そのオブジェクトを JSON 文字列に変換して、サイズをバイト単位で取得します。1 回の API コールが 1 MB の制限を超えないようにしてください。例えば、1 つの UsageRecord リクエストの最大サイズが 200 KB の場合、リクエストの一部として 5 つを超える UsageRecords を送信しないでください (200 KB * 5 = 1 MB)。

購入者エクスペリエンス

次の表は、購入者が AccountId および BusinessUnit ベンダータグを有効化した後の購入者エクスペリエンスの例を示しています。

この例では、購入者は [コスト使用状況レポート] で割り当てられた使用量を確認できます。ベンダー計測タグにはプレフィックス “aws:marketplace:isv” が使用されます。購入者は、請求情報とコスト管理の、[コスト配分タグ]、[AWS生成コスト配分タグ] でそれらを有効化できます。

[コスト使用状況レポート] の最初と最後の行は、出品者が Metering Service に送信する内容に関係します ([販売者のエクスペリエンス](#) 例を参照)。

コスト使用状況レポート (簡略版)

ProductCode	購入者	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	70	2222	オペレーション
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	30	3333	財務

ProductCode	購入者	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	20	4444	IT
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	20	5555	マーケティング
xyz	111122223333	ネットワーク: 検査対象 (GB) あたり	30	1111	マーケティング

コード例については、[使用量割り当てタグ付きの BatchMeterUsage のコード例 \(オプション\)](#) を参照してください。

を使用した使用権限の確認 AWS Marketplace Entitlement Service

SaaS 契約製品の場合、販売者の製品は AWS Marketplace Entitlement Service を呼び出し、[GetEntitlements](#) API を使用して、顧客の使用権限を取得します。製品は、そのアカウントの後続の使用を AWS Marketplace エンタイトルメントサービスに対して検証する必要があります。例えば、顧客がアカウントに 10 人のユーザーをプロビジョニングする場合、製品は AWS Marketplace その容量に対する使用権限サービスをチェックする必要があります。

製品に対するお客様の使用権限を確認するには、AWS Marketplace 使用権限サービスで `GetEntitlements` オペレーションを使用します。AWS Marketplace 使用権限サービスは、米国東部 (バージニア北部) リージョンでのみ利用でき、`entitlement.marketplace.us-east-1.amazonaws.com` または `entitlement-marketplace.us-east-1.api.aws` からアクセスできます。

`GetEntitlements` は、顧客の AWS アカウント ID、顧客識別子、ディメンションをフィルターとして受け入れます。ProductCode は必須パラメータです。このオペレーションは、使用権限のページ分割されたリストを返します。結果には、その使用権限が有効である最小期間

を示す ExpirationDate フィールドがあります。顧客が自動更新をセットアップしている場合、ExpirationDate フィールドの日付は更新日付です。

コード例については、[SaaS 製品統合のコード例](#) を参照してください。

ユーザー操作に関する使用権限の取得

次の例は、ユーザー操作に関する使用権限を取得するプロセスを理解するのに役立ちます。

例: ユーザーベースの製品

所定の顧客に対していくつかのアカウントが存在できる製品を提供しています。その顧客はダッシュボードで新しいユーザーを (たとえば、認証情報を割り当てるために) プロビジョニングできます。顧客が新しいユーザーをプロビジョニングすると、製品は GetEntitlements を呼び出して、そのキャパシティがあることを確認します。そうでない場合は、AWS Marketplace Metering Service を呼び出して、追加のユーザーに請求することができます。

例: データストレージ製品

顧客が特定量のデータを暗号化形式または非暗号化形式で保存できる製品を提供しています。顧客は、製品での既存のデータ量および割り当て済みのデータ量が表示されるダッシュボードを表示できます。ダッシュボードでは、GetEntitlements を使用して割り当て量を取得します。

SaaS 製品統合チェックリスト

Software as a Service (SaaS) 製品を公開する前に、必要な設定を完了したか確認する必要があります。次の SaaS 統合チェックリストを使用すると、必要な設定が完了したことを確認できます。

カテゴリ	要件
アクセス	使用する AWS AWS Marketplace アカウントを含む販売者登録フォームを送信しました。
アクセス	利用規約、銀行口座、および W8/W9 税金フォームを含めて、販売者登録を完了した。
アクセス	登録済みアカウントのクロス AWS Marketplace アカウントロールを設定しました。
製品	AWS Marketplace 管理ポータルでの製品リクエストフォームの記入を完了した。

カテゴリ	要件
製品	AMMP の製品作成ウィザードのメモタブでテスト用の AWS アカウント IDs を提供しました。
製品	[製品] タブで .txt 形式の EULA の URL を指定した。
製品	から製品コードと Amazon SNS トピック情報を受け取りました AWS Marketplace。
製品	Amazon SNS トピックを予約購読し、Amazon SNS トピックを予約購読するための Amazon SQS キューを作成した。
請求ソリューション	SaaS サブスクリプション製品の顧客に対する 1 時間ごとの計測レコードを BatchMeterUsage オペレーションに送信できることを検証した。追加使用量に対する計測レコードを SaaS 契約製品の顧客ごとに送信できることを確認した。
請求ソリューション	SaaS AWS Marketplace 契約製品の使用権限管理サービスから顧客の使用権限を検証できることを検証しました。
請求ソリューション	テストアカウントに対して生成された請求書で、想定どおりに費用が表示されることを検証した。
請求ソリューション	無効な顧客 ID やキャンセル済みサブスクリプションなどの状況をテストした。
製品	発行 AWS Marketplace のために製品リクエストを送信しました。
Registration (登録)	HTTP POST リクエストを受け入れることができる HTTPS 登録ページを実装した。

カテゴリ	要件
Registration (登録)	新規の顧客登録を受け入れることができることを検証した。
Registration (登録)	登録トークンを Cookie に保存していないことを検証した。
Registration (登録)	ResolveCustomer を使用して AWS トークン CustomerIdentifier から ProductCode と を取得していることを検証しました。
Registration (登録)	から受け取った登録トークンを遅延 AWS なく解決できることを検証しました。
Registration (登録)	Gmail などの E メールサービスのアドレスを使用した登録をブロックしていないことをテストした。
Registration (登録)	不完全な登録および複数回の登録試行を受け入れることができることをテストした。
サブスクリプション	unsubscribe-pending メッセージおよび unsubscribe-success メッセージを処理できることをテストした。
サブスクリプション	unsubscribe-pending メッセージを受信してから 1 時間以内に最終的な計測レコードが送信されることを検証した。
セキュリティ	AWS ルートアカウントに API キーがなく、強力なパスワードがあり、ハードウェア多要素認証 (MFA) デバイスに関連付けられていることを検証しました。すべての管理アクセスは、AWS Identity and Access Management (IAM) で作成された ID を通じて行われます。共有アカウントがありません。

カテゴリ	要件
セキュリティ	Amazon Elastic Compute Cloud (Amazon EC2) へのすべてのプログラムによるアクセスに IAM ロールが使用されていることを確認した。スクリプト、ヘッダー、ソースコードに認証情報がハードコードされていないことを確認した。
セキュリティ	包括的なログ記録およびログの統合が維持されていることを検証した。
セキュリティ	アプリケーションサービスと、データベースやファイルシステムへのアクセスを分離する、パブリックとプライベートのサブネットの境界が明確に定義されていることを確認した。機密データを区別し、パブリックデータとプライベートデータを分離する、明確なデータクラスが定義されていることを確認した。
セキュリティ	スケジュールされたキーローテーションを使用した、転送中および休止中のプライベートデータの暗号化が実施されていることを確認した。
セキュリティ	セキュリティインシデントツールが導入および利用されていること、およびタイムリーな調査と復旧に対応するインシデント対応演習が定期的にスケジュールされていることを確認した。
信頼性	システムが、必要に応じてスケールアップ/ダウンして需要の変化に適応できること、および負荷分散を採用して高いパフォーマンスを確保していることを確認した。システムが、必要に応じてエッジベースのキャッシュを提供できることを確認した。

カテゴリ	要件
信頼性	復旧時間と目標復旧時点が規定されていること、および災害復旧が定期的にスケジュールされていることを検証した。コンポーネントの障害が自動トリガーと通知によって自己修復されることを確認した。

での SaaS 製品のレポート AWS Marketplace

AWS Marketplace は、サブスクライバー、財務状況、使用状況、税金に関するデータを含む Software as a Service (SaaS) 製品のレポートを生成します。次の表に利用可能なツールについての情報を示します。レポートを使用すると、顧客ベースについてのインサイトを得ることができ、売上や税金などの財務をより深く理解できるようになります。詳細については、「[the section called “販売者レポート”](#)」を参照してください。以下の表に、SaaS 製品の財務状況がどのようにレポートされるかを示します。

レポートを行う	SaaS の内容
日別ビジネスレポート	前払い契約料金は [Fees (料金)] セクションに表示されます。 計測使用量料金は [Usage (使用量)] セクションに表示されます。
月別収益レポート	前払い契約料金は [Annual subscriptions (年間サブスクリプション)] セクションに表示されます。 計測使用量料金は [Billing and revenue data (請求および収益データ)] セクションに表示されます。
販売補償レポート	前払い契約料金と月別の追加使用量料金は個別の明細項目として表示されます。

レポートを行う	SaaS の内容
顧客サブスクリプションレポート	新しい SaaS 契約は [Annual subscriptions (年間サブスクリプション)] セクションに表示されます。 新しい SaaS サブスクリプションは [Hourly/monthly subscriptions (時間単位または月単位のサブスクリプション)] セクションに表示されます。

SaaS 製品統合のコード例

次のコード例を使用して、Software as a Service (SaaS) 製品を、製品の公開と保守に必要な AWS Marketplace APIs と統合できます。詳細については、次のセクションを参照してください。

トピック

- [ResolveCustomer コード例](#)
- [GetEntitlement コード例](#)
- [BatchMeterUsage コード例](#)
- [使用量割り当てタグ付きの BatchMeterUsage のコード例 \(オプション\)](#)

ResolveCustomer コード例

次のコード例は、すべての料金モデルに関連しています。Python の例によって、x-amzn-marketplace-token トークンが

CustomerIdentifier、ProductCode、CustomerAWSAccountId に交換されま

す。CustomerAWSAccountId は、サブスクリプションに関連付けられた AWS アカウント ID です。このコードは、AWS Marketplace 管理ポータルからリダイレクトされたときに、登録ウェブサイト上のアプリケーションで実行されます。リダイレクトは、トークンを含む POST リクエストです。

ResolveCustomer の詳細については、「AWS Marketplace Metering Service API リファレンス」の「[ResolveCustomer](#)」を参照してください。

```
# Import AWS Python SDK and urllib.parse
```

```
import boto3
import urllib.parse as urlparse

# Resolving Customer Registration Token
formFields = urlparse.parse_qs(postBody)
regToken = formFields['x-amzn-marketplace-token'][0]

# If regToken present in POST request, exchange for customerID
if (regToken):
    marketplaceClient = boto3.client('meteringmarketplace')
    customerData = marketplaceClient.resolve_customer(RegistrationToken=regToken)
    productCode = customerData['ProductCode']
    customerID = customerData['CustomerIdentifier']
    customerAWSAccountId = customerData['CustomerAWSAccountId']

    # TODO: Store customer information
    # TODO: Validate no other accounts share the same customerID
```

レスポンスの例

```
{
  'CustomerIdentifier': 'string',
  'CustomerAWSAccountId': 'string',
  'ProductCode': 'string'
}
```

GetEntitlement コード例

次のコード例は、契約を持つ SaaS 製品および消費料金モデルを持つ SaaS 契約に関連しています。Python の例では、顧客がアクティブな使用権限を持っていることを確認します。

GetEntitlement について詳しくは、AWS Marketplace 使用権限管理サービス API リファレンスの「[GetEntitlement](#)」を参照してください。

```
# Import AWS Python SDK
import boto3

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

# Filter entitlements for a specific customerID
#
```

```
# productCode is supplied after the AWS Marketplace Ops team has published
# the product to limited
#
# customerID is obtained from the ResolveCustomer response
entitlement = marketplaceClient.get_entitlements({
    'ProductCode': 'productCode',
    'Filter' : {
        # Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
        'CUSTOMER_IDENTIFIER': [
            'customerID',
        ]
        # Option 2: Using CustomerAWSAccountID (preferred)
        # 'CUSTOMER_AWS_ACCOUNT_ID': [
        #     'awsAccountID',
        # ]
    },
    'NextToken' : 'string',
    'MaxResults': 123
})

# TODO: Verify the dimension a customer is subscribed to and the quantity,
# if applicable
```

レスポンスの例

戻り値は、AWS Marketplace 管理ポータルで製品を作成したときに作成されたディメンションに対応します。

```
{
  "Entitlements": [
    {
      "CustomerIdentifier": "string",
      "CustomerAWSAccountID": "string",
      "Dimension": "string",
      "ExpirationDate": number,
      "ProductCode": "string",
      "Value": {
        "BooleanValue": boolean,
        "DoubleValue": number,
        "IntegerValue": number,
        "StringValue": "string"
      }
    }
  ]
}
```

```
],  
  "NextToken": "string"  
}
```

BatchMeterUsage コード例

次のコード例は、SaaS サブスクリプションおよび消費料金モデルを使用した契約に関連しますが、消費のない SaaS 契約製品には該当しません。Python の例では、従pay-as-you-goの料金を顧客に請求 AWS Marketplace するために、計測レコードを に送信します。

```
# NOTE: Your application will need to aggregate usage for the  
#       customer for the hour and set the quantity as seen below.  
# AWS Marketplace can only accept records for up to an hour in the past.  
#  
# productCode is supplied after the AWS Marketplace Ops team has  
# published the product to limited  
#  
# You can use either:  
# - customerID from the ResolveCustomer response (deprecated after Dec 31, 2025)  
# - AWS account ID of the buyer  
  
# Import AWS Python SDK  
import boto3  
from datetime import datetime  
  
# Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)  
usageRecord = [  
    {  
        'Timestamp': datetime(2015, 1, 1),  
        'CustomerIdentifier': 'customerID',  
        'Dimension': 'string',  
        'Quantity': 123  
    }  
]  
  
# Option 2: Using CustomerAWSAccountID (preferred)  
# usageRecord = [  
#     {  
#         'Timestamp': datetime(2015, 1, 1),  
#         'CustomerAWSAccountID': 'awsAccountID',  
#         'Dimension': 'string',  
#         'Quantity': 123  
#     }  
# ]
```

```
# ]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(
    UsageRecords=usageRecord,
    ProductCode='productCode'
)
```

BatchMeterUsage の詳細については、AWS Marketplace Metering Service API リファレンスの「[BatchMeterUsage](#)」を参照してください。

レスポンスの例

```
{
  'Results': [
    {
      'UsageRecord': {
        'Timestamp': datetime(2015, 1, 1),
        'CustomerIdentifier': 'string',
        'CustomerAWSAccountID': 'string',
        'Dimension': 'string',
        'Quantity': 123
      },
      'MeteringRecordId': 'string',
      'Status': 'Success' | 'CustomerNotSubscribed' | 'DuplicateRecord'
    },
  ],
  'UnprocessedRecords': [
    {
      'Timestamp': datetime(2015, 1, 1),
      'CustomerIdentifier': 'string',
      'CustomerAWSAccountID': 'string',
      'Dimension': 'string',
      'Quantity': 123
    }
  ]
}
```

使用量割り当てタグ付きの **BatchMeterUsage** のコード例 (オプション)

次のコード例は、SaaS サブスクリプションと使用料金モデルを含む契約に関連していますが、使用されていない SaaS 契約製品には関連していません。Python の例では、従量制料金を顧客に請求するために、適切な使用量割り当てタグを含む計測レコードを AWS Marketplace に送信します。

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited
#
# You can use either:
# - customerID from the ResolveCustomer response (deprecated after Dec 31, 2025)
# - AWS account ID of the buyer

# Import AWS Python SDK
import boto3
import time

# Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
usageRecords = [
    {
        "Timestamp": int(time.time()),
        "CustomerIdentifier": "customerID",
        "Dimension": "Dimension1",
        "Quantity": 3,
        "UsageAllocations": [
            {
                "AllocatedUsageQuantity": 2,
                "Tags": [
                    { "Key": "BusinessUnit", "Value": "IT" },
                    { "Key": "AccountId", "Value": "*****" },
                ]
            },
            {
                "AllocatedUsageQuantity": 1,
                "Tags": [
                    { "Key": "BusinessUnit", "Value": "Finance" },
                    { "Key": "AccountId", "Value": "*****" },
                ]
            }
        ]
    },
]
```

```

    ]
  }
]

# Option 2: Using CustomerAWSAccountID (preferred)
# usageRecords = [
#     {
#         "Timestamp": int(time.time()),
#         "CustomerAWSAccountID": "awsAccountID",
#         "Dimension": "Dimension1",
#         "Quantity": 3,
#         "UsageAllocations": [
#             {
#                 "AllocatedUsageQuantity": 2,
#                 "Tags": [
#                     { "Key": "BusinessUnit", "Value": "IT" },
#                     { "Key": "AccountId", "Value": "*****" },
#                 ]
#             },
#             {
#                 "AllocatedUsageQuantity": 1,
#                 "Tags": [
#                     { "Key": "BusinessUnit", "Value": "Finance" },
#                     { "Key": "AccountId", "Value": "*****" },
#                 ]
#             },
#         ]
#     }
# ]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(
    UsageRecords=usageRecords,
    ProductCode="testProduct"
)

```

BatchMeterUsage の詳細については、AWS Marketplace Metering Service API リファレンスの「[BatchMeterUsage](#)」を参照してください。

レスポンスの例

```
{
```



```

"Results": [
  {
    "Timestamp": "1634691015",
    "CustomerIdentifier": "customerID",
    "CustomerAWSAccountID": "awsAccountID",
    "Dimension": "Dimension1",
    "Quantity": 3,
    "UsageAllocations": [
      {
        "AllocatedUsageQuantity": 2,
        "Tags": [
          { "Key": "BusinessUnit", "Value": "IT" },
          { "Key": "AccountId", "Value": "*****" }
        ]
      },
      {
        "AllocatedUsageQuantity": 1,
        "Tags": [
          { "Key": "BusinessUnit", "Value": "Finance" },
          { "Key": "AccountId", "Value": "*****" }
        ]
      }
    ],
    "MeteringRecordId": "8fjef98ejf",
    "Status": "Success"
  }
],
"UnprocessedRecords": [
  {
    "Timestamp": "1634691015",
    "CustomerIdentifier": "customerID",
    "CustomerAWSAccountID": "awsAccountID",
    "Dimension": "Dimension1",
    "Quantity": 3,
    "UsageAllocations": []
  }
]
}

```

を使用して Amazon VPC 経由で製品を配信する AWS PrivateLink

AWS Marketplace は AWS PrivateLink、Amazon ネットワークを使用して、販売する製品へのアクセスを購入者に提供 AWS のサービス できる をサポートしています AWS Marketplace。以下のセク

ションでは、AWS PrivateLink テクノロジーを使用して Amazon Virtual Private Cloud (VPC) エンドポイントを介して製品を設定および配信するプロセスの概要を説明します。

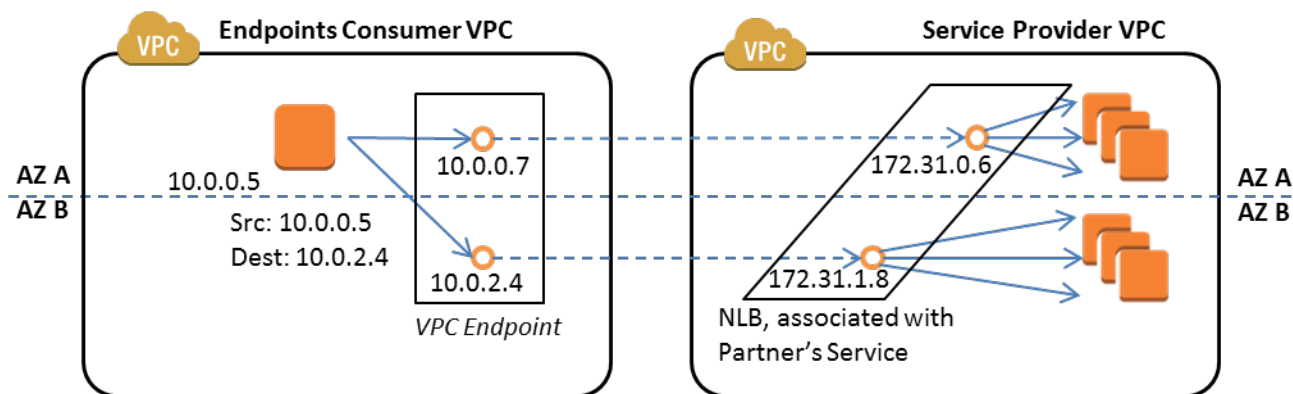
このトピックでは、複数の AWS サービスと AWS Marketplace 環境に関する実務上の知識があることを前提としています。

トピック

- [序章](#)
- [製品の設定](#)
- [製品を に送信する AWS Marketplace](#)
- [VPC エンドポイントへの購入者のアクセス](#)
- [付録: チェックリスト](#)

序章

AWS Marketplace 販売者は、Amazon VPC エンドポイントを介して購入者にサービスへのアクセスを提供できます。このアプローチにより、購入者は [AWS PrivateLink](#) テクノロジーを使用して Amazon ネットワーク経由で販売者のサービスにアクセスできます。AWS Marketplace を使用してこのサービスを作成および配信する場合、購入者は でサービスを検出できます AWS Marketplace。購入者は、VPC エンドポイントを作成するための使用可能なサービスのリストで、製品を見つけることもできます。



[VPC エンドポイント](#) は、インターネット、NAT デバイス、VPN 接続、または を介したアクセスを必要とせずに、AWS VPC と別の AWS サービスとの間にプライベート接続を作成できるようにする仮想デバイスです AWS Direct Connect。を使用してエンドポイントサービスを作成し AWS Marketplace、購入者がこのテクノロジーを使用してサービスに接続できるようにします。この接続

方法は、インターネット経由ではなく、Amazon のプライベートネットワーク経由でサービスにアクセスするため、購入者にとってより安全です。

サービスを提供するリージョンごとに、サービスエンドポイントを作成することにより、リソースを新規作成するか既存のリソースを使用して VPC を設定し、サービスインスタンスと Network Load Balancer をセットアップして、サービスを Network Load Balancer に登録します。これらのステップを完了し、サービスをテストしたら、設定情報を [AWS Marketplace Seller Operations](#) チームに提供します。

AWS では、購入者が VPC エンドポイントを作成するときに使用できるプライベート DNS 名を指定することをお勧めします。

購入者は VPC エンドポイントを作成するときに、オプションとして、プライベート DNS 名を有効にすることができます。このオプションを選択すると、購入者の VPC サービスによって [プライベートホストゾーン](#) が設定されます。プライベート DNS 名を指定すると、購入者はこれを使用して販売者のサービスに接続するように VPC エンドポイントを設定できます。購入者のプライベートホストゾーンで、プライベート DNS 名 (api.example.com) は、エンドポイントサービス用にランダムに生成された DNS 名 (vpce-1111111111111111-yyy-yyyyy.api.vpce.example.com) をポイントします。購入者の EC2 インスタンスは、複数の異なる VPC 間で同じ統一された DNS 名 (api.example.com) を呼び出します。また、パブリックとプライベートの DNS 名が同じであれば、購入者は VPC の内部または外部からサービスにアクセスする際に同じパブリック名を使用できます。

を通じてサービスを利用できるようにする方法については AWS Marketplace、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。AWS Marketplace 購入者がサービスをサブスクライブして VPC エンドポイントを作成すると、サービスが AWS Marketplace サービスの下に表示されます。AWS Marketplace Seller Operations チームは、VPC エンドポイントの作成時にサービスを簡単に検出できるように、使いやすい DNS 名を使用します。

製品は SaaS(Software as a Service) 製品として作成されます。計測と請求は、他の AWS Marketplace SaaS 製品と同じです。

製品の設定

Amazon VPC エンドポイント経由で製品を使用できるように設定するには:

1. 既存の [Amazon VPC](#) を作成または使用します。
2. 製品の [Amazon EC2](#) インスタンス (複数可) を作成 (または既存のインスタンスを使用) します。
3. 製品を提供する各リージョンで [Network Load Balancer](#) を作成します。AWS では、あるリージョンのすべての [アベイラビリティゾーン \(AZ\)](#) を含めることをお勧めします。

4. Amazon VPC コンソール、CLI、またはサポートされている SDK を使用して、VPC エンドポイントサービスを作成します。
5. Network Load Balancer 経由でサービスにアクセスできることを確認します。
6. わかりやすい DNS 名の [証明書](#)を [AWS Certificate Manager \(ACM\) にリクエスト](#)します。ACM は証明書を発行する前に、証明書リクエストのドメイン名の所有者または管理者を検証します。
7. api.vpce.example.com などのユーザーフレンドリーな DNS 名のサブドメインを、AWS Marketplace Seller Operations チームから提供されたネームサーバーに委任します。DNS システムでは、DNS 名 (vpce-0ac6c347a78c90f8.api.vpce.example.com など) をパブリックに解決できるように、このサブドメインを AWS Marketplace Seller Operations チームが提供する Amazon Route 53 ネームサーバーを指すネームサーバー (NS) リソースレコードを作成する必要があります。
8. 購入者の AWS アカウントへのアクセスを許可します。

注意: サポートされている SDK または次の CLI コマンドを使用して、アカウントへのアクセスを自動化できます: `aws vpcev2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-0123456789abcdef1 --add-allowed-principals arn:aws:iam::111111111111:root arn:aws:iam::222222222222:root`。

製品を に送信する AWS Marketplace

サービスを に公開するプロセス中に AWS Marketplace、AWS Marketplace Seller Operations チームと協力して作業します。PrivateLink 対応製品を送信するには:

1. 次の情報を [AWS Marketplace Seller Operations](#) チームに E メールで送信します。
 - a. エンドポイントの作成に使用されるエンドポイントと AWS アカウント。エンドポイントは次のようになります: `com.amazonaws.vpce.us-east-1.vpce-svc-0daa010345a21646`
 - b. ユーザーが使いやすいサービスの DNS 名。これは、AWS Marketplace 購入者が製品にアクセスするために使用する DNS 名です。
 - c. 証明書のリクエストに使用した AWS アカウントと、購入者が VPC エンドポイントへのアクセスに使用するプライベート DNS 名。

Seller AWS Marketplace Operations チームは、登録するサービスに使用する会社の ID と DNS 名 (api.vpce.example.com など) を検証します。検証後、DNS 名はデフォルトのベースエンドポイントの DNS 名よりも優先されます。

VPC エンドポイントへの購入者のアクセス

AWS Marketplace VPC エンドポイントを作成する購入者は、次のような状況でサービスを検出できません。

- このページで前述した販売者のプロセスに従って、製品を新規作成したか、既存の製品を使用した場合。
- 購入者がサービスをサブスクライブした場合。
- 許可された AWS アカウントのリストに購入者のアカウントを追加しました。

購入者が VPC エンドポイントを作成する場合、オプションとして、プライベートホストゾーンを VPC と関連付けることができます。ホストゾーンにはサービスのデフォルトのプライベート DNS 名 (など) のレコードセットが含まれており、VPC のエンドポイントネットワークインターフェイスのプライベート IP アドレスに解決されます。

AWS Marketplace サービスを含む購入者がホストするエンドポイントは、すべてのアカウントにアクセス許可 (「*」アクセス許可) を付与できます。ただし、この方法を使用すると、サービス名で検索しない限り、Describe 呼び出しやコンソールにサービスが表示されません。Describe 呼び出しでサービスを表示するには、購入者の AWS アカウントをサービスによって許可リストに明示的に追加する必要があります。

サービスにアクセスする場合、購入者は次の操作を行います。

1. でサービスを検出してサブスクライブします AWS Marketplace。
2. AWS Command Line Interface (AWS CLI)、API、または Amazon VPC コンソールを使用してサービスを検出し、使用するサブネットと AZs でサービスに接続するための VPC エンドポイントを確立します。エンドポイントは、サブネットに Elastic Network Interface として表示されます。ローカル IP アドレス、リージョンおよびゾーンの DNS 名がエンドポイントに割り当てられます。

クライアント側の DNS 名	名前
リージョン別	Vpce<0dc9a211a78c90f8>.api.vpce.example.com
IAD2 (1a)	us-east-1a-Vpce<0dc9a211a78c90f8>.api.vpce.example.com

クライアント側の DNS 名	名前
IAD2 (1b)	us-east-1b-Vpce<0dc9a211a78c90f8>.api .vpce.example.com

デフォルトのプライベート DNS 名を指定し、購入者が VPC エンドポイントの作成時に [プライベート DNS 名を有効にする] (プライベートホストゾーンを関連付ける) を選択すると、サービスに接続するためのリージョンのデフォルトのプライベート DNS 名が購入者に表示されます。

名前	エイリアス	[Alias Hosted Zone ID]	(注意)
api.example.com	vpce<0dc9a211a78c90f8>.api.vpce. example.com	Z00AABBCCDD	IAD1 IAD2

付録: チェックリスト

次のチェックリストを使用して、AWS Marketplace Seller Operations チームに送信する前に、製品を設定してテストしていることを確認します。

製品の作成チェックリスト

- VPC を作成 (または既存を使用) し、設定します。
- VPC 内にネットワークロードバランサーを作成して設定します。
- VPC エンドポイントサービスを作成して、ネットワークロードバランサーにサービスを登録します。
- VPC エンドポイントの設定に使用した AWS アカウント ID を AWS Marketplace Seller Operations チームに渡します。
- デフォルトのエンドポイントサービス名 (com.amazonaws.vpce.us-east-1.vpce-svc-0bbb070044a2164 など) を AWS Marketplace Seller Operations チームに提供します。
- ランダムに生成されたサービス DNS 名を上書きするには、ユーザーの使いやすいサービス DNS 名 (必須) を指定します。ユーザーの使いやすいサービス DNS 名に使用されているサブドメインの ACM から SSL 証明書をリクエストします。これらの証明書と、リクエストに使用した AWS アカウント ID を AWS Marketplace Seller Operations チームに提供します。

- 推奨: プライベート DNS 名を提供します。
- AWS PrivateLink テクノロジーを使用してサービスに接続するオプションを AWS Marketplace 購入者に通知し、許可するプロセスを作成します。購入者の AWS アカウント ID を許可されたアカウントのリストに追加します。 IDs

製品のテスト

- サービスが設定済みで、検出可能であることを確認します。
- サービスが Network Load Balancer を介して検出可能であることを確認します。
- 購入者が VPC エンドポイントを作成し、サービスにアクセスできることを確認します。サービスのセットアップに使用した AWS アカウントではない、所有しているアカウントを使用します。

のプロフェッショナルサービス製品 AWS Marketplace

販売者は、AWS Marketplace 購入者にプロフェッショナルサービスを提供できます。プロフェッショナルサービスには、AWS Marketplaceの AWS サービスおよび製品の使用方法に関して、評価、移行、サポート、管理、および他の人をトレーニングするサービスが含まれています。販売者は、提供するサービスを説明する製品オファーを作成し、顧客と交渉して契約条項を作成し、AWS Marketplaceを通じてサービスのカスタムオファーを作成します。

Note

独立系ソフトウェアベンダー (ISV)、チャネルパートナー、またはコンサルティングパートナーは、チャネルパートナーのプライベートオファーを使用して、別のパートナーにプロフェッショナルサービス製品の再販を許可することもできます。詳細については、「[Creating a resell opportunity for a channel partner as an ISV](#)」を参照してください。

購入者は、カテゴリでプロフェッショナルサービスを選択し、配信方法でプロフェッショナルサービスを選択し、パブリッシャー、料金モデル、料金単位別に検索を絞り込むことで、AWS Marketplace カタログでプロフェッショナルサービス製品を見つけることができます。AWS 請求書に記載されているサービスに対して課金されます。などのツールを使用して AWS Cost Explorer、支払いを一元化し、コストを管理できます。

プロフェッショナルサービス製品の詳細については、以下を参照してください。

- [プロフェッショナルサービス製品の開始方法](#)
- [プロフェッショナルサービス製品の詳細の提供](#)
- [プロフェッショナルサービス製品の要件](#)
- [ISV としてのチャネルパートナーへの再販オポチュニティの作成](#)

以下の動画では、AWS Marketplaceのプロフェッショナルサービス製品の管理について説明しています。

[でプロフェッショナルサービス製品を管理します AWS Marketplace。](#)

ヘルプの利用

プロフェッショナルサービス製品のサポートについては、の事業開発パートナー AWS Marketplace または [AWS Marketplace Seller Operations](#) チームにお問い合わせください。

でのプロフェッショナルサービス製品の開始方法 AWS Marketplace

販売者は、AWS Marketplace 購入者にプロフェッショナルサービスを提供できます。プロフェッショナルサービスには、AWS Marketplaceの AWS サービスおよび製品の使用方法に関して、評価、移行、サポート、管理、および他の人をトレーニングするサービスが含まれています。製品を定義すると、購入者はあなたが提供するサービスと、そのサービスを受けるためにあなたの会社を選ぶべき理由について知ることができます。次に AWS Marketplace は、購入者があなたに連絡できるようにします。契約に合意すると、販売者は購入者が固定料金でサービスを購入できるプライベートオファーを作成します。以下のセクションでは、プロフェッショナルサービス製品の使用を開始する方法を示し、最初の製品を作成して、お客様に提供するために必要な手順について説明します。

以下の動画では、AWS Marketplaceのプロフェッショナルサービス製品の表示について説明しています。

トピック

- [前提条件](#)
- [プロフェッショナルサービス製品の作成](#)
- [プライベートオファーを作成する](#)
- [製品情報の編集](#)
- [製品の料金の編集](#)
- [製品の可視性の編集](#)
- [プロフェッショナルサービス製品の削除](#)

前提条件

プロフェッショナルサービスを で販売するには AWS Marketplace、以下の前提条件を満たす必要があります。

- AWS Marketplace 管理ポータルにアクセスできます。これは、販売者として登録し、販売する製品を管理するために使用するツールです AWS Marketplace。へのアクセスの詳細については AWS

Marketplace 管理ポータル、「」を参照してください[AWS Marketplace 販売者のポリシーとアクセス許可](#)。

- AWS Marketplace 販売者として登録し、税金と銀行情報を送信します。支払いを受け取るには、英国以外の銀行口座を指定する必要があります。販売者になる方法については、「[AWS Marketplace 販売者としての開始方法](#)」を参照してください。
- のサービスまたは少なくとも 1 つのパブリック製品に関連するプロフェッショナル AWS サービス製品を提供する必要があります AWS Marketplace。あなたの製品がそれらの製品を直接サポートするか、それらの製品のサブスクリプションを促進するサービスを提供している必要があります。

Note

製品は、評価、実装、マネージドサービス、プレミアムサポート、トレーニングのいずれかの主要なカテゴリに出品されている必要があります。

プロフェッショナルサービス製品のガイドラインの詳細については、「[でのプロフェッショナルサービス製品の要件 AWS Marketplace](#)」を参照してください。

プロフェッショナルサービス製品の作成

次の手順では、AWS Marketplace 管理ポータルで新しいプロフェッショナルサービス製品を作成する方法について説明しています。

プロフェッショナルサービス製品を作成するには

1. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
2. [製品] メニューから [プロフェッショナルサービス] を選択します。このページには、既に作成済みのすべてのプロフェッショナルサービス製品と、それらの製品の作成または変更のために送信したリクエストが表示されます。
3. [プロフェッショナルサービス製品] タブで、[プロフェッショナルサービス製品の作成] を選択します。
4. [製品の作成] ページで、製品の情報を入力し、[送信] を選択します。入力する必要がある詳細については、「[でのプロフェッショナルサービス製品の詳細の提供 AWS Marketplace](#)」を参照してください。
5. (オプション) の「製品」メニューから AWS Marketplace 管理ポータル「プロフェッショナルサービス」を選択し、「リクエスト」タブを選択します。製品リクエストに正しい [製品のタイ

トル] が表示され、[リクエストの状態] が [レビュー中] になっていることを確認します。数分以内に製品が限定プレビューモードで作成されるはずです。

Note

[プロフェッショナルサービス] ページの [リクエスト] タブに戻って、いつでもリクエストの状態を確認できます。作成プロセスでエラーが発生した場合は、ここに表示されます。リクエストを選択すると、リクエストの詳細を確認したり、エラーを修正したりできます。

製品が最初に公開された時点では、あなたの AWS アカウント (製品の作成に使用したアカウント) のみがアクセスできます。プロフェッショナルサービスページから製品を表示する場合は、で表示 AWS Marketplace を選択して、購入者 AWS Marketplace 向けの に表示される製品の詳細を表示できます。この詳細リストは、プライベートオファーを拡張しない限り、AWS Marketplace 他のユーザーは利用できません。

製品を一般公開する方法については、「[製品の可視性の編集](#)」を参照してください。

プライベートオファーを作成する

潜在的な購入者が製品を見ると AWS Marketplace、直接購入することはできません。サブスクライブしようとする、[プライベートオファー](#) をリクエストするようにリダイレクトされます。は AWS Marketplace 販売者アカウントのルートユーザーの E メールアドレスに E AWS Marketplace メールメッセージを送信し、顧客がプライベートオファーをリクエストしたことを通知します。次の手順では、このリクエストに応答する方法を説明します。

Note

を通じて 250,000 USD を超えるプライベートオファーを作成する場合は AWS Marketplace 管理ポータル、追加の承認が必要になる場合があります。詳細については、[AWS Marketplace 管理ポータル](#) 経由で Private Offer Success Team (POST)、Vendor Finance Success (VFS)、または Seller Reporting チームにお問い合わせください。

プロフェッショナルサービス製品のプライベートオファーを作成するには

1. リクエストに関して疑問がある場合は、顧客に連絡して解決してください。AWS Marketplace でプライベートオファーを作成する前に、オファー条件に同意します。購入者は製品を購入する義務はないため、オファーを作成する前に同意しておくのは理にかなっています。
2. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
3. メニューから [オファー] を選択し、[プライベートオファーの作成] を選択します。
4. [プライベートオファーの作成] ページで、プライベートオファーを作成する製品を選択します。現在入手可能な製品のみが含まれます。
5. プライベートオファーを延長する [購入者アカウント ID] を入力します。1 つのプライベートオファーに最大 25 のアカウントを含めることができます。購入者がオファーのリクエスト機能を使用した場合、受信したメールメッセージには、リクエストするアカウントの購入者アカウント ID が含まれます。
6. 購入者に製品の分割払いをできるようにするかどうかを選択します。通常、短期契約は 1 回払いで支払われます。長期契約には分割払いのオプションがある場合がありますが、これは購入者との契約の一部です。[次へ] を選択します。
7. 以下の情報を含む、オファーの詳細を入力します。
 - [カスタムオファー名] - あなたと顧客がオファーを認識できるような詳細情報を入力します。会社名または製品名と製品の説明を含めます。名前、電話番号、住所など、個人を特定できる情報は含めないでください。
 - [契約終了日] - 同意したサービスの終了日。例えば、1 年間のサポートを提供する場合は、サービスが利用可能になる日から 1 年後の日付を入力します。
 - [製品ディメンション] - 提供しているサービスの料金と単位は次のとおりです。
 - 一括払いオファー - 各ディメンションとそれに関連する料金を記載できます (例えば、シルバー、ゴールド、プラチナというディメンションを設定できます)。購入者は希望に応じて選択して支払うことができます。
 - 支払いスケジュールを含むオファー - 1 つのディメンションを選択し、各支払いの金額と日付を記載した支払いスケジュールを提供する必要があります。

Note

0 USD オファーを作成する場合は、確認のため [0 USD 料金を有効にする] を選択する必要があります。この予防措置は、誤って無料オファーを作成することを防ぐのに役立ちます。

- [サービス契約] - 顧客とのサービス契約を定義するドキュメント。アップロードするドキュメント (テキスト形式または PDF 形式) は 1 つの PDF 文書にまとめられるため、内容を理解するためにファイル名が不要であることを確認してください。
 - [オファーの有効期限日] - オファーの有効期限が切れる日付です。これにより購入者がオファーを承諾するまでの期間が決まりますが、プロフェッショナルサービスがいつ利用可能になるかとは関係ありません。オファーの作成後にオファーの有効期限を延長できます。
8. オプションの編集が終わったら、[次へ] を選択します。
9. [オファーを確認] ページで、オファーの詳細が正しいことを確認し、[オファーを作成] を選択します。
- 
- Note**
- オファーが公開されるまで時間がかかる場合があります。公開されたオファーは、[オファーを管理] ページで確認できます。(まだ承認されていない) オファーを編集する必要がある場合は、そのページから編集できます。
10. オファーが公開され、[プライベートオファーの管理] ページに表示されたら、そのオファーの [アクション] メニューから [オファー URL をコピー] を選択し、購入者に E メールメッセージで送信して承認してもらいます。

製品情報の編集

次の手順では、AWS Marketplace 管理ポータルの既存のプロフェッショナルサービス製品の製品情報を編集する方法を説明します。

製品情報を編集するには

1. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
2. [製品] メニューから [プロフェッショナルサービス] を選択します。このページには、既に作成済みのすべてのプロフェッショナルサービス製品と、それらの製品の作成または変更のために発行済みのリクエストが表示されます。
3. 編集する既存の製品を選択します。[変更をリクエスト] メニューから、[製品情報を更新] を選択します。
4. 詳細情報を変更します。編集できるフィールドの詳細については、「[でのプロフェッショナルサービス製品の詳細の提供 AWS Marketplace](#)」を参照してください。
5. [送信] を選択してリクエストを作成します。

6. (オプション) プロフェッショナルサービスページのリクエストタブにまだ表示されていない場合は、「の商品」メニューから AWS Marketplace 管理ポータル「プロフェッショナルサービス」を選択し、「リクエスト」タブを選択します。リクエストに正しい [製品のタイトル] が表示され、[リクエストの状態] が [レビュー中] になっていることを確認します。数分以内に、リクエストした変更が製品に反映されます。エラーがある場合は、こちらで確認し、エラーを修正して編集内容を再送信することができます。

製品の料金の編集

次の手順では、AWS Marketplace 管理ポータルの既存のプロフェッショナルサービス製品の料金情報を編集する方法を説明します。

製品の料金を編集するには

1. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
2. [製品] メニューから [プロフェッショナルサービス] を選択します。このページには、既に作成済みのすべてのプロフェッショナルサービス製品と、それらの製品の作成または変更のために送信したリクエストが表示されます。
3. 編集する既存の製品を選択してから、[変更をリクエスト] メニューから [料金設定ディメンションを更新] を選択します。

Note

を通じてのみ新しい料金ディメンションを追加できます AWS Marketplace 管理ポータル。以前に作成したディメンションを変更または削除するには、[AWS Marketplace Seller Operations チーム](#)に連絡してリクエストを伝えてください。リクエストには、製品 ID と、変更または削除を希望するディメンションの詳細を記載してください。

4. 必要な新しい料金ディメンションを追加します。料金フィールドの詳細については、「[でのプロフェッショナルサービス製品の詳細の提供 AWS Marketplace](#)」を参照してください。
5. [送信] を選択してリクエストを作成します。
6. (オプション) の「製品」メニューから AWS Marketplace 管理ポータル「プロフェッショナルサービス」を選択し、「リクエスト」タブを選択します。リクエストに正しい [製品のタイトル] が表示され、[リクエストの状態] が [レビュー中] になっていることを確認します。数分以内に、リクエストした変更が製品に反映されます。エラーがある場合は、こちらで確認し、エラーを修正して編集内容を再送信することができます。

製品の可視性の編集

デフォルトでは、製品は公開範囲が限定され、新しい製品はあなたのアカウントにのみ表示されます。他のテストアカウントを追加、または製品を AWS Marketplace で公開することができます。次の手順では、AWS Marketplace 管理ポータル の既存のプロフェッショナルサービス製品の可視性を編集する方法を説明します。

製品の可視性を編集するには

1. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
2. [製品] メニューから [プロフェッショナルサービス] を選択します。このページには、作成済みのすべてのプロフェッショナルサービス製品が表示されます。
3. 可視性を編集または更新する既存の製品を選択します。[変更をリクエスト] メニューから、[製品の可視性の更新] を選択します。
4. オプションを選択して製品の可視性を更新し、[送信] を選択してレビュー用にリクエストを送信します。
5. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。ステータスが失敗の場合は、リクエスト名を選択して検出された問題を確認します。

Note

製品を公開 AWS Marketplace カタログに表示するには、AWS Marketplace Seller Operations チームによる製品レビューを行い、製品が製品ガイドラインを満たしていることを確認する必要があります (「」を参照 [でのプロフェッショナルサービス製品の要件 AWS Marketplace](#))。リクエストが完了するには数日かかる場合があります。

プロフェッショナルサービス製品の削除

次の手順では、AWS Marketplace 管理ポータル で既存のプロフェッショナルサービス製品を削除する方法について説明しています。

 Note

プロフェッショナルサービス製品を削除しても、アクティブなプライベートオファーには影響しません。

製品を削除するには

1. ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#) にサインインします。
2. [製品] メニューから [プロフェッショナルサービス] を選択します。このページには、作成済みのすべてのプロフェッショナルサービス製品が表示されます。
3. 削除する既存の製品を選択します。[変更をリクエスト] メニューから、[製品の可視性の更新] を選択します。
4. 製品の可視性を更新する可視性オプションで [制限] を選択し、[送信] を選択します。
5. [リクエスト] タブの [リクエストステータス] が [レビュー中] と表示されていることを確認します。リクエストが完了すると、ステータスが [成功] になります。

 Note

リクエストが完了するには数日かかる場合があります。アクティブなオファーを持つ製品は、最後のアクティブなサブスクリプションまたは契約が完了するまで制限状態に移行され、その後削除されます AWS Marketplace。制限付きステータスは、既存のユーザーが引き続き製品を使用できることを意味します。ただし、製品は一般には公開されなくなり、新規ユーザーも利用できなくなります。

でのプロフェッショナルサービス製品の詳細の提供 AWS Marketplace

プロフェッショナルサービス製品を に公開するときは AWS Marketplace、製品メタデータを指定する必要があります。製品メタデータには、可視性と料金に関する製品設定と、購入者にプロフェッショナルサービス製品に関する詳細情報を提供する製品情報が含まれます。製品の説明、追加リソース、料金ディメンション、製品の可視性を編集できます。以下のセクションでは、製品の詳細を準備または編集する際に使用できる情報について説明します。

Note

プロフェッショナルサービス製品のガイドラインと要件については、「[でのプロフェッショナルサービス製品の要件 AWS Marketplace](#)」を参照してください。

トピック

- [製品の説明](#)
- [追加リソース](#)
- [サポート情報](#)
- [料金ディメンション](#)
- [製品の可視性](#)

製品の説明

製品詳細の製品説明セクションは、製品の中核です。潜在的な購入者が購入を決定できるように、製品について説明しています。製品詳細のこのセクションには、以下のデータが含まれます。

- [製品のタイトル] - 製品の名前。これは製品を識別するために使用され、製品ページと検索結果に表示されます。製品に対して意味のある名前を選択します。内で一意である必要があります AWS Marketplace。
- SKU – (オプション) 製品を追跡するために使用されます AWS Marketplace。この情報はあなた自身が使用するためのもので、購入者には見えません。
- 簡単な説明 – 製品カタログのタイルと製品タイトルの下に表示される AWS Marketplace 製品の簡潔な説明。
- [詳しい説明] - 購入者に製品の詳細を説明する、長い形式の説明です。製品の機能、利点、使用方法、および製品に固有のその他情報を一覧表示します。情報を理解して容易に目を通せるようにするため、用意されている書式を使用してください。
- 製品ロゴ – このフィールドは、製品を表すイメージファイルを指すパブリック Amazon S3 URL です。ファイルは、.png、.jpg、.gif 形式で、背景は透明または白、5 MB 未満、幅と高さは 110 ~ 10,000 ピクセルである必要があります。ロゴは製品送信中にアップロードされ、に保存されます AWS Marketplace。URL の内容を変更しても、送信 AWS Marketplace 後に でロゴは変更されません。

Note

指定する Amazon S3 URL は公開されている必要があります。これは、ファイルが存在する Amazon S3 バケットのプロパティです。詳細については、Amazon Simple Storage Service コンソールユーザーガイドの[Amazon S3バケットのパブリックアクセス設定を編集する方法](#)を参照してください。

- [ハイライト] - 製品の主な機能や差別化要因を説明する、1~3 つの短い特徴です。少なくとも 1 つのハイライトが必要です。
- [製品カテゴリ] - 提供するサービスの種類です。少なくとも 1 つ、最大 3 つのカテゴリを選択する必要があります。選択できるカテゴリは多数ありますが、プロフェッショナルサービス製品には以下のうち少なくとも 1 つが含まれている必要があります。

評価

顧客の現在の運用環境を評価して、その組織に適したソリューションを見つけます。

実装

サードパーティー製ソフトウェアの設定、セットアップ、デプロイを支援します。

Premium Support

顧客のニーズに合わせて設計された、エキスパートによるガイダンスやサポートを受けることができます。

Managed Services

顧客に代わって、エンドツーエンドで環境を管理します。

トレーニング

顧客の従業員がベストプラクティスを学ぶのに役立つ、エキスパートによるカスタマイズされたワークショップ、プログラム、教育ツールを提供します。

- [検索結果のキーワード] - 購入者が製品の検索に使用できるキーワードを 3 つまで提供します。キーワードは、カンマ区切りのリストで最大 250 文字まで指定できます。
- 関連製品 – オプション – サービス AWS Marketplace が または サポートする から少なくとも 1 つのパブリック製品を含めます。AWS Marketplace は、製品の詳細ページまたはそれらの製品の関連製品に表示する製品を選択するときに、これらの製品を入力として使用します。

追加リソース

製品詳細の [こちらをご確認ください] セクションには、顧客に役立つように作成したリソースへのリンクを記載できます。これは、オンラインで保存される 1~3 つのダウンロード可能なリソースのオプションセットです。リソースの例としては、製品情報シート、ホワイトペーパー、製品マニュアルなどがあります。リソースごとに、リソースの名前と URL を指定します。

サポート情報

このセクションはフォーマットされたテキストフィールドで、サービスに提供するサポートについて説明できます。

お客様は、サービスの使用、トラブルシューティング、返金のリクエスト (該当する場合) などの問題に対するサポートを期待しています。製品のサポートには、E メールアドレス、電話番号、またはサポート URL (複数可) のいずれかを指定する必要があります。購入者が製品を購入するときに期待できるサポートレベルの説明を含めます。

料金ディメンション

プロフェッショナルサービスの料金ディメンションは、あなたが提供するパッケージです。例えば、シルバー、ゴールド、プラチナのサポートを提供する場合があります。または、10 時間、20 時間、または 50 時間のコンサルティングを提供することもできます。提供するディメンション (少なくとも 1 つ、最大 24 個) ごとに、名前と説明を指定します。購入者と直接やり取りして製品のプライベートオファーを作成する場合は、これらのディメンションの実際の料金を設定します。

Note

料金ディメンションの使用方法和料金の設定方法については、「[プライベートオファーを作成する](#)」を参照してください。

製品の可視性

リリースされた製品は、自分のアカウント AWS Marketplace のみ、テストアカウントの小さなセット、またはすべての AWS アカウントで表示できます。デフォルトでは、製品はプライベートリリースで公開されます。製品の表示設定を変更するには、「[製品の可視性の編集](#)」を参照してください。

でのプロフェッショナルサービス製品の要件 AWS Marketplace

AWS Marketplace は、のすべての製品やサービスの要件を維持します AWS Marketplace。これらの要件は、顧客に安全、安心、および信頼できる、整理されたデジタルカタログを提供するのに役立ちます。販売者には、特定の製品のニーズを満たすために、必要に応じて追加の規制やプロトコルの導入を検討するよう奨励しています。

AWS Marketplace は、送信時にすべての製品と関連するメタデータを確認し、現在の AWS Marketplace 要件を満たしているか、超えているかを確認します。当社のセキュリティ要件を満たすため、私たちはこれらの要件を見直して調整します。さらに、AWS Marketplace は製品を継続的に見直し、これらの要件に対する変更を満たしていることを確認します。製品がコンプライアンスに違反した場合、AWS Marketplace は、製品の更新についてお客様に連絡します。場合によっては、問題が解決されるまで、新しい受信者が製品を一時的に利用できないことがあります。

トピック

- [製品設定ガイドライン](#)
- [顧客情報の要件](#)
- [製品使用ガイドライン](#)
- [アーキテクチャガイドライン](#)

製品設定ガイドライン

すべてのプロフェッショナルサービス製品は、次の製品設定ガイドラインに従う必要があります。

- すべての料金ディメンションは、AWS Marketplaceを使用して提供および請求される実際のサービスに関連するものでなければなりません。
- 製品は、評価、実装、マネージドサービス、プレミアムサポート、トレーニングのいずれかのカテゴリに出品されている必要があります。
- 必要なプロフェッショナルサービスカテゴリ以外にも、提供するサービスに適合する他の適切なカテゴリを選択して、製品を正しく分類する必要があります。
- 製品のロゴは、AWS ロゴや無関係なサードパーティーのロゴと混同するように設計しないでください。
- 製品の詳細には、攻撃的または不適切な内容を含めないでください。 <https://aws.amazon.com/aup/> AWS で入手できる適正利用規約に準拠している必要があります。

- プロフェッショナルサービス製品は、 にリストされているサードパーティーソフトウェア製品に関連するサービスを直接サポートまたは提供 AWS Marketplace するか、お客様が の導入または管理に関連する特定の成果を達成できるように支援する必要があります AWS クラウド。

顧客情報の要件

すべてのプロフェッショナルサービス製品は、次の顧客情報の要件に従う必要があります。

- プロフェッショナルサービス製品は、記載されているディメンションで完全に請求する必要があります AWS Marketplace。
- クレジットカードや銀行口座情報など、 のプロフェッショナルサービス製品出品に関する顧客支払い情報は AWS Marketplace 、 いくつの場合も収集できません。
- プロフェッショナルサービス製品 AWS と組み合わせて から提供されるサブスクライバーまたはサブスクライバー候補情報は、 AWS Marketplace 販売者の利用規約に従ってのみ使用する必要があります。

製品使用ガイドライン

すべてのプロフェッショナルサービス製品は、次の製品使用ガイドラインに従う必要があります。

- プロフェッショナルサービスのリストを通じて顧客から連絡があった場合は、2 営業日以内にその顧客に連絡する必要があります。顧客がプライベートオファーを受け入れると、プライベートオファーに説明がない限り、2 日以内にその顧客に連絡するか、次のステップを伝える必要があります。
- 顧客は、サービスの使用、トラブルシューティング、返金のリクエスト (該当する場合) などの問題について、簡単にサポートを受けることができる必要があります。サポートの問い合わせオプションは、フルフィルメントランディングページで指定する必要があります。サポートの説明には、顧客が期待できるサポートレベルに関する記述を含める必要があります。
- 製品の概要には、明確な価値提案、主な機能、詳細なドキュメントへのリンク、提供されるサービスの購入前および購入後のサポートの明確な定義を含める必要があります。
- 製品には明確でわかりやすいサービス条件が必要です。

アーキテクチャガイドライン

すべてのプロフェッショナルサービス製品は、次のアーキテクチャガイドラインに従う必要があります。

- プロフェッショナルサービス製品は、AWS サービスまたは少なくとも 1 つのパブリック AWS Marketplace 製品 (最大 4 つ) に関連している必要があります。この製品では、関連製品のサービスを直接提供したり、関連製品のサブスクライバーを増やすサービスを提供します。
- 提供されるサービスのために顧客のインフラストラクチャにリソースを追加する必要がある場合は、次のガイドラインに従う必要があります。
 - AWS Security Token Service または AWS Identity and Access Management (IAM) を使用するなど、安全な方法でリソースをプロビジョニングします。
 - プロビジョニングされたすべての AWS サービスの説明、IAM ポリシーステートメント、IAM ロールまたはユーザーがカスタマーアカウントでどのようにデプロイおよび使用されるかなど、追加のドキュメントを提供します。
- 顧客が、AWS Marketplace 取引とは別に追加の AWS インフラストラクチャコストが発生した場合、追加のインフラストラクチャ料金を支払う責任があることを説明する通知を製品説明に含めます。
- 製品によってエージェントがデプロイされる場合は、そのエージェントを AWS アカウントでデプロイする方法の説明について顧客に提供する必要があります。

でのプロフェッショナルサービス製品の料金 AWS Marketplace

AWS Marketplace 販売者は、プロフェッショナルサービス製品のプライベートオファーを受け入れることができます。プライベートオファーは、製品の購入に使用される交渉された用語です AWS Marketplace。販売者は、購入者と交渉してプライベートオファーを作成します。プライベートオファーは、で利用可能な料金モデルの 1 つです AWS Marketplace。販売者のプライベートオファーでは、複数年契約とカスタム期間契約のオプションがあります。このトピックでは、プロフェッショナルサービス製品の料金とプライベートオファーについて詳しく説明します。

複数年契約とカスタム期間契約の詳細については、[AWS Marketplace 製品のプライベートオファーの準備](#)および[プライベートオファーの分割プラン](#)を参照してください。

1 つの製品につき 1 つの料金設定ができます。AWS Marketplace 製品の料金の詳細については、「」を参照してください [の製品料金 AWS Marketplace](#)。

プライベートオファーの仕組み

プライベートオファーは、[AWS Marketplace 管理ポータル](#) の オファーページから作成および管理できます。オファー製品を指定して、一意の ID と URL を生成します。プライベートオファーの料金プランを作成し、法的条件と販売ドキュメントを追加し、オファーを特定の購入者 AWS アカウントに拡張します。オファーは、オファーを作成したアカウントにのみ表示されます。

プライベートオファーを作成し、購入者になる可能性がある相手に通知すると、相手はオファーを表示して受け入れることができます。オファーを表示するには、購入者がオファー AWS アカウントを受け取った にサインインする必要があります。

 Note

リンクされたアカウントまたは管理者アカウントにオファーを拡張しない限り、購入者はオファーを表示できません。オファーでサービスの制限を強制することはできないため、製品に制限がない限り、購入者は交渉された価格で製品を好きなだけ使用できます。

プライベートオファーの作成については、「[Creating and managing private offers](#)」を参照してください。

プライベートオファーは販売者レポートで追跡されます。詳細については、「[Reporting for private offers](#)」および「[Seller reports guide](#)」を参照してください。

データ製品

AWS Data Exchange は、AWS のお客様が AWS クラウド でファイルベースのデータセットを安全に交換できるようにするサービスです。プロバイダーとして、AWS Data Exchange は、データ配信、エンタイトルメント、課金テクノロジーを構築して維持する必要を排除します。AWS Data Exchange のプロバイダーは、AWS のお客様に連絡し、既存のお客様にサブスクリプションをより効率的に付与するために、セキュアかつ透過的で信頼性に優れたチャネルを利用できます。AWS Data Exchange プロバイダーになるためのプロセスでは、適格性を判断するためのステップがいくつか必要になります。

データ製品には、次の部分があります。

- 製品の詳細 - この情報は、サブスクライバーになる可能性があるユーザーが製品の内容を理解するのに役立ちます。これには、名前、説明 (簡単な説明と詳しい説明の両方)、ロゴ画像、およびサポート連絡先情報が含まれます。製品の詳細はプロバイダーによって記入されます。
- 製品のオファー - AWS Data Exchange で製品を利用できるようにするには、プロバイダーがパブリックオファーを定義する必要があります。これには、価格と期間、データサブスクリプション契約、返金ポリシー、カスタムオファーの作成オプションが含まれます。
- データセット - 製品には、1 つ、または複数のデータセットを含めることができます。データセットは、ファイルベースのデータコンテンツの動的なセットです。データセットは動的で、リビジョンを使用してバージョン管理されます。各リビジョンには複数のアセットを含めることができます。

資格要件などの詳細については、「AWS Data Exchange ユーザーガイド」の「[AWS Data Exchange でデータ製品を提供する](#)」を参照してください。

での公開用に製品を送信する AWS Marketplace

製品を AWS Marketplace で使用可能にするには、製品送信プロセスを使用します。製品は、1 つの料金構造を持つ 1 つの Amazon マシンイメージ (AMI) などのシンプルなものでも、AWS CloudFormation テンプレートを使用した複雑なものでも、複雑な料金オプションや支払いスケジュールでもかまいません。製品提供を定義し、次の 2 つの方法のいずれか AWS Marketplace 管理ポータル で 通じて送信します。

- [製品] タブの使用 - 比較的複雑度が低い製品の場合、[製品] タブを使用して、リクエストを完全に定義し送信します。
- [アセット] タブの使用 - 複雑度が高く、より多くの定義を必要とする製品の場合、製品積載フォーム (PLF) をダウンロードし、製品の詳細を追加して、[ファイルのアップロード] オプションを使用して完成したフォームをアップロードします。

Note

データ製品プロバイダーは、AWS Data Exchange コンソールを使用して製品を公開する必要があります。詳細については、「AWS Data Exchange ユーザーガイド」の「[新製品の公開](#)」を参照してください。

最初は [製品] タブを使用して開始し、どちらの方法を使用するかを決定することをお勧めします。次の表では、リクエストの送信に使用する設定と方法が示されています。最初の列は製品の料金モデルです。他の 3 つの列では、製品が顧客にデプロイされる方法が示されています。

料金モデル	単一ノード AMI を使用して開始された製品	で起動された製品 AWS CloudFormation	Software as a Service (SaaS) として開始された製品
Bring-Your-Own-License (BYOL)	[製品] タブ	[アセット] タブ	
空き	[製品] タブ	[アセット] タブ	
時間単位払い	[製品] タブ	[アセット] タブ	

料金モデル	単一ノード AMI を使用して開始された製品	で起動された製品 AWS CloudFormation	Software as a Service (SaaS) として開始された製品
年間契約で時間単位での支払い	[製品] タブ	[アセット] タブ	
月額	[製品] タブ	[アセット] タブ	
月間で毎時	[アセット] タブ	[アセット] タブ	
有料使用料 (AWS Marketplace Metering Service)	[製品] タブ	[アセット] タブ	
契約料金	[製品] タブ		
SaaS サブスクリプション			[製品] タブ
SaaS 契約			[製品] タブ
SaaS レガシー			[アセット] タブ

製品を個別に送信することも、製品積載フォームを使用して複数の製品を送信したり、製品のアップデートを同時に実行したりすることもできます。[製品] タブを使用して、複数の製品を送信することはできません。送信する製品とその送信方法が不明な場合は、[製品] タブから始めます。送信で問題が発生した場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

トピック

- [\[製品\] タブの使用](#)
- [会社と製品のロゴ要件](#)
- [有料の再パッケージ版ソフトウェアを送信するための要件](#)
- [ハードウェアコンポーネントを持つ製品の要件](#)
- [AWS CloudFormationが発表した製品 \(無料または有料\) または使用量ベースの有料 AMI 製品](#)
- [製品の変更と更新](#)
- [タイミングと心構え](#)

- [への AMIs の送信 AWS Marketplace](#)
- [最終チェックリスト](#)

[製品] タブの使用

[製品] タブにアクセスするには、AWS Marketplace 管理ポータルにログインします。[製品] タブで、管理する製品のタイプに応じて、[サーバー]、[SaaS]、または [機械学習] のいずれかを選択します。

現在の製品を含む、その製品タイプのダッシュボードが表示されます。[リクエスト] タブを選択すると、ダッシュボードには未処理のリクエストと、完了済みのリクエスト履歴が表示されます。製品リクエストの作成を開始したら、進行中の作業を保存し、必要に応じていくつかの異なるセッションでリクエストを作成することができます。

製品リクエストを送信すると、AWS Marketplace チームはそれをレビューします。リクエストした製品タイプの製品ページで、リクエストのステータスをモニタリングできます。新しい製品の場合、リクエストの公開が承認されると、限定掲載の URL が届きます。この URL を使用して送信をプレビューし、承認します。製品オファーは、送信を承認するまで公開されません。

製品に対する更新をリクエストした場合、変更を確認し承認する必要なく、更新が公開されます。これには、バージョンの追加または削除、メタデータの変更などが含まれます。

リクエストのステータスは [リクエスト] タブで追跡します。タブには、次のいずれかが表示されます。

- [下書き] - リクエストプロセスを開始しましたが、リクエストを送信していません。
- [送信済み] - リクエストを完了および送信して確認中です。
- 必要なアクション – AWS Marketplace チームはリクエストを確認し、詳細情報が必要です。
- 承認が必要 – AWS Marketplace チームは製品の制限付き出品 URL を作成しました。AWS Marketplace が公開する前に URL を確認して、承認または却下する必要があります。承認すると、サイトの公開中にステータスが公開保留中になります。却下すると、ステータスは下書きに戻り、リクエストを変更することができます。
- 公開保留中 – リクエストのモックアップを承認し、製品を AWS Marketplace 公開しています。
- [期限切れ] - リクエストプロセスを開始しましたが、6 か月以内に完了しなかったため、要求が期限切れになりました。

送信済みというステータスのエントリがある場合は、送信を取り消すことができます。下書きというステータスのエントリがある場合は、リクエストを取り消すことができます。これにより、最初からやり直すことができます。下書きのエントリを削除すると、エントリはリクエスト履歴タブに移動します。

に製品を追加するには AWS GovCloud (US) AWS リージョン、[アクティブな AWS GovCloud \(米国\) アカウントがあり](#)、エクスポートコントロールの要件を含む要件に準拠 AWS GovCloud (US) している必要があります。

会社と製品のロゴ要件

AWS Marketplaceをブラウジングした際のユーザーエクスペリエンスに統一感を持たせるため、顧客の企業ロゴおよび製品ロゴは次の AWS Marketplace のガイドラインに準拠する必要があります。

製品ロゴ仕様 - 製品のロゴイメージは、背景は透過または白、120 ~ 640 ピクセルのサイズで比率は 1:1 または 2:1 (幅) である必要があります。

企業ロゴ仕様 - 企業ロゴイメージは、背景が透過で、220 x 220 ピクセルのサイズ、すべての辺の内側に 10 ピクセルの余白を持たせることができます。

有料の再パッケージ版ソフトウェアを送信するための要件

再パッケージされたソフトウェアのリストを送信する前に、以下の要件を満たす必要があります。この場合、再パッケージされたソフトウェアには、オープンソース AMI、または Windows を搭載した AMI などの別のベンダーによって作成されたソフトウェアが含まれます。

要件

- 製品のタイトルには、再パッケージ化による付加価値を示す必要があります。製品のタイトルの例: 強化済み <製品>、パッケージが追加された <製品>、または <製品 2> 上の <製品 1>。
- 製品のタイトルには、ドキュメントでサポートされていない他の言語を含めないでください。例えば、製品のタイトルに認定、オリジナル、無料といった言葉を使用することはできません。ただし、提供する製品詳細でこれらが裏付けられている場合を除きます。
- 商品の簡単な説明には、商品の料金を要約した明確な説明を含める必要があります。簡単な説明は、この商品には...の料金がかかりますというフレーズで始める必要があります。例えば、製品に販売者からのサポート料金が含まれている場合、製品説明に「この製品には販売者サポートの料金がかかります」と明記する必要があります。

- 製品のロゴは、販売者登録時に使用した会社のロゴと同じでなければなりません。製品のロゴが会社のロゴと異なる可能性があるのは、正式なソフトウェアロゴを使用する場合のみで、元のソフトウェアベンダーから明示的な許可を受ける必要があります。明示的な許可を得た場合は、変更リクエストのメモセクション (または製品ロードフォームを使用する場合は [ファイルのアップロード] ページの [簡単な説明を入力] フィールド) にそのドキュメントへのリンクを含める必要があります。
- AMI 製品の場合、AMI 名を元の製品から再利用することはできません。AMI 名は販売者名で始まり、[販売者名] [name-given-to-ami] の形式に従う必要があります。

AMI 名が命名規則に準拠していない場合は、AWS コンソールから AMI をコピーして名前を変更できます。詳細については、「Amazon EC2 ユーザーガイド」の「[Amazon EC2 AMI のコピー](#)」を参照してください。

有料表示が自社製ではないスタンドアロンソフトウェア製品に関するもので、製品に知的財産が追加されていない (例えば、追加のソフトウェアライブラリのバンドルや特別な設定の追加など) 場合は、前述の要件に加えて、以下の要件も満たす必要があります。

- 製品のタイトルには、販売者名 (前述の付加価値と共に) を含める必要があります。販売者名とは、販売者登録時に使用する名前です。例えば、<販売者> によるメンテナンスサポートがある <製品> などです。
- 製品の詳細な説明の最初の行は、これは再パッケージされたソフトウェア製品で、追加料金がかかります... というフレーズで始まる必要があります。(または、オープンソースの場合、これは再パッケージされたオープンソースソフトウェア製品で、追加料金がかかります...)。また、詳細な説明には、課金対象を要約した明確な説明と、それらの機能を説明する追加の詳細を記載する必要があります。例えば、追加サポートの料金を請求するオープンソース製品の長い説明は、これは再パッケージされたオープンソースソフトウェア製品であり、{SLA の詳細} のサポートには追加料金がかかりますように始まる場合があります。

ハードウェアコンポーネントを持つ製品の要件

ハードウェア製品の販売は許可されていません AWS Marketplace。ハードウェアコンポーネント (SIM カード、スマートデバイス、IoT デバイス、センサーなど) を必要とするソフトウェア製品を送信する場合は、以下の要件を満たす必要があります。

- ハードウェアコンポーネントは販売できません AWS Marketplace。
- ハードウェアコンポーネントのコストは、製品の定価に含めることはできません。

- 製品の [製品概要] セクションには、次のステートメントを含める必要があります。「この製品で必要となる可能性のあるハードウェアは、個別に取得する必要があります。」詳細については、製品の詳細ページを参照してください。

AWS CloudFormationが発表した製品 (無料または有料) または使用量ベースの有料 AMI 製品

Note

一部の料金モデルでは、このセクションで説明する製品ロードフォームを使用して CloudFormation 製品で AMI を発行する必要がなくなりました。販売者ポータル[のサーバー製品](#)ページで CloudFormation を使用して Amazon マシンイメージ (AMI) または AMI を作成し、すぐに製品ロードフォームをダウンロードするように求められない場合は、[「AMI ベースの製品の作成」](#)および[CloudFormation テンプレートをリストに追加する](#)」を参照してください。

製品ロードフォーム (PLF) を使用して、AWS Marketplace 顧客が AWS CloudFormation テンプレートを使用して起動する製品を送信します。PLF は AWS Marketplace 管理ポータルから入手できます。

製品を送信するおおまかな手順を以下に示します。

- 料金モデルを選択します。
- 製品ロードフォーム (PLF) (Microsoft Excel スプレッドシート) をダウンロードします。
- 製品ロードフォームに入力します。
- フォームの指示に従って、製品を送信します。

各手順の詳細については、掲載されている順序でセクションを展開します。

料金モデルの選択

製品の料金モデルを選択する必要があります。選択したモデルは、PLF に入力した料金情報をコントロールします。サポートされている料金モデルについては、本ガイドの[「の AMI 製品の料金 AWS Marketplace」](#)を参照してください。

新製品の PLF のダウンロード

1. [AWS Marketplace 管理ポータル](#)を起動します。
2. [アセット] タブでの右側のペインで、[\[CloudFormation 製品を含む単一の AMI\]](#) のリンクを選択します。

フォームがブラウザに表示されます。

3. [ファイルのダウンロード] を選択し、Excel でファイルを編集します。

—または—

Microsoft OneDrive がある場合は、[コピーの編集] を選択します。これにより、PLF が OneDrive に保存され、編集できるようになります。

Note

スプレッドシートには、いくつかのサンプル製品が含まれています。フォームを送信する前に、これらを削除する必要があります。

既存の製品の PLF のダウンロード

フォームをダウンロードするには

- AWS Marketplace Dashboard を起動し、製品ロードフォームのダウンロードを選択します。



スプレッドシートには、以前のバージョンの製品のデータが含まれています。そのデータは変更せず、次の空白の行に新しい製品を追加します。

PLF の入力

以下の手順では、製品ロードフォーム (PLF) を入力する方法について説明します。この手順は、新規および既存の製品に適用されます。

フォームに入力するには

1. [SKU] 列から [返金およびキャンセルポリシー] の列で、製品に関するすべての情報を入力します。

Note

[製品アクセス指示] の列には、詳細で明確な使用方法を入力する必要があります。本ガイドの「[の AMI および コンテナ 製品の 使用手順の作成 AWS Marketplace](#)」に記載されている要件に従ってください。

2. [タイプ] 列から [エンドポイント URL の相対 URL] 列で、AMI の必要な情報を入力します。

Important

AMI は AWS Marketplace と共有する必要があります。これを行うには、本ガイドの「[への AMIs の送信 AWS Marketplace](#)」の手順に従います。

3. エンドユーザーライセンス契約 URL 列には、標準の AWS Marketplace 契約へのリンクが表示されます。この契約を使用したり、使用する EULA へのリンクを入力したりすることができます。リンクを入力する場合は、Amazon S3 バケットからのリンクのように、お客様が EULA をダウンロードできるようにする必要があります。

標準契約の詳細については、本ガイドの「[での標準化された契約の使用 AWS Marketplace](#)」を参照してください。

4. us-east-1 Availability to Make available for all future instance types 列 AWS で、使用する **FALSE** リージョンごとに **TRUE** または を入力します。

Note

GovCloud リージョンには追加の要件があります。例えば、GovCloud リージョンを使用するには、GovCloud アカウントを所有している必要があります。詳細については、「AWS GovCloud ユーザーガイド」の「[Getting set up](#)」を参照してください。

5. [推奨インスタンスタイプ] 列で、推奨インスタンスタイプを受け入れるか、リストから別のインスタンスタイプを選択します。使用するリージョンでインスタンスタイプが使用可能であることを確認します。

 Note

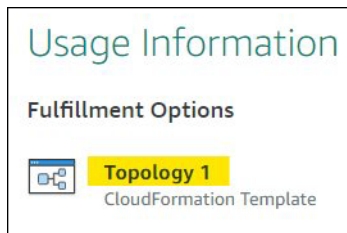
- ほとんどのお客様は、推奨インスタンスタイプを受け入れます。
- インスタンスタイプが製品と同じリージョンで使用可能であることを確認する必要があります。

6. [推奨インスタンスタイプ] と [含める国] の列で、インスタンスタイプに **TRUE** と入力します。これにより、インスタンスタイプがアクティブ化されます。残りのタイプには **FALSE** と入力します。インスタンスのタイプの詳細については、<https://aws.amazon.com/ec2/instance-types/> を参照してください。
7. [含める国] 列と [除外する国] 列で、含める国または除外する国の 2 文字の国コード (US など) を入力します。
8. [料金モデル] 列に、製品の料金モデルを入力します。

以下のリストで、料金モデルと入力が必要な他の列について説明します。

- BYOL URL – ライセンス URL を入力します。料金情報を入力する必要はありません。
 - 時間単位 – TRUE に設定したインスタンスタイプの料金を入力します。他のすべての列は空のままにします。これに関連する列は、[a1.medium 時間料金] から [z1d.metal 時間料金] までです。
 - 時間単位年間料金 – 前の手順の列に加えて、[a1.medium 年間料金] から [z1d.metal 年間料金] までの列を入力します。TRUE に設定したインスタンスタイプの料金を入力します。その他の列は空白のままでもかまいません。
 - 使用状況 – [FCP カテゴリ] から [FCP Dimension24 レート] の列に使用状況ディメンションに関する情報を入力します。
 - 契約 – [契約カテゴリ] から [契約 Dimension24 36-Month レート] の列に契約ディメンションに関連する情報を入力します。
9. [セキュリティグループルール 1] から [セキュリティグループルール 12] の列に、製品のセキュリティグループに関する情報を入力します。tcp, #, #, 0.0.0.0/0 フォーマットに従って入力してください。例えば、SSH には tcp, 22, 22, 0.0.0.0/0 を使用し、RDP には tcp, 3389, 3389, 0.0.0.0/0 を使用します。
 10. クラスターと AWS リソーストポロジ 1: クラスターとリソーストポロジ 3: アーキテクチャ図 URL AWS 列に、製品の CloudFormation データを入力します。次のデータを入力する必要があります。

- トポロジータイトル – デプロイまたはフルフィルメントオプションのタイトル。タイトルは、[フルフィルメントオプション] セクションの製品の詳細ページに表示されます。例:



- [料金の見積もり] 列に、値を含む [AWS 計算ツール](#)へのリンクを入力します。
- 短い説明と長い説明 – デプロイオプションの説明を入力します。
- テンプレート URL – Cloudformation テンプレートをダウンロード可能なリンクを提供します。
- アーキテクチャ図 – CloudFormation のトポロジアーキテクチャ図をダウンロード可能なリンクを提供します。各デプロイオプションには、スタックの起動を示す個別の図が必要です。図は、[アーキテクチャ図](#) に記載されている要件に従う必要があります。

PLF の送信

次の手順では、入力した PLF を送信する方法について説明します。

1. [AWS Marketplace 管理ポータル](#)にサインインします。
2. [アセット] タブで、[ファイルのアップロード] を選択します。
3. [ファイルのアップロード] ページで、PLF と AWS CloudFormation テンプレートをアップロードします。ファイルアップローダーは、安全な転送メカニズムと送信されたファイルの履歴を提供します。アップローダーは自動的に通知し AWS Marketplace、ポリシーとセキュリティのコンプライアンス、ソフトウェアの脆弱性、製品の使いやすさについて送信内容を確認します。リクエストについて質問や問題がある場合、チームは E メールメッセージを送信します。

製品の更新

製品ロードフォーム (PLF) を使用して作成した製品の場合は、PLF を使用して製品に変更を加えます。完了した元の PLF を変更することができます。利用できない場合は、新しい PLF から開始することもできます。[製品] タブを使用するのと同様に、新しいバージョンの追加や既存のバージョンの削除をしたり、さらに料金、インスタンスタイプ、リージョンの可用性、メタデータを更新すること

ができます 更新を行うには、新製品の場合と同じ方法で、更新された製品を準備します。製品アップデートの準備が整ったら、以下のステップに従います。

1. 既存の PLF を使用するか、[アセット] タブの、[AWS Marketplace 管理ポータル](#) から [ファイルのアップロード] を選択します。[製品積載フォームおよび販売者ガイド] で、製品の PLF をダウンロードできます。
2. PLF で製品を更新します。
3. [AWS Marketplace 管理ポータル](#) の [アセット] タブで、[ファイルのアップロード] をクリックします。
4. [ファイルのアップロード] ページで、更新した PLF と AWS CloudFormation テンプレートをアップロードします。ファイルアップローダーは、安全な転送メカニズムと送信されたファイルの履歴を提供します。アップローダーは、リクエストの処理を開始するように AWS Marketplace チームに自動的に通知します。送信の説明を追加します (新しいバージョンの追加、価格の変更、メタデータの変更など)。

製品の送信は、ポリシーとセキュリティのコンプライアンス、ソフトウェアの脆弱性、および製品のユーザビリティについて確認されます。リクエストについて質問や問題がある場合、AWS Marketplace チームは E メールメッセージで連絡します。既存の製品ページに対する更新は、追加の確認を受けずに直接処理されリリースされます。

製品の変更と更新

販売者はいつでも製品の変更を送信することができ、前述のように処理されます。ただし、90 日または 120 日ごとに、または変更が保留中の場合にのみ変更を加えることができます。例としては、料金変更 AWS リージョン やインスタンスタイプの変更などがあります。一般的な変更は次のとおりです。

- 新バージョン - ソフトウェアの新バージョン、パッチまたはアップデートのロールアウト。お客様のリクエストに応じて、新しいバージョンの可用性について AWS Marketplace コンテンツをサブスクライブしているお客様に通知したり、お客様に代わってアップグレード手順を送信したりできます。
- メタデータの変更 - 製品情報の変更 (説明、URL、および使用方法)。
- 料金の変更 - 料金設定額の変更。リクエストが完了した後に、現在のお客様への通知が送信されます。通知が送信されると、料金変更は 90 日間の期間の翌月の 1 日に有効になります。例えば、3 月 16 日に変更を行うと、90 日後の 6 月 16 日ごろになりますが、料金変更は翌月の初日に行われます。実際の変更日は 7 月 1 日です。

- 料金モデルの変更 - 料金モデルの変更 (例えば、毎時、無料、年間で毎時)。すべての料金モデルの変更がサポートされているわけではなく、モデルを変更するリクエストはすべて AWS Marketplace チームで確認および承認されなければなりません。無料モデルから有料モデルへの変更は、既存のお客様に大きな影響を与えます。もう 1 つの選択肢は、追加機能を備えた新しい製品を提案し、現在のお客様に移行を促すことです。
- リージョンまたはインスタンスの変更 - インスタンスのタイプやリージョンの追加または削除。
- 製品のテイクダウン - から製品ページを削除 AWS Marketplace して、新規顧客がサブスクライブしないようにします。リクエストが完了した後に、現在のお客様への通知が送信されます。

タイミングと心構え

可能な限り迅速にリクエストを処理するように努めていますが、リクエストには複数の反復と販売者と AWS Marketplace チームによるレビューが必要になる場合があります。プロセスの完了に必要な時間のガイダンスについては、以下を参照してください。

- 通常、合計リクエスト時間はカレンダー時間の 2~4 週間です。複数の繰り返しと製品メタデータおよびソフトウェアの調整により、リクエストや製品が複雑なほど時間がかかることがあります。
- Amazon では、リクエストの優先順位をつけるために、計画されているイベントまたはリリースの少なくとも 45 日前に、完了した製品リクエストと AMI が必要です。

リクエストに関するご質問がございましたら、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

への AMIs の送信 AWS Marketplace

に構築および送信されるすべての AMIs AWS Marketplace は、すべての製品ポリシーに従う必要があります。送信する前に、AMI の最終チェックをいくつかお勧めします。

- システムから、すべてのデフォルトのパスワード、認可キー、キーペア、セキュリティキー、またはその他の認証情報など、すべてのユーザー認証情報を削除します。
- ルートログインが無効になっている、またはロックされていることを確認します。sudo アクセスアカウントのみが許可されます。
- AWS GovCloud (US) リージョンにデプロイする AMI を提出する場合は、[アクティブな AWS GovCloud アカウントを持つ](#)ことが必要で、該当する輸出管理要件などの [AWS GovCloud の要件](#)に同意する必要があります。

AMI セルフサービススキャン

AWS Marketplace は、内で AMI スキャンを提供します AWS Marketplace 管理ポータル。新製品の場合、システムはスキャンを自動的に実行し、送信時に結果を提供します。既存の製品の新しいバージョンについては、「バージョンの追加」のテスト機能を使用してスキャンを開始します。このプロセスは通常 1 時間未満で完了し、単一の便利な場所で包括的なフィードバックを提供します。

この新しいセルフサービススキャンで AMI を共有してスキャンするには、次のようにします。

1. <https://aws.amazon.com/marketplace/management/products/server> に移動します。
2. 製品タイトル列で、製品を選択します。
3. 製品詳細ページの「バージョン」セクションで「バージョンの追加」のテストを選択します。
4. 「バージョンの追加」ページで、必要なバージョン情報と検証パラメータを入力します。
5. [テストの実行] を選択します。
6. スキャン結果を表示します。

AMI が正常にスキャンされたら、現在のプロセスに従って、製品ロードフォーム (PLF) [をアップロード](#)して AWS Marketplace Seller Operations チームに送信できます。問題がある場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

AMI をセルフサービススキャンリストに含めるには、AMI が us-east-1 (バージニア北部) リージョンにあり、AWS Marketplace 販売者アカウントによって所有されている必要があります。他のアカウントにへのアクセスを許可する必要がある場合は AWS Marketplace 管理ポータル、それらのアカウントを販売者として登録する必要があります。詳細については、「[AWS Marketplace 販売者としての登録](#)」を参照してください。

AMI クローン作成と製品コード割り当て

AMI が送信されると、はソフトウェアが利用可能であることを示すリージョンごとにクローンされた AMIs AWS Marketplace を作成します。このクローン作成および発行プロセス中に、AWS Marketplace はクローン AMI に製品コードをアタッチします。製品コードは、アクセスの制御と使用量の測定の両方に使用されます。すべての送信は、この AMI クローン作成プロセスを経なければなりません。

最終チェックリスト

製品の公開が遅れるのを防ぐには、製品リクエストを送信する前にこのチェックリストを使用してください。

製品の使用

- 本稼働の準備完了。
- 時間またはその他の制限による製品の使用を制限しません。
- 1-Click 履行体験との互換性。
- クライアントアプリケーションを含め、製品を使用するために必要なものはすべてソフトウェアに含まれます。
- デフォルトのユーザーはランダム化されたパスワードを使用し、初期ユーザーの作成では、インスタンス ID などのインスタンスに固有の値を使用してインスタンスを使用する権限が購入者に付与されていることを確認する必要があります。

無料製品にするか、有料製品にするか

- 製品を使用するために追加のライセンスは必要ありません。
- 有料の再パッケージソフトウェアは を満たしています AWS Marketplace [有料の再パッケージ版ソフトウェアを送信するための要件](#)。
- 購入者は、製品を使用するために個人を特定できる情報 (E メールアドレスなど) を提供する必要はありません。

AMI の準備

- ハードウェア仮想マシン (HVM) の仮想化と 64 ビットアーキテクチャを活用します。
- 既知の脆弱性、マルウェア、またはウイルスは含まれていません
- 購入者には、AMI へのオペレーティングシステムレベルの管理アクセス権があります。
- でテスト「バージョンの追加 AWS Marketplace 管理ポータル」を使用して AMI をスキャンします。

Windows AMI の場合

- 「Amazon EC2 ユーザーガイド」の「[EC2Config サービスを使用して Windows インスタンスを設定する](#)」で説明している Ec2ConfigService の最新バージョンを使用します。
- Ec2SetPassword、Ec2WindowsActivate、Ec2HandleUserData プラグインは、「Amazon EC2 ユーザーガイド」の「[EC2Config サービスを使用して Windows インスタンスを設定する](#)」で説明されているように有効になっています。
- ゲストアカウントまたはリモートデスクトップのユーザーアカウントが存在しないことを確認します

Linux AMI の場合

- ルートログインはロックされ、無効になっています。
- 承認されたキー、デフォルトのパスワード、またはその他の認証情報が含まれません。
- すべての必須フィールドが入力されています。
- すべての値が指定された文字制限内にあります。
- すべての URL がエラーなしで読み込まれます。
- 製品イメージは、少なくとも 110 ピクセル幅で、1:1 ~ 2:1 の比率です。
- 料金は、有効なすべてのインスタンスタイプ (料金モデルが hourly、hourly_monthly、および hourly_annual) で指定されます。
- 月単位料金設定が指定されています (hourly_monthly および monthly pricing モデルの場合)。

自動 AMI 構築に関する質問やコメントがある場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

での製品のマーケティング AWS Marketplace

製品の認識を高め、製品ページに直接トラフィックを誘導することで AWS Marketplace、製品の成功に貢献できます AWS Marketplace。以下のセクションでは、AWS Marketplaceに出品した製品の販売に役立つ情報とサポートを提供します。

トピック

- [180 日間の GTM アカデミー](#)
- [製品が入手可能になったことの発表](#)
- [AWS Marketplace メッセージング](#)
- [のレビュー AWS Marketplace](#)
- [へのリンク AWS Marketplace](#)
- [AWS Marketplace ブランディングの使用](#)
- [販売 AWS 者として での購入について](#)
- [で製品に直接リンクする AWS Marketplace](#)
- [プレスリリース](#)
- [AWS Marketplace 商標使用ガイドライン](#)

180 日間の GTM アカデミー

180 日間の GTM アカデミーは、すべての AWS Marketplace 販売者が利用でき、セルフサービス go-to-market (GTM) リソースを提供し、のサービスに対する需要生成キャンペーンの構築、アクティブ化、追跡を支援します AWS Marketplace。次のようにできます：

- リスティングの需要を迅速に追跡できます。
- と AWS Marketplace 価値提案をメッセージングに統合することで、マーケティング投資収益率を向上させ AWS、カスタマーメッセージングを強化します。
- [AWS Marketplace 販売促進プログラムガイドプログラムガイド](#)の達成に向けて、またはその範囲内で進めることができます。

180 日間の GTM アカデミーには、[AWS Marketplace 管理ポータル](#) の [Marketplace リソース] セクションからアクセスできます。

製品が入手可能になったことの発表

製品の可用性を広く発表することをお勧めします AWS Marketplace。これは、プレスリリース、ツイート、ブログ、その他の任意のメディアチャネル経由で行うことができます。含まれる可能性のあるサンプルテキストと、商標の使用に関するガイドラインと指示、およびプレスリリースの発行について説明しました。

公開する前に、ブログ、ツイート、その他のプレスリリース以外の発表を確認し、AWS メッセージングやブランドのガイドラインや音声との一貫性を確保します。レビューのリクエストを AWS アカウントマネージャーに送信します。レビューが完了するまでに、最大で 10 営業日かかります。ツイート、ブログ、またはプレスリリースを投稿したらお知らせください。その可視性を高めるために再掲載できるよう最善を尽くします。

AWS Marketplace メッセージング

顧客とのコミュニケーションでは、AWS Marketplaceを使用して製品を購入する目的、目標、および利点について説明することができます。AWS Marketplaceを参照するときは、次のメッセージを使用します。

とは AWS Marketplace

AWS Marketplace は、で実行されるソフトウェアとサービスを簡単に検索、比較、すぐに使用を開始できるオンラインストアです AWS。の訪問者 AWS Marketplace は 1-Click デプロイを使用して、事前設定されたソフトウェアをすばやく起動し、使用した分だけを時間単位または月単位で支払うことができます。は請求と支払い AWS を処理し、ソフトウェア料金はお客様の AWS 請求書に表示されます。

顧客が を購入する理由 AWS Marketplace

ソフトウェアの検索とデプロイは難しい場合があります。AWS Marketplace は、データベースやアプリケーションサーバー、IoT ソリューション、デベロッパーツール、ビジネスアプリケーションなどのソフトウェアインフラストラクチャを含む、商用および無料の IT およびビジネスソフトウェアを幅広く提供しています。AWS Marketplace のお客様は、オプションの比較、レビューの読み取り、必要なソフトウェアをすばやく見つけることができます。その後、1-Click または AWS Marketplace 管理ポータルを使用して、独自の Amazon Elastic Compute Cloud インスタンスにデプロイできます。

ソフトウェア価格は Web サイトに明確に掲載されており、既にアマゾン ウェブ サービスに登録されている支払い方法で、お客様はほとんどのソフトウェアを即座に購入することができます。ソフトウェア料金は、AWS インフラストラクチャ料金と同じ月額請求書に表示されます。

ソフトウェアまたは SaaS 販売者が販売する理由 AWS Marketplace

では AWS Marketplace、で実行されるサービスを持つ Software and Software as a Service (SaaS) 販売者は、顧客意識の向上、デプロイの簡素化、自動請求の恩恵 AWS を受けることができます。

AWS Marketplace は、で実行されるソフトウェアおよびサービスのソフトウェアおよび SaaS 販売者が、個々のソフトウェア開発者から大企業まで、数十万の顧客に製品を公開することで、新しい AWS 顧客 AWS を見つけるのに役立ちます。

を販売 AWS Marketplace することで、独立系ソフトウェアベンダー (ISVs) は、コストのかかるコード変更を行わずに、ソフトウェアの時間単位の請求を追加できます。Amazon マシンイメージ (AMI) をにアップロード AWS し、時間単位のコストを提供するだけです。請求はによって管理され AWS Marketplace、販売者は使用量の計測、顧客アカウントの管理、支払いの処理の責任を解放されるため、ソフトウェア開発者は優れたソフトウェアの構築に集中する時間が長くなります。

さらに、お客様にとっては、ソフトウェアの事前構成済みイメージを簡単にデプロイして、新規お客様のオンボーディングを簡素化することができます。

のレビュー AWS Marketplace

AWS Marketplace は、顧客が製品に関するレビューを送信できるようにします。また、独立した製品レビューをキュレートする企業間マーケットプレイスである G2 などのシンジケートレビュー担当者が、同社のシンジケートレビューを AWS Marketplace に統合する機能も提供します。

AWS Marketplace カスタマーレビューは、購入者向けユーザーガイドに記載されているレビューガイドラインを満たす必要があります。レビュー送信は、レビュー基準を満たしていることを確認するために送信がレビュー AWS Marketplace されるまでにリリースされません。レビューガイドラインについて詳しくは、「[製品レビュー](#)」を参照してください。シンジケートレビュー組織は独自のプロセスを使用してレビューを検証し、リリース AWS Marketplace 前にによってレビューされません。製品の配信レビューが商品レビューガイドラインを満たしていないと思われる場合や、商品のレビューに好ましくないコンテンツが含まれていると思われる場合は、[AWS Marketplace Seller Operations](#) にお問い合わせください。

レビュー担当者は、5 つの星評価システムに基づいて製品の星評価を提供することもできます。評価は、製品の全体的な星評価を算出するために平均されます。シンジケートレビューには星の評価を含

めることもできますが、シンジケートレビューの星の評価は、AWS 顧客の星の評価では平均化されません。

製品レビュー機能に関するその他の重要なポイントを以下に示します。

- から製品レビューを削除することはできません AWS Marketplace。ただし、コメントが製品レビューを管理するレビュー基準を満たしている限り、任意のレビューにコメントを残すことができます。
- レビューがレビューのガイドラインを満たしていないか、好ましくないコンテンツが含まれていると思われる場合は、[AWS Marketplace Seller Operations](#) にお問い合わせになり、懸念事項を説明してください。
- AWS で製品を検索するお客様は、評価、検証済みレビュー、外部ソースレビューに基づいて結果を検索およびフィルタリング AWS Marketplace できます。 の AWS お客様は、検索結果に外部ソースの評価と AWS 顧客評価が表示されます。
- 製品のシンジケートレビューは AWS Marketplace 、無料で自動的に追加されます。レビューは自動的に追加されるため、シンジケートレビューを追加するためにリクエストを送信する必要はありません。
- 製品のシンジケートレビューがない場合は、シンジケートレビュー担当者に連絡し、そのプロセスに従って製品のレビューを受けることができます。たとえば G2 では、そのウェブサイトアクセスし、製品ページに移動してレビュープロセスを開始できます。

へのリンク AWS Marketplace

企業のウェブで、製品について説明し、促進する存在感を出すことができます。製品は で実行でき AWS 、 を使用して購入できることを強調することをお勧めします AWS Marketplace。お客様がソフトウェアを発見してデプロイするプロセスを簡略化するため、お客様の製品を結びつけるための手順について説明しました。

AWS Marketplace ブランディングの使用

Note

ここにリストされているガイドとアセットを使用するには、AWS パートナーである必要があります。AWS パートナーになる方法については、[AWS 「パートナーネットワークに参加する」](#) を参照してください。

AWS Marketplace は、ソフトウェア、データ、またはサービスの公開に役立つメッセージングガイダンスとブランドアセットを提供します。出品を昇格させる場合は AWS Marketplace、次のリソースをダウンロードできます。

- AWS Partner Creative and Messaging Guide 2023
- ブランドアセットで利用可能 AWS Marketplace

リソースをダウンロードするには

1. [AWS パートナーセントラル](#)にサインインします。
2. ブラウザのアドレスバーに次の URL を貼り付け、ページを開きます: [AWS Partner Creative and Messaging Guide 2023](#)。
3. [ダウンロード] を選択します。
4. ブランドアセット [で利用可能な AWS Marketplace](#) に対して、ステップ 2 と 3 を繰り返します。

販売 AWS 者として での購入について

で購入 AWSすると、AWS Marketplace 販売者は独自のウェブサイトで AWS Marketplace 製品を表示および販売できます。AWS Marketplace 販売者は AWS Marketplace call-to-action (CTA) ボタンや APIs と統合して、販売者所有のウェブサイトや AWS 強力なソフトウェア検出と調達エクスペリエンスを構築できます。これには、独立系ソフトウェアベンダー (ISVs)、チャネルパートナー (CPs)、ディストリビューター (DPs) など、AWS 販売者が所有しているウェブプロパティが含まれます。

Note

AWS 機能での購入の詳細については、「[での購入 AWS](#)」を参照してください。

CTA ボタンには以下が含まれます。

- で購入する AWS
- でオファーを表示する AWS
- で無料で試す AWS
- プライベートオファーをリクエストする
- デモをリクエストする

で購入 AWS すると、次の利点があります。

- AWS ウェブサイトでの共同ブランドと検出: ウェブサイトで、 を活用した共同ブランドエクスペリエンスを作成します AWS Marketplace。顧客は、ウェブサイト AWS Marketplace の から厳選された製品リストを検索したり、製品をフィルタリングしたり、レビューにアクセスしたり、カスタム料金をリクエストしたり、アカウントを使用して自信を持って購入したりできます AWS。
- AWS 顧客の調達速度: ウェブサイトの Buy with AWS ボタンを使用すると、顧客は無料トライアルにすばやくサブスクライブしたり、AWS アカウントを使用して購入したりできます。で購入 AWS すると、 を通じて合理化され、共同ブランド化された購入エクスペリエンスがお客様に提供されます AWS Marketplace。
- で既存のオペレーションを活用する AWS Marketplace: ウェブサイトを通じて の検出と調達の利点 AWS Marketplace をお客様に拡張し、 の既存のプロセスを活用して、追加の労力をかけずに出品、支払い、サポート、インサイト AWS Marketplace を管理します。

AWS 前提条件付きで購入する

で購入を有効にするには AWS、以下を実行する必要があります。

- で [販売者として登録](#)されている AWS Marketplace。
- の認可された販売者またはリセラー AWS Marketplace であり、[販売者のサービス条件 AWS Marketplace](#)に同意します。
- [AWS 商標ガイドライン](#)に従ってください。
- [AWS 商標ライセンスによる購入](#)に従ってください。
- に少なくとも 1 つのパブリック SaaS 製品出品がある AWS Marketplace。

Note

現在、での購入は SaaS 製品のみ AWS をサポートしています。

- Buy with AWS branding ガイドラインに従ってください。
- [AWS Partner Central](#) アカウントにログインして、ブランド要素とマーケティングアセットについて従うべき AWS Marketplace ブランドアセットと仕様を含むマーケティングツールキットをダウンロードします。
- [ブランド要素とマーケティングアセットの使用手順については、AWS クリエイティブおよびメッセージングでの購入ガイドライン](#)を参照してください。

[AWS での購入: 販売者としての開始方法](#)

ブランドの変更を確認する

Buy with AWS experience で起動する前に、ブランドレビューリクエストを送信するには、次のステップを実行します。

1. Commercial Marketplace に移動 AWS Marketplace 管理ポータル して [サポートチケットを送信する](#)
 - a. カテゴリとして で購入 AWSを選択する
 - b. サブカテゴリとしてブランドレビューを選択します。
2. チケットに以下を入力します。
 - a. プライマリ E メールアドレスと販売者 ID。
 - b. 「リクエストの詳細を入力する」セクションで、次の操作を行います。
 - i. AWS 統合で購入を実装する URL
 - ii. AWS Marketplace オペレーションチームが確認するためのサンドボックス、ライブサイト、モックアップへのリンク。これを含めない場合、チケットの送信後に運用チームがリクエストします。
3. AWS Marketplace 販売者は、AWS ブランドガイドラインに従って Buy と照らし合わせて実装を評価します。コンプライアンスを確保するためのフィードバックを提供します。起動する前に、すべてのフィードバックに対処し、承認を得る必要があります。

AWS レポートで購入する

AWS メトリクスで購入すると、販売者はコンバージョンファネルを包括的に把握できます。AWS 販売者による購入ダッシュボードは、ユーザーエンゲージメントを測定するための詳細なインサイトを提供します。販売者は、 を通じて AWS Buy with metrics にアクセスできます [AWS Marketplace 管理ポータル](#)。

AWS メトリクスでの購入ダッシュボードを表示するには：

1. [AWS Marketplace 管理ポータル](#) にログインします。
2. Insights に移動します。
3. Marketing をクリックします。

4. で購入 AWS をクリックします。

または、アカウントをリンクした販売者は、Partner Central を通じて AWS メトリクスにアクセスできます。これらのメトリクスを表示するには、[AWS Partner Central](#) にログインし、分析セクションに移動します。

で製品に直接リンクする AWS Marketplace

ウェブサイトまたはマーケティング資料にディープリンクを追加して、特定の製品ページに顧客を誘導します AWS Marketplace。以下のフォーマットを使用します。

```
https://aws.amazon.com/marketplace/pp/prodview-prodview ID
```

prodview ID を製品の一意の英数字コードに置き換えます AWS Marketplace。例:

Example

```
https://aws.amazon.com/marketplace/pp/prodview-1234567890abcdef
```

[aws.amazon.com/marketplace](#) でアプリケーションを検索すると、URL に prodview ID が表示されます。prodview ID の検索については、アカウントマネージャーにお問い合わせください。

Note

リンクを使用する前にテストして、お客様が正しいページに移動できることを確認します。

プレスリリース

任意のチャネル AWS Marketplace を通じて、での製品の可用性を発表することをお勧めします。ただし、参照するすべてのプレスリリースは、公開または発表を行う前に、Amazon によってレビューおよび承認 AWS Marketplace される必要があります。発表することをお勧めしますが、AWS Marketplace 販売者との共同プレスリリースをサポートすることはできません。case-by-case で、からの引用を含むプレスリリースをサポートします AWS。見積りは、にリストされている新しい製品やサービスを発表 AWS Marketplace するか、が使用する顧客リファレンスを含むなど、いくつかの条件を満たす必要があります AWS Marketplace。

すべてのプレスリリースのドラフトは、お客様が作成しなければなりません。次の見出しをお勧めします。〔製品名を挿入〕が利用可能になりました AWS Marketplace。整合性を保つために、このドキュメントのメッセージを使用します。

プレスリリースは、以下のようにする必要があります:

- 発表が Amazon.com にどのように関連しているかを明確かつ正確に説明する
- 顧客と AWS のロールを明確にする
- お客様を重視し、お客様の利益を高める

プレスリリースは、以下のことは禁止です:

- 関係を表す際、パートナー、パートナーシップ、アライアンス、「協業」、「提携」などの用語を使用する。合意、連携、関係をお勧めする。
- 事前合意なしに、アマゾン ウェブ サービス エグゼクティブからの引用を含めます。
- 販売予測を含めるか、企業共通のウェブサイトを参照する場合を除き、出品者ごとに「.com」を使用する。
- Amazon.com の「アソシエイト」としてお客様の組織を参照する。これは、オンラインアフィリエイトプログラム [の Amazon Associates](#) と混同する可能性があるためです。
- Amazon.com に関する専有情報を開示するか、弊社の株式ティッカーシンボルを参照する。

プレスリリースをテキスト形式でアカウントマネージャーに送信して、レビューしてもらいます。さらに、商標を使用する前に、[Amazon Web Services の商標ガイドライン](#)を確認してください。AWS Marketplace 商標に固有のガイドラインは、以下のセクションに記載されています。

AWS Marketplace 商標使用ガイドライン

本ガイドラインは、Amazon.com, Inc. および/またはその関連会社 (「Amazon」) によって事前に承認されたマテリアルにおける AWS Marketplace ロゴおよび商標 (各「商標」および総称して「商標」) の使用に適用されます。本ガイドラインはどのようなときにも厳格に厳守していただくものとし、本ガイドラインに違反して商標を使用すると、商標の使用に関連するライセンスは自動的に解除されます。

1. Amazon が明示的に承認した目的のためにのみ商標を使用することができ、お客様は、(i) お客様が商標を使用することに関する Amazon とのすべての契約の合意事項の最新版 (総称して「契約

- 書」) を遵守、(ii) 本ガイドラインの最新版を遵守、および (iii) Amazon が商標の使用に際して随時発行する可能性のあるその他の利用規約、条件、またはポリシーを遵守します。
2. 弊社はお客様が使用できる承認された商標イメージを提供します。お客様は、以下を含みますがこれらに限定されない、商標の割合、色、フォントの変更、商標の追加や削除など、どのような方法によっても商標を変更することはできません。
 3. 本契約で特別に許可された商標を使用する以外に、Amazon による後援または裏書を意味するような方法で商標を使用することはできません。
 4. Amazon、その製品またはサービスを誹謗するために商標を使用することはできません。また商標の Amazon の信用をおとしめたり、損なったり、傷つけたりする方法を使用することはできません。
 5. 商標は、商標の各面とその他のビジュアル、グラフィックまたはテキストの要素間に適度な間隔を置いて、単独に表示されなければなりません。どのような場合でも、商標は、商標の可読性または表示を妨害するような背景に置かれてはなりません。
 6. 商標を表示するマテリアルには、AWS Marketplace 「および AWS Marketplace ロゴは Amazon.com, Inc. またはその関連会社の商標です。
 7. お客様は、商標に対するすべての権利は Amazon の独占的財産であり、商標の使用を通じて構築されたすべての信用は、Amazon のメリットのために効力が生ずることを認めます。商標に関する Amazon の権利または所有権と矛盾するどのような措置も講じません。

Amazon は、単独の裁量で行使可能な権利を有し、これらのガイドラインや承認された商標をいつでも変更し、アクセス許可なく、または本ガイドラインに準拠しない使用に対しては適切な措置を講じる権利を留保します。このガイドラインについてご不明な点がございましたら、trademarks@amazon.com までお問い合わせください。または、書面で以下までお問い合わせください。

Amazon.com, Inc., Attention: Trademarks

PO Box 81226

Seattle, WA 98108-1226

AWS Marketplace イベントの販売者通知

AWS Marketplace 製品に関連する通知は、E メール、Amazon EventBridge イベント、Amazon Simple Notification Service (Amazon SNS) トピックを通じてタイムリーに受信できます。AWS Marketplace 例えば、プライベートオファー、プロフェッショナルサービス製品のリクエスト、定期的なスキャンリマインダーに関連する通知を受け取ることができます。このトピックでは、利用可能な通知とイベントオプションの概要について説明します。

- E メール通知 – 内では AWS Marketplace、販売者、購入者、独立系ソフトウェアベンダー (ISVs) は E メール通知を受信できます。はルートアカウント AWS Marketplace を使用して、自動 E メールを AWS アカウントの E メールにリアルタイムで送信し、AWS Marketplace オファーと契約を更新します。通知のカスタム E メールエイリアスを追加したり、E メール通知の受信者のサブスクリプションを解除したりすることもできます。詳細については、「[AWS Marketplace イベントの E メール通知の管理](#)」を参照してください。
- Amazon SNS トピック – 製品のカスタマーサブスクリプションの変更に関する通知を受け取るには、製品の作成時に AWS Marketplace 提供される の Amazon SNS トピックをサブスクライブできます。例えば、顧客がプライベートオファーをいつ受け入れるかを知ることができます。詳細については、製品タイプの Amazon SNS トピックを参照してください。
 - [Software as a service \(SaaS\) 製品](#)
 - [Amazon マシンイメージ \(AMI\) 製品](#)
 - [コンテナ製品](#)
- EventBridge – EventBridge を使用して AWS Marketplace、オファーの作成時など、で発生するアクションのイベントを受信できます。イベントには、ID、有効期限、製品の詳細などの情報が含まれます。詳細については、[Amazon EventBridge イベント](#) および「[Amazon EventBridge ユーザーガイド](#)」を参照してください。

以下のトピックでは、AWS Marketplaceの通知とイベントの詳細について説明します。

トピック

- [AWS Marketplace イベントの E メール通知の管理](#)
- [Amazon EventBridge イベント](#)

AWS Marketplace イベントの E メール通知の管理

AWS Marketplace は、オファー、契約、サブスクリプション、製品、セキュリティ、請求と支払い、および Private Marketplace の更新に関する E メール通知を送信します。独立系ソフトウェアベンダー (ISVs)、AWS Marketplace チャンネルパートナー、顧客は E メール通知を受け取ることができます。オファーと契約の更新のために送信される E メール通知の例と詳細については、「[イベントタイプ](#)」を参照してください。

AWS Marketplace は、AWS アカウントの[ルートユーザー](#)に関連付けられた E メールアドレスに E メール通知を送信します。AWS アカウントに関連付けられた E メールアドレスを更新するには、[AWS「アカウントの主要連絡先の更新」](#)を参照してください。通知の[カスタム E メールエイリアスを追加](#)したり、E メール通知の[受信者のサブスクリプションを解除](#)したりすることもできます。

Note

E AWS Marketplace メールが見つからない場合は、スパムフォルダを確認するか、E メール設定を調整します。からの E メール通知 AWS Marketplace は から送信されます no-reply@marketplace.aws。通知メールは、Google や Yahoo などのプロバイダーによってフィルタリングされる場合があります。手順については、「[正当なメールが迷惑メールに振り分けられないようにする \(Google\)](#)」、または「[Block and unblock email addresses in Yahoo Mail](#)」を参照してください。

トピック

- [イベントタイプ](#)
- [フィールドの説明](#)
- [通知の管理](#)

イベントタイプ

以下のイベントタイプは、機械学習製品を除くすべての製品および価格タイプの E メール通知でサポートされています。

オファー

次の表にオファーイベントを示します。オファーは、購入者による製品の使用に関する一連の条件です。詳細については、「[AWS Marketplace 製品のプライベートオファーの準備](#)」を参照してください。

イベント	イベント	受取人	タイトル	フィールド
オ ファー 公開 CP 通知 CPPO	AWS Marketplace チャネル パートナー のプ ライベ ートオ ファーの 公開。	チャネ ルパー トナー	AWS Marketplace チャネル パート ナー のプ ライベ ートオ ファーが 公開さ れました	顧客 AWS アカウント ID、ISV 名、ISV AWS アカ ウント ID、製品名、製品 ID、オファー ID、オポチュ ニティ ID、オファー公開日、オファーの有効期限
オ ファー 公開 ISV 通知 CPPO	AWS Marketplace チャネル パート ナー のプ ライベ ートオ ファーの 公開。	ISV	AWS Marketplace チャネル パート ナー のプ ライベ ートオ ファーが 公開さ れました	顧客 AWS アカウント IDs、チャネルパートナー、 チャネルパートナー AWS アカウント ID、製品名、 製品 ID、オファー名、オファー ID、オポチュニティ ID、オファー公開日、オファーの有効期限、卸売価 格

Eメール	イベント	受取人	タイトル	フィールド
OfferPublicationSCPv1.0	販売者は AWS Marketplace チャネル パートナーの プライベート オファーを 発行し ます。	ISV または AWS Marketplace チャネル パートナー	プライベート オファーの 公開	顧客 AWS アカウント IDs、製品名、製品 ID、オファー名、オファー ID、オファー公開日、オファーの有効期限、合計契約価値
リセラーのオポチュニティの有効期限切れ	AWS Marketplace チャネル パートナーの 販売承認の有効期限が切れます。	AWS Marketplace チャネル パートナー	オファー: 期限切れの 販売承認の表示	ISV、製品名、製品 ID

Ｅメール	イベント	受取人	タイトル	フィールド
リセラーのオポチュニティの取り消し	ISV は AWS Marketplace チャネルパートナーの販売承認を取り消します。	AWS Marketplace チャネルパートナー	オファー: 非アクティブ化された販売承認の表示	ISV、製品名、製品 ID
リセラーオポチュニティの作成	ISV は、AWS Marketplace チャネルパートナーの販売承認または機会を作成します。	AWS Marketplace チャネルパートナー	オファー: 新しい販売承認のプライベートオファーを作成する	ISV、製品名、製品 ID、販売認可期間

契約

ISVsと AWS Marketplace チャネルパートナーは、購入者がパブリックオファー、プライベートオファー、または AWS Marketplace チャネルパートナーのプライベートオファーを受け入れると、E メール通知を受け取ります。オファーが承諾されると、契約が作成されます。E メール通知は、契約ライフサイクルイベントにも送信されます。これには、購入者の承諾後の契約の開始、キャンセル、

交換、障害が含まれます。次の表に契約イベントに送信される E メールを示します。E メール通知は現在、一部の製品タイプで利用できます。特定の製品タイプにはまだこの機能がない場合があります。

E メール	イベント	受取人	タイトル	フィールド
オファーの承諾 – プライベート オファー または プリックス オファー	AWS Marketplace 顧客は プリックス オファー または プライベート オファーを受け入れます。	ISV	お客様は AWS Marketplace オファーを承諾しました。	お客様の会社名、お客様のアカウント AWS ID、製品名、製品 ID、契約 ID、契約開始日、契約終了日、契約承諾日、購入金額
Offer Accepted – AWS Marketplace チャネルパートナーのプライベート オファー。	AWS Marketplace お客様は AWS Marketplace チャネルパートナーのプライベート オファーを受け入れます。	ISV	お客様は AWS Marketplace チャネルパートナーのプライベート オファーを受け入れました。	お客様の会社名、お客様のアカウント AWS ID、チャネルパートナー名、チャネルパートナー AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID、契約開始日、契約終了日、契約承諾日、卸売価格

Eメール	イベント	受取人	タイトル	フィールド
	受け入れます。			
Offer Accepted – AWS Marketplace チャネル パートのプラ イベート オファー。	AWS Marketplace お客様は AWS Marketplace チャネル パートのプラ イベート オファーを受け入れます。	AWS Marketplace チャネル パート ナー	お客様は AWS Marketplace チャネル パートのプラ イベート オファーを受け入れました。	お客様の会社名、お客様の AWS アカウント ID、ISV 名、ISV AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID、契約開始日、契約終了日、契約承諾日、卸売価格、マージン、購入金額

イベント	受取人	タイトル	フィールド
契約開始 - プライベートオファー	ISV	AWS Marketplace 契約が開始されました	顧客 AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID、契約開始日、契約終了日、購入金額

Eメール	イベント	受取人	タイトル	フィールド
契約開始 – AWS Marketplace チャネルパートナーのプライベートオファー	契約またはサブスクリプション製品の AWS Marketplace 契約が AWS Marketplace チャネルパートナーのプライベートオファーから開始されました。	ISV	AWS Marketplace 契約が開始されました。	顧客 AWS アカウント ID、チャネルパートナー名、チャネルパートナー AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID、契約開始日、契約終了日、卸売価格

エメール	イベント	受取人	タイトル	フィールド
契約開始 – プライベートオファー	AWS Marketplace チャネル パートナー のプライベートオファーを承諾すると、契約またはサブスクリプション製品の契約が開始されます。	AWS Marketplace チャネル パートナー	AWS Marketplace 契約が開始されました。	顧客 AWS アカウント ID、ISV 名、ISV AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID、契約開始日、契約終了日、マージン、購入金額

Eメール	イベント	受取人	タイトル	フィールド
契約 キャンセル プライ ベート オファ ーま たはパ ブリッ クオ ファー	プライ ベート オファ ー契約 または パブリ ックオ ファー 契約の キャン セル。	ISV	AWS Marketpla ce 契約 がキャ ンセル されま した。	顧客 AWS アカウント ID、製品名、製品 ID、オ ファー名、オファー ID、契約 ID、契約開始日、契約 終了日
契約 キャン セル – AWS Marketpla ce チャ ネル パート ナー のプラ イベー トオフ アー	AWS Marketpla ce チャ ネル パート ナー のプラ イベー トオフ アー 契約の キャン セル。	ISV	AWS Marketpla ce 契約 がキャ ンセル されま した。	顧客 AWS アカウント ID、チャネルパートナー、 チャネルパートナー AWS アカウント ID、製品名、 製品 ID、オファー名、オファー ID、契約 ID、契約 開始日、契約終了日

Eメール	イベント	受取人	タイトル	フィールド
契約のキャンセルチャネルパートナーのプライベントオフアー	AWS Marketplace チャネルパートナーのプライベントオフアー 契約のキャンセル。	AWS Marketplace チャネルパートナー	AWS Marketplace 契約がキャンセルされました。	顧客 AWS アカウント ID、ISV、ISV AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID、契約開始日、契約終了日
コンプライアンスエラー – 登録販売者	顧客の支払いが失敗したため、AWS Marketplace 契約は失敗します。	オファーを作成した ISV または AWS Marketplace チャネルパートナー	必要なアクション： AWS Marketplace 契約作成の失敗	サブスクライブ AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID

イベント	受取人	タイトル	フィールド
支払いエラー - 登録販売者	顧客の支払いが失敗したため、AWS Marketplace 契約は失敗します。	オファーを作成した ISV または AWS Marketplace チャネルパートナー	AWS Marketplace 契約の支払い失敗 顧客 AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、契約 ID
契約の置き換え - プライベートオファー	AWS Marketplace 契約は別の契約に置き換えられます。	ISV	AWS Marketplace 契約が置き換えられました 会社名、顧客 AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、新しい契約 ID、契約開始日、契約終了日、新しい購入金額
契約の置き換え - チャネルパートナーのプライベートオファー	AWS Marketplace 契約は別の契約に置き換えられます。	ISV	AWS Marketplace 契約が置き換えられました 会社名、顧客 AWS アカウント ID、チャネルパートナー、チャネルパートナー AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、新しい契約 ID、契約開始日、契約終了日、卸売価格

Eメール	イベント	受取人	タイトル	フィールド
契約の置き換え – チャンネルパートナーのプライベートオファー	AWS Marketplace 契約は別の契約に置き換えられます。	AWS Marketplace チャンネルパートナー	AWS Marketplace 契約が置き換えられました	会社名、顧客 AWS アカウント ID、ISV 名、ISV AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、新しい契約 ID、契約開始日、契約終了日、卸売価格、マージン、新しい購入金額

支払い

ISVsとチャンネルパートナーは、銀行口座の詳細が無効であるために支払いが一時停止された場合、Eメール通知を受け取ります。

Eメール	イベント	受取人	タイトル	フィールド
支払いの一時停止 – 無効な銀行口座情報	AWS Marketplace は、銀行口座の詳細が無効であるため、販売者への支払いを一時停止	ISV と AWS Marketplace チャンネルパートナー	支払いの一時停止	リソースARN

Eメール	イベント	受取人	タイトル	フィールド

止します。

Resource ARN フィールドには、無効な銀行口座 Amazon リソースナンバー (ARN) が表示されます。この問題は、AWS Marketplace 管理ポータル (AMMP) に[銀行口座を追加](#)することで修正できます。

銀行口座の詳細の追加

AWS Marketplace 管理ポータル (AMMP) に銀行口座の詳細を追加して、支払いにおける無効な銀行口座 ARN エラーを修正します。

銀行口座の詳細を追加するには

1. AWS Marketplace 販売者アカウント[AWS Marketplace 管理ポータル](#)でサインインします。
2. メニューバーで、**設定**を選択します。
3. **支払い情報**タブを選択します。
4. 「銀行口座」セクションで、「銀行口座の追加」を選択します。
5. 銀行口座の詳細を追加し、正確性を確認します。詳細については、「[ステップ 3: 銀行口座情報を入力する](#)」を参照してください。
6. 購入者から支払いを受け取るには、**支払い方法**で**支払い方法の追加**を選択します。
7. Know Your Customer (KYC) と銀行口座を確認するように求められたら、**検証ステップ**に従います。
8. 変更内容を保存します。

フィールドの説明

次の表に [オファー](#) テーブルおよび [契約](#) テーブルで参照されるフィールドの説明を示します。

フィールド	説明			
顧客の会社名	サブスクリバの会社名。			

フィールド	説明			
顧客 AWS アカウント ID	製品をサブスクライブしている AWS アカウントの ID。			
ISV 名	販売者の会社名。			
ISV AWS アカウント ID	販売者の AWS アカウントの ID。			
製品名	製品のタイトル。			
製品 ID	ソフトウェア製品のわかりやすい一意の識別子。			
オファー名	オファーのタイトル。			
オファー ID	購入者が署名したオファーの識別子。			
オファーの可視性	オファーがパブリック、プライベート、またはエンタープライズ契約オファーのいずれになるか。			

フィールド	説明			
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意の契約データフィードリファレンス。			
契約開始日	お客様の製品サブスクリプションの開始日 (MM-DD-YYYY 形式)。将来の日付の契約の場合、この日付は受理日とは異なる場合があります。			
契約受理日	お客様が製品をサブスクライブした日付 (MM-DD-YYYY 形式)。			
契約終了日	契約の有効期限が切れる日付 (MM-DD-YYYY 形式)。従量制/従量制料金サブスクリプションの場合、この日付は JAN-1-9999 に設定されます。			

フィールド	説明			
契約終了日	契約の有効期限が切れる日付 (MM-DD-YYYY 形式)。従量制/従量制料金サブスクリプションの場合、この日付は JAN-1-9999 に設定されます。			
購入額	契約の合計額と呼ばれる、契約の推定コスト。これは、SaaS、プロフェッショナルサービス、サーバー製品タイプ、契約または年間料金タイプに適用されます。			
チャネルパートナーの会社名	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウント名。			

フィールド	説明			
チャネルパートナー AWS アカウント ID	ISV から製品またはサービスを購入して顧客に再販した AWS Marketplace チャネルパートナーの AWS アカウントの ID。			
卸売価格	製品を AWS Marketplace チャネルパートナーに再販するための ISV からの卸売コスト。			
通貨コード	契約の推定費用に関連付けられたオファー料金の通貨です。			
新しい契約 ID	更新またはアップグレードされた契約の契約 ID。			
オファーの公開日	販売者がオファーを公開した日付。			
オファーの有効期限	オファーの有効期限が切れる日付。			

フィールド	説明			
オポチュニティ ID	登録されたオポチュニティの一意の識別子。			
販売承認期間	販売承認で指定されているように、リセラーが割引を使用してオファーを作成することを許可されている期間。			

通知の管理

以下のトピックでは、イベントの E メール通知を管理する方法について説明します。

E メールアドレスの追加または更新

AWS Marketplace 管理ポータルを使用して、カスタム E メール通知に最大 10 件の E メールアドレスを追加できます。

E メールアドレスを追加または更新するには

1. [AWS Marketplace 管理ポータル](#) にサインインします。
2. [設定] から [通知] タブを選択します。
3. [カスタム通知用 E メール] で [メールアドレスを追加] を選択します。
4. [受信者の詳細] では、[E メールアドレス] フィールドにカスタム E メールアドレスを入力します。
5. (オプション) [新しい受信者を追加] を選択し、別の E メールアドレスを追加します (合計 10 個まで)。
6. [Submit] を選択してください。

通知から受信者登録を解除する

E メールアドレスを削除すると、受信者はカスタム E メール通知の購読を解除できます。

イベント通知から受信者登録を解除するには

1. [AWS Marketplace 管理ポータル](#) にサインインします。
2. [設定] から [通知] タブを選択します。
3. [カスタム通知用 E メール] で [メールアドレスを更新] を選択します。
4. [受信者の詳細] で [削除] を選択して E メールアドレスを削除します。
5. [Submit] を選択してください。

Note

E メール内のリンクを使用して登録を解除することもできます。

Amazon EventBridge イベント

販売者は、Amazon EventBridge を使用して AWS Marketplace のイベントの通知を受信できます。たとえば、オファーの作成 AWS Marketplace 時に からイベントを受け取ることができます。イベントには、ID、有効期限、製品の詳細などの詳細が含まれます。EventBridge は、アプリケーションをさまざまなソースからのデータに接続するために使用できるイベントバスサービスです。詳細については「[Amazon EventBridge ユーザーガイド](#)」を参照してください。以下のセクションでは、EventBridge コンソールの Marketplace カタログサービスのイベントに関する詳細情報を提供します。

トピック

- [新しいオファーのイベント](#)
- [変更セットのイベント](#)
- [セキュリティ概要レポートのイベント](#)
- [支払いのイベント](#)

このトピック

販売者によるアクション	受信したイベント	関連トピック
独立系ソフトウェアベンダー (ISV) がオファーを作成し、オファーを購入可能にする	Offer Released	the section called “新しいオファーのイベント”
ISV の製品をチャネルパートナーが使用してオファーを作成する	Offer Released	the section called “新しいオファーのイベント”
チャネルパートナーがオファーを作成する	Offer Released	the section called “新しいオファーのイベント”
変更セットが成功する	Change Set Succeeded	the section called “変更セットのイベント”
変更セットが失敗する	Change Set Failed	the section called “変更セットのイベント”
変更セットがキャンセルされる	Change Set Cancelled	the section called “変更セットのイベント”
ISV の製品でセキュリティの脆弱性が検出された	Products Security Report Created	the section called “セキュリティ概要レポートのイベント”

新しいオファーのイベント

販売者がオファーを作成して、オファーを購入できるようにすると、購入者は次の詳細タイプのイベントを受け取ります: Offer Released。

Note

EventBridge ルールの作成の詳細については、「Amazon EventBridge ユーザーガイド」の「[Amazon EventBridge ルール](#)」を参照してください。

ISV が作成した新しいオファーのイベント本文の例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Offer Released",
  "source": "aws.marketplacecatalog",
  "account": "123456789012",
  "time": "2023-08-26T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Offer/offer-1234567890123"
  ],
  "detail": {
    "requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
    "catalog": "AWSMarketplace",
    "offer": {
      "id": "offer-1234567890123",
      "arn": "arn:aws:catalog:us-east-1:123456789012:Offer/offer-1234567890123",
      "name": "Offer Name",
      "expirationDate": "2025-08-26T00:00:00Z"
    },
    "product": {
      "id": "bbbbaaaa-abcd-1111-abcd-666666666666",
      "arn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/SaaSProduct/bbbbbaaaa-abcd-1111-abcd-666666666666",
      "title": "Product Title"
    },
    "manufacturer": {
      "accountId": "123456789012",
      "name": "Manufacturer Account Name"
    },
    "sellerOfRecord": {
      "accountId": "123456789012",
      "name": "Seller Account Name"
    },
    "targetedBuyerAccountIds": [
      "999988887777",
      "111122223333"
    ]
  }
}
```


以下は、ISV の製品をチャネルパートナーが使用してオファーを作成した場合のイベント本文の例です。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Offer Released",
  "source": "aws.marketplacecatalog",
  "account": "123456789012",
  "time": "2023-08-26T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aws-marketplace:us-east-1:987654321098:AWSMarketplace/Offer/offer-1234567890123"
  ],
  "detail": {
    "requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
    "catalog": "AWSMarketplace",
    "offer": {
      "id": "offer-1234567890123",
      "arn": "arn:aws:catalog:us-east-1:987654321098:Offer/offer-1234567890123",
      "name": "Offer Name",
      "expirationDate": "2025-08-26T00:00:00Z"
    },
    "product": {
      "id": "bbbbaaaa-abcd-1111-abcd-666666666666",
      "arn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/SaaSProduct/bbbbaaaa-abcd-1111-abcd-666666666666",
      "title": "Product Title"
    },
    "manufacturer": {
      "accountId": "123456789012",
      "name": "Manufacturer Account Name"
    },
    "sellerOfRecord": {
      "accountId": "987654321098",
      "name": "Seller Account Name"
    },
    "targetedBuyerAccountIds": ["999988887777", "111122223333"],
  }
}
```

以下は、チャネルパートナーがオファーを作成した場合のイベント本文の例です。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Offer Released",
  "source": "aws.marketplacecatalog",
  "account": "987654321098",
  "time": "2023-08-26T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aws-marketplace:us-east-1:987654321098:AWSMarketplace/Offer/offer-1234567890123"
  ],
  "detail": {
    "requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
    "catalog": "AWSMarketplace",
    "offer": {
      "id": "offer-1234567890123",
      "arn": "arn:aws:catalog:us-east-1:987654321098:Offer/offer-1234567890123",
      "name": "Offer Name",
      "expirationDate": "2025-08-26T00:00:00Z"
    },
    "product": {
      "id": "bbbbaaaa-abcd-1111-abcd-666666666666",
      "arn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/SaaSProduct/bbbbbaaaa-abcd-1111-abcd-666666666666",
      "title": "Product Title"
    },
    "manufacturer": {
      "accountId": "123456789012",
      "name": "Manufacturer Account Name"
    },
    "sellerOfRecord": {
      "accountId": "987654321098",
      "name": "Seller Account Name"
    },
    "targetedBuyerAccountIds": ["999988887777", "111122223333"],
  }
}
```

変更セットのイベント

変更セットが完了すると、販売者、チャネルパートナー、プライベートマーケットプレイス管理者はイベントを受信することができます。AWS Marketplace Catalog API は、変更セットが完了し、ステータスが成功、失敗、またはキャンセルされたときにイベントを送信します。これらのイベントのソースは `aws.marketplacecatalog` で、詳細タイプの値は `Change Set Succeeded`、`Change Set Failed`、`Change Set Cancelled` のいずれかになります。

Note

変更セットの詳細については、「AWS Marketplace Catalog API リファレンス」の「[変更セットの使用](#)」を参照してください。

各イベントには、変更セット ID、変更セット名、イベント詳細タイプ、失敗コード (失敗したリクエストの場合)、リクエストの開始時刻と終了時刻などの変更リクエストの詳細が含まれます。これにより、`DescribeChangeSet` アクションを継続的にクエリしたり、変更リクエストのステータス AWS Marketplace 管理ポータル についてを確認したりすることなく、変更セットをモニタリングできます。

Note

EventBridge ルールの作成の詳細については、「Amazon EventBridge ユーザーガイド」の「[Amazon EventBridge ルール](#)」を参照してください。

以下は、`Change Set Succeeded` 詳細タイプのイベント本文の例です。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Change Set Succeeded",
  "source": "aws.marketplacecatalog",
  "account": "123456789012",
  "time": "2022-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/76yesvf8y165pa4f98td2crtg"
  ],
```

```
"detail": {
  "requestId" : "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
  "Catalog": "AWSMarketplace",
  "ChangeSetId": "76yesvf8y165pa4f98td2crtg",
  "ChangeSetName": "Create my product",
  "StartTime": "2018-02-27T13:45:22Z",
  "EndTime": "2018-02-27T14:55:22Z"
}
```

以下は、Change Set Failed 詳細タイプのイベント本文の例です。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Change Set Failed",
  "source": "aws.marketplacecatalog",
  "account": "123456789012",
  "time": "2022-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/76yesvf8y165pa4f98td2crtg"
  ],
  "detail": {
    "requestId" : "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
    "Catalog": "AWSMarketplace",
    "ChangeSetId": "76yesvf8y165pa4f98td2crtg",
    "ChangeSetName": "Create my product",
    "StartTime": "2018-02-27T13:45:22Z",
    "EndTime": "2018-02-27T14:55:22Z",
    "FailureCode": "CLIENT_ERROR"
  }
}
```

以下は、Change Set Cancelled 詳細タイプのイベント本文の例です。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Change Set Cancelled",
  "source": "aws.marketplacecatalog",
  "account": "123456789012",
```

```
"time": "2022-11-01T13:12:22Z",
"region": "us-east-1",
"resources": [
  "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/76yesvf8y165pa4f98td2crtg"
],
"detail": {
  "requestId" : "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
  "Catalog": "AWSMarketplace",
  "ChangeSetId": "76yesvf8y165pa4f98td2crtg",
  "ChangeSetName": "Create my product",
  "StartTime": "2018-02-27T13:45:22Z",
  "EndTime": "2018-02-27T14:55:22Z"
}
}
```

セキュリティ概要レポートのイベント

販売者の製品でセキュリティの脆弱性が検出されると、概要レポートイベントと、未解決の製品問題の定期的なリマインダーを受け取ることができます。これらのイベントのソースは `aws.marketplacecatalog` で、詳細タイプは `Products Security Report Created` です。

各イベントには、検出された問題のある製品とバージョンの数、影響を受ける最新バージョンの数、およびこれらの製品またはバージョンの一時的な制限を防ぐために解決が必要な日付の概要が含まれます。

Note

EventBridge ルールの作成の詳細については、「Amazon EventBridge ユーザーガイド」の「[Amazon EventBridge ルール](#)」を参照してください。

セキュリティイベントの管理の詳細については、「AWS Blog」のブログ記事「[How to improve the security of your product catalog in AWS Marketplace](#)」を参照してください。

以下は、`Products Security Report Created` 詳細タイプのイベント本文の例です。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Products Security Report Created",
  "source": "aws.marketplacecatalog",
```

```
"account": "123456789012",
"time": "2023-10-31T00:00:00Z",
"region": "us-east-1",
"resources": [],
"detail": {
  "numberOfProductsWithIssues": 1,
  "numberOfVersionsWithIssues": 1,
  "numberOfLatestVersionsWithIssues": 1,
  "newIssuesFound": true,
  "upcomingResolutionDueDate": "2023-12-01T00:00:00Z",
  "requestId": "533fa17d-3e97-5051-bcaf-1fae45fb3f8b"
}
}
```

支払いのイベント

無効な銀行口座の詳細が原因で販売者の銀行口座への支払いが失敗した場合、AWS Marketplace ISVsとチャネルパートナーはイベントを受け取る可能性があります。

次の JSON イベントコードでは、これらのイベントの `source` 値は `aws.marketplace`、`detail-type` 値は `Disbursement Paused`。 `resources` 値は、無効な銀行口座 Amazon リソースナンバー (ARN) を示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Disbursement Paused",
  "source": "aws.marketplace",
  "account": "<account id of end user>",
  "time": "2022-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:payments:us-east-1:1234567890:paymentinstrument:123"],
  "detail": {
    "requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
    "catalog": "AWSMarketplace"
  }
}
```

無効な銀行口座の詳細の問題を解決するには、AWS Marketplace 管理ポータルに銀行口座の詳細を追加します。手順については、「[銀行口座の詳細を追加するには](#)」を参照してください。

Amazon EventBridge ルールの作成の詳細については、[「Amazon EventBridge ユーザーガイド」の「Amazon EventBridge のルール」](#)を参照してください。 EventBridge

の販売者レポート、データフィード、ダッシュボード AWS Marketplace

AWS Marketplace 販売者は、次のツールを使用して製品の販売に関する情報を収集および分析できます。

- が自動的に AWS Marketplace 生成され、登録されたすべての AWS Marketplace 販売者が利用できる [レポート](#)。
- [API](#)。これらのレポートの各セクションにアクセスできます。
- [データフィード](#)。レポートに示された取引の顧客情報を識別するための追加の顧客情報を提供します。
- [ダッシュボード](#)。財務データへのアクセスと分析に役立つチャート、グラフ、インサイトを利用できる [Amazon QuickSight](#) が用意されています。

AWS Marketplace は、次の事項に従いながら、レポート、データフィード、ダッシュボードにできるだけ多くのデータを提供します。

- 顧客データの保護に関する Amazon の標準と規定。
- 購入者が AWS Marketplace で製品を購入する際に承諾する利用規約。販売者は、購入者のデータを安全に管理し、購入者からリクエストがあり次第、データを削除することが契約上義務付けられています。

Note

レポートの通知 E メールをキャンセルするには、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

のレポート、データフィード、ダッシュボードの詳細については AWS Marketplace、以下のトピックを参照してください。

トピック

- [の販売者配信データフィード AWS Marketplace](#)
- [販売者がレポートする AWS Marketplace](#)

- [補足レポート](#)
- [販売者ダッシュボード](#)

の販売者配信データフィード AWS Marketplace

AWS Marketplace は、構造化された up-to-date 製品および顧客情報を AWS Marketplace システムから販売者 Amazon S3 バケットに送信し、販売者所有のビジネスインテリジェンスツール間で ETL (抽出、変換、ロード) するためのメカニズムとしてデータフィードを提供します。データフィードは、カンマ区切り値 (CSV) ファイルを収集し、これを指定先の暗号化された Amazon S3 バケットに配信します。データフィードは 1 日以内に生成され、前日の 24 時間のデータが含まれます。このセクションでは、データフィードの概要を示し、データフィードにアクセスして使用方法について説明します。以降のセクションでは、各データフィードについて説明します。

取引データは双時構造で配信および追加されるため、販売者は両方のタイムスタンプ付きの 2 つのタイムラインに沿ってデータを保存し、クエリを実行できます。

- 有効時間: 現実世界で事実が発生した日時 (「知ったこと」)
- システム時間: その事実がデータベースに記録された日時 (「知ったとき」)

データフィードは、UTC 午前 0 時に毎日配信され、前日からの 24 時間分のデータを含む更新が実行されます。更新は、サブスクライブしている顧客、請求を受けている顧客、または支払い AWS によって定義できます。

トピック

- [AWS Marketplace データフィードのストレージと構造](#)
- [データフィードへのアクセス](#)
- [データフィードを使用したデータ収集と分析](#)
- [データフィードテーブルの概要](#)
- [データフィードクエリの例](#)
- [データフィード](#)

AWS Marketplace データフィードのストレージと構造

AWS Marketplace は、構造化された up-to-date 製品および顧客情報を AWS Marketplace システムから販売者所有のビジネスインテリジェンスツール間の ETL (抽出、変換、ロード) 用の販売者

Amazon S3 バケットに送信するメカニズムとして、データフィードを提供します。このトピックでは、データフィードの構造とストレージについて詳述します。

データフィードは、カンマ区切り値 (CSV) ファイルを収集し、これを指定先の暗号化された Amazon S3 バケットに配信します。CSV ファイルの特性は次のとおりです。

- [4180 標準](#)に準拠しています。
- 文字エンコーディングは UTF-8 (BOM なし) です。
- カンマは、値間の区切り文字として使用されます。
- フィールドは二重引用符でエスケープ
- \n は改行文字です。
- 日付は UTC タイムゾーンで報告され、日時形式は ISO 8601 に従い、精度は 1 秒以内です。
- すべての *_period_start_date 値および *_period_end_date 値は包括的です。つまり、23:59:59 は任意の日の最後のタイムスタンプです。
- すべての金銭情報フィールドの先頭に通貨フィールドが付きます。
- 金銭情報フィールドでは、小数点の区切り文字としてピリオド (.) を使用し、3 桁の区切り文字としてカンマ (,) を使用しません。

データフィードは次のように生成されて保存されます。

- データフィードは 1 日以内に生成され、前日の 24 時間のデータが含まれます。
- Amazon S3 バケットの場合、データフィードは次の形式を使用して月別に整理されます。

bucket-name/data-feed-name_version/year=YYYY/month=MM/data.csv

- 毎日のデータフィードが生成されると、当月の既存の CSV ファイルに追加されます。新しい月が始まると、データフィードごとに新しい CSV ファイルが生成されます。
- データフィードの情報は、2010 年 1 月 1 日から 2020 年 4 月 30 日 (当日を含む) にバックフィルされ、year=2010/month=01 サブフォルダの [CSV ファイル](#)で確認できます。

特定のデータフィードの当月のファイルに列ヘッダーのみが含まれ、データが含まれていない場合があります。これは、当月のフィードに新しいエントリがなかったことを意味します。製品フィードなど、更新頻度が低いデータフィードで発生する場合があります。このような場合、データはバックフィルされたフォルダで利用できます。

- Amazon S3 では、[Amazon S3 ライフサイクルポリシー](#)を作成して、バケットにファイルを保持する期間を管理できます。

- 暗号化された Amazon SNS Amazon S3を設定できます。通知を設定する方法の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[Amazon SNS の開始方法](#)」を参照してください。

データの履歴化

各データフィールドには、データの履歴を示す列があります。valid_to を除き、これらの列はすべてのデータフィールドに共通です。これらは共通の履歴スキーマとして含まれており、データのクエリに役立ちます。

列名	説明
valid_from	主キーの値が他のフィールドの値に関連して有効である最初の日付。
valid_to	この列は 住所 データフィールドにのみ表示され、常に空白です。
insert_date	レコードがデータフィールドに挿入された日付。
update_date	レコードが最後に更新された日付。
delete_date	この列は常に空白です。

これらの列の例を次に示します。

valid_from	valid_to	insert_date	update_date	delete_date
2018-12-12T02:00:00Z		2018-12-12T02:00:00Z	2018-12-12T02:00:00Z	
2019-03-29T03:00:00Z		2019-03-29T03:00:00Z	2019-03-29T03:00:00Z	
2019-03-29T03:00:00Z		2019-03-29T03:00:00Z	2019-04-28T03:00:00Z	

valid_from と update_date フィールドが一緒になって双時データモデルを形成します。valid_from フィールドは、その名のとおり、アイテムがいつ有効になるかがわかります。項目が編集された場合、フィールドにはそれぞれ、update_date が異なり、valid_from の日付が同じレコードが複数含まれている可能性があります。例えば、ある項目の現在の値を調べるには、最新の valid_from の日付があるレコードのリストから最新の update_date のレコードを検索します。

上の例では、レコードは元々 2018 年 12 月 12 日に作成されたものです。その後、2019 年 3 月 29 日に変更されました (レコード内のアドレスが変更された場合など)。その後、2019 年 4 月 28 日に、アドレスの変更が修正されました (valid_from は変更されず、update_date は変更されました)。アドレスを修正すると (まれなイベント)、レコードが元の valid_from の日付から遡って変更されるため、そのフィールドは変更されませんでした。最新の valid_from レコードを検索するクエリでは 2 つのレコードが返され、最新の update_date から現在の実際のレコードが得られます。

データフィードへのアクセス

を使用すると AWS Marketplace、データフィードを、構造化された up-to-date 製品および顧客情報を AWS Marketplace システムから、販売者所有のビジネスインテリジェンスツール間の ETL (抽出、変換、ロード) 用の Amazon S3 バケットに送信するメカニズムとして使用できます。暗号化された Amazon S3 バケットでデータフィードを受信するように環境を設定する必要があります。このトピックでは、データフィードにアクセスしてサブスクリプションを解除する方法について説明します。

トピック

- [データフィードにアクセスする](#)
- [データフィードポリシー](#)
- [データフィードからサブスクリプションを解除する](#)

データフィードにアクセスする

1. SQL と ETL (Extract/Transform/Load) の経験があるビジネスインテリジェンスまたはデータエンジニアを割り当てます。また、API の設定経験も必要です。
2. Amazon Simple Storage Service バケットとデータフィードのサブスクリプションを設定します。Marketplace 製品出品に関連付けられた AWS 販売者アカウント ID を使用します。これを行うには、[この YouTube 動画を視聴](#)するか、次の手順を実行します。

ビデオと手順では、設定を簡素化する[AWS CloudFormation テンプレート](#)の使用方法が説明されています。

- a. ウェブブラウザを開いて [AWS Marketplace 管理ポータル](#) にサインインし、「[顧客データストレージの設定](#)」に進みます。
- b. AWS CloudFormation テンプレートを使用してリソースを作成するを選択し、別のウィンドウで AWS CloudFormation コンソールでテンプレートを開きます。
- c. テンプレートで以下を指定し、[Next (次へ)] を選択します。
 - スタック名 - データフィードへのアクセスを有効にするために作成するリソースのコレクション。
 - Amazon S3 バケット名 - データフィードを保存する先のバケット。
 - (オプション) Amazon SNS トピック名 – Amazon Simple Storage Service バケットで通知を受信するためのトピック。
- d. [Review (確認)] ページで入力内容を確認し、[Create stack (スタックの作成)] を選択します。これにより、CloudFormation のステータスと詳細が記載された新しいページが開きます。
- e. リソースタブから、CloudFormation ページから次のリソースの Amazon リソースネーム (ARNs) を、AWS Marketplace [顧客データストレージの設定](#) ページのフィールドにコピーします。
 - データフィードを保存する先の Amazon S3 バケット
 - AWS KMS Amazon S3 バケットを暗号化するための キー
 - (オプション) が Amazon SNS3 バケットに新しいデータを AWS 配信するときに通知を受信するための Amazon SNS トピック Amazon S3
- f. [Set up customer data storage (顧客データストレージの設定)] ページで、[Submit (送信)] を選択します。
- g. (オプション) CloudFormation テンプレートによって作成されたポリシーを編集します。詳細については、「[データフィードポリシー](#)」を参照してください。

これで、データフィードにサブスクライブしました。次回にデータフィードが生成されると、データにアクセスできます。

3. ETL (Extract/Transform/Load) オペレーションを使用すると、データフィードをデータウェアハウスまたはリレーショナルデータベースに接続します。

Note

データツールにはさまざまな機能があります。ツールの機能に合わせて統合を設定するには、ビジネスインテリジェンスエンジニアまたはデータエンジニアを関与させる必要があります。

4. SQL クエリを実行または作成するには、データツールでプライマリキーと外部キーを適用するようにデータフィードを設定します。各データフィードは一意のテーブルを表し、エンティティ関係を使用してデータスキーマ内のすべてのデータフィードを設定する必要があります。テーブルとエンティティの関係の詳細については、このガイドの「[データフィードテーブルの概要](#)」を参照してください。
5. Amazon Simple Notification Service をセットアップして、データウェアハウスまたはリレーショナルデータベースを自動的に更新します。各一意のフィードのデータが Amazon SNS/Amazon S3通知を設定できます。これらの通知を使用すると、販売者データツールがこの機能をサポートし、データフィードを介して新しいデータを受信する際に、販売者データウェアハウスを自動更新できます。詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[Amazon SNS の開始方法](#)」を参照してください。

通知の例

```
{
  "mainExecutionId": "1bc08b11-ab4b-47e1-866a-9c8f38423a98",
  "executionId": "52e862a9-42d2-41e0-8010-810af84d39b1",
  "subscriptionId": "27ae3961-b13a-44bc-a1a7-365b2dc181fd",
  "processedFiles": [],
  "executionStatus": "SKIPPED",
  "errors": [],
  "feedType": "[data feed name]"
}
```

通知には次の executionStatus ステータスがあります。

- SKIPPED – 販売者に対するその日の新しいデータはありません。
- COMPLETED – フィードに新しいデータを提供しました。
- FAILED – フィード配信に問題があります。

6. SQL クエリを実行してセットアップを検証します。この[ガイドの サンプルクエリ](#)、または GitHub のクエリは、<https://github.com/aws-samples/aws-marketplace-api-samples/tree/main/seller-data-feeds/queries> で使用できます。

Note

このガイドのサンプルクエリは Athena AWS 用に書かれています。ツールで使用するクエリは変更する必要がある場合があります。

7. ビジネスユーザーがデータを使用する場所を決定します。例えば、以下のことが可能です。
- データウェアハウスまたは SQL データベースから .csv データをエクスポートします。
 - PowerBI や Tableau などの視覚化ツールにデータを接続します。
 - Salesforce、Infor、Netsuite などの CRM、ERP、財務ツールにデータをマッピングします。

AWS CloudFormation テンプレートの詳細については、「AWS CloudFormation ユーザーガイド」の[AWS CloudFormation 「テンプレートの使用」](#)を参照してください。

データフィードポリシー

Amazon S3 バケットが CloudFormation テンプレートによって作成されると、そのバケット、AWS KMS キー、Amazon SNS トピックにアタッチされたアクセスポリシーが作成されます。このポリシーにより、AWS Marketplace レポートサービスはデータフィード情報をバケットと SNS トピックに書き込むことができます。各ポリシーには次のようなセクションがあります (この例は Amazon S3 バケットのものです)。

```
{
  "Sid": "AwsMarketplaceDataFeedsAccess",
  "Effect": "Allow",
  "Principal": {
    "Service": "reports.marketplace.amazonaws.com"
  },
  "Action": [
    "s3:ListBucket",
    "s3:GetObject",
    "s3:PutObject",
    "s3:GetEncryptionConfiguration",
    "s3:GetBucketAcl",
    "s3:PutObjectAcl"
  ],
```

```

    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket",
      "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ],
  },

```

このポリシーでは、AWS Marketplace は `reports.marketplace.amazonaws.com` サービスプリンシパルを使用して Amazon S3 バケットにデータをプッシュします。CloudFormation テンプレートでは、`amzn-s3-demo-bucket` が指定されています。

AWS Marketplace レポートサービスが Amazon S3 AWS KMS または Amazon SNS を呼び出すと、バケットに書き込む予定のデータの ARN が提供されます。バケットに書き込まれるデータが、ユーザーに代わって書き込まれるデータだけにするには、ポリシーの条件に `aws:SourceArn` を指定します。次の例では、`account-id` を自分の AWS アカウントの ID に置き換える必要があります。

```

{
  "Sid": "AwsMarketplaceDataFeedsAccess",
  "Effect": "Allow",
  "Principal": {
    "Service": "reports.marketplace.amazonaws.com"
  },
  "Action": [
    "s3:ListBucket",
    "s3:GetObject",
    "s3:PutObject",
    "s3:GetEncryptionConfiguration",
    "s3:GetBucketAcl",
    "s3:PutObjectAcl"
  ],
  "Resource": [
    "arn:aws:s3:::amzn-s3-demo-bucket",
    "arn:aws:s3:::amzn-s3-demo-bucket/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "account-id",
      "aws:SourceArn": [
        "arn:aws:marketplace::account-id:AWSMarketplace/SellerDataSubscription/DataFeeds_V1",
        "arn:aws:marketplace::account-id:AWSMarketplace/SellerDataSubscription/Example-Report"
      ]
    }
  }
}

```



```
    },  
    },  
}
```

データフィードからサブスクリプションを解除する

ウェブブラウザを開き、[AWS Marketplace 管理ポータル](#)にサインインします。次に、「[お問い合わせ](#)」[ページ](#)に移動して、サブスクリプション解除リクエストを AWS Marketplace Seller Operations チームに送信します。受信解除のリクエストの処理が完了するのに、最大で 10 営業日かかる場合があります。

データフィードを使用したデータ収集と分析

AWS Marketplace は、構造化された up-to-date 製品および顧客情報を AWS Marketplace システムから販売者 Amazon S3 バケットに送信し、販売者所有のビジネスインテリジェンスツール間で ETL (抽出、変換、ロード) するためのメカニズムとしてデータフィードを提供します。Amazon S3 バケットのデータが使用可能な場合、次の方法でデータフィードを使用できます。

- [データフィードへのアクセス](#) で作成した Amazon S3 バケットから .CSV ファイルをダウンロードして、スプレッドシートでデータを表示できるようにします。
- ETL (抽出、変換、ロード)、SQL クエリ、ビジネス分析ツールを使用して、データを収集して分析します。

AWS サービスを使用してデータを収集および分析したり、.CSV ベースのデータセットの分析を実行できるサードパーティーのツールを使用したりできます。

データを収集および分析するためのデータフィードの詳細については、次の例を参照してください。

例: AWS サービスを使用してデータを収集および分析する

次の手順では、データフィードの送信先を Amazon S3 バケットとするように環境を設定済みであり、バケットにデータフィードが含まれていることを前提としています。

データフィードからデータを収集して分析するには

1. [AWS Glue コンソール](#)から、データフィードを格納する Amazon S3 バケットに接続する[クローラーを作成](#)し、必要なデータを抽出して、AWS Glue Data Catalog にメタデータテーブルを作成します。

詳細については AWS Glue、「[AWS Glue デベロッパーガイド](#)」を参照してください。

2. [Athena コンソール](#)から、[AWS Glue Data Catalogのデータに対して SQL クエリを実行](#)します。

Athena の詳細については、「[Amazon Athena ユーザーガイド](#)」を参照してください。

3. [QuickSight コンソール](#)から[分析を作成し](#)、データの[ビジュアルを作成](#)します。

QuickSight の詳細については、[Amazon QuickSight ユーザーガイド](#)」を参照してください。

AWS サービスを使用してデータフィード内のデータを収集および分析する方法の詳細な例については、AWS Marketplace ブログの「[販売者データフィード配信サービス、Amazon Athena、QuickSight を使用して販売者レポートを作成する](#)」を参照してください。

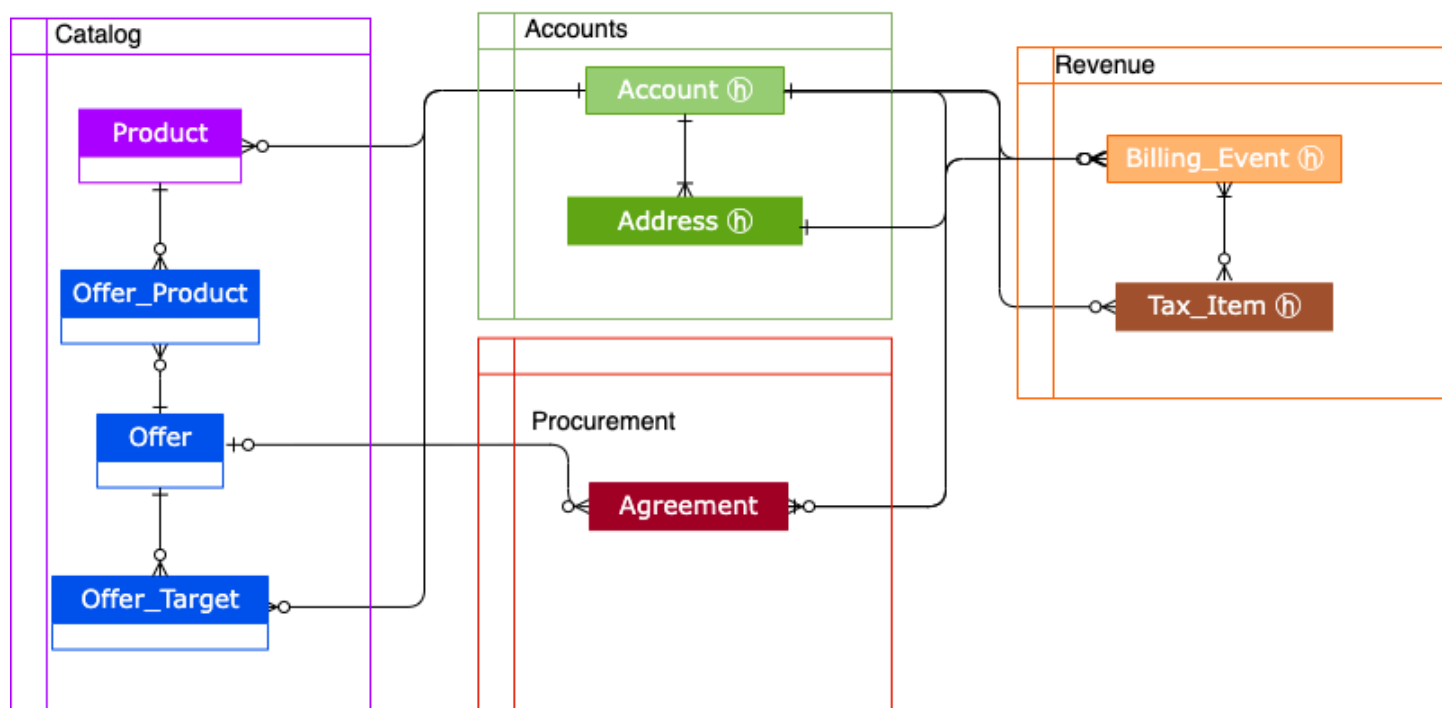
データフィードテーブルの概要

AWS Marketplace は、データフィードを一連のテーブルとして提供し、結合してクエリのコンテキストを増やすことができます。

AWS Marketplace は、データフィードで次の一般的なドメインまたは関心のあるカテゴリを提供します。

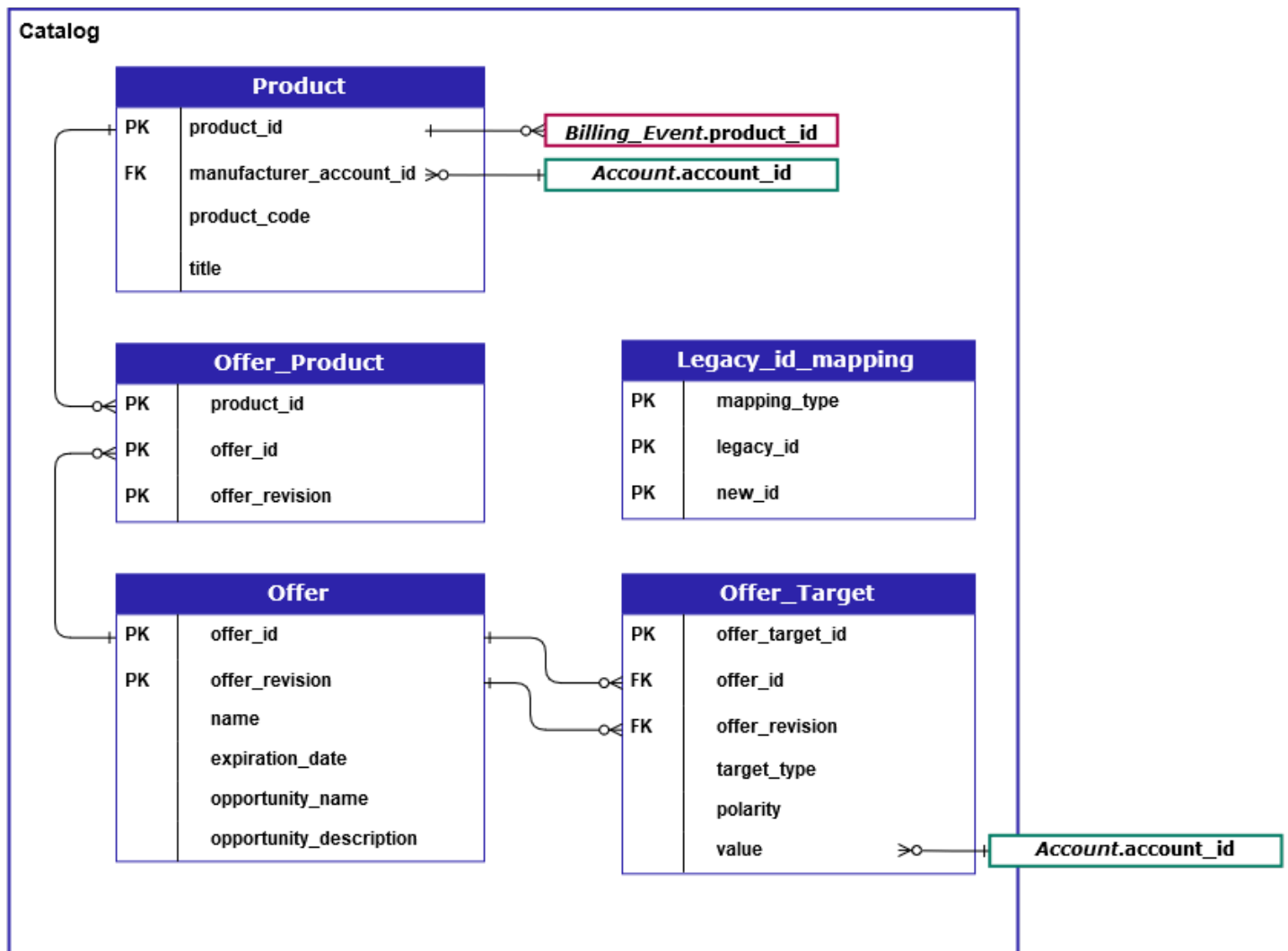
- カタログ - アカウント内の製品やオファーに関する情報が含まれます。
- アカウント - で製品を提供または購入するアカウント AWS Marketplace (チャネルパートナーや購入者など、お客様自身のアカウントまたはお客様が作業する相手のアカウント) に関する情報が含まれます。
- 収益 - 請求、支払い、税金に関する情報が含まれます。
- 調達 - 登録販売者として作成した製品オファーの契約に関する情報が含まれます。

この図は、カタログ、アカウント、収益ドメインのテーブルを示しています。



カタログ関連テーブル

次の図は、カタログドメイン内のテーブルとテーブル内のフィールド間の関係を示しています。



Product、Offer_Product、Offer、Offer_Target、Legacy_id_mapping テーブルはカタログドメインにあります。

Offer_Target テーブルにはターゲットの account_id の値フィールドが含まれますが、target_type 値が account である場合に限りです。

Legacy_id_mapping テーブルは現在のデータには使用されません。

Note

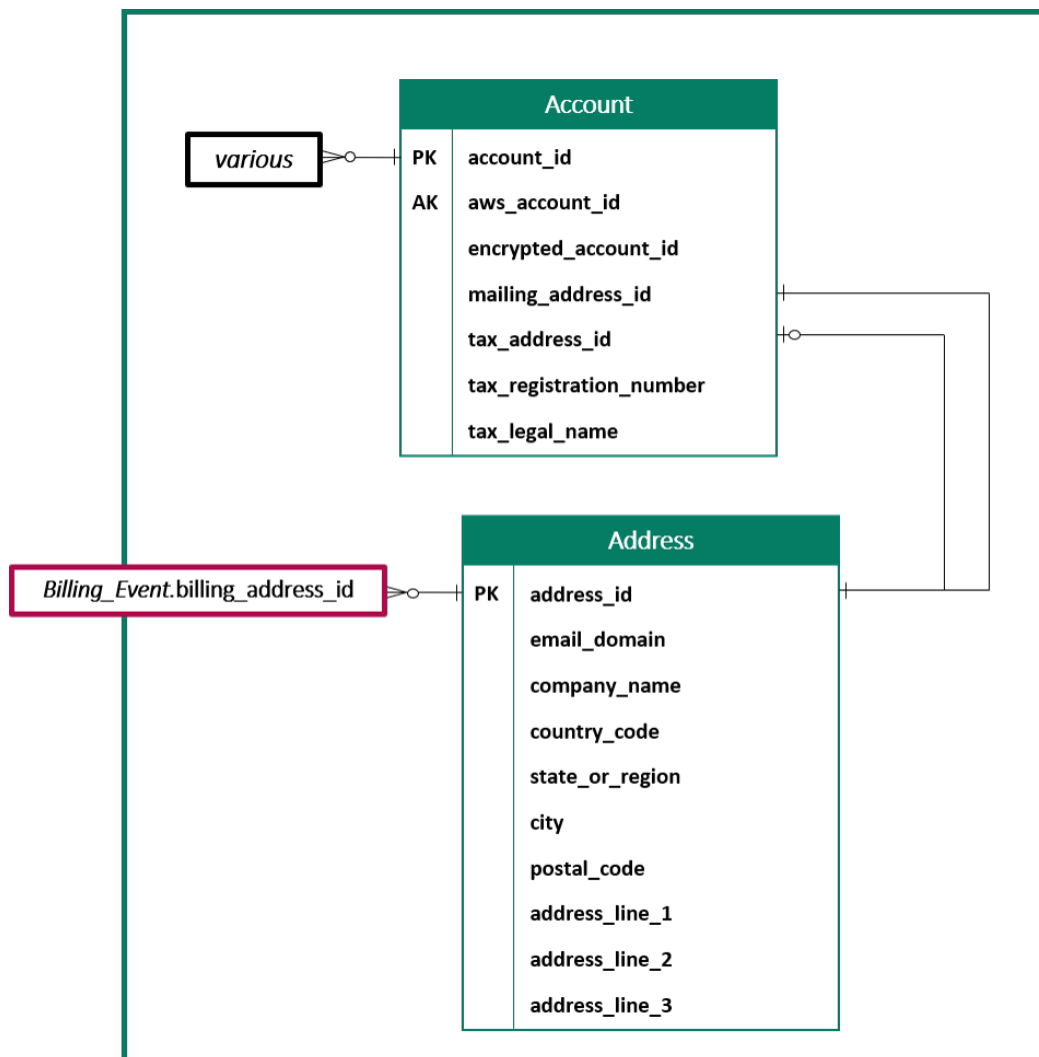
テーブル内の各フィールドの説明や作成できる結合など、これらのテーブルの詳細については、以下のトピックを参照してください。

- [製品データフィード](#)
- [オファー製品データフィード](#)

- [オフアーデータフィード](#)
- [オフアーターゲットデータフィード](#)
- [レガシーマッピングデータフィード](#)

勘定科目関連テーブル

次の図は、アカウントドメイン内のとのテーブル間の関係、Account および Address テーブル内のフィールドを示しています。



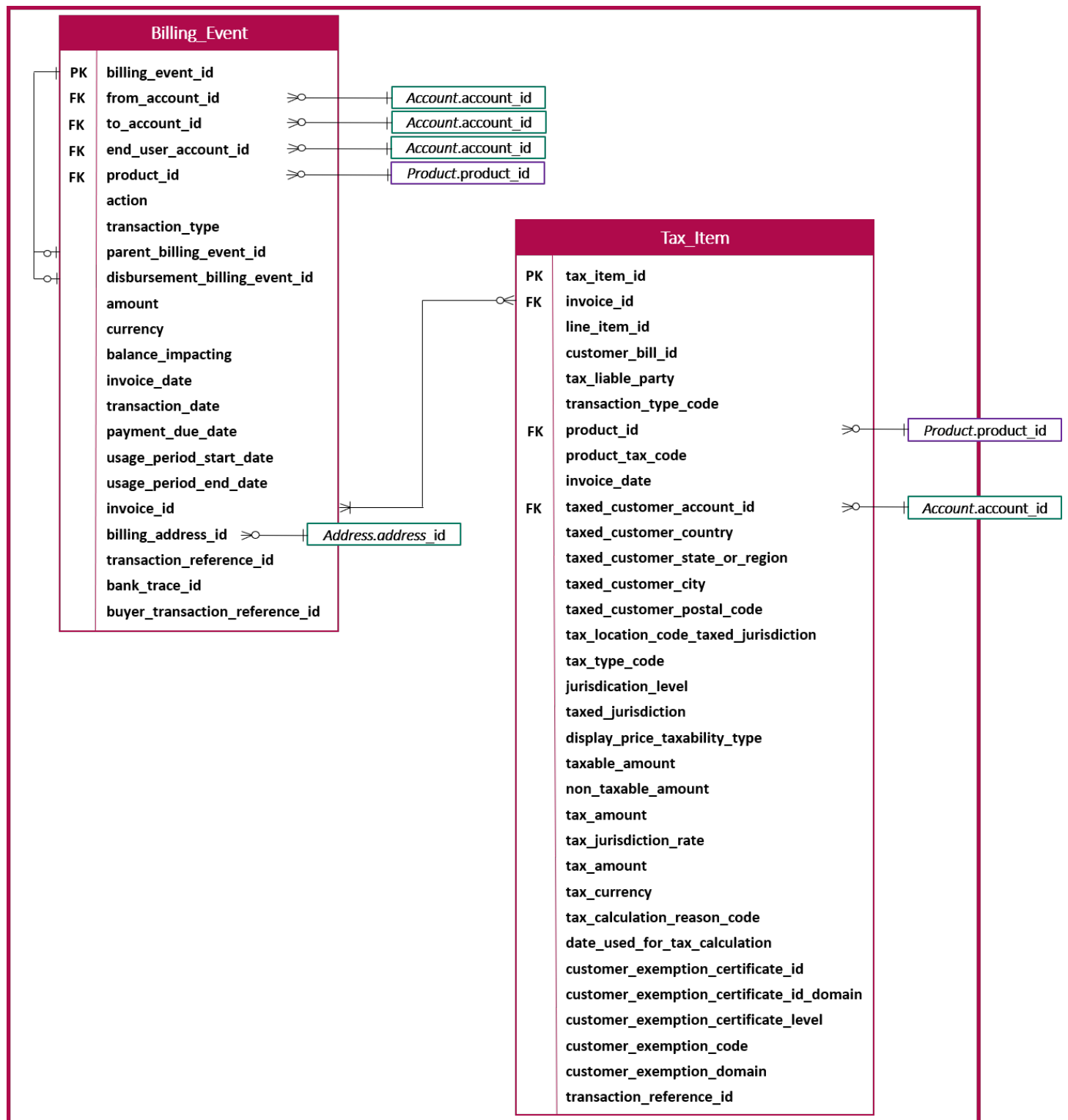
Note

テーブル内の各フィールド説明や作成できる結合を含むこれらのテーブルの詳細については、次のトピックを参照してください。

- [アカウントデータフィード](#)
- [住所データフィード](#)

収益関連テーブル

次の図は、収益ドメイン内のとのテーブル間の関係、Billing_Event および Tax_Item テーブル内のフィールドを示しています。Billing_Event テーブルには、支払いと請求イベントに関する情報が含まれています。



Note

テーブル内の各フィールド説明や作成できる結合を含むこれらのテーブルの詳細については、次のトピックを参照してください。

- [請求イベントデータフィード](#)
- [税品目データフィード](#)

調達関連テーブル

次の図は、調達ドメインの契約テーブル内のフィールドを示しています。

Note

テーブル内の各フィールドの説明や作成できる結合など、これらのテーブルの詳細については、このガイドの「[契約データフィード](#)」を参照してください。

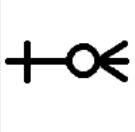
以下のセクションでは、各ドメインのエンティティ関係 (ER) 図を示します。各 ER 図には、テーブルと各テーブル内のフィールド、およびテーブルの結合に使用できるフィールドが示されています。

Note

このセクションの ER 図には、すべてのデータフィードに共通するフィールドは含まれていません。共通フィールドの詳細については、「[AWS Marketplace データフィードのストレージと構造](#)」を参照してください。

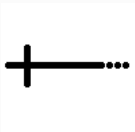
次の表では、ER 図に示されている記号について説明します。

記号	説明
PK	プライマリキー - テーブルのプライマリキー。valid_from および update_date フィールドと一緒に使用すると、一意になります。これらのフィールドと一緒に使用する方法の詳細については、「 データの履歴化 」を参照

記号	説明
	してください。複数のフィールドがプライマリキーとしてマークされている場合、それらのフィールドが一緒になってプライマリキーになります。
FK	外部キー - 別のテーブルのプライマリキーを表すフィールド。テーブル内で必ずしも一意である必要はありません。 Note 現在のテーブルのレコードに対応するレコードが外部テーブルにない場合、外部キーが空白になることがあります。
AK	代替キー - テーブル内のキーとして使用できるキー。プライマリキーと同じ一意性のルールに従います。
	コネクタ - フィールド間の線は接続を表し、テーブルを結合するために使用できる 2 つのフィールドです。線の端は接続の種類を表します。この例は 1 対多の接続を表しています。

コネクタタイプ

次のテーブルは、各コネクタが持つことができる端部のタイプを示しています。

コネクタタイプ	説明
	1 対 n - この端があるコネクタは、結合のこちら側に値が 1 つだけある結合を表します。

コネクタタイプ	説明
	0 または 1 対 n - この端のコネクタは、結合のこちら側に値が 0 または 1 つある結合を表します。
	0 以上対 n - この端のコネクタは、結合のこちら側に 0 個、1 個、または多数の値を持つ結合を表します。
	1 以上対 n - この端があるコネクタは、結合のこちら側に 1 つまたは複数の値がある結合を表します。

データフィードクエリの例

このセクションでは、が提供するデータフィードを使用した複雑なクエリの例を示します AWS Marketplace。これらの例は、AWS Marketplace 管理ポータルから得られる [販売者がでレポートする AWS Marketplace](#) と似ています。これらのクエリをカスタマイズして、必要な他のレポートを作成できます。

クエリの例

- [契約と更新](#)
- [請求済み収益](#)
- [未払い請求書または支払い済み請求書](#)
- [課税請求書](#)
- [製品別の支払い](#)
- [販売補償レポート](#)

契約と更新

契約および更新データを検索するには、次の例のような一連のクエリを実行できます。クエリは相互に構築され、[契約と更新] ダッシュボード、粒度の細かいデータセクションを作成します。図のように例を使用するか、データやユースケースに合わせてカスタマイズできます。

クエリのコメントは、クエリの動作と変更方法について説明しています。

Query currently under development.

請求済み収益

請求書データを検索するには、次の例のような一連のクエリを実行できます。クエリは相互に構築され、請求収益レポートを作成します。図のように例を使用するか、データやユースケースに合わせてカスタマイズできます。

クエリのコメントは、クエリの動作と変更方法について説明しています。

```
-- Billed revenue report

-- General note: When executing this query we are assuming that the data ingested in
the database uses
-- two time axes (the valid_from column and the update_date column).
-- See documentation for more details: https://docs.aws.amazon.com/marketplace/latest/userguide/data-feed.html#data-feed-details

-- An account_id has several valid_from dates (each representing a separate revision of
the data)
-- but because of bi-temporality, an account_id + valid_from tuple can appear multiple
times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
with accounts_with_uni_temporal_data as (
  select
    account_id,
    aws_account_id,
    encrypted_account_id,
    mailing_address_id,
    tax_address_id,
    tax_legal_name,
    from_iso8601_timestamp(valid_from) as valid_from,
    tax_registration_number
  from
    (
      select
        account_id,
        aws_account_id,
        encrypted_account_id,
```

```

        mailing_address_id,
        tax_address_id,
        tax_legal_name,
        valid_from,
        delete_date,
        tax_registration_number,
        row_number() over (partition by account_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
    from
        accountfeed_v1
)
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

accounts_with_history as (
    with accounts_with_history_with_extended_valid_from as (
        select
            account_id,
            -- sometimes, this columns gets imported as a "bigint" and loses heading 0s ->
            casting to a char and re-adding heading 0s (if need be)
            substring('000000000000' || cast(aws_account_id as varchar), -12) as aws_account_id,
            encrypted_account_id,
            mailing_address_id,
            tax_address_id,
            tax_legal_name,
            -- The start time of account valid_from is extended to '1970-01-01 00:00:00',
            because:
            -- ... in tax report transformations, some tax line items with invoice_date
            cannot
            -- ... fall into the default valid time range of the associated account
            case
                when lag(valid_from) over (partition by account_id order by valid_from asc) is
null
                    then cast('1970-01-01 00:00:00' as timestamp)
                    else valid_from
            end as valid_from
        from accounts_with_uni_temporal_data
    )
    select
        account_id,

```

```
aws_account_id,  
encrypted_account_id,  
mailing_address_id,  
tax_address_id,  
tax_legal_name,  
valid_from,  
coalesce(  
    lead(valid_from) over (partition by account_id order by valid_from asc),  
    cast('2999-01-01 00:00:00' as timestamp)  
) as valid_to  
from  
    accounts_with_history_with_extended_valid_from  
)  
  
-- An address_id has several valid_from dates (each representing a separate revision of  
the data)  
-- but because of bi-temporality, an account_id + valid_from tuple can appear multiple  
times with a different update_date.  
-- We are only interested in the most recent tuple (ie, uni-temporal model)  
address_with_uni_temporal_data as (  
    select  
        from_iso8601_timestamp(valid_from) as valid_from,  
        address_id,  
        company_name,  
        email_domain,  
        country_code,  
        state_or_region,  
        city,  
        postal_code,  
        row_num  
    from  
    (  
        select  
            valid_from,  
            update_date,  
            delete_date,  
            address_id,  
            company_name,  
            email_domain,  
            country_code,  
            state_or_region,  
            city,  
            postal_code,
```

```
        row_number() over (partition by address_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
    from
        addressfeed_v1
    )
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

-- We are only interested in the most recent tuple (BTW: a given address is not
  supposed to change over time but when bugs ;-) so this query mainly does nothing)
address_with_latest_revision as (
    select
        valid_from,
        address_id,
        company_name,
        email_domain,
        country_code,
        state_or_region,
        city,
        postal_code,
        row_num_latest_revision
    from
    (
        select
            valid_from,
            address_id,
            company_name,
            email_domain,
            country_code,
            state_or_region,
            city,
            postal_code,
            row_number() over (partition by address_id order by valid_from desc) as
row_num_latest_revision
        from
            address_with_uni_temporal_data
    )
    where
        row_num_latest_revision = 1
),
```

```

accounts_with_history_with_company_name as (
  select
    awh.account_id,
    awh.aws_account_id,
    awh.encrypted_account_id,
    awh.mailing_address_id,
    awh.tax_address_id,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when address.company_name = '' then null else address.company_name end,
      awh.tax_legal_name) as mailing_company_name,
    address.email_domain,
    awh.valid_from,
    -- For BYOL, the agreement might be accepted (using some external non-AWS system or
    manual process) days before
    -- that BYOL agreement is entered into AWS Marketplace by the buyer. Therefore, the
    buyer is permitted to manually
    -- enter a backdated acceptance date, which might predate the point in time when
    the account was created.
    -- To work around this, we need to adjust the valid_from of the account to be
    -- earlier than the earliest possible backdated BYOL agreement acceptance date.
    case
      when lag(awh.valid_from) over (partition by aws_account_id order by
awh.valid_from asc) is null
      then date_add('Day', -212, awh.valid_from)
      -- 212 is the longest delay between acceptance_date of the agreement and the
account start_Date
      else awh.valid_from
    end as valid_from_adjusted,
    awh.valid_to
  from accounts_with_history as awh
  left join address_with_latest_revision as address on
    awh.mailing_address_id = address.address_id and awh.mailing_address_id is not null
),

-- An agreement_id has several valid_from dates (each representing an agreement
revision)
-- but because of bi-temporality, an agreement_id + valid_from tuple can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
agreements_with_uni_temporal_data as (
  select
    agreement_id,

```

```

    origin_offer_id,
    proposer_account_id,
    acceptor_account_id,
    agreement_revision,
    from_iso8601_timestamp(valid_from) as valid_from,
    from_iso8601_timestamp(start_date) as start_date,
    from_iso8601_timestamp(end_date) as end_date,
    from_iso8601_timestamp(acceptance_date) as acceptance_date,
    agreement_type,
    previous_agreement_id,
    agreement_intent
from
(
    select
        --empty value in Athena shows as '', change all '' value to null
        case when agreement_id = '' then null else agreement_id end as agreement_id,
        origin_offer_id,
        proposer_account_id,
        acceptor_account_id,
        agreement_revision,
        valid_from,
        delete_date,
        start_date,
        end_date,
        acceptance_date,
        agreement_type,
        previous_agreement_id,
        agreement_intent,
        row_number() over (partition by agreement_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
    from
        -- TODO change to agreementfeed_v1 when Agreement Feed is GA'ed
        agreementfeed
)
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

agreements_with_history as (
    with agreements_with_window_functions as (
        select

```



```

        agreement_id,
        origin_offer_id as offer_id,
        proposer_account_id,
        acceptor_account_id,
        agreement_revision,
        start_date,
        end_date,
        acceptance_date,
        -- The start time of agreement valid_from is extended to '1970-01-01 00:00:00',
because:
        -- ... in usage report transformations, some usage line items with usage_date
cannot
        -- ... fall into the default valid time range of the associated agreement
        case
            when lag(valid_from) over (PARTITION BY agreement_id order by valid_from asc)
is null
            then timestamp '1970-01-01 00:00:00'
            else valid_from
        end as valid_from,
        coalesce(
            lead(valid_from) over (partition by agreement_id order by valid_from asc),
            timestamp '2999-01-01 00:00:00'
        ) as valid_to,
        rank() over (partition by agreement_id order by valid_from asc) version,
        agreement_type,
        previous_agreement_id,
        agreement_intent
    from
        agreements_with_uni_temporal_data
)
select
    agreement_id,
    offer_id,
    proposer_account_id,
    acceptor_account_id,
    agreement_revision,
    start_date,
    end_date,
    acceptance_date,
    valid_from,
    case
        when version=1 and valid_from<timestamp '2023-03-03 06:16:08.743' then
timestamp '1970-01-01'

```

```
-- The following 60 minute adjustment is to handle special case where When
Renewal happens for a contract
    when version=1 then date_add('minute',-60,valid_from)
    else valid_from
end as valid_from_adjusted,
valid_to,
agreement_type,
previous_agreement_id,
agreement_intent
from
    agreements_with_window_functions
),

-- An offer_id has several valid_from dates (each representing an offer revision)
-- but because of bi-temporality, an offer_id + valid_from tuple can appear multiple
times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
offers_with_uni_temporal_data as (
    select
        from_iso8601_timestamp(valid_from) as valid_from,
        from_iso8601_timestamp(update_date) as update_date,
        from_iso8601_timestamp(delete_date) as delete_date,
        offer_id,
        offer_revision,
        name,
        expiration_date,
        opportunity_id,
        opportunity_name,
        opportunity_description,
        seller_account_id
    from
        (
            select
                valid_from,
                update_date,
                delete_date,
                offer_id,
                offer_revision,
                name,
                expiration_date,
                opportunity_id,
                opportunity_name,
                opportunity_description,
                seller_account_id,
```

```

        row_number() over (partition by offer_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
    from
        offerfeed_v1
)
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

-- Here, we build the validity time range (adding valid_to on top of valid_from) of
each offer revision.
-- We will use it to get Offer name at invoice time.
-- NB: If you'd rather get "current" offer name, un-comment
"offers_with_latest_revision"
offers_with_history as (
    select
        offer_id,
        offer_revision,
        name,
        opportunity_id,
        opportunity_name,
        opportunity_description,
        valid_from,
        -- When we try to look up an offer revision as at the acceptance date of a BYOL
agreement, we run into a problem.
        -- For BYOL, the agreement might be accepted (using some external non-AWS system or
manual process) days before
        -- that BYOL agreement is entered into AWS Marketplace by the buyer. Therefore, the
buyer is permitted to manually
        -- enter a backdated acceptance date, which might predate the point in time when
the first revision of the offer
        -- was created. To work around this, we need to adjust the valid_from on the first
revision of the offer to be
        -- earlier than the earliest possible backdated BYOL agreement acceptance date.
        case
            when lag(valid_from) over (partition by offer_id order by valid_from asc) is null
and valid_from < cast('2021-04-01' as timestamp)
            then date_add('Day', -3857, valid_from)
            -- 3857 is the longest delay between acceptance_date of an agreement and the
first revision of the offer

```

```

        when lag(valid_from) over (partition by offer_id order by valid_from asc) is null
and valid_from >= cast('2021-04-01' as timestamp)
        then date_add('Day', -1460, valid_from)
        --after 2021 for the two offers we need to adjust for 2 more years
        else valid_from
    end as valid_from_adjusted,
    coalesce(
        lead(valid_from) over (partition by offer_id order by valid_from asc),
        cast('2999-01-01 00:00:00' as timestamp))
    as valid_to
from offers_with_uni_temporal_data
),
-- provided for reference only if you are interested into get "current" offer name
-- (ie. not used afterwards)
offers_with_latest_revision as (
    select
        offer_id,
        offer_revision,
        name,
        opportunity_name,
        opportunity_description,
        valid_from,
        null valid_to
    from
    (
        select
            offer_id,
            offer_revision,
            name,
            opportunity_name,
            opportunity_description,
            valid_from,
            null valid_to,
            row_number() over (partition by offer_id order by valid_from desc) as
row_num_latest_revision
        from
            offers_with_uni_temporal_data
    )
    where
        row_num_latest_revision = 1
),

-- An offer_target_id has several valid_from dates (each representing an offer
revision)

```

```
-- but because of bi-temporality, an offer_target_id + valid_from tuple can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
offer_targets_with_uni_temporal_data as (
  select
    from_iso8601_timestamp(valid_from) as valid_from,
    from_iso8601_timestamp(update_date) as update_date,
    from_iso8601_timestamp(delete_date) as delete_date,
    offer_target_id,
    offer_id,
    offer_revision,
    target_type,
    polarity,
    value
  from
    (
      select
        valid_from,
        update_date,
        delete_date,
        offer_target_id,
        offer_id,
        offer_revision,
        target_type,
        polarity,
        value,
        row_number() over (partition by offer_target_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
      from
        offertargetfeed_v1
    )
  where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

offer_target_type as (
  select
    offer_id,
    offer_revision,
    substring(
```

```

    -- The first character indicates the priority (lower value means higher
precedence):
    min(
        case
            when offer_target.target_type='BuyerAccounts' then '1Private'
            when offer_target.target_type='ParticipatingPrograms' then '2Program:' ||
cast(offer_target.value as varchar)
            when offer_target.target_type='CountryCodes' then '3GeoTargeted'
            -- well, there is no other case today, but rather be safe...
            else '40ther Targeting'
        end
    ),
    -- Remove the first character that was only used for th priority in the "min"
aggregate function:
    2
    ) as offer_target
from
    offer_targets_with_uni_temporal_data as offer_target
group by
    offer_id,
    offer_revision
),

offers_with_history_with_target_type as (
    select
        offer.offer_id,
        offer.offer_revision,
        -- even though today it is not possible to combine several types of targeting in a
single offer, let's ensure the query is still predictable if this gets possible in the
future
        max(
            case
                when off_tgt.offer_target is null then 'Public'
                else off_tgt.offer_target
            end
        ) as offer_target,
        min(offer.name) as name,
        min(offer.opportunity_name) as opportunity_name,
        min(offer.opportunity_description) as opportunity_description,
        offer.valid_from,
        offer.valid_from_adjusted,
        offer.valid_to,
        offer.opportunity_id
    from

```

```

    offers_with_history as offer
left join offer_target_type as off_tgt on
    offer.offer_id = off_tgt.offer_id
    and offer.offer_revision = off_tgt.offer_revision
group by
    offer.offer_id,
    offer.offer_revision,
    offer.valid_from,
    offer.valid_from_adjusted,
    offer.valid_to,
    offer.opportunity_id
),

-- provided for reference only if you are interested into get "current" offer targets
-- (ie. not used afterwards)
offers_with_latest_revision_with_target_type as (
    select
        offer.offer_id,
        offer.offer_revision,
        -- even though today it is not possible to combine several types of targeting in a
        single offer, let's ensure the query is still predictable if this gets possible in the
        future
        max(
            distinct
            case
                when off_tgt.target_type is null then 'Public'
                when off_tgt.target_type='BuyerAccounts' then 'Private'
                when off_tgt.target_type='ParticipatingPrograms' then 'Program:||'
            cast(off_tgt.value as varchar)
            case
                when off_tgt.target_type='CountryCodes' then 'GeoTargeted'
                -- well, there is no other case today, but rather be safe...
                else 'Other Targeting'
            end
        ) as offer_target,
        min(offer.name) as name,
        min(offer.opportunity_name) as opportunity_name,
        min(offer.opportunity_description) as opportunity_description,
        offer.valid_from,
        offer.valid_to
    from
        offers_with_latest_revision offer
    -- left joining because public offers don't have targets
    left join offer_targets_with_uni_temporal_data off_tgt on
        offer.offer_id=off_tgt.offer_id and offer.offer_revision=off_tgt.offer_revision

```

```
group by
    offer.offer_id,
    offer.offer_revision,
    -- redundant with offer_revision, as each revision has a dedicated valid_from (but
    cleaner in the group by)
    offer.valid_from,
    offer.valid_to
),

-- A product_id has several valid_from dates (each representing a product revision),
-- but because of bi-temporality, each product_id + valid_from tuple can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
products_with_uni_temporal_data as (
    select
        from_iso8601_timestamp(valid_from) as valid_from,
        from_iso8601_timestamp(update_date) as update_date,
        from_iso8601_timestamp(delete_date) as delete_date,
        product_id,
        manufacturer_account_id,
        product_code,
        title
    from
        (
            select
                valid_from,
                update_date,
                delete_date,
                product_id,
                manufacturer_account_id,
                product_code,
                title,
                row_number() over (partition by product_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
            from
                productfeed_v1
        )
    where
        -- keep latest ...
        row_num = 1
        -- ... and remove the soft-deleted one.
        and (delete_date is null or delete_date = '')
),
```



```
products_with_history as (  
  select  
    product_id,  
    title,  
    valid_from,  
    -- Offerv2 can have upto 50 years and Offerv3 is upto 5 years of past date  
    case  
      when lag(valid_from) over (partition by product_id order by valid_from asc) is  
null and valid_from < cast('2021-04-01' as timestamp)  
        then date_add('Day', -3857, valid_from)  
      -- 3827 is the longest delay between acceptance_date of an agreement and the  
product  
      -- we are keeping 3857 as a consistency between the offers and products  
      when lag(valid_from) over (partition by product_id order by valid_from asc) is  
null and valid_from >= cast('2021-04-01' as timestamp)  
        then date_add('Day', -2190, valid_from)  
      --after 2021 for the two offers we need to adjust for 2 more years  
      else valid_from  
    end as valid_from_adjusted,  
    coalesce(  
      lead(valid_from) over (partition by product_id order by valid_from asc),  
      cast('2999-01-01 00:00:00' as timestamp)  
    ) as valid_to,  
    product_code,  
    manufacturer_account_id  
  from  
    products_with_uni_temporal_data  
) ,  
  
legacy_products as (  
  select  
    legacy_id,  
    new_id  
  from  
    legacyidmappingfeed_v1  
  where  
    mapping_type='PRODUCT'  
  group by  
    legacy_id,  
    new_id  
) ,  
  
-- A given billing_event_id represents an accounting event and thus has only one  
valid_from date,
```

```
-- but because of bi-temporality, a billing_event_id (+ its valid_from) can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
billing_events_with_uni_temporal_data as (
  select
    billing_event_id,
    valid_from,
    update_date,
    delete_date,
    invoice_date,
    transaction_type,
    transaction_reference_id,
    parent_billing_event_id,
    bank_trace_id,
    broker_id,
    product_id,
    disbursement_billing_event_id,
    action,
    from_account_id,
    to_account_id,
    end_user_account_id,
    billing_address_id,
    amount,
    currency,
    balance_impacting,
    --empty value in Athena shows as '', change all '' value to null
    case when agreement_id = '' then null else agreement_id end as agreement_id,
    invoice_id,
    payment_due_date,
    usage_period_start_date,
    usage_period_end_date,
    buyer_transaction_reference_id,
    row_num
  from
    (
      select
        billing_event_id,
        from_iso8601_timestamp(valid_from) as valid_from,
        from_iso8601_timestamp(update_date) as update_date,
        delete_date,
        from_iso8601_timestamp(invoice_date) as invoice_date,
        transaction_type,
        transaction_reference_id,
        parent_billing_event_id,
```

```

-- casting in case data was imported as number
cast(bank_trace_id as varchar) as bank_trace_id,
broker_id,
product_id,
disbursement_billing_event_id,
action,
from_account_id,
to_account_id,
end_user_account_id,
billing_address_id,
-- casting in case data was imported as varchar
cast(amount as decimal(38,6)) as amount,
currency,
balance_impacting,
agreement_id,
invoice_id,
case when payment_due_date is null or payment_due_date = '' then null else
from_iso8601_timestamp(payment_due_date) end as payment_due_date,
from_iso8601_timestamp(usage_period_start_date) as usage_period_start_date,
from_iso8601_timestamp(usage_period_end_date) as usage_period_end_date,
buyer_transaction_reference_id,
row_number() over (partition by billing_event_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
from
    billingeventfeed_v1
)
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

-- Here we select the account_id of the current seller (We identify this by looking for
the to_account_id related to revenue transactions).
-- We will use it later to distinguish own agreements from agreements generated by
channel partners.
seller_account as (
    select
        from_account_id as seller_account_id
    from
        billing_events_with_uni_temporal_data bill
    where

```

```

-- Assumption here is only seller will pay listing fee. As of 12/21/2021, there are
cases that Channel partner have 0 listing fee for CPP0, so the amount could be 0.
bill.transaction_type like 'AWS_REV_SHARE' and amount <= 0 and action = 'INVOICED'
group by
-- from_account_id is always the same for all those "listing fee" transactions ==
the seller of record himself.
-- If this view returns more than 1 record, the overall query will fail (on
purpose). Please contact AWS Marketplace if this happens.
from_account_id
),

billing_event_with_business_flags as (
select
  bl.billing_event_id,
  bl.end_user_account_id,
  bl.agreement_id,
  agreement.proposer_account_id,
  agreement.offer_id,
  agreement.acceptor_account_id,
  case
    -- For AWS and BALANCE_ADJUSTMENT, the billing event feed will show the "AWS
Marketplace" account as the
    -- receiver of the funds and the seller as the payer. We are not interested in
this information here.
    -- Null values will be ignored by the `max` aggregation function.
    when bl.transaction_type like 'AWS%' then null
    -- For BALANCE_ADJUSTMENT, payer is seller themselves
    when bl.invoice_id is null then bl.to_account_id
    -- We get the payer of the invoice from *any* transaction type that is not AWS
and not BALANCE_ADJUSTMENT (because they are the same for a given end user + agreement
+ product).
    else bl.from_account_id
  end as payer_account_id,
  bl.product_id,
  bl.action,
  bl.transaction_type,
  bl.parent_billing_event_id,
  bl.disbursement_billing_event_id,
  bl.amount,
  bl.currency,
  bl.balance_impacting,
  bl.invoice_date,
  bl.payment_due_date,
  bl.usage_period_start_date,

```

```

    bl.usage_period_end_date,
    bl.invoice_id,
    bl.billing_address_id,
    bl.transaction_reference_id,
    bl.buyer_transaction_reference_id,
    case when disbursement.bank_trace_id = 'EMEA_MP_TEST_TRACE_ID' then null else
disbursement.bank_trace_id end as bank_trace_id,
    case when disbursement.bank_trace_id = 'EMEA_MP_TEST_TRACE_ID' then null else
disbursement.invoice_date end as disbursement_date,
    disbursement.billing_event_id as disbursement_id,
    -- We will use disbursement_id_or_invoiced as part of the PK, so it cannot be null:
    coalesce(
        --empty value in Athena shows as '', change all '' value to null
        case when disbursement.billing_event_id = '' then null else
disbursement.billing_event_id end,
        '<invoiced>') as disbursement_id_or_invoiced,
    bl.broker_id,
    case
        when bl.invoice_id is null /* transaction_type = 'BALANCE_ADJUSTMENT' */
            then (select seller_account_id from seller_account) || ':' ||
cast(bl.invoice_date as varchar)
        else bl.buyer_transaction_reference_id
            || '-' || case when bl.agreement_id is null or bl.agreement_id = '' then ' ' else
bl.agreement_id end
            || '-' || case when bl.end_user_account_id is null or bl.end_user_account_id = ''
then ' ' else bl.end_user_account_id end
            || '-' || coalesce(cast(bl.usage_period_start_date as varchar), ' ')
            || '-' || coalesce(cast(bl.usage_period_end_date as varchar), ' ')
        end as internal_buyer_line_item_id,
    bl.buyer_transaction_reference_id <> bl.transaction_reference_id as
is_seller_invoice,
    case when bl.transaction_type = 'SELLER_REV_SHARE' and (select seller_account_id
from seller_account) <> bl.to_account_id then true else false end as is_cog,
    case when bl.transaction_type in('SELLER_REV_SHARE_CREDIT',
'SELLER_REV_SHARE_REFUND') and (select seller_account_id from seller_account) <>
bl.to_account_id then true else false end as is_cog_refund,
    --TODO: replace below logic once we can create a logic the identify reseller/
manufacturer without depending on agreement feed
    case when agreement.proposer_account_id <> (select seller_account_id from
seller_account) then true else false end as is_manufacturer_view_of_reseller
from
    billing_events_with_uni_temporal_data as bl
left join billing_events_with_uni_temporal_data as disbursement on
    disbursement.transaction_type like 'DISBURSEMENT%'

```

```

        and disbursement.action = 'DISBURSED'
        and disbursement.transaction_type IN ('DISBURSEMENT', 'DISBURSEMENT_FAILURE')
        and bl.disbursement_billing_event_id = disbursement.billing_event_id
    left join agreements_with_history as aggrement on
        bl.agreement_id = aggrement.agreement_id
        and bl.invoice_date >= aggrement.valid_from_adjusted
        and bl.invoice_date < aggrement.valid_to
    left join accounts_with_history awh on
        bl.to_account_id = awh.account_id
        and bl.invoice_date >= awh.valid_from
        and bl.invoice_date < awh.valid_to
where
    bl.transaction_type not like 'DISBURSEMENT%' and
    (bl.agreement_id is null or bl.agreement_id = ''
    or aggrement.agreement_id is not null)
),

-- listagg function in athena does not support partitioning, grouping here and then
-- joining to the main query
seller_invoice_list as (
    select
        internal_buyer_line_item_id,
        listagg(case when not is_seller_invoice then null else invoice_id end, ',') within
        group (order by case when not is_seller_invoice then null else cast(invoice_date as
        varchar) end) as seller_invoice_id_or_null,
        listagg(case when not is_seller_invoice then null else cast(invoice_date as
        varchar) end, ',') within group (order by case when not is_seller_invoice then null
        else cast(invoice_date as varchar) end) as seller_invoice_date_or_null
    from
        (
            -- listagg function in athena does not support ordering by another field when
            -- distinct is used,
            -- here we first select distinct invoices and then do the listagg order by
            invoice_date
            select distinct internal_buyer_line_item_id, is_seller_invoice, invoice_id,
            invoice_date
            from billing_event_with_business_flags) distinct_invoices
    group by internal_buyer_line_item_id
    order by internal_buyer_line_item_id
),

billing_event_with_categorized_transaction as (
-- Use the flags that were created in the previous transformation in more calculated
columns:

```

```
-- NOTE: This transformation has no joins and no window functions
select
    billing_event_id,
    end_user_account_id,
    agreement_id,
    proposer_account_id,
    offer_id,
    acceptor_account_id,
    case when is_cog or is_cog_refund then null else payer_account_id end as
payer_account_id,
    product_id,
    action,
    transaction_type,
    parent_billing_event_id,
    disbursement_billing_event_id,
    amount,
    currency,
    balance_impacting,
    invoice_date,
    payment_due_date,
    usage_period_start_date,
    usage_period_end_date,
    invoice_id,
    billing_address_id,
    transaction_reference_id,
    buyer_transaction_reference_id,
    bank_trace_id,
    disbursement_date,
    disbursement_id,
    disbursement_id_or_invoiced,
    broker_id,
    bl.internal_buyer_line_item_id,
    is_seller_invoice,
    is_cog,
    is_cog_refund,
    is_manufacturer_view_of_reseller,

    -- Buyer/seller columns:
    case when is_seller_invoice then null else invoice_id end as
buyer_invoice_id_or_null,
    seller_invoices.seller_invoice_id_or_null,
    case when is_seller_invoice then null else invoice_date end as
buyer_invoice_date_or_null,
    seller_invoices.seller_invoice_date_or_null,
```

```

-- Categorized amounts by transaction type:
case when transaction_type = 'SELLER_REV_SHARE' and not is_cog then amount else 0
end as gross_revenue,
case when transaction_type in ('SELLER_REV_SHARE_REFUND', 'SELLER_REV_SHARE_CREDIT')
and not is_cog_refund then amount else 0 end as gross_refund,
case when transaction_type = 'SELLER_REV_SHARE' and is_cog then amount else 0
end as cogs,
case when transaction_type in ('SELLER_REV_SHARE_REFUND', 'SELLER_REV_SHARE_CREDIT')
and is_cog_refund then amount else 0 end as cogs_refund,
case when transaction_type = 'AWS_REV_SHARE' then amount else 0 end as
aws_rev_share,
case when transaction_type in ('AWS_REV_SHARE_REFUND', 'AWS_REV_SHARE_CREDIT') then
amount else 0 end as aws_refund_share,
case when transaction_type = 'AWS_TAX_SHARE' and not is_seller_invoice then
amount else 0 end as aws_tax_share, -- AWS tax share from _buyer_ invoice
case when transaction_type = 'AWS_TAX_SHARE' and is_seller_invoice then
amount else 0 end as aws_tax_share_listing_fee, -- AWS tax share from _seller_ invoice
case when transaction_type = 'AWS_TAX_SHARE_REFUND' and not is_seller_invoice
then amount else 0 end as aws_tax_share_refund,
case when transaction_type = 'AWS_TAX_SHARE_REFUND' and is_seller_invoice
then amount else 0 end as aws_tax_share_refund_listing_fee,
case when transaction_type = 'SELLER_TAX_SHARE' then amount else 0 end as
seller_tax_share,
case when transaction_type = 'SELLER_TAX_SHARE_REFUND' then amount else 0 end as
seller_tax_share_refund,
case when transaction_type = 'BALANCE_ADJUSTMENT' then amount else 0 end as
balance_adjustment,
case when transaction_type = 'SELLER_REV_SHARE_CREDIT' then amount else 0 end as
seller_rev_credit,
case when transaction_type = 'AWS_REV_SHARE_CREDIT' then amount else 0 end as
aws_ref_fee_credit
from
  billing_event_with_business_flags as bl
left join seller_invoice_list as seller_invoices
  on bl.internal_buyer_line_item_id = seller_invoices.internal_buyer_line_item_id
),

line_items_aggregated as (
-- This transformation has the only "group by" in all of these transformations.
-- NOTE: This transformation has no joins and no window functions
select
  internal_buyer_line_item_id,
  disbursement_id,

```



```

disbursement_id_or_invoiced,
product_id,
broker_id,
currency,
agreement_id,
proposer_account_id,
acceptor_account_id,
max(payer_account_id) as payer_account_id,
offer_id,
end_user_account_id,
usage_period_start_date,
usage_period_end_date,
max(payment_due_date) payment_due_date,
buyer_transaction_reference_id,
bank_trace_id,
disbursement_date,
max(billing_address_id) as billing_address_id,

-- Buyer/seller columns:
max(buyer_invoice_id_or_null) as buyer_invoice_id,
max(seller_invoice_id_or_null) as seller_invoice_id,
max(buyer_invoice_date_or_null) as buyer_invoice_date,
max(seller_invoice_date_or_null) as seller_invoice_date,

-- Categorized amounts by transaction type:
-- When disbursement_id_or_invoiced = '<invoiced>', these are invoiced amounts
-- When disbursement_id_or_invoiced <> '<invoiced>' these are disbursed amounts for
_this_ specific disbursement_id
sum(gross_revenue) as gross_revenue_this_disbursement_id_or_invoiced,
sum(gross_refund) as gross_refund_this_disbursement_id_or_invoiced,
sum(cogs) as cogs_this_disbursement_id_or_invoiced,
sum(cogs_refund) as cogs_refund_this_disbursement_id_or_invoiced,
sum(aws_rev_share) as aws_rev_share_this_disbursement_id_or_invoiced,
sum(aws_refund_share) as aws_refund_share_this_disbursement_id_or_invoiced,
sum(aws_tax_share) as aws_tax_share_this_disbursement_id_or_invoiced,
sum(aws_tax_share_listing_fee) as
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,
sum(aws_tax_share_refund) as aws_tax_share_refund_this_disbursement_id_or_invoiced,
sum(aws_tax_share_refund_listing_fee) as
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,
sum(seller_tax_share) as seller_tax_share_this_disbursement_id_or_invoiced,
sum(seller_tax_share_refund) as
seller_tax_share_refund_this_disbursement_id_or_invoiced,
sum(balance_adjustment) as balance_adjustment_this_disbursement_id_or_invoiced,

```

```

    sum(seller_rev_credit) as seller_rev_credit_this_disbursement_id_or_invoiced,
    sum(aws_ref_fee_credit) as aws_ref_fee_credit_this_disbursement_id_or_invoiced
from
    billing_event_with_categorized_transaction as billing_categorized
group by
    internal_buyer_line_item_id,
    disbursement_id,
    disbursement_id_or_invoiced,
    broker_id,
    -- The following columns are included the in group by but they are intentionally
    omitted from the PK.
    -- These columns should have the _same_ values for each record in the PK.
    product_id,
    currency,
    agreement_id,
    proposer_account_id,
    acceptor_account_id,
    offer_id,
    end_user_account_id,
    usage_period_start_date,
    usage_period_end_date,
    buyer_transaction_reference_id,
    bank_trace_id,
    disbursement_date
),

-- listagg function in athena does not support partitioning, grouping here and then
joining to the main query
disbursement_list as (
    select
        internal_buyer_line_item_id,
        listagg(cast(disbursement_date as varchar),',') within group (order by
        cast(disbursement_date as varchar)) as disbursement_date_list,
        listagg(bank_trace_id,',') within group (order by cast(disbursement_date as
        varchar)) as disburse_bank_trace_id_list
    from (
        -- listagg function in athena does not support ordering by another field when
        distinct is used,
        -- here we first select distinct bank_trace_ids and then do the listagg order by
        disbursement_date
        select distinct internal_buyer_line_item_id, disbursement_date, bank_trace_id
    from billing_event_with_business_flags) distinct_disbursements
    group by internal_buyer_line_item_id
    order by internal_buyer_line_item_id

```

```
),  
  
line_items_with_window_functions as (  
--add flag next step compare gross_revenue and gross_revenue_disbursed or gross_refund  
and gross_refund_disbursed  
select  
    line_item.internal_buyer_line_item_id,  
    disbursement_id,  
    disbursement_id_or_invoiced,  
    product_id,  
    broker_id,  
    currency,  
    agreement_id,  
    proposer_account_id,  
    acceptor_account_id,  
    -- when there's aws_rev_Share adjustment/refund to a seller_rev_share invoice, it  
    can happen that for the same aws_rev_share invoice_id, there are multiple disbursement  
    events,  
    -- using windows function to map payer_account_id of seller_rev_share to all  
    corresponding aws_rev_Share  
    max(payer_account_id) over (partition by line_item.internal_buyer_line_item_id) as  
    payer_account_id,  
    offer_id,  
    end_user_account_id,  
    usage_period_start_date,  
    usage_period_end_date,  
    payment_due_date,  
    bank_trace_id,  
    disbursement_date,  
    billing_address_id,  
  
    -- Buyer/seller columns:  
    max(buyer_invoice_id) over (partition by line_item.internal_buyer_line_item_id) as  
    buyer_invoice_id,  
    seller_invoice_id,  
    max(buyer_invoice_date) over (partition by line_item.internal_buyer_line_item_id)  
    as buyer_invoice_date,  
    seller_invoice_date,  
  
    -- When disbursement_id_or_invoiced = '<invoiced>', these are actually invoiced  
    amounts  
    -- When disbursement_id_or_invoiced <> '<invoiced>' these are disbursed amounts for  
    _this_ specific disbursement_id  
    gross_revenue_this_disbursement_id_or_invoiced,
```

```

gross_refund_this_disbursement_id_or_invoiced,
cogs_this_disbursement_id_or_invoiced,
cogs_refund_this_disbursement_id_or_invoiced,
aws_rev_share_this_disbursement_id_or_invoiced,
aws_refund_share_this_disbursement_id_or_invoiced,
aws_tax_share_this_disbursement_id_or_invoiced,
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,
aws_tax_share_refund_this_disbursement_id_or_invoiced,
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,
seller_tax_share_this_disbursement_id_or_invoiced,
seller_tax_share_refund_this_disbursement_id_or_invoiced,
balance_adjustment_this_disbursement_id_or_invoiced,
seller_rev_credit_this_disbursement_id_or_invoiced,
aws_ref_fee_credit_this_disbursement_id_or_invoiced,
-- IMPORTANT: All window functions partitioned by internal_buyer_line_item_id:

-- Invoiced amounts, categorized by transaction type:
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
gross_revenue_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end)over
(partition by line_item.internal_buyer_line_item_id) as gross_revenue_invoiced,
    sum(case when disbursement_id_or_invoiced = '<invoiced>' then
gross_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as gross_refund_invoiced,
    sum(case when disbursement_id_or_invoiced = '<invoiced>' then
cogs_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_invoiced,
    sum(case when disbursement_id_or_invoiced = '<invoiced>' then
cogs_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_refund_invoiced,
    sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_rev_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_rev_share_invoiced,
    sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_refund_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
aws_refund_share_invoiced,
    sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_tax_share_invoiced,
    sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_listing_fee_invoiced,

```

```

sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_listing_fee_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
seller_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
seller_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_refund_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
balance_adjustment_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
balance_adjustment_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
seller_rev_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_rev_credit_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_ref_fee_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_ref_fee_credit_invoiced,

-- Total disbursed amounts (for all disbursement_id values), categorized by
transaction type:
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
gross_revenue_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as gross_revenue_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
gross_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as gross_refund_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
cogs_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
cogs_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_refund_disbursed,

```

```

sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
aws_rev_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_rev_share_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
aws_refund_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
aws_refund_share_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
aws_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_tax_share_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_listing_fee_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
aws_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_listing_fee_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
seller_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
seller_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_refund_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
balance_adjustment_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
balance_adjustment_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
seller_rev_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_rev_credit_disbursed,
sum(case when disbursement_id_or_invoiced <> '<invoiced>' then
aws_ref_fee_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_ref_fee_credit_disbursed,

-- aggregate multiple disbursement

```

```

    max(disbursement_date) over (partition by line_item.internal_buyer_line_item_id) as
    last_disbursement_date,
    first_value(case when disbursement_id_or_invoiced =
'<invoiced>' then null else disbursement_id_or_invoiced end)
over(partition by line_item.internal_buyer_line_item_id order by
coalesce(disbursement_date,cast('1900-01-01' as timestamp)) desc rows between
unbounded preceding and unbounded following) as last_disbursement_id,
    first_value(bank_trace_id) over (partition by line_item.internal_buyer_line_item_id
order by coalesce(disbursement_date,cast('1900-01-01' as timestamp)) desc rows between
unbounded preceding and unbounded following) as last_disburse_bank_trace_id,
    disb_list.disbursement_date_list,
    disb_list.disburse_bank_trace_id_list
from
    line_items_aggregated as line_item
    left join disbursement_list disb_list
        on line_item.internal_buyer_line_item_id = disb_list.internal_buyer_line_item_id
),

cpo_offer_id as (
    select
        -- Channel partner offers do not exist in offertargetfeed_v1 table (as per legal
        requirement), causing cppo offer be defined as 'Public' in previous step, we will
        convert them back to 'Private' in next step
        offer_id
    from
        offers_with_uni_temporal_data
    where
        -- seller_account_id is null means the ISV owns the offer
        seller_account_id is not null
        and seller_account_id <> (select seller_account_id from seller_account)
    group by
        offer_id
),

line_items_with_window_functions_enrich_offer_product_address as (
    select
        internal_buyer_line_item_id,
        disbursement_id,
        disbursement_id_or_invoiced,
        line.product_id,
        legacy_product.legacy_id as legacy_product_id,
        products.title as product_title,
        line.broker_id,
        line.currency,

```

```

    line.end_user_account_id,
    acc_enduser.encrypted_account_id as end_user_encrypted_account_id,
    acc_enduser.aws_account_id as end_user_aws_account_id,
    acc_payer.aws_account_id as payer_aws_account_id,
    acc_payer.encrypted_account_id payer_encrypted_account_id,
    line.agreement_id,
    agreement.agreement_revision,
    line.proposer_account_id,
    case when offer.offer_id like 'aiqoffer-%' then null else agreement.start_date end
as Agreement_Start_Date,
    case when offer.offer_id like 'aiqoffer-%' then null else agreement.end_date end as
Agreement_End_Date,
    case when offer.offer_id like 'aiqoffer-%' then null else agreement.acceptance_date
end as Agreement_Acceptance_Date,
    case when offer.offer_id like 'aiqoffer-%' then null else agreement.valid_from end
as agreement_updated_date,
    case when offer.offer_id like 'aiqoffer-%' then null else
line.usage_period_start_date end as Usage_Period_Start_Date,
    case when offer.offer_id like 'aiqoffer-%' then null else
line.usage_period_end_date end as Usage_Period_End_Date,

    line.acceptor_account_id,
    acc_subscriber.aws_account_id as subscriber_aws_account_id,
    acc_subscriber.encrypted_account_id as subscriber_encrypted_account_id,
    offer.offer_id,
    case
        when offer.offer_id in (
            select distinct offer_id
            from cppo_offer_id)
        then 'Private'
        else offer.offer_target
    end as offer_target,
    offer.name offer_name,
    offer.opportunity_name offer_opportunity_name,
    offer.opportunity_description offer_opportunity_description,
    offer.opportunity_id,
    payment_due_date,
    line.bank_trace_id,
    disbursement_date,
    billing_address_id,
    buyer_invoice_id,
    seller_invoice_id,
    buyer_invoice_date,
    seller_invoice_date,

```



```
gross_revenue_this_disbursement_id_or_invoiced,  
gross_refund_this_disbursement_id_or_invoiced,  
cogs_this_disbursement_id_or_invoiced,  
cogs_refund_this_disbursement_id_or_invoiced,  
aws_rev_share_this_disbursement_id_or_invoiced,  
aws_refund_share_this_disbursement_id_or_invoiced,  
aws_tax_share_this_disbursement_id_or_invoiced,  
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,  
seller_tax_share_this_disbursement_id_or_invoiced,  
seller_tax_share_refund_this_disbursement_id_or_invoiced,  
balance_adjustment_this_disbursement_id_or_invoiced,  
seller_rev_credit_this_disbursement_id_or_invoiced,  
aws_ref_fee_credit_this_disbursement_id_or_invoiced,  
gross_revenue_invoiced,  
gross_refund_invoiced,  
cogs_invoiced,  
cogs_refund_invoiced,  
aws_rev_share_invoiced,  
aws_refund_share_invoiced,  
aws_tax_share_invoiced,  
aws_tax_share_listing_fee_invoiced,  
aws_tax_share_refund_invoiced,  
aws_tax_share_refund_listing_fee_invoiced,  
seller_tax_share_invoiced,  
seller_tax_share_refund_invoiced,  
balance_adjustment_invoiced,  
seller_rev_credit_invoiced,  
aws_ref_fee_credit_invoiced,  
gross_revenue_disbursed,  
gross_refund_disbursed,  
cogs_disbursed,  
cogs_refund_disbursed,  
aws_rev_share_disbursed,  
aws_refund_share_disbursed,  
aws_tax_share_disbursed,  
aws_tax_share_listing_fee_disbursed,  
aws_tax_share_refund_disbursed,  
aws_tax_share_refund_listing_fee_disbursed,  
seller_tax_share_disbursed,  
seller_tax_share_refund_disbursed,  
balance_adjustment_disbursed,  
seller_rev_credit_disbursed,
```

```

aws_ref_fee_credit_disbursed,
last_disbursement_date,
last_disbursement_id,
last_disburse_bank_trace_id,
disbursement_date_list,
disburse_bank_trace_id_list,
products.product_code,
acc_products.aws_account_id as manufacturer_aws_account_id,
products.manufacturer_account_id,
--add subscriber and payer addressID, payer address preference order: tax address >
billing address > mailing address, subscriber address preference order: tax address >
mailing address
coalesce (
    --empty value in Athena shows as '', change all '' value to null in order to
follow the preference order logic above
    case when acc_subscriber.tax_address_id = '' then null else
acc_subscriber.tax_address_id end,
    case when acc_subscriber.mailing_address_id = '' then null else
acc_subscriber.mailing_address_id end) as subscriber_address_id,
coalesce (
    case when acc_payer.tax_address_id = '' then null else acc_payer.tax_address_id
end,
    case when line.billing_address_id = '' then null else line.billing_address_id
end,
    case when acc_payer.mailing_address_id = '' then null else
acc_payer.mailing_address_id end) as payer_address_id,
coalesce (
    case when acc_enduser.tax_address_id = '' then null else
acc_enduser.tax_address_id end,
    case when line.billing_address_id = '' then null else line.billing_address_id
end,
    case when acc_enduser.mailing_address_id = '' then null else
acc_enduser.mailing_address_id end) as end_user_address_id
from
    line_items_with_window_functions as line
left join agreements_with_history as agreement on
    line.agreement_id = agreement.agreement_id and line.buyer_invoice_date >=
agreement.valid_from_adjusted and line.buyer_invoice_date<agreement.valid_to
left join offers_with_history_with_target_type as offer on
    line.offer_id = offer.offer_id and line.buyer_invoice_date >= offer.valid_from
and line.buyer_invoice_date<offer.valid_to
left join products_with_history as products on
    line.product_id = products.product_id and line.buyer_invoice_date >=
products.valid_from_adjusted and line.buyer_invoice_date<products.valid_to

```

```

left join legacy_products as legacy_product on
    line.product_id = legacy_product.new_id
left join accounts_with_history_with_company_name as acc_payer on
    line.payer_account_id = acc_payer.account_id and line.buyer_invoice_date >=
acc_payer.valid_from and line.buyer_invoice_date<acc_payer.valid_to
left join accounts_with_history_with_company_name as acc_enduser on
    line.end_user_account_id = acc_enduser.account_id and line.buyer_invoice_date
>= acc_enduser.valid_from and line.buyer_invoice_date<acc_enduser.valid_to
left join accounts_with_history_with_company_name as acc_subscriber on
    line.acceptor_account_id = acc_subscriber.account_id
and line.buyer_invoice_date >= acc_subscriber.valid_from and
line.buyer_invoice_date<acc_subscriber.valid_to
left join accounts_with_history_with_company_name as acc_products on
    products.manufacturer_account_id = acc_products.account_id
and line.buyer_invoice_date >= acc_products.valid_from and
line.buyer_invoice_date<acc_products.valid_to

),

line_items_with_window_functions_enrich_offer_product_address_name as (
    select
        line.internal_buyer_line_item_id,
        disbursement_id,
        disbursement_id_or_invoiced,
        product_id,
        legacy_product_id,
        product_title,
        broker_id,
        currency,
        end_user_address_id,
        end_user_account_id,
        end_user_encrypted_account_id,
        end_user_aws_account_id,
        add_enduser.company_name end_user_company_name,
        add_enduser.email_domain end_user_email_domain,
        add_enduser.city end_user_city,
        add_enduser.state_or_region end_user_state,
        add_enduser.country_code end_user_country,
        add_enduser.postal_code end_user_postal_code,
        payer_aws_account_id,
        payer_encrypted_account_id,
        payer_address_id,
        add_payer.company_name payer_company_name,
        add_payer.email_domain payer_email_domain,

```

```
add_payer.city payer_city,
add_payer.state_or_region payer_state,
add_payer.country_code payer_country,
add_payer.postal_code payer_postal_code,
agreement_id,
agreement_revision,
agreement_start_date,
agreement_end_date,
agreement_acceptance_date,
agreement_updated_date,
case when proposer_account_id = (select seller_account_id from seller_account) then
null else acc_proposer.aws_account_id end as reseller_aws_account_id,
case when proposer_account_id = (select seller_account_id from seller_account) then
null else acc_proposer.mailing_company_name end as reseller_company_name,
usage_period_start_date,
usage_period_end_date,
proposer_account_id,
acc_proposer.aws_account_id as proposer_aws_account_id,
acceptor_account_id,
subscriber_aws_account_id,
subscriber_encrypted_account_id,
subscriber_address_id,
add_subscriber.company_name subscriber_company_name,
add_subscriber.email_domain subscriber_email_domain,
add_subscriber.city subscriber_city,
add_subscriber.state_or_region subscriber_state,
add_subscriber.country_code subscriber_country,
add_subscriber.postal_code subscriber_postal_code,
offer_id,
offer_target,
offer_name,
offer_opportunity_name,
offer_opportunity_description,
opportunity_id,
payment_due_date,
bank_trace_id,
disbursement_date,
billing_address_id,
max(buyer_invoice_id)as buyer_invoice_id,
max(seller_invoice_id)as seller_invoice_id,
max(buyer_invoice_date)as buyer_invoice_date,
max(seller_invoice_date)as seller_invoice_date,
gross_revenue_this_disbursement_id_or_invoiced,
gross_refund_this_disbursement_id_or_invoiced,
```

```
cogs_this_disbursement_id_or_invoiced,
cogs_refund_this_disbursement_id_or_invoiced,
aws_rev_share_this_disbursement_id_or_invoiced,
aws_refund_share_this_disbursement_id_or_invoiced,
aws_tax_share_this_disbursement_id_or_invoiced,
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,
aws_tax_share_refund_this_disbursement_id_or_invoiced,
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,
seller_tax_share_this_disbursement_id_or_invoiced,
seller_tax_share_refund_this_disbursement_id_or_invoiced,
balance_adjustment_this_disbursement_id_or_invoiced,
seller_rev_credit_this_disbursement_id_or_invoiced,
aws_ref_fee_credit_this_disbursement_id_or_invoiced,
(gross_revenue_this_disbursement_id_or_invoiced +
gross_refund_this_disbursement_id_or_invoiced +
aws_rev_share_this_disbursement_id_or_invoiced +
aws_refund_share_this_disbursement_id_or_invoiced +
seller_tax_share_this_disbursement_id_or_invoiced +
seller_tax_share_refund_this_disbursement_id_or_invoiced
+ cogs_this_disbursement_id_or_invoiced +
cogs_refund_this_disbursement_id_or_invoiced +
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced +
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced) as
seller_net_revenue_this_disbursement_id_or_invoiced,
gross_revenue_invoiced,
gross_refund_invoiced,
cogs_invoiced,
cogs_refund_invoiced,
aws_rev_share_invoiced,
aws_refund_share_invoiced,
aws_tax_share_invoiced,
aws_tax_share_listing_fee_invoiced,
aws_tax_share_refund_invoiced,
aws_tax_share_refund_listing_fee_invoiced,
seller_tax_share_invoiced,
seller_tax_share_refund_invoiced,
balance_adjustment_invoiced,
seller_rev_credit_invoiced,
aws_ref_fee_credit_invoiced,
gross_revenue_disbursed,
gross_refund_disbursed,
cogs_disbursed,
cogs_refund_disbursed,
aws_rev_share_disbursed,
```

```

aws_refund_share_disbursed,
aws_tax_share_disbursed,
aws_tax_share_listing_fee_disbursed,
aws_tax_share_refund_disbursed,
aws_tax_share_refund_listing_fee_disbursed,
seller_tax_share_disbursed,
seller_tax_share_refund_disbursed,
balance_adjustment_disbursed,
seller_rev_credit_disbursed,
aws_ref_fee_credit_disbursed,
(gross_revenue_invoiced + gross_revenue_disbursed) as uncollected_gross_revenue,
-- net revenue = gross revenue - listing fee - tax - cogs
(gross_revenue_invoiced + gross_refund_invoiced + aws_rev_share_invoiced
+ aws_refund_share_invoiced + seller_tax_share_invoiced +
seller_tax_share_refund_invoiced + cogs_invoiced + cogs_refund_invoiced +
aws_tax_share_listing_fee_invoiced + aws_tax_share_refund_listing_fee_invoiced) as
seller_net_revenue,
(gross_revenue_invoiced + gross_refund_invoiced + aws_rev_share_invoiced
+ aws_refund_share_invoiced + seller_tax_share_invoiced +
seller_tax_share_refund_invoiced + cogs_invoiced + cogs_refund_invoiced +
aws_tax_share_listing_fee_invoiced + aws_tax_share_refund_listing_fee_invoiced
+ gross_revenue_disbursed + gross_refund_disbursed + aws_rev_share_disbursed
+ aws_refund_share_disbursed + seller_tax_share_disbursed +
seller_tax_share_refund_disbursed + cogs_disbursed + cogs_refund_disbursed +
aws_tax_share_listing_fee_disbursed + aws_tax_share_refund_listing_fee_disbursed) as
uncollected_seller_net_revenue,
last_disbursement_date,
last_disbursement_id,
last_disburse_bank_trace_id,
disbursement_date_list,
disburse_bank_trace_id_list,
product_code,
manufacturer_aws_account_id,
manufacturer_account_id,
acc_manu.mailing_company_name as manufacturer_company_name,
cast(null as varchar) as AR_Period,
case
  when (
    (gross_revenue_invoiced <> 0 and gross_revenue_invoiced = -1 *
gross_revenue_disbursed)
    or (gross_refund_invoiced <> 0 and gross_refund_invoiced = -1 *
gross_refund_disbursed)
    or (balance_adjustment_invoiced <> 0 and balance_adjustment_invoiced = -1 *
balance_adjustment_disbursed)

```

```

        or (seller_tax_share_refund_invoiced <> 0 and seller_tax_share_refund_invoiced
= -1 * seller_tax_share_refund_disbursed)
        or (gross_revenue_invoiced = 0 and gross_refund_invoiced = 0 and
balance_adjustment_invoiced = 0 and seller_tax_share_refund_invoiced = 0 and
last_disbursement_id is not null)) then 'Yes'
        when gross_revenue_disbursed = 0 and gross_refund_disbursed = 0 and
balance_adjustment_disbursed = 0 and seller_tax_share_disbursed = 0 and
seller_tax_share_refund_disbursed = 0 then 'No'
        else 'Partial'
    end as Disbursement_Flag
from line_items_with_window_functions_enrich_offer_product_address as line
left join accounts_with_history_with_company_name as acc_manu on
    line.manufacturer_account_id = acc_manu.account_id and line.buyer_invoice_date >=
acc_manu.valid_from_adjusted and line.buyer_invoice_date <= acc_manu.valid_to
left join accounts_with_history_with_company_name as acc_proposer on
    line.proposer_account_id = acc_proposer.account_id and line.buyer_invoice_date >=
acc_proposer.valid_from and line.buyer_invoice_date<acc_proposer.valid_to
left join address_with_latest_revision as add_payer on
    line.payer_address_id = add_payer.address_id
left join address_with_latest_revision as add_subscriber on
    line.subscriber_address_id = add_subscriber.address_id
left join address_with_latest_revision as add_enduser on
    line.end_user_address_id = add_enduser.address_id
group by
    line.internal_buyer_line_item_id,
    disbursement_id,
    disbursement_id_or_invoiced,
    product_id,
    legacy_product_id,
    product_title,
    broker_id,
    currency,
    end_user_address_id,
    end_user_account_id,
    end_user_encrypted_account_id,
    end_user_aws_account_id,
    add_enduser.company_name,
    add_enduser.email_domain,
    add_enduser.city,
    add_enduser.state_or_region,
    add_enduser.country_code,
    add_enduser.postal_code,
    payer_aws_account_id,
    payer_encrypted_account_id,

```

```
payer_address_id,  
add_payer.company_name,  
add_payer.email_domain,  
add_payer.city,  
add_payer.state_or_region,  
add_payer.country_code,  
add_payer.postal_code,  
agreement_id,  
agreement_revision,  
case when proposer_account_id = (select seller_account_id from seller_account) then  
null else acc_proposer.aws_account_id end,  
case when proposer_account_id = (select seller_account_id from seller_account) then  
null else acc_proposer.mailing_company_name end,  
agreement_start_date,  
agreement_end_date,  
agreement_acceptance_date,  
agreement_updated_date,  
usage_period_start_date,  
usage_period_end_date,  
acceptor_account_id,  
subscriber_aws_account_id,  
subscriber_encrypted_account_id,  
subscriber_address_id,  
add_subscriber.company_name,  
add_subscriber.email_domain,  
add_subscriber.city,  
add_subscriber.state_or_region,  
add_subscriber.country_code,  
add_subscriber.postal_code,  
offer_id,  
offer_target,  
offer_name,  
offer_opportunity_name,  
offer_opportunity_description,  
opportunity_id,  
payment_due_date,  
bank_trace_id,  
disbursement_date,  
billing_address_id,  
gross_revenue_this_disbursement_id_or_invoiced,  
gross_refund_this_disbursement_id_or_invoiced,  
cogs_this_disbursement_id_or_invoiced,  
cogs_refund_this_disbursement_id_or_invoiced,  
aws_rev_share_this_disbursement_id_or_invoiced,
```



```
aws_refund_share_this_disbursement_id_or_invoiced,  
aws_tax_share_this_disbursement_id_or_invoiced,  
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,  
seller_tax_share_this_disbursement_id_or_invoiced,  
seller_tax_share_refund_this_disbursement_id_or_invoiced,  
balance_adjustment_this_disbursement_id_or_invoiced,  
seller_rev_credit_this_disbursement_id_or_invoiced,  
aws_ref_fee_credit_this_disbursement_id_or_invoiced,  
gross_revenue_invoiced,  
gross_refund_invoiced,  
cogs_invoiced,  
cogs_refund_invoiced,  
aws_rev_share_invoiced,  
aws_refund_share_invoiced,  
aws_tax_share_invoiced,  
aws_tax_share_listing_fee_invoiced,  
aws_tax_share_refund_invoiced,  
aws_tax_share_refund_listing_fee_invoiced,  
seller_tax_share_invoiced,  
seller_tax_share_refund_invoiced,  
balance_adjustment_invoiced,  
seller_rev_credit_invoiced,  
aws_ref_fee_credit_invoiced,  
gross_revenue_disbursed,  
gross_refund_disbursed,  
cogs_disbursed,  
cogs_refund_disbursed,  
aws_rev_share_disbursed,  
aws_refund_share_disbursed,  
aws_tax_share_disbursed,  
aws_tax_share_listing_fee_disbursed,  
aws_tax_share_refund_disbursed,  
aws_tax_share_refund_listing_fee_disbursed,  
seller_tax_share_disbursed,  
seller_tax_share_refund_disbursed,  
balance_adjustment_disbursed,  
seller_rev_credit_disbursed,  
aws_ref_fee_credit_disbursed,  
last_disbursement_date,  
last_disbursement_id,  
last_disburse_bank_trace_id,  
disbursement_date_list,
```

```

    disburse_bank_trace_id_list,
    product_code,
    manufacturer_aws_account_id,
    manufacturer_account_id,
    acc_manu.mailing_company_name,
    proposer_account_id,
    acc_proposer.aws_account_id
),

billed_revenue as (
    select
        -----
        -- Invoice Info --
        -----
        buyer_invoice_date as Invoice_Date,
        Payment_Due_Date as Payment_Due_Date,
        concat(
            'Net ',
            case
                when abs(date_diff('Day', Payment_due_date, buyer_invoice_date))>180 then
'180+'
                else cast(abs(date_diff('Day', Payment_due_date, buyer_invoice_date)) as
varchar)
            end,
            ' days'
        ) as payment_terms,
        buyer_invoice_id as Invoice_ID,
        coalesce(
            --empty value in Athena shows as '', change all '' value to null
            case when seller_invoice_id = '' then null else seller_invoice_id end,
            'Not applicable') as Listing_Fee_Invoice_ID,

        -----
        --End user Information --
        -----
        coalesce(
            --empty value in Athena shows as '', change all '' value to null
            case when End_User_Company_Name = '' then null else End_User_Company_Name end,
            'Not available') as End_User_Company_Name,
        End_User_AWS_Account_ID,
        End_User_Encrypted_Account_ID,
        End_User_Email_Domain,
        End_User_City,
        End_User_State as End_User_State_or_Region,

```

```

End_User_Country,
End_User_Postal_Code,
End_User_Address_ID,

-----
--Subscriber Information --
-----

case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_Company_Name is null or Subscriber_Company_Name = '' then 'Not
provided'
  else Subscriber_Company_Name
  end as Subscriber_Company_Name,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  else Subscriber_AWS_Account_ID
  end as Subscriber_AWS_Account_ID,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  else Subscriber_Encrypted_Account_ID
  end as Subscriber_Encrypted_Account_ID,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_Email_Domain is null or Subscriber_Email_Domain = '' then 'Not
provided'
  else Subscriber_Email_Domain
  end as Subscriber_Email_Domain,
case
  when Agreement_id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_City is null or Subscriber_City = '' then 'Not provided'
  else Subscriber_City
  end as Subscriber_City,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_State is null or Subscriber_State = '' then 'Not provided'
  else Subscriber_State
  end as Subscriber_State_or_Region,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_Country is null or Subscriber_Country = '' then 'Not provided'
  else Subscriber_Country
  end as Subscriber_Country,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'

```

```

        when Subscriber_Postal_Code is null or Subscriber_Postal_Code = '' then 'Not
provided'
        else Subscriber_Postal_Code
        end as Subscriber_Postal_Code,
    case
        when Agreement_ID is null or Agreement_ID = '' then 'Not available'
        when Subscriber_Address_ID is null or Subscriber_Address_ID = '' then 'Not
provided'
        else Subscriber_Address_ID
        end as Subscriber_Address_ID,

    -----
    -- Procurement Info --
    -----

    -- product title at time of invoice. It is possible that the title changes over
time and therefore there may be multiple product titles mapped to a single product id.
    coalesce(
        --empty value in Athena shows as '', change all '' value to null
        case when Product_Title = '' then null else Product_Title end,
        'Not provided') as Product_Title,
    -- offer name at time of invoice. It is possible that the name changes over time
therefore there may be multiple offer names mapped to a single offer id.
    case
        when Agreement_Id is null or Agreement_ID = '' then 'Not available'
        when (Offer_Name is null or Offer_Name = '') and Offer_Target = 'Public' then
'Not applicable'
        else Offer_Name
        end as Offer_Name,
    case
        when Agreement_Id is null or Agreement_ID = ''
        then 'Not available'
        else Offer_ID
        end as Offer_ID,
    -- offer target at time of invoice.,
    case
        when Agreement_Id is null or Agreement_ID = '' then 'Not available'
        else Offer_Target
        end as Offer_Visibility,
    coalesce(
        --empty value in Athena shows as '', change all '' value to null
        case when Agreement_ID = '' then null else Agreement_ID end,
        'Not available') as Agreement_ID,
    Agreement_Start_Date,
    Agreement_Acceptance_Date,

```

```

Agreement_End_Date,
Usage_Period_Start_Date,
Usage_Period_End_Date,

-----
-- Disbursement Info --
-----

case
  when Disbursement_Flag = 'Yes' then 'Disbursed'
  when Disbursement_Flag = 'No' then 'Not Disbursed'
  else 'Other'
end as Disbursement_Status,
last_disbursement_date as disbursement_date,
case
  when Disbursement_Flag = 'No' then 'Not applicable'
  when disburse_bank_trace_id_list is null or disburse_bank_trace_id_list = '' then
'Not available'
  else disburse_bank_trace_id_list
end as disburse_bank_trace_id,

-----
-- Revenues --
-----

-- We are rounding the sums using 2 decimal precision
-- Note that the rounding method might differ between SQL implementations.
-- The monthly revenue report is using RoundingMode.HALF_UP. This might create tiny
discrepancies between this SQL output
-- and the legacy report
round(gross_revenue_invoiced,2) as Gross_Revenue,
round(gross_refund_invoiced,2) as Gross_Refund,
round(aws_rev_share_invoiced,2) as Listing_Fee,
round(aws_refund_share_invoiced,2) as Listing_Fee_Refund,
truncate(
  case
    when gross_revenue_invoiced != 0 then abs(aws_rev_share_invoiced/
gross_revenue_invoiced)
    when gross_refund_invoiced != 0 then abs(aws_refund_share_invoiced/
gross_refund_invoiced)
    else 0
  end
  ,4) as Listing_Fee_Percentage,
round(seller_tax_share_invoiced,2) as Seller_Tax_Share,
round(seller_tax_share_refund_invoiced,2) as Seller_Tax_Share_Refund,
round(aws_tax_share_invoiced,2) as AWS_Tax_Share,

```

```

round(aws_tax_share_refund_invoiced,2) as AWS_Tax_Share_Refund,
round(aws_tax_share_listing_fee_invoiced,2) as AWS_Tax_Share_Listing_Fee,
round(aws_tax_share_refund_listing_fee_invoiced,2) as
AWS_Tax_Share_Refund_Listing_Fee,
round(cogs_invoiced,2) as Wholesale_cost,
round(cogs_refund_invoiced,2) as Wholesale_cost_Refund,
round(seller_net_revenue,2) as Seller_Net_Revenue,
currency as Currency,

substring(internal_buyer_line_item_id,1,strpos(internal_buyer_line_item_id,'-')-1)
as Transaction_Reference_ID,
broker_id as AWS_seller_of_record,

-----
-- Resale info --
-----
case
  when Opportunity_Id is null or Opportunity_Id = '' then
    case
      when Offer_Target = 'Public' then 'Not applicable'
      when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
      else null
    end
  else Opportunity_Id
end as Resale_authorization_ID,
case
  when Offer_Opportunity_Name is null or Offer_Opportunity_Name = '' then
    case
      when Offer_Target = 'Public' then 'Not applicable'
      when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
      else null
    end
  else Offer_Opportunity_Name
end as Resale_authorization_name,
case
  when Offer_Opportunity_Description is null or Offer_Opportunity_Description = ''
then
    case
      when Offer_Target = 'Public' then 'Not applicable'
      when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not null
and Agreement_Id != '') then 'Not applicable'
      else null

```

```

        end
        else Offer_Opportunity_Name
    end as Resale_authorization_description,
    case
        when (Reseller_AWS_Account_ID is not null and Reseller_AWS_Account_ID != '')
            and (Reseller_Company_Name is null or Reseller_Company_Name = '') then 'Not
available'
        when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
            and (opportunity_id is null or opportunity_id = '') then 'Not applicable'
        when (select seller_account_id from seller_account) <>
manufacturer_aws_account_id
            and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
        else Reseller_Company_Name
    end as Reseller_Company_Name,
    case
        when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
            and (Opportunity_Id is null or Opportunity_Id = '') then 'Not applicable'
        when (select seller_account_id from seller_account) <>
manufacturer_aws_account_id
            and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
        else Reseller_AWS_Account_ID
    end as Reseller_AWS_Account_ID,

-----
-- Payer Information --
-----
coalesce(
    --empty value in Athena shows as '', change all '' value to null
    case when Payer_Company_Name = '' then null else Payer_Company_Name end,
    'Not available') as Payer_Company_Name,
Payer_AWS_Account_ID,
Payer_Encrypted_Account_ID,
Payer_Email_Domain,
Payer_City,
Payer_State as Payer_State_or_Region,
Payer_Country,
Payer_Postal_Code,
Payer_Address_ID,

-----
-- ISV Information --
-----

```

```

    manufacturer_aws_account_id as ISV_Account_ID,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when Manufacturer_Company_Name = '' then null else Manufacturer_Company_Name
    end,
    'Not available') as ISV_Company_Name,

    -----
    -- Products info --
    -----
    Legacy_Product_ID,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when Product_ID = '' then null else Product_ID end,
      'Not provided') as Product_ID,
    Product_Code
  from
    line_items_with_window_functions_enrich_offer_product_address_name as line
  where disbursement_id_or_invoiced = '<invoiced>'

)

select *
from billed_revenue
where invoice_date >= date_add('DAY', -90, current_date)
--where invoice_date between cast('2023-01-01' as timestamp) and cast('2024-03-01' as
timestamp)

```

未払い請求書または支払い済み請求書

未払い請求書または支払い済み請求書を検索するには、次の例のような一連のクエリを実行できます。クエリは相互に構築され、集金と支払いレポートを作成します。図のように例を使用するか、データやユースケースに合わせてカスタマイズできます。

クエリのコメントは、クエリの動作と変更方法について説明しています。

```

-- Collections and disbursements report

-- General note: When running this query, we assume that the data ingested in the
database uses
-- two time axes (the valid_from column and the update_date column).

```



```
-- See documentation for more details: https://docs.aws.amazon.com/marketplace/latest/
userguide/data-feed.html#data-feed-details

-- An account_id has several valid_from dates (each representing a separate revision of
the data)
-- but because of bi-temporality, an account_id + valid_from tuple can appear multiple
times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
with accounts_with_uni_temporal_data as (
    select
        account_id,
        aws_account_id,
        encrypted_account_id,
        mailing_address_id,
        tax_address_id,
        tax_legal_name,
        from_iso8601_timestamp(valid_from) as valid_from,
        tax_registration_number
    from
        (
            select
                account_id,
                aws_account_id,
                encrypted_account_id,
                mailing_address_id,
                tax_address_id,
                tax_legal_name,
                valid_from,
                delete_date,
                tax_registration_number,
                row_number() over (partition by account_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
            from
                accountfeed_v1
        )
    where
        -- keep latest ...
        row_num = 1
        -- ... and remove the soft-deleted one.
        and (delete_date is null or delete_date = '')
),

accounts_with_history as (
    with accounts_with_history_with_extended_valid_from as (
```

```

select
    account_id,
    -- sometimes, this columns gets imported as a "bigint" and loses heading 0s ->
    casting to a char and re-adding heading 0s (if need be)
    substring('000000000000' || cast(aws_account_id as varchar), -12) as aws_account_id,
    encrypted_account_id,
    mailing_address_id,
    tax_address_id,
    tax_legal_name,
    -- The start time of account valid_from is extended to '1970-01-01 00:00:00',
because:
    -- ... in tax report transformations, some tax line items with invoice_date
cannot
    -- ... fall into the default valid time range of the associated account
case
    when lag(valid_from) over (partition by account_id order by valid_from asc) is
null
        then cast('1970-01-01 00:00:00' as timestamp)
        else valid_from
    end as valid_from
from accounts_with_uni_temporal_data
)
select
    account_id,
    aws_account_id,
    encrypted_account_id,
    mailing_address_id,
    tax_address_id,
    tax_legal_name,
    valid_from,
    coalesce(
        lead(valid_from) over (partition by account_id order by valid_from asc),
        cast('2999-01-01 00:00:00' as timestamp)
    ) as valid_to
from
    accounts_with_history_with_extended_valid_from
),

-- An address_id has several valid_from dates (each representing a separate revision of
the data)
-- but because of bi-temporality, an account_id + valid_from tuple can appear multiple
times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
address_with_uni_temporal_data as (

```

```
select
  from_iso8601_timestamp(valid_from) as valid_from,
  address_id,
  company_name,
  email_domain,
  country_code,
  state_or_region,
  city,
  postal_code,
  row_num
from
(
  select
    valid_from,
    update_date,
    delete_date,
    address_id,
    company_name,
    email_domain,
    country_code,
    state_or_region,
    city,
    postal_code,
    row_number() over (partition by address_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
  from
    addressfeed_v1
)
where
  -- keep latest ...
  row_num = 1
  -- ... and remove the soft-deleted one.
  and (delete_date is null or delete_date = '')
),

-- We are only interested in the most recent tuple (BTW: a given address is not
  supposed to change over time but when bugs ;-) so this query mainly does nothing)
address_with_latest_revision as (
  select
    valid_from,
    address_id,
    company_name,
    email_domain,
    country_code,
```

```
state_or_region,
city,
postal_code,
row_num_latest_revision
from
(
  select
    valid_from,
    address_id,
    company_name,
    email_domain,
    country_code,
    state_or_region,
    city,
    postal_code,
    row_number() over (partition by address_id order by valid_from desc) as
row_num_latest_revision
  from
    address_with_uni_temporal_data
)
where
  row_num_latest_revision = 1
),

accounts_with_history_with_company_name as (
  select
    awh.account_id,
    awh.aws_account_id,
    awh.encrypted_account_id,
    awh.mailing_address_id,
    awh.tax_address_id,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when address.company_name = '' then null else address.company_name end,
      awh.tax_legal_name) as mailing_company_name,
    address.email_domain,
    awh.valid_from,
    -- For BYOL, the agreement might be accepted (using some external non-AWS system or
manual process) days before
    -- that BYOL agreement is entered into AWS Marketplace by the buyer. Therefore, the
buyer is permitted to manually
    -- enter a backdated acceptance date, which might predate the point in time when
the account was created.
    -- To work around this, we need to adjust the valid_from of the account to be
```

```

-- earlier than the earliest possible backdated BYOL agreement acceptance date.
case
  when lag(awh.valid_from) over (partition by aws_account_id order by
awh.valid_from asc) is null
    then date_add('Day', -212, awh.valid_from)
    -- 212 is the longest delay between acceptance_date of the agreement and the
account start_date
    else awh.valid_from
  end as valid_from_adjusted,
  awh.valid_to
from accounts_with_history as awh
left join address_with_latest_revision as address on
  awh.mailing_address_id = address.address_id and awh.mailing_address_id is not null
),

-- An agreement_id has several valid_from dates (each representing an agreement
revision)
-- but because of bi-temporality, an agreement_id + valid_from tuple can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
agreements_with_uni_temporal_data as (
  select
    agreement_id,
    origin_offer_id,
    proposer_account_id,
    acceptor_account_id,
    agreement_revision,
    from_iso8601_timestamp(valid_from) as valid_from,
    from_iso8601_timestamp(start_date) as start_date,
    from_iso8601_timestamp(end_date) as end_date,
    from_iso8601_timestamp(acceptance_date) as acceptance_date,
    agreement_type,
    previous_agreement_id,
    agreement_intent
  from
    (
      select
        --empty value in Athena shows as '', change all '' value to null
        case when agreement_id = '' then null else agreement_id end as agreement_id,
        origin_offer_id,
        proposer_account_id,
        acceptor_account_id,
        agreement_revision,
        valid_from,

```

```

        delete_date,
        start_date,
        end_date,
        acceptance_date,
        agreement_type,
        previous_agreement_id,
        agreement_intent,
        row_number() over (partition by agreement_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
    from
        -- TODO change to agreementfeed_v1 when Agreement Feed is GA'ed
        agreementfeed
    )
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

agreements_with_history as (
    with agreements_with_window_functions as (
        select
            agreement_id,
            origin_offer_id as offer_id,
            proposer_account_id,
            acceptor_account_id,
            agreement_revision,
            start_date,
            end_date,
            acceptance_date,
            -- The start time of agreement valid_from is extended to '1970-01-01 00:00:00',
because:
            -- ... in usage report transformations, some usage line items with usage_date
cannot
            -- ... fall into the default valid time range of the associated agreement
case
                when lag(valid_from) over (PARTITION BY agreement_id order by valid_from asc)
is null
                then timestamp '1970-01-01 00:00:00'
                else valid_from
            end as valid_from,
            coalesce(
                lead(valid_from) over (partition by agreement_id order by valid_from asc),

```

```

        timestamp '2999-01-01 00:00:00'
    ) as valid_to,
    rank() over (partition by agreement_id order by valid_from asc) version,
    agreement_type,
    previous_agreement_id,
    agreement_intent
from
    agreements_with_uni_temporal_data
)
select
    agreement_id,
    offer_id,
    proposer_account_id,
    acceptor_account_id,
    agreement_revision,
    start_date,
    end_date,
    acceptance_date,
    valid_from,
    case
        when version=1 and valid_from < timestamp '2023-03-03 06:16:08.743' then
timestamp '1970-01-01'
        -- The following 60 minute adjustment is to handle special case where When
Renewal happens for a contract
        when version=1 then date_add('minute',-60,valid_from)
        else valid_from
    end as valid_from_adjusted,
    valid_to,
    agreement_type,
    previous_agreement_id,
    agreement_intent
from
    agreements_with_window_functions
),

-- An offer_id has several valid_from dates (each representing an offer revision)
-- but because of bi-temporality, an offer_id + valid_from tuple can appear multiple
times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
offers_with_uni_temporal_data as (
    select
        from_iso8601_timestamp(valid_from) as valid_from,
        from_iso8601_timestamp(update_date) as update_date,
        from_iso8601_timestamp(delete_date) as delete_date,

```

```
offer_id,
offer_revision,
name,
expiration_date,
opportunity_id,
opportunity_name,
opportunity_description,
seller_account_id
from
(
  select
    valid_from,
    update_date,
    delete_date,
    offer_id,
    offer_revision,
    name,
    expiration_date,
    opportunity_id,
    opportunity_name,
    opportunity_description,
    seller_account_id,
    row_number() over (partition by offer_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
  from
    offerfeed_v1
)
where
  -- keep latest ...
  row_num = 1
  -- ... and remove the soft-deleted one.
  and (delete_date is null or delete_date = '')
),

-- Here, we build the validity time range (adding valid_to on top of valid_from) of
each offer revision.
-- We will use it to get Offer name at invoice time.
-- NB: If you'd rather get "current" offer name, un-comment
"offers_with_latest_revision"
offers_with_history as (
  select
    offer_id,
    offer_revision,
    name,
```



```

    opportunity_id,
    opportunity_name,
    opportunity_description,
    valid_from,
    -- When we try to look up an offer revision as at the acceptance date of a BYOL
    agreement, we run into a problem.
    -- For BYOL, the agreement might be accepted (using some external non-AWS system or
    manual process) days before
    -- that BYOL agreement is entered into AWS Marketplace by the buyer. Therefore, the
    buyer is permitted to manually
    -- enter a backdated acceptance date, which might predate the point in time when
    the first revision of the offer
    -- was created. To work around this, we need to adjust the valid_from on the first
    revision of the offer to be
    -- earlier than the earliest possible backdated BYOL agreement acceptance date.
    case
        when lag(valid_from) over (partition by offer_id order by valid_from asc) is null
    and valid_from < cast('2021-04-01' as timestamp)
        then date_add('Day', -3857, valid_from)
        -- 3857 is the longest delay between acceptance_date of an agreement and the
    first revision of the offer
        when lag(valid_from) over (partition by offer_id order by valid_from asc) is null
    and valid_from >= cast('2021-04-01' as timestamp)
        then date_add('Day', -1460, valid_from)
        --after 2021 for the two offers we need to adjust for 2 more years
        else valid_from
    end as valid_from_adjusted,
    coalesce(
        lead(valid_from) over (partition by offer_id order by valid_from asc),
        cast('2999-01-01 00:00:00' as timestamp))
    as valid_to
from offers_with_uni_temporal_data
),
-- provided for reference only if you are interested into get "current" offer name
-- (ie. not used afterwards)
offers_with_latest_revision as (
    select
        offer_id,
        offer_revision,
        name,
        opportunity_name,
        opportunity_description,
        valid_from,
        null valid_to

```

```
from
(
  select
    offer_id,
    offer_revision,
    name,
    opportunity_name,
    opportunity_description,
    valid_from,
    null valid_to,
    row_number() over (partition by offer_id order by valid_from desc) as
row_num_latest_revision
  from
    offers_with_uni_temporal_data
)
where
  row_num_latest_revision = 1
),

-- An offer_target_id has several valid_from dates (each representing an offer
revision)
-- but because of bi-temporality, an offer_target_id + valid_from tuple can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
offer_targets_with_uni_temporal_data as (
  select
    from_iso8601_timestamp(valid_from) as valid_from,
    from_iso8601_timestamp(update_date) as update_date,
    from_iso8601_timestamp(delete_date) as delete_date,
    offer_target_id,
    offer_id,
    offer_revision,
    target_type,
    polarity,
    value
  from
  (
    select
      valid_from,
      update_date,
      delete_date,
      offer_target_id,
      offer_id,
      offer_revision,
```

```

        target_type,
        polarity,
        value,
        row_number() over (partition by offer_target_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
    from
        offertargetfeed_v1
)
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

offer_target_type as (
    select
        offer_id,
        offer_revision,
        substring(
            -- The first character indicates the priority (lower value means higher
precedence):
            min(
                case
                    when offer_target.target_type='BuyerAccounts' then '1Private'
                    when offer_target.target_type='ParticipatingPrograms' then '2Program:'||
cast(offer_target.value as varchar)
                    when offer_target.target_type='CountryCodes' then '3GeoTargeted'
                    -- well, there is no other case today, but rather be safe...
                    else '40ther Targeting'
                end
            ),
            -- Remove the first character that was only used for th priority in the "min"
aggregate function:
            2
        ) as offer_target
    from
        offer_targets_with_uni_temporal_data as offer_target
    group by
        offer_id,
        offer_revision
),

offers_with_history_with_target_type as (

```

```
select
    offer.offer_id,
    offer.offer_revision,
    -- even though today it is not possible to combine several types of targeting in a
single offer, let's ensure the query is still predictable if this gets possible in the
future
    max(
        case
            when off_tgt.offer_target is null then 'Public'
            else off_tgt.offer_target
        end
    ) as offer_target,
    min(offer.name) as name,
    min(offer.opportunity_name) as opportunity_name,
    min(offer.opportunity_description) as opportunity_description,
    offer.valid_from,
    offer.valid_from_adjusted,
    offer.valid_to,
    offer.opportunity_id
from
    offers_with_history as offer
left join offer_target_type as off_tgt on
    offer.offer_id = off_tgt.offer_id
    and offer.offer_revision = off_tgt.offer_revision
group by
    offer.offer_id,
    offer.offer_revision,
    offer.valid_from,
    offer.valid_from_adjusted,
    offer.valid_to,
    offer.opportunity_id
),

-- provided for reference only if you are interested into get "current" offer targets
-- (ie. not used afterwards)
offers_with_latest_revision_with_target_type as (
    select
        offer.offer_id,
        offer.offer_revision,
        -- even though today it is not possible to combine several types of targeting in a
single offer, let's ensure the query is still predictable if this gets possible in the
future
        max(
            distinct
```

```

        case
            when off_tgt.target_type is null then 'Public'
            when off_tgt.target_type='BuyerAccounts' then 'Private'
            when off_tgt.target_type='ParticipatingPrograms' then 'Program:||'
cast(off_tgt.value as varchar)
            when off_tgt.target_type='CountryCodes' then 'GeoTargeted'
            -- well, there is no other case today, but rather be safe...
            else 'Other Targeting'
        end
    ) as offer_target,
    min(offer.name) as name,
    min(offer.opportunity_name) as opportunity_name,
    min(offer.opportunity_description) as opportunity_description,
    offer.valid_from,
    offer.valid_to
from
    offers_with_latest_revision offer
-- left joining because public offers don't have targets
left join offer_targets_with_uni_temporal_data off_tgt on
    offer.offer_id=off_tgt.offer_id and offer.offer_revision=off_tgt.offer_revision
group by
    offer.offer_id,
    offer.offer_revision,
    -- redundant with offer_revision, as each revision has a dedicated valid_from (but
cleaner in the group by)
    offer.valid_from,
    offer.valid_to
),

-- A product_id has several valid_from dates (each representing a product revision),
-- but because of bi-temporality, each product_id + valid_from tuple can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
products_with_uni_temporal_data as (
    select
        from_iso8601_timestamp(valid_from) as valid_from,
        from_iso8601_timestamp(update_date) as update_date,
        from_iso8601_timestamp(delete_date) as delete_date,
        product_id,
        manufacturer_account_id,
        product_code,
        title
    from
    (

```

```
select
  valid_from,
  update_date,
  delete_date,
  product_id,
  manufacturer_account_id,
  product_code,
  title,
  row_number() over (partition by product_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
from
  productfeed_v1
)
where
  -- keep latest ...
  row_num = 1
  -- ... and remove the soft-deleted one.
  and (delete_date is null or delete_date = '')
),

products_with_history as (
  select
    product_id,
    title,
    valid_from,
    -- Offerv2 can have upto 50 years and Offerv3 is upto 5 years of past date
    case
      when lag(valid_from) over (partition by product_id order by valid_from asc) is
null and valid_from < cast('2021-04-01' as timestamp)
        then date_add('Day', -3857, valid_from)
      -- 3827 is the longest delay between acceptance_date of an agreement and the
product
      -- we are keeping 3857 as a consistency between the offers and products
      when lag(valid_from) over (partition by product_id order by valid_from asc) is
null and valid_from >= cast('2021-04-01' as timestamp)
        then date_add('Day', -2190, valid_from)
      --after 2021 for the two offers we need to adjust for 2 more years
      else valid_from
    end as valid_from_adjusted,
    coalesce(
      lead(valid_from) over (partition by product_id order by valid_from asc),
      cast('2999-01-01 00:00:00' as timestamp)
    ) as valid_to,
    product_code,
```

```
    manufacturer_account_id
  from
    products_with_uni_temporal_data
),

legacy_products as (
  select
    legacy_id,
    new_id
  from
    legacyidmappingfeed_v1
  where
    mapping_type='PRODUCT'
  group by
    legacy_id,
    new_id
),

-- A given billing_event_id represents an accounting event and thus has only one
-- valid_from date,
-- but because of bi-temporality, a billing_event_id (+ its valid_from) can appear
-- multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
billing_events_with_uni_temporal_data as (
  select
    billing_event_id,
    valid_from,
    update_date,
    delete_date,
    invoice_date,
    transaction_type,
    transaction_reference_id,
    parent_billing_event_id,
    bank_trace_id,
    broker_id,
    product_id,
    disbursement_billing_event_id,
    action,
    from_account_id,
    to_account_id,
    end_user_account_id,
    billing_address_id,
    amount,
    currency,
```

```
balance_impacting,
--empty value in Athena shows as '', change all '' value to null
case when agreement_id = '' then null else agreement_id end as agreement_id,
invoice_id,
payment_due_date,
usage_period_start_date,
usage_period_end_date,
buyer_transaction_reference_id,
row_num
from
(
select
    billing_event_id,
    from_iso8601_timestamp(valid_from) as valid_from,
    from_iso8601_timestamp(update_date) as update_date,
    delete_date,
    from_iso8601_timestamp(invoice_date) as invoice_date,
    transaction_type,
    transaction_reference_id,
    parent_billing_event_id,
    -- casting in case data was imported as number
    cast(bank_trace_id as varchar) as bank_trace_id,
    broker_id,
    product_id,
    disbursement_billing_event_id,
    action,
    from_account_id,
    to_account_id,
    end_user_account_id,
    billing_address_id,
    -- casting in case data was imported as varchar
    cast(amount as decimal(38,6)) as amount,
    currency,
    balance_impacting,
    agreement_id,
    invoice_id,
    case when payment_due_date is null or payment_due_date = '' then null else
from_iso8601_timestamp(payment_due_date) end as payment_due_date,
    from_iso8601_timestamp(usage_period_start_date) as usage_period_start_date,
    from_iso8601_timestamp(usage_period_end_date) as usage_period_end_date,
    buyer_transaction_reference_id,
    row_number() over (partition by billing_event_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
from
```



```

        billingeventfeed_v1
    )
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

-- Here we select the account_id of the current seller (We identify this by looking for
the to_account_id related to revenue transactions).
-- We will use it later to distinguish own agreements from agreements generated by
channel partners.
seller_account as (
    select
        from_account_id as seller_account_id
    from
        billing_events_with_uni_temporal_data bill
    where
        -- Assumption here is only seller will pay listing fee. As of 12/21/2021, there are
cases that Channel partner have 0 listing fee for CPP0, so the amount could be 0.
        bill.transaction_type like 'AWS_REV_SHARE' and amount <= 0 and action = 'INVOICED'
    group by
        -- from_account_id is always the same for all those "listing fee" transactions ==
the seller of record himself.
        -- If this view returns more than 1 record, the overall query will fail (on
purpose). Please contact AWS Marketplace if this happens.
        from_account_id
),

billing_event_with_business_flags as (
    select
        bl.billing_event_id,
        bl.end_user_account_id,
        bl.agreement_id,
        agreement.proposer_account_id,
        agreement.offer_id,
        agreement.acceptor_account_id,
        case
            -- For AWS and BALANCE_ADJUSTMENT, the billing event feed will show the "AWS
Marketplace" account as the
            -- receiver of the funds and the seller as the payer. We are not interested in
this information here.
            -- Null values will be ignored by the `max` aggregation function.

```

```

    when bl.transaction_type like 'AWS%' then null
    -- For BALANCE_ADJUSTMENT, payer is seller themselves
    when bl.invoice_id is null then bl.to_account_id
    -- We get the payer of the invoice from *any* transaction type that is not AWS
    and not BALANCE_ADJUSTMENT (because they are the same for a given end user + agreement
+ product).
    else bl.from_account_id
end as payer_account_id,
bl.product_id,
bl.action,
bl.transaction_type,
bl.parent_billing_event_id,
bl.disbursement_billing_event_id,
bl.amount,
bl.currency,
bl.balance_impacting,
bl.invoice_date,
bl.payment_due_date,
bl.usage_period_start_date,
bl.usage_period_end_date,
bl.invoice_id,
bl.billing_address_id,
bl.transaction_reference_id,
bl.buyer_transaction_reference_id,
case when disbursement.bank_trace_id = 'EMEA_MP_TEST_TRACE_ID' then null else
disbursement.bank_trace_id end as bank_trace_id,
case when disbursement.bank_trace_id = 'EMEA_MP_TEST_TRACE_ID' then null else
disbursement.invoice_date end as disbursement_date,
disbursement.billing_event_id as disbursement_id,
-- We will use disbursement_id_or_invoiced as part of the PK, so it cannot be null:
coalesce(
    --empty value in Athena shows as '', change all '' value to null
    case when disbursement.billing_event_id = '' then null else
disbursement.billing_event_id end,
    '<invoiced>') as disbursement_id_or_invoiced,
bl.broker_id,
case
    when bl.invoice_id is null /* transaction_type = 'BALANCE_ADJUSTMENT' */
        then (select seller_account_id from seller_account) || ':' ||
cast(bl.invoice_date as varchar)
    else bl.buyer_transaction_reference_id
        || '-' || case when bl.agreement_id is null or bl.agreement_id = '' then ' ' else
bl.agreement_id end

```

```

        || '-' || case when bl.end_user_account_id is null or bl.end_user_account_id = ''
then ' ' else bl.end_user_account_id end
        || '-' || coalesce(cast(bl.usage_period_start_date as varchar), ' ')
        || '-' || coalesce(cast(bl.usage_period_end_date as varchar), ' ')
    end as internal_buyer_line_item_id,
    bl.buyer_transaction_reference_id <> bl.transaction_reference_id as
is_seller_invoice,
    case when bl.transaction_type = 'SELLER_REV_SHARE' and (select seller_account_id
from seller_account) <> bl.to_account_id then true else false end as is_cog,
    case when bl.transaction_type in('SELLER_REV_SHARE_CREDIT',
'SELLER_REV_SHARE_REFUND') and (select seller_account_id from seller_account) <>
bl.to_account_id then true else false end as is_cog_refund,
    --TODO: replace below logic once we can create a logic the identify reseller/
manufacturer without depending on agreement feed
    case when aggrement.proposer_account_id <> (select seller_account_id from
seller_account) then true else false end as is_manufacturer_view_of_reseller
from
    billing_events_with_uni_temporal_data as bl
left join billing_events_with_uni_temporal_data as disbursement on
    disbursement.transaction_type like 'DISBURSEMENT%'
    and disbursement.action = 'DISBURSED'
    and disbursement.transaction_type IN ('DISBURSEMENT', 'DISBURSEMENT_FAILURE')
    and bl.disbursement_billing_event_id = disbursement.billing_event_id
left join agreements_with_history as aggrement on
    bl.agreement_id = aggrement.agreement_id
    and bl.invoice_date >= aggrement.valid_from_adjusted
    and bl.invoice_date < aggrement.valid_to
left join accounts_with_history awh on
    bl.to_account_id = awh.account_id
    and bl.invoice_date >= awh.valid_from
    and bl.invoice_date < awh.valid_to
where
    bl.transaction_type not like 'DISBURSEMENT%' and
    (bl.agreement_id is null or bl.agreement_id = ''
    or aggrement.agreement_id is not null)
),

-- listagg function in athena does not support partitioning, grouping here and then
joining to the main query
seller_invoice_list as (
    select
        internal_buyer_line_item_id,

```

```
listagg(case when not is_seller_invoice then null else invoice_id end,',' ) within
group (order by case when not is_seller_invoice then null else cast(invoice_date as
varchar) end) as seller_invoice_id_or_null,
listagg(case when not is_seller_invoice then null else cast(invoice_date as
varchar) end,',' ) within group (order by case when not is_seller_invoice then null
else cast(invoice_date as varchar) end) as seller_invoice_date_or_null
from
(
-- listagg function in athena does not support ordering by another field when
distinct is used,
-- here we first select distinct invoices and then do the listagg order by
invoice_date
select distinct internal_buyer_line_item_id, is_seller_invoice, invoice_id,
invoice_date
from billing_event_with_business_flags) distinct_invoices
group by internal_buyer_line_item_id
order by internal_buyer_line_item_id
),

billing_event_with_categorized_transaction as (
-- Use the flags that were created in the previous transformation in more calculated
columns:
-- NOTE: This transformation has no joins and no window functions
select
billing_event_id,
end_user_account_id,
agreement_id,
proposer_account_id,
offer_id,
acceptor_account_id,
case when is_cog or is_cog_refund then null else payer_account_id end as
payer_account_id,
product_id,
action,
transaction_type,
parent_billing_event_id,
disbursement_billing_event_id,
amount,
currency,
balance_impacting,
invoice_date,
payment_due_date,
usage_period_start_date,
usage_period_end_date,
```

```

invoice_id,
billing_address_id,
transaction_reference_id,
buyer_transaction_reference_id,
bank_trace_id,
disbursement_date,
disbursement_id,
disbursement_id_or_invoiced,
broker_id,
bl.internal_buyer_line_item_id,
is_seller_invoice,
is_cog,
is_cog_refund,
is_manufacturer_view_of_reseller,

-- Buyer/seller columns:
case when is_seller_invoice then null else invoice_id end as
buyer_invoice_id_or_null,
seller_invoices.seller_invoice_id_or_null,
case when is_seller_invoice then null else invoice_date end as
buyer_invoice_date_or_null,
seller_invoices.seller_invoice_date_or_null,

-- Categorized amounts by transaction type:
case when transaction_type = 'SELLER_REV_SHARE' and not is_cog then amount else 0
end as gross_revenue,
case when transaction_type in ('SELLER_REV_SHARE_REFUND', 'SELLER_REV_SHARE_CREDIT')
and not is_cog_refund then amount else 0 end as gross_refund,
case when transaction_type = 'SELLER_REV_SHARE' and is_cog then amount else 0
end as cogs,
case when transaction_type in ('SELLER_REV_SHARE_REFUND', 'SELLER_REV_SHARE_CREDIT')
and is_cog_refund then amount else 0 end as cogs_refund,
case when transaction_type = 'AWS_REV_SHARE' then amount else 0 end as
aws_rev_share,
case when transaction_type in ('AWS_REV_SHARE_REFUND', 'AWS_REV_SHARE_CREDIT') then
amount else 0 end as aws_refund_share,
case when transaction_type = 'AWS_TAX_SHARE' and not is_seller_invoice then
amount else 0 end as aws_tax_share, -- AWS tax share from _buyer_ invoice
case when transaction_type = 'AWS_TAX_SHARE' and is_seller_invoice then
amount else 0 end as aws_tax_share_listing_fee, -- AWS tax share from _seller_ invoice
case when transaction_type = 'AWS_TAX_SHARE_REFUND' and not is_seller_invoice
then amount else 0 end as aws_tax_share_refund,
case when transaction_type = 'AWS_TAX_SHARE_REFUND' and is_seller_invoice
then amount else 0 end as aws_tax_share_refund_listing_fee,

```

```
    case when transaction_type = 'SELLER_TAX_SHARE' then amount else 0 end as
seller_tax_share,
    case when transaction_type = 'SELLER_TAX_SHARE_REFUND' then amount else 0 end as
seller_tax_share_refund,
    case when transaction_type = 'BALANCE_ADJUSTMENT' then amount else 0 end as
balance_adjustment,
    case when transaction_type = 'SELLER_REV_SHARE_CREDIT' then amount else 0 end as
seller_rev_credit,
    case when transaction_type = 'AWS_REV_SHARE_CREDIT' then amount else 0 end as
aws_ref_fee_credit
from
    billing_event_with_business_flags as bl
    left join seller_invoice_list as seller_invoices
        on bl.internal_buyer_line_item_id = seller_invoices.internal_buyer_line_item_id
),

line_items_aggregated as (
-- This transformation has the only "group by" in all of these transformations.
-- NOTE: This transformation has no joins and no window functions
select
    internal_buyer_line_item_id,
    disbursement_id,
    disbursement_id_or_invoiced,
    product_id,
    broker_id,
    currency,
    agreement_id,
    proposer_account_id,
    acceptor_account_id,
    max(payer_account_id) as payer_account_id,
    offer_id,
    end_user_account_id,
    usage_period_start_date,
    usage_period_end_date,
    max(payment_due_date) payment_due_date,
    buyer_transaction_reference_id,
    bank_trace_id,
    disbursement_date,
    max(billing_address_id) as billing_address_id,

    -- Buyer/seller columns:
    max(buyer_invoice_id_or_null) as buyer_invoice_id,
    max(seller_invoice_id_or_null) as seller_invoice_id,
    max(buyer_invoice_date_or_null) as buyer_invoice_date,
```

```

max(seller_invoice_date_or_null) as seller_invoice_date,

-- Categorized amounts by transaction type:
-- When disbursement_id_or_invoiced = '<invoiced>',    these are invoiced amounts
-- When disbursement_id_or_invoiced <> ''<invoiced>' these are disbursed amounts
for _this_ specific disbursement_id
sum(gross_revenue) as gross_revenue_this_disbursement_id_or_invoiced,
sum(gross_refund) as gross_refund_this_disbursement_id_or_invoiced,
sum(cogs) as cogs_this_disbursement_id_or_invoiced,
sum(cogs_refund) as cogs_refund_this_disbursement_id_or_invoiced,
sum(aws_rev_share) as aws_rev_share_this_disbursement_id_or_invoiced,
sum(aws_refund_share) as aws_refund_share_this_disbursement_id_or_invoiced,
sum(aws_tax_share) as aws_tax_share_this_disbursement_id_or_invoiced,
sum(aws_tax_share_listing_fee) as
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,
sum(aws_tax_share_refund) as aws_tax_share_refund_this_disbursement_id_or_invoiced,
sum(aws_tax_share_refund_listing_fee) as
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,
sum(seller_tax_share) as seller_tax_share_this_disbursement_id_or_invoiced,
sum(seller_tax_share_refund) as
seller_tax_share_refund_this_disbursement_id_or_invoiced,
sum(balance_adjustment) as balance_adjustment_this_disbursement_id_or_invoiced,
sum(seller_rev_credit) as seller_rev_credit_this_disbursement_id_or_invoiced,
sum(aws_ref_fee_credit) as aws_ref_fee_credit_this_disbursement_id_or_invoiced
from
    billing_event_with_categorized_transaction as billing_categorized
group by
    internal_buyer_line_item_id,
    disbursement_id,
    disbursement_id_or_invoiced,
    broker_id,
    -- The following columns are included the in group by but they are intentionally
    omitted from the PK.
    -- These columns should have the _same_ values for each record in the PK.
    product_id,
    currency,
    agreement_id,
    proposer_account_id,
    acceptor_account_id,
    offer_id,
    end_user_account_id,
    usage_period_start_date,
    usage_period_end_date,
    buyer_transaction_reference_id,

```

```
    bank_trace_id,
    disbursement_date
),

-- listagg function in athena does not support partitioning, grouping here and then
  joining to the main query
disbursement_list as (
  select
    internal_buyer_line_item_id,
    listagg(cast(disbursement_date as varchar),',') within group (order by
cast(disbursement_date as varchar)) as disbursement_date_list,
    listagg(bank_trace_id,',') within group (order by cast(disbursement_date as
varchar)) as disburse_bank_trace_id_list
  from (
    -- listagg function in athena does not support ordering by another field when
distinct is used,
    -- here we first select distinct bank_trace_ids and then do the listagg order by
disbursement_date
    select distinct internal_buyer_line_item_id, disbursement_date, bank_trace_id
  from billing_event_with_business_flags) distinct_disbursements
  group by internal_buyer_line_item_id
  order by internal_buyer_line_item_id
),

line_items_with_window_functions as (
--add flag next step compare gross_revenue and gross_revenue_disbursed or gross_refund
and gross_refund_disbursed
  select
    line_item.internal_buyer_line_item_id,
    disbursement_id,
    disbursement_id_or_invoiced,
    product_id,
    broker_id,
    currency,
    agreement_id,
    proposer_account_id,
    acceptor_account_id,
    -- when there's aws_rev_Share adjustment/refund to a seller_rev_share invoice, it
can happen that for the same aws_rev_share invoice_id, there are multiple disbursement
events,
    -- using windows function to map payer_account_id of seller_rev_share to all
corresponding aws_rev_Share
    max(payer_account_id) over (partition by line_item.internal_buyer_line_item_id) as
payer_account_id,
```



```

offer_id,
end_user_account_id,
usage_period_start_date,
usage_period_end_date,
payment_due_date,
bank_trace_id,
disbursement_date,
billing_address_id,

-- Buyer/seller columns:
max(buyer_invoice_id) over (partition by line_item.internal_buyer_line_item_id) as
buyer_invoice_id,
seller_invoice_id,
max(buyer_invoice_date) over (partition by line_item.internal_buyer_line_item_id)
as buyer_invoice_date,
seller_invoice_date,

-- When disbursement_id_or_invoiced = '<invoiced>', these are actually invoiced
amounts
-- When disbursement_id_or_invoiced <> '<invoiced>' these are disbursed amounts
for _this_ specific disbursement_id
gross_revenue_this_disbursement_id_or_invoiced,
gross_refund_this_disbursement_id_or_invoiced,
cogs_this_disbursement_id_or_invoiced,
cogs_refund_this_disbursement_id_or_invoiced,
aws_rev_share_this_disbursement_id_or_invoiced,
aws_refund_share_this_disbursement_id_or_invoiced,
aws_tax_share_this_disbursement_id_or_invoiced,
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,
seller_tax_share_this_disbursement_id_or_invoiced,
seller_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,
balance_adjustment_this_disbursement_id_or_invoiced,
seller_rev_credit_this_disbursement_id_or_invoiced,
aws_ref_fee_credit_this_disbursement_id_or_invoiced,
-- IMPORTANT: All window functions partitioned by internal_buyer_line_item_id:

-- Invoiced amounts, categorized by transaction type:
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
gross_revenue_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end)over
(partition by line_item.internal_buyer_line_item_id) as gross_revenue_invoiced,

```

```
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
gross_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as gross_refund_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
cogs_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
cogs_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_refund_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_rev_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_rev_share_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_refund_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
aws_refund_share_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_tax_share_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_listing_fee_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_listing_fee_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
seller_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
seller_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_refund_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
balance_adjustment_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
balance_adjustment_invoiced,
```

```

sum(case when disbursement_id_or_invoiced = '<invoiced>' then
seller_rev_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_rev_credit_invoiced,
sum(case when disbursement_id_or_invoiced = '<invoiced>' then
aws_ref_fee_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_ref_fee_credit_invoiced,

-- Total disbursed amounts (for all disbursement_id values), categorized by
transaction type:
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
gross_revenue_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as gross_revenue_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
gross_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as gross_refund_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
cogs_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
cogs_refund_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as cogs_refund_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
aws_rev_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_rev_share_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
aws_refund_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
aws_refund_share_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
aws_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6)) end) over
(partition by line_item.internal_buyer_line_item_id) as aws_tax_share_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_listing_fee_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
aws_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_disbursed,
sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced else cast(0 as

```

```

decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_tax_share_refund_listing_fee_disbursed,
    sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
seller_tax_share_this_disbursement_id_or_invoiced else cast(0 as decimal(38,6))
end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_disbursed,
    sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
seller_tax_share_refund_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_tax_share_refund_disbursed,
    sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
balance_adjustment_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
balance_adjustment_disbursed,
    sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
seller_rev_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
seller_rev_credit_disbursed,
    sum(case when disbursement_id_or_invoiced '<>' '<invoiced>' then
aws_ref_fee_credit_this_disbursement_id_or_invoiced else cast(0 as
decimal(38,6)) end) over (partition by line_item.internal_buyer_line_item_id) as
aws_ref_fee_credit_disbursed,

    -- aggregate multiple disbursement
    max(disbursement_date) over (partition by line_item.internal_buyer_line_item_id) as
last_disbursement_date,
    first_value(case when disbursement_id_or_invoiced =
''<invoiced>' then null else disbursement_id_or_invoiced end)
over(partition by line_item.internal_buyer_line_item_id order by
coalesce(disbursement_date,cast('1900-01-01' as timestamp)) desc rows between
unbounded preceding and unbounded following) as last_disbursement_id,
    first_value(bank_trace_id) over (partition by line_item.internal_buyer_line_item_id
order by coalesce(disbursement_date,cast('1900-01-01' as timestamp)) desc rows between
unbounded preceding and unbounded following) as last_disburse_bank_trace_id,
    disb_list.disbursement_date_list,
    disb_list.disburse_bank_trace_id_list
from
    line_items_aggregated as line_item
    left join disbursement_list disb_list
        on line_item.internal_buyer_line_item_id = disb_list.internal_buyer_line_item_id
),

cpo_offer_id as (
    select

```

```

-- Channel partner offers do not exist in offertargetfeed_v1 table (as per legal
requirement), causing cppo offer be defined as 'Public' in previous step, we will
convert them back to 'Private' in next step
    offer_id
from
    offers_with_uni_temporal_data
where
    -- seller_account_id is null means the ISV owns the offer
    seller_account_id is not null
    and seller_account_id '<>' (select seller_account_id from seller_account)
group by
    offer_id
),

line_items_with_window_functions_enrich_offer_product_address as (
    select
        internal_buyer_line_item_id,
        disbursement_id,
        disbursement_id_or_invoiced,
        line.product_id,
        legacy_product.legacy_id as legacy_product_id,
        products.title as product_title,
        line.broker_id,
        line.currency,
        line.end_user_account_id,
        acc_enduser.encrypted_account_id as end_user_encrypted_account_id,
        acc_enduser.aws_account_id as end_user_aws_account_id,
        acc_payer.aws_account_id as payer_aws_account_id,
        acc_payer.encrypted_account_id payer_encrypted_account_id,
        line.agreement_id,
        agreement.agreement_revision,
        line.proposer_account_id,
        case when offer.offer_id like 'aiqoffer-%' then null else agreement.start_date end
as Agreement_Start_Date,
        case when offer.offer_id like 'aiqoffer-%' then null else agreement.end_date end as
Agreement_End_Date,
        case when offer.offer_id like 'aiqoffer-%' then null else agreement.acceptance_date
end as Agreement_Acceptance_Date,
        case when offer.offer_id like 'aiqoffer-%' then null else agreement.valid_from end
as agreement_updated_date,
        case when offer.offer_id like 'aiqoffer-%' then null else
line.usage_period_start_date end as Usage_Period_Start_Date,
        case when offer.offer_id like 'aiqoffer-%' then null else
line.usage_period_end_date end as Usage_Period_End_Date,

```

```
line.acceptor_account_id,  
acc_subscriber.aws_account_id as subscriber_aws_account_id,  
acc_subscriber.encrypted_account_id as subscriber_encrypted_account_id,  
offer.offer_id,  
case  
  when offer.offer_id in (  
    select distinct offer_id  
    from cppo_offer_id)  
  then 'Private'  
  else offer.offer_target  
end as offer_target,  
offer.name offer_name,  
offer.opportunity_name offer_opportunity_name,  
offer.opportunity_description offer_opportunity_description,  
offer.opportunity_id,  
payment_due_date,  
line.bank_trace_id,  
disbursement_date,  
billing_address_id,  
buyer_invoice_id,  
seller_invoice_id,  
buyer_invoice_date,  
seller_invoice_date,  
gross_revenue_this_disbursement_id_or_invoiced,  
gross_refund_this_disbursement_id_or_invoiced,  
cogs_this_disbursement_id_or_invoiced,  
cogs_refund_this_disbursement_id_or_invoiced,  
aws_rev_share_this_disbursement_id_or_invoiced,  
aws_refund_share_this_disbursement_id_or_invoiced,  
aws_tax_share_this_disbursement_id_or_invoiced,  
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,  
seller_tax_share_this_disbursement_id_or_invoiced,  
seller_tax_share_refund_this_disbursement_id_or_invoiced,  
balance_adjustment_this_disbursement_id_or_invoiced,  
seller_rev_credit_this_disbursement_id_or_invoiced,  
aws_ref_fee_credit_this_disbursement_id_or_invoiced,  
gross_revenue_invoiced,  
gross_refund_invoiced,  
cogs_invoiced,  
cogs_refund_invoiced,  
aws_rev_share_invoiced,
```

```

aws_refund_share_invoiced,
aws_tax_share_invoiced,
aws_tax_share_listing_fee_invoiced,
aws_tax_share_refund_invoiced,
aws_tax_share_refund_listing_fee_invoiced,
seller_tax_share_invoiced,
seller_tax_share_refund_invoiced,
balance_adjustment_invoiced,
seller_rev_credit_invoiced,
aws_ref_fee_credit_invoiced,
gross_revenue_disbursed,
gross_refund_disbursed,
cogs_disbursed,
cogs_refund_disbursed,
aws_rev_share_disbursed,
aws_refund_share_disbursed,
aws_tax_share_disbursed,
aws_tax_share_listing_fee_disbursed,
aws_tax_share_refund_disbursed,
aws_tax_share_refund_listing_fee_disbursed,
seller_tax_share_disbursed,
seller_tax_share_refund_disbursed,
balance_adjustment_disbursed,
seller_rev_credit_disbursed,
aws_ref_fee_credit_disbursed,
last_disbursement_date,
last_disbursement_id,
last_disburse_bank_trace_id,
disbursement_date_list,
disburse_bank_trace_id_list,
products.product_code,
acc_products.aws_account_id as manufacturer_aws_account_id,
products.manufacturer_account_id,
--add subscriber and payer addressID, payer address preference order: tax
address>billing address>mailing address, subscriber address preference order: tax
address> mailing address
    coalesce (
        --empty value in Athena shows as '', change all '' value to null in order to
follow the preference order logic above
        case when acc_subscriber.tax_address_id = '' then null else
acc_subscriber.tax_address_id end,
        case when acc_subscriber.mailing_address_id = '' then null else
acc_subscriber.mailing_address_id end) as subscriber_address_id,
    coalesce (

```

```

        case when acc_payer.tax_address_id = '' then null else acc_payer.tax_address_id
end,
        case when line.billing_address_id = '' then null else line.billing_address_id
end,
        case when acc_payer.mailing_address_id = '' then null else
acc_payer.mailing_address_id end) as payer_address_id,
        coalesce (
            case when acc_enduser.tax_address_id = '' then null else
acc_enduser.tax_address_id end,
            case when line.billing_address_id = '' then null else line.billing_address_id
end,
            case when acc_enduser.mailing_address_id = '' then null else
acc_enduser.mailing_address_id end) as end_user_address_id
from
    line_items_with_window_functions as line
left join agreements_with_history as agreement on
    line.agreement_id = agreement.agreement_id and line.buyer_invoice_date >=
agreement.valid_from_adjusted and line.buyer_invoice_date < agreement.valid_to
left join offers_with_history_with_target_type as offer on
    line.offer_id = offer.offer_id and line.buyer_invoice_date >= offer.valid_from
and line.buyer_invoice_date < offer.valid_to
left join products_with_history as products on
    line.product_id = products.product_id and line.buyer_invoice_date >=
products.valid_from_adjusted and line.buyer_invoice_date < products.valid_to
left join legacy_products as legacy_product on
    line.product_id = legacy_product.new_id
left join accounts_with_history_with_company_name as acc_payer on
    line.payer_account_id = acc_payer.account_id and line.buyer_invoice_date >=
acc_payer.valid_from and line.buyer_invoice_date < acc_payer.valid_to
left join accounts_with_history_with_company_name as acc_enduser on
    line.end_user_account_id = acc_enduser.account_id and line.buyer_invoice_date
>= acc_enduser.valid_from and line.buyer_invoice_date < acc_enduser.valid_to
left join accounts_with_history_with_company_name as acc_subscriber on
    line.acceptor_account_id = acc_subscriber.account_id and
line.buyer_invoice_date >= acc_subscriber.valid_from and line.buyer_invoice_date <
acc_subscriber.valid_to
left join accounts_with_history_with_company_name as acc_products on
    products.manufacturer_account_id = acc_products.account_id and
line.buyer_invoice_date >= acc_products.valid_from and line.buyer_invoice_date <
acc_products.valid_to
),

line_items_with_window_functions_enrich_offer_product_address_name as (

```



```
select
  line.internal_buyer_line_item_id,
  disbursement_id,
  disbursement_id_or_invoiced,
  product_id,
  legacy_product_id,
  product_title,
  broker_id,
  currency,
  end_user_address_id,
  end_user_account_id,
  end_user_encrypted_account_id,
  end_user_aws_account_id,
  add_enduser.company_name end_user_company_name,
  add_enduser.email_domain end_user_email_domain,
  add_enduser.city end_user_city,
  add_enduser.state_or_region end_user_state,
  add_enduser.country_code end_user_country,
  add_enduser.postal_code end_user_postal_code,
  payer_aws_account_id,
  payer_encrypted_account_id,
  payer_address_id,
  add_payer.company_name payer_company_name,
  add_payer.email_domain payer_email_domain,
  add_payer.city payer_city,
  add_payer.state_or_region payer_state,
  add_payer.country_code payer_country,
  add_payer.postal_code payer_postal_code,
  agreement_id,
  agreement_revision,
  agreement_start_date,
  agreement_end_date,
  agreement_acceptance_date,
  agreement_updated_date,
  case when proposer_account_id = (select seller_account_id from seller_account) then
null else acc_proposer.aws_account_id end as reseller_aws_account_id,
  case when proposer_account_id = (select seller_account_id from seller_account) then
null else acc_proposer.mailing_company_name end as reseller_company_name,
  usage_period_start_date,
  usage_period_end_date,
  proposer_account_id,
  acc_proposer.aws_account_id as proposer_aws_account_id,
  acceptor_account_id,
  subscriber_aws_account_id,
```

```
subscriber_encrypted_account_id,  
subscriber_address_id,  
add_subscriber.company_name subscriber_company_name,  
add_subscriber.email_domain subscriber_email_domain,  
add_subscriber.city subscriber_city,  
add_subscriber.state_or_region subscriber_state,  
add_subscriber.country_code subscriber_country,  
add_subscriber.postal_code subscriber_postal_code,  
offer_id,  
offer_target,  
offer_name,  
offer_opportunity_name,  
offer_opportunity_description,  
opportunity_id,  
payment_due_date,  
bank_trace_id,  
disbursement_date,  
billing_address_id,  
max(buyer_invoice_id)as buyer_invoice_id,  
max(seller_invoice_id)as seller_invoice_id,  
max(buyer_invoice_date)as buyer_invoice_date,  
max(seller_invoice_date)as seller_invoice_date,  
gross_revenue_this_disbursement_id_or_invoiced,  
gross_refund_this_disbursement_id_or_invoiced,  
cogs_this_disbursement_id_or_invoiced,  
cogs_refund_this_disbursement_id_or_invoiced,  
aws_rev_share_this_disbursement_id_or_invoiced,  
aws_refund_share_this_disbursement_id_or_invoiced,  
aws_tax_share_this_disbursement_id_or_invoiced,  
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,  
seller_tax_share_this_disbursement_id_or_invoiced,  
seller_tax_share_refund_this_disbursement_id_or_invoiced,  
balance_adjustment_this_disbursement_id_or_invoiced,  
seller_rev_credit_this_disbursement_id_or_invoiced,  
aws_ref_fee_credit_this_disbursement_id_or_invoiced,  
(gross_revenue_this_disbursement_id_or_invoiced +  
gross_refund_this_disbursement_id_or_invoiced +  
aws_rev_share_this_disbursement_id_or_invoiced +  
aws_refund_share_this_disbursement_id_or_invoiced +  
seller_tax_share_this_disbursement_id_or_invoiced +  
seller_tax_share_refund_this_disbursement_id_or_invoiced
```

```

    + cogs_this_disbursement_id_or_invoiced +
cogs_refund_this_disbursement_id_or_invoiced +
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced +
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced) as
seller_net_revenue_this_disbursement_id_or_invoiced,
    gross_revenue_invoiced,
    gross_refund_invoiced,
    cogs_invoiced,
    cogs_refund_invoiced,
    aws_rev_share_invoiced,
    aws_refund_share_invoiced,
    aws_tax_share_invoiced,
    aws_tax_share_listing_fee_invoiced,
    aws_tax_share_refund_invoiced,
    aws_tax_share_refund_listing_fee_invoiced,
    seller_tax_share_invoiced,
    seller_tax_share_refund_invoiced,
    balance_adjustment_invoiced,
    seller_rev_credit_invoiced,
    aws_ref_fee_credit_invoiced,
    gross_revenue_disbursed,
    gross_refund_disbursed,
    cogs_disbursed,
    cogs_refund_disbursed,
    aws_rev_share_disbursed,
    aws_refund_share_disbursed,
    aws_tax_share_disbursed,
    aws_tax_share_listing_fee_disbursed,
    aws_tax_share_refund_disbursed,
    aws_tax_share_refund_listing_fee_disbursed,
    seller_tax_share_disbursed,
    seller_tax_share_refund_disbursed,
    balance_adjustment_disbursed,
    seller_rev_credit_disbursed,
    aws_ref_fee_credit_disbursed,
    (gross_revenue_invoiced + gross_revenue_disbursed) as uncollected_gross_revenue,
    -- net revenue = gross revenue - listing fee - tax - cogs
    (gross_revenue_invoiced + gross_refund_invoiced + aws_rev_share_invoiced
+ aws_refund_share_invoiced + seller_tax_share_invoiced +
seller_tax_share_refund_invoiced + cogs_invoiced + cogs_refund_invoiced +
aws_tax_share_listing_fee_invoiced + aws_tax_share_refund_listing_fee_invoiced) as
seller_net_revenue,
    (gross_revenue_invoiced + gross_refund_invoiced + aws_rev_share_invoiced
+ aws_refund_share_invoiced + seller_tax_share_invoiced +

```

```

seller_tax_share_refund_invoiced + cogs_invoiced + cogs_refund_invoiced +
aws_tax_share_listing_fee_invoiced + aws_tax_share_refund_listing_fee_invoiced
    + gross_revenue_disbursed + gross_refund_disbursed + aws_rev_share_disbursed
+ aws_refund_share_disbursed + seller_tax_share_disbursed +
seller_tax_share_refund_disbursed + cogs_disbursed + cogs_refund_disbursed +
aws_tax_share_listing_fee_disbursed + aws_tax_share_refund_listing_fee_disbursed) as
uncollected_seller_net_revenue,
    last_disbursement_date,
    last_disbursement_id,
    last_disburse_bank_trace_id,
    disbursement_date_list,
    disburse_bank_trace_id_list,
    product_code,
    manufacturer_aws_account_id,
    manufacturer_account_id,
    acc_manu.mailing_company_name as manufacturer_company_name,
    cast(null as varchar) as AR_Period,
    case
        when (
            (gross_revenue_invoiced '<>0 and gross_revenue_invoiced = -1 *
gross_revenue_disbursed)
            or (gross_refund_invoiced '<> 0 and gross_refund_invoiced = -1 *
gross_refund_disbursed)
            or (balance_adjustment_invoiced '<> 0 and balance_adjustment_invoiced = -1 *
balance_adjustment_disbursed)
            or (seller_tax_share_refund_invoiced '<> 0 and seller_tax_share_refund_invoiced
= -1 * seller_tax_share_refund_disbursed)
            or (gross_revenue_invoiced = 0 and gross_refund_invoiced = 0 and
balance_adjustment_invoiced = 0 and seller_tax_share_refund_invoiced = 0 and
last_disbursement_id is not null)) then 'Yes'
            when gross_revenue_disbursed = 0 and gross_refund_disbursed = 0 and
balance_adjustment_disbursed = 0 and seller_tax_share_disbursed = 0 and
seller_tax_share_refund_disbursed = 0 then 'No'
            else 'Partial'
        end as Disbursement_Flag
from line_items_with_window_functions_enrich_offer_product_address as line
left join accounts_with_history_with_company_name as acc_manu on
    line.manufacturer_account_id = acc_manu.account_id and line.buyer_invoice_date >=
acc_manu.valid_from_adjusted and line.buyer_invoice_date <= acc_manu.valid_to
left join accounts_with_history_with_company_name as acc_proposer on
    line.proposer_account_id = acc_proposer.account_id and line.buyer_invoice_date >=
acc_proposer.valid_from and line.buyer_invoice_date < acc_proposer.valid_to
left join address_with_latest_revision as add_payer on
    line.payer_address_id = add_payer.address_id

```

```
left join address_with_latest_revision as add_subscriber on
  line.subscriber_address_id = add_subscriber.address_id
left join address_with_latest_revision as add_enduser on
  line.end_user_address_id = add_enduser.address_id
group by
  line.internal_buyer_line_item_id,
  disbursement_id,
  disbursement_id_or_invoiced,
  product_id,
  legacy_product_id,
  product_title,
  broker_id,
  currency,
  end_user_address_id,
  end_user_account_id,
  end_user_encrypted_account_id,
  end_user_aws_account_id,
  add_enduser.company_name,
  add_enduser.email_domain,
  add_enduser.city,
  add_enduser.state_or_region,
  add_enduser.country_code,
  add_enduser.postal_code,
  payer_aws_account_id,
  payer_encrypted_account_id,
  payer_address_id,
  add_payer.company_name,
  add_payer.email_domain,
  add_payer.city,
  add_payer.state_or_region,
  add_payer.country_code,
  add_payer.postal_code,
  agreement_id,
  agreement_revision,
  case when proposer_account_id = (select seller_account_id from seller_account) then
null else acc_proposer.aws_account_id end,
  case when proposer_account_id = (select seller_account_id from seller_account) then
null else acc_proposer.mailing_company_name end,
  agreement_start_date,
  agreement_end_date,
  agreement_acceptance_date,
  agreement_updated_date,
  usage_period_start_date,
  usage_period_end_date,
```

```
acceptor_account_id,  
subscriber_aws_account_id,  
subscriber_encrypted_account_id,  
subscriber_address_id,  
add_subscriber.company_name,  
add_subscriber.email_domain,  
add_subscriber.city,  
add_subscriber.state_or_region,  
add_subscriber.country_code,  
add_subscriber.postal_code,  
offer_id,  
offer_target,  
offer_name,  
offer_opportunity_name,  
offer_opportunity_description,  
opportunity_id,  
payment_due_date,  
bank_trace_id,  
disbursement_date,  
billing_address_id,  
gross_revenue_this_disbursement_id_or_invoiced,  
gross_refund_this_disbursement_id_or_invoiced,  
cogs_this_disbursement_id_or_invoiced,  
cogs_refund_this_disbursement_id_or_invoiced,  
aws_rev_share_this_disbursement_id_or_invoiced,  
aws_refund_share_this_disbursement_id_or_invoiced,  
aws_tax_share_this_disbursement_id_or_invoiced,  
aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_this_disbursement_id_or_invoiced,  
aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,  
seller_tax_share_this_disbursement_id_or_invoiced,  
seller_tax_share_refund_this_disbursement_id_or_invoiced,  
balance_adjustment_this_disbursement_id_or_invoiced,  
seller_rev_credit_this_disbursement_id_or_invoiced,  
aws_ref_fee_credit_this_disbursement_id_or_invoiced,  
gross_revenue_invoiced,  
gross_refund_invoiced,  
cogs_invoiced,  
cogs_refund_invoiced,  
aws_rev_share_invoiced,  
aws_refund_share_invoiced,  
aws_tax_share_invoiced,  
aws_tax_share_listing_fee_invoiced,  
aws_tax_share_refund_invoiced,
```

```

aws_tax_share_refund_listing_fee_invoiced,
seller_tax_share_invoiced,
seller_tax_share_refund_invoiced,
balance_adjustment_invoiced,
seller_rev_credit_invoiced,
aws_ref_fee_credit_invoiced,
gross_revenue_disbursed,
gross_refund_disbursed,
cogs_disbursed,
cogs_refund_disbursed,
aws_rev_share_disbursed,
aws_refund_share_disbursed,
aws_tax_share_disbursed,
aws_tax_share_listing_fee_disbursed,
aws_tax_share_refund_disbursed,
aws_tax_share_refund_listing_fee_disbursed,
seller_tax_share_disbursed,
seller_tax_share_refund_disbursed,
balance_adjustment_disbursed,
seller_rev_credit_disbursed,
aws_ref_fee_credit_disbursed,
last_disbursement_date,
last_disbursement_id,
last_disburse_bank_trace_id,
disbursement_date_list,
disburse_bank_trace_id_list,
product_code,
manufacturer_aws_account_id,
manufacturer_account_id,
acc_manu.mailing_company_name,
proposer_account_id,
acc_proposer.aws_account_id
),
invoiced_not_disbursed as(
  select
    --we will filter on rownum =1 in next step,
    -- means internal_buyer_line_item_id, there's only '<invoiced>' record, no
    disbursement_id linked
    *,
    max(case when disbursement_id_or_invoiced = '<invoiced>' then 1 else 2 end)
      over (partition by internal_buyer_line_item_id) rownum
  from line_items_with_window_functions_enrich_offer_product_address_name as line_items
),

```

```

collections_and_disbursements as (
  select
    -----
    -- Invoice Info --
    -----
    buyer_invoice_date as Invoice_Date,
    Payment_Due_Date as Payment_Due_Date,
    concat(
      'Net ',
      case
        when abs(date_diff('Day', Payment_due_date, buyer_invoice_date))>180 then
'180+'
        else cast(abs(date_diff('Day', Payment_due_date, buyer_invoice_date)) as
varchar)
      end,
      ' days'
    ) as payment_terms,
    buyer_invoice_id as Invoice_ID,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when seller_invoice_id = '' then null else seller_invoice_id end,
      'Not applicable') as Listing_Fee_Invoice_ID,

    -----
    --End user Information --
    -----
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when End_User_Company_Name = '' then null else End_User_Company_Name end,
      'Not available') as End_User_Company_Name,
    End_User_AWS_Account_ID,
    End_User_Encrypted_Account_ID,
    End_User_Email_Domain,
    End_User_City,
    End_User_State as End_User_State_or_Region,
    End_User_Country,
    End_User_Postal_Code,
    End_User_Address_ID,

    -----
    --Subscriber Information --
    -----
    case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'

```



```
    when Subscriber_Company_Name is null or Subscriber_Company_Name = '' then 'Not
provided'
    else Subscriber_Company_Name
end as Subscriber_Company_Name,
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    else Subscriber_AWS_Account_ID
end as Subscriber_AWS_Account_ID,
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    else Subscriber_Encrypted_Account_ID
end as Subscriber_Encrypted_Account_ID,
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    when Subscriber_Email_Domain is null or Subscriber_Email_Domain = '' then 'Not
provided'
    else Subscriber_Email_Domain
end as Subscriber_Email_Domain,
case
    when Agreement_id is null or Agreement_ID = '' then 'Not available'
    when Subscriber_City is null or Subscriber_City = '' then 'Not provided'
    else Subscriber_City
end as Subscriber_City,
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    when Subscriber_State is null or Subscriber_State = '' then 'Not provided'
    else Subscriber_State
end as Subscriber_State_or_Region,
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    when Subscriber_Country is null or Subscriber_Country = '' then 'Not provided'
    else Subscriber_Country
end as Subscriber_Country,
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    when Subscriber_Postal_Code is null or Subscriber_Postal_Code = '' then 'Not
provided'
    else Subscriber_Postal_Code
end as Subscriber_Postal_Code,
case
    when Agreement_ID is null or Agreement_ID = '' then 'Not available'
    when Subscriber_Address_ID is null or Subscriber_Address_ID = '' then 'Not
provided'
    else Subscriber_Address_ID
```

```

end as Subscriber_Address_ID,

-----
-- Procurement Info --
-----

-- product title at time of invoice. It is possible that the title changes over
time and therefore there may be multiple product titles mapped to a single product id.
coalesce(
  --empty value in Athena shows as '', change all '' value to null
  case when Product_Title = '' then null else Product_Title end,
  'Not provided') as Product_Title,
-- offer name at time of invoice. It is possible that the name changes over time
therefore there may be multiple offer names mapped to a single offer id.
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when (Offer_Name is null or Offer_Name = '') and Offer_Target = 'Public' then
'Not applicable'
  else Offer_Name
end as Offer_Name,
case
  when Agreement_Id is null or Agreement_ID = ''
  then 'Not available'
  else Offer_ID
end as Offer_ID,
-- offer visibility at time of invoice.,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  else Offer_Target
end as Offer_Visibility,
coalesce(
  --empty value in Athena shows as '', change all '' value to null
  case when Agreement_ID = '' then null else Agreement_ID end,
  'Not available') as Agreement_ID,
Agreement_Start_Date,
Agreement_Acceptance_Date,
Agreement_End_Date,

Usage_Period_Start_Date,
Usage_Period_End_Date,

-----
-- Disbursement Info --
-----

case

```

```

        when Disbursement_Flag = 'Yes' then 'Disbursed'
        when Disbursement_Flag = 'No' then 'Not Disbursed'
        else 'Other'
    end as Disbursement_Status,
    last_disbursement_date as disbursement_date,
    case
        when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date))
        else null
    end as Disbursement_Time,
    case
        when Disbursement_Flag = 'No' then 'Not applicable'
        when bank_trace_id is null or bank_trace_id = '' then 'Not available'
        else bank_trace_id
    end as disburse_bank_trace_id,

-----
-- Revenues --
-----

-- We are rounding the sums using 2 decimal precision
-- Note that the rounding method might differ between SQL implementations.
-- The monthly revenue report is using RoundingMode.HALF_UP. This might create tiny
discrepancies between this SQL output
-- and the legacy report
round(-1 * gross_revenue_this_disbursement_id_or_invoiced,2) as Gross_Revenue,
round(-1 * gross_refund_this_disbursement_id_or_invoiced,2) as Gross_Refund,
round(-1 * aws_rev_share_this_disbursement_id_or_invoiced,2) as Listing_Fee,
round(-1 * aws_refund_share_this_disbursement_id_or_invoiced,2) as
Listing_Fee_Refund,
truncate(
    case
        when gross_revenue_this_disbursement_id_or_invoiced !=
= 0 then abs(aws_rev_share_this_disbursement_id_or_invoiced/
gross_revenue_this_disbursement_id_or_invoiced)
        when gross_refund_this_disbursement_id_or_invoiced !=
0 then abs(aws_refund_share_this_disbursement_id_or_invoiced/
gross_refund_this_disbursement_id_or_invoiced)
        else 0
    end
    ,4) as Listing_Fee_Percentage,
round(-1 * seller_tax_share_this_disbursement_id_or_invoiced,2) as
Seller_Tax_Share,
round(-1 * seller_tax_share_refund_this_disbursement_id_or_invoiced,2) as
Seller_Tax_Share_Refund,

```

```

    round(-1 * aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,2) as
AWS_Tax_Share_Listing_Fee,
    round(-1 * aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,2) as
AWS_Tax_Share_Refund_Listing_Fee,
    round(-1 * cogs_this_disbursement_id_or_invoiced,2) as Wholesale_cost,
    round(-1 * cogs_refund_this_disbursement_id_or_invoiced,2) as
Wholesale_cost_Refund,
    round(-1 * seller_net_revenue_this_disbursement_id_or_invoiced,2) as
Seller_Net_Revenue,
    currency as Currency,

    substring(internal_buyer_line_item_id,1,strpos(internal_buyer_line_item_id,'-')-1)
as Transaction_Reference_ID,
    broker_id as AWS_seller_of_record,

-----
-- Resale info --
-----
case
    when Opportunity_Id is null or Opportunity_Id = '' then
        case
            when Offer_Target = 'Public' then 'Not applicable'
            when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
            else null
        end
    else Opportunity_Id
end as Resale_authorization_ID,
case
    when Offer_Opportunity_Name is null or Offer_Opportunity_Name = '' then
        case
            when Offer_Target = 'Public' then 'Not applicable'
            when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
            else null
        end
    else Offer_Opportunity_Name
end as Resale_authorization_name,
case
    when Offer_Opportunity_Description is null or Offer_Opportunity_Description = ''
then
        case
            when Offer_Target = 'Public' then 'Not applicable'

```

```

        when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
        else null
    end
    else Offer_Opportunity_Description
end as Resale_authorization_description,
case
    when (Reseller_AWS_Account_ID is not null and Reseller_AWS_Account_ID != '')
        and (Reseller_Company_Name is null or Reseller_Company_Name = '') then 'Not
available'
    when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
        and (opportunity_id is null or opportunity_id = '') then 'Not applicable'
    when (select seller_account_id from seller_account) '<>
manufacturer_aws_account_id
        and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
    else Reseller_Company_Name
end as Reseller_Company_Name,
case
    when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
        and (Opportunity_Id is null or Opportunity_Id = '') then 'Not applicable'
    when (select seller_account_id from seller_account) '<>
manufacturer_aws_account_id
        and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
    else Reseller_AWS_Account_ID
end as Reseller_AWS_Account_ID,

-----
-- Payer Information --
-----
coalesce(
    --empty value in Athena shows as '', change all '' value to null
    case when Payer_Company_Name = '' then null else Payer_Company_Name end,
    'Not available') as Payer_Company_Name,
Payer_AWS_Account_ID, -- "Customer AWS Account Number" in legacy report
Payer_Encrypted_Account_ID,
Payer_Email_Domain,
Payer_City,
Payer_State as Payer_State_or_Region,
Payer_Country,
Payer_Postal_Code,
Payer_Address_ID,

```

```

-----
-- ISV Information --
-----
manufacturer_aws_account_id as ISV_Account_ID,
coalesce(
  --empty value in Athena shows as '', change all '' value to null
  case when Manufacturer_Company_Name = '' then null else Manufacturer_Company_Name
end,
  'Not available') as ISV_Company_Name,

-----
-- Products info --
-----
Legacy_Product_ID,
coalesce(
  --empty value in Athena shows as '', change all '' value to null
  case when Product_ID = '' then null else Product_ID end,
  'Not provided') as Product_ID,
-- this is to get the legacy product id https://sim.amazon.com/issues/MP-
INSIGHTS-2561
Product_Code,

  case when Disbursement_Flag = 'Yes' then round(-1 *
seller_net_revenue_this_disbursement_id_or_invoiced,2) else 0 end as
Disbursed_Net_Revenue,
  case when Disbursement_Flag = 'No' then round(-1 *
seller_net_revenue_this_disbursement_id_or_invoiced,2) else 0 end as
Undisbursed_Net_Revenue,
  case
    when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <= 0 then 'Not due'
    when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=30 then '1-30 days late'
    when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=60 then '31-60 days late'
    when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=90 then '61-90 days late'
    when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=120 then '91-120 days late'

```

```

        when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end >=121 then '121+ days late'
        else null
        end as Disbursement_Period
from
    line_items_with_window_functions_enrich_offer_product_address_name as line
where disbursement_id_or_invoiced != '<invoiced>'

union

select
    -----
    -- Invoice Info --
    -----
    buyer_invoice_date as Invoice_Date,
    Payment_Due_Date as Payment_Due_Date,
    concat(
        'Net ',
        case
            when abs(date_diff('Day', Payment_due_date, buyer_invoice_date)) >180 then
'180+'
            else cast(abs(date_diff('Day', Payment_due_date, buyer_invoice_date)) as
varchar)
        end,
        ' days'
    ) as payment_terms,
    buyer_invoice_id as Invoice_ID,
    coalesce(
        --empty value in Athena shows as '', change all '' value to null
        case when seller_invoice_id = '' then null else seller_invoice_id end,
        'Not applicable') as Listing_Fee_Invoice_ID,

    -----
    --End user Information --
    -----
    coalesce(
        --empty value in Athena shows as '', change all '' value to null
        case when End_User_Company_Name = '' then null else End_User_Company_Name end,
        'Not available') as End_User_Company_Name,
    End_User_AWS_Account_ID,
    End_User_Encrypted_Account_ID,
    End_User_Email_Domain,
    End_User_City,

```

```

End_User_State as End_User_State_or_Region,
End_User_Country,
End_User_Postal_Code,
End_User_Address_ID,

-----
--Subscriber Information --
-----

case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_Company_Name is null or Subscriber_Company_Name = '' then 'Not
provided'
  else Subscriber_Company_Name
end as Subscriber_Company_Name,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  else Subscriber_AWS_Account_ID
end as Subscriber_AWS_Account_ID,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  else Subscriber_Encrypted_Account_ID
end as Subscriber_Encrypted_Account_ID,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_Email_Domain is null or Subscriber_Email_Domain = '' then 'Not
provided'
  else Subscriber_Email_Domain
end as Subscriber_Email_Domain,
case
  when Agreement_id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_City is null or Subscriber_City = '' then 'Not provided'
  else Subscriber_City
end as Subscriber_City,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_State is null or Subscriber_State = '' then 'Not provided'
  else Subscriber_State
end as Subscriber_State_or_Region,
case
  when Agreement_Id is null or Agreement_ID = '' then 'Not available'
  when Subscriber_Country is null or Subscriber_Country = '' then 'Not provided'
  else Subscriber_Country
end as Subscriber_Country,
case

```



```

    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    when Subscriber_Postal_Code is null or Subscriber_Postal_Code = '' then 'Not
provided'
    else Subscriber_Postal_Code
end as Subscriber_Postal_Code,
case
    when Agreement_ID is null or Agreement_ID = '' then 'Not available'
    when Subscriber_Address_ID is null or Subscriber_Address_ID = '' then 'Not
provided'
    else Subscriber_Address_ID
end as Subscriber_Address_ID,

-----
-- Procurement Info --
-----

coalesce(
    --empty value in Athena shows as '', change all '' value to null
    case when Product_Title = '' then null else Product_Title end,
    'Not provided') as Product_Title,
    -- offer name at time of invoice. It is possible that the name changes over time
therefore there may be multiple offer names mapped to a single offer id.
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    when (Offer_Name is null or Offer_Name = '') and Offer_Target = 'Public' then
'Not applicable'
    else Offer_Name
end as Offer_Name,
case
    when Agreement_Id is null or Agreement_ID = ''
    then 'Not available'
    else Offer_ID
end as Offer_ID,
    -- offer visibility at time of invoice.,
case
    when Agreement_Id is null or Agreement_ID = '' then 'Not available'
    else Offer_Target
end as Offer_Visibility,
coalesce(
    --empty value in Athena shows as '', change all '' value to null
    case when Agreement_ID = '' then null else Agreement_ID end,
    'Not available') as Agreement_ID,
    --case when Agreement_Id is null or Agreement_Id = '' then cast(null as timestamp)
else Agreement_Start_Date end as Agreement_Start_Date,

```

```

--case when Agreement_Id is null or Agreement_Id = '' then cast(null as timestamp)
else Agreement_End_Date end as Agreement_End_Date,
--case when Agreement_Id is null or Agreement_Id = '' then cast(null as timestamp)
else Agreement_Acceptance_Date end as Agreement_Acceptance_Date,
Agreement_Start_Date,
Agreement_Acceptance_Date,
Agreement_End_Date,

Usage_Period_Start_Date,
Usage_Period_End_Date,

-----
-- Disbursement Info --
-----
case
  when Disbursement_Flag = 'Yes' then 'Disbursed'
  when Disbursement_Flag = 'No' then 'Not Disbursed'
  else 'Other'
end as Disbursement_Status,
last_disbursement_date as disbursement_date,
case
  when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date))
  else null
end as Disbursement_Time,
case
  when Disbursement_Flag = 'No' then 'Not applicable'
  when bank_trace_id is null or bank_trace_id = '' then 'Not available'
  else bank_trace_id
end as disburse_bank_trace_id,

-----
-- Revenues --
-----
-- We are rounding the sums using 2 decimal precision
-- Note that the rounding method might differ between SQL implementations.
-- The monthly revenue report is using RoundingMode.HALF_UP. This might create tiny
discrepancies between this SQL output
-- and the legacy report
round(gross_revenue_this_disbursement_id_or_invoiced,2) as Gross_Revenue,
round(gross_refund_this_disbursement_id_or_invoiced,2) as Gross_Refund,
round(aws_rev_share_this_disbursement_id_or_invoiced,2) as Listing_Fee,
round(aws_refund_share_this_disbursement_id_or_invoiced,2) as Listing_Fee_Refund,
truncate(

```

```

        case
            when gross_revenue_this_disbursement_id_or_invoiced !
= 0 then abs(aws_rev_share_this_disbursement_id_or_invoiced/
gross_revenue_this_disbursement_id_or_invoiced)
            when gross_refund_this_disbursement_id_or_invoiced !=
0 then abs(aws_refund_share_this_disbursement_id_or_invoiced/
gross_refund_this_disbursement_id_or_invoiced)
            else 0
        end
        ,4) as Listing_Fee_Percentage,
        round(seller_tax_share_this_disbursement_id_or_invoiced,2) as Seller_Tax_Share,
        round(seller_tax_share_refund_this_disbursement_id_or_invoiced,2) as
Seller_Tax_Share_Refund,
        round(aws_tax_share_listing_fee_this_disbursement_id_or_invoiced,2) as
AWS_Tax_Share_Listing_Fee,
        round(aws_tax_share_refund_listing_fee_this_disbursement_id_or_invoiced,2) as
AWS_Tax_Share_Refund_Listing_Fee,
        round(cogs_this_disbursement_id_or_invoiced,2) as Wholesale_cost,
        round(cogs_refund_this_disbursement_id_or_invoiced,2) as Wholesale_cost_Refund,
        round(seller_net_revenue_this_disbursement_id_or_invoiced,2) as Seller_Net_Revenue,
        currency as Currency,

        substring(internal_buyer_line_item_id,1,strpos(internal_buyer_line_item_id,'-')-1)
as Transaction_Reference_ID,
        broker_id as AWS_seller_of_record,

        -----
        -- Resale info --
        -----
        case
            when Opportunity_Id is null or Opportunity_Id = '' then
                case
                    when Offer_Target = 'Public' then 'Not applicable'
                    when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
                    else null
                end
            else Opportunity_Id
        end as Resale_authorization_ID,
        case
            when Offer_Opportunity_Name is null or Offer_Opportunity_Name = '' then
                case
                    when Offer_Target = 'Public' then 'Not applicable'

```

```

        when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
        else null
        end
        else Offer_Opportunity_Name
    end as Resale_authorization_name,
    case
        when Offer_Opportunity_Description is null or Offer_Opportunity_Description = ''
then
        case
            when Offer_Target = 'Public' then 'Not applicable'
            when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
            else null
            end
        else Offer_Opportunity_Description
    end as Resale_authorization_description,
    case
        when (Reseller_AWS_Account_ID is not null and Reseller_AWS_Account_ID != '')
and (Reseller_Company_Name is null or Reseller_Company_Name = '') then 'Not
available'
        when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
and (opportunity_id is null or opportunity_id = '') then 'Not applicable'
        when (select seller_account_id from seller_account) '<>
manufacturer_aws_account_id
        and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
        else Reseller_Company_Name
    end as Reseller_Company_Name,
    case
        when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
and (Opportunity_Id is null or Opportunity_Id = '') then 'Not applicable'
        when (select seller_account_id from seller_account) '<>
manufacturer_aws_account_id
        and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
        else Reseller_AWS_Account_ID
    end as Reseller_AWS_Account_ID,

-----
-- Payer Information --
-----
coalesce(
    --empty value in Athena shows as '', change all '' value to null

```

```

        case when Payer_Company_Name = '' then null else Payer_Company_Name end,
        'Not available') as Payer_Company_Name,
Payer_AWS_Account_ID, -- "Customer AWS Account Number" in legacy report
Payer_Encrypted_Account_ID,
Payer_Email_Domain,
Payer_City,
Payer_State as Payer_State_or_Region,
Payer_Country,
Payer_Postal_Code,
Payer_Address_ID,

-----
-- ISV Information --
-----
manufacturer_aws_account_id as ISV_Account_ID,
coalesce(
    --empty value in Athena shows as '', change all '' value to null
    case when Manufacturer_Company_Name = '' then null else Manufacturer_Company_Name
end,
    'Not available') as ISV_Company_Name,

-----
-- Products info --
-----
-- product title at time of invoice. It is possible that the title changes over
time and therefore there may be multiple product titles mapped to a single product id.
Legacy_Product_ID,
coalesce(
    --empty value in Athena shows as '', change all '' value to null
    case when Product_ID = '' then null else Product_ID end,
    'Not provided') as Product_ID,
-- this is to get the legacy product id https://sim.amazon.com/issues/MP-
INSIGHTS-2561
Product_Code,

    case when Disbursement_Flag = 'Yes' then
round(seller_net_revenue_this_disbursement_id_or_invoiced,2) else 0 end as
Disbursed_Net_Revenue,
    case when Disbursement_Flag = 'No' then
round(seller_net_revenue_this_disbursement_id_or_invoiced,2) else 0 end as
Undisbursed_Net_Revenue,
    case

```

```

        when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <= 0 then 'Not due'
        when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=30 then '1-30 days late'
        when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=60 then '31-60 days late'
        when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=90 then '61-90 days late'
        when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end <=120 then '91-120 days late'
        when case when Disbursement_Flag = 'Yes' then date_diff('DAY',
date_trunc('DAY',payment_due_date), date_trunc('DAY',last_disbursement_date)) else
null end >=121 then '121+ days late'
        else null
    end as Disbursement_Period
from
    invoiced_not_disbursed
where rownum = 1

)

select *
from collections_and_disbursements
where payment_due_date >= date_add('DAY', -90, current_date)
--where payment_due_date between cast('2023-01-01' as timestamp) and cast('2024-12-31'
as timestamp)

--where disbursement_date >= date_add('DAY', -90, current_date)
--where disbursement_date between cast('2023-01-01' as timestamp) and cast('2024-12-31'
as timestamp)

```

課税請求書

課税請求書を検索するには、次の例のような一連のクエリを実行できます。クエリは相互に構築され、課税レポートを作成します。図のように例を使用するか、データやユースケースに合わせてカスタマイズできます。

クエリのコメントは、クエリの動作と変更方法について説明しています。

```
-- Taxation report

-- General note: When executing this query we are assuming that the data ingested in
the database is using
-- two time axes (the valid_from column and the update_date column).
-- See documentation for more details: https://docs.aws.amazon.com/marketplace/latest/userguide/data-feed.html#data-feed-details

-- An account_id has several valid_from dates (each representing a separate revision of
the data)
-- but because of bi-temporality, an account_id + valid_from tuple can appear multiple
times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
with accounts_with_uni_temporal_data as (
  select
    account_id,
    aws_account_id,
    encrypted_account_id,
    mailing_address_id,
    tax_address_id,
    tax_legal_name,
    from_iso8601_timestamp(valid_from) as valid_from,
    tax_registration_number
  from
    (
      select
        account_id,
        aws_account_id,
        encrypted_account_id,
        mailing_address_id,
        tax_address_id,
        tax_legal_name,
        valid_from,
        delete_date,
        tax_registration_number,
        row_number() over (partition by account_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
      from
        accountfeed_v1
    )
  where
    -- keep latest ...
```

```

    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

accounts_with_history as (
  with accounts_with_history_with_extended_valid_from as (
    select
      account_id,
      -- sometimes, this columns gets imported as a "bigint" and loses heading 0s ->
      casting to a char and re-adding heading 0s (if need be)
      substring('000000000000' || cast(aws_account_id as varchar), -12) as aws_account_id,
      encrypted_account_id,
      mailing_address_id,
      tax_address_id,
      tax_legal_name tax_legal_name,
      -- The start time of account valid_from is extended to '1970-01-01 00:00:00',
      because:
      -- ... in tax report transformations, some tax line items with invoice_date
      cannot
      -- ... fall into the default valid time range of the associated account
      CASE
        WHEN LAG(valid_from) OVER (PARTITION BY account_id ORDER BY valid_from ASC) IS
        NULL
          THEN CAST('1970-01-01 00:00:00' as timestamp)
        ELSE valid_from
      END AS valid_from
    from
      (select * from accounts_with_uni_temporal_data ) as account
  )
  select
    account_id,
    aws_account_id,
    encrypted_account_id,
    mailing_address_id,
    tax_address_id,
    tax_legal_name,
    valid_from,
    coalesce(
      lead(valid_from) over (partition by account_id order by valid_from asc),
      cast('2999-01-01 00:00:00' as timestamp)
    ) as valid_to
  from
    accounts_with_history_with_extended_valid_from

```



```
),

-- A product_id has several valid_from dates (each representing a product revision),
-- but because of bi-temporality, each product_id + valid_from tuple can appear
-- multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
products_with_uni_temporal_data as (
    select
        from_iso8601_timestamp(valid_from) as valid_from,
        from_iso8601_timestamp(update_date) as update_date,
        from_iso8601_timestamp(delete_date) as delete_date,
        product_id,
        manufacturer_account_id,
        product_code,
        title
    from
        (
            select
                valid_from,
                update_date,
                delete_date,
                product_id,
                manufacturer_account_id,
                product_code,
                title,
                row_number() over (partition by product_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
            from
                productfeed_v1
        )
    where
        -- keep latest ...
        row_num = 1
        -- ... and remove the soft-deleted one.
        and (delete_date is null or delete_date = '')
),

products_with_history as (
    select
        product_id,
        title,
        valid_from,
        case
```

```

        when lag(valid_from) over (partition by product_id order by valid_from asc) is
null and valid_from < cast('2021-04-01' as timestamp)
            then date_add('Day', -3857, valid_from)
        -- 3827 is the longest delay between acceptance_date of an agreement and the
product
        -- we are keeping 3857 as a consistency between the offers and products
        when lag(valid_from) over (partition by product_id order by valid_from asc) is
null and valid_from >= cast('2021-04-01' as timestamp)
            then date_add('Day', -2190, valid_from)
        --after 2021 for the two offers we need to adjust for 2 more years
        else valid_from end as valid_from_adjusted,
    coalesce(
        lead(valid_from) over (partition by product_id order by valid_from asc),
        cast('2999-01-01 00:00:00' as timestamp)
    ) as valid_to,
    product_code,
    manufacturer_account_id
from
    products_with_uni_temporal_data
),

-- A tax_item_id has several valid_from dates (each representing a product revision),
-- but because of bi-temporality, each tax_item_id + valid_from tuple can appear
multiple times with a different update_date.
-- We are only interested in the most recent tuple (ie, uni-temporal model)
tax_items_with_uni_temporal_data as (
    select
        from_iso8601_timestamp(valid_from) as valid_from,
        from_iso8601_timestamp(update_date) as update_date,
        delete_date,
        cast(tax_item_id as varchar) as tax_item_id,
        cast(invoice_id as varchar) as invoice_id,
        cast(line_item_id as varchar) as line_item_id,
        cast(customer_bill_id as varchar) as customer_bill_id,
        tax_liable_party,
        transaction_type_code,
        product_id,
        product_tax_code,
        from_iso8601_timestamp(invoice_date) as invoice_date,
        taxed_customer_account_id,
        taxed_customer_country,
        taxed_customer_state_or_region,
        taxed_customer_city,
        taxed_customer_postal_code,

```

```
tax_location_code_taxed_jurisdiction,  
tax_type_code,  
jurisdiction_level,  
taxed_jurisdiction,  
display_price_taxability_type,  
tax_jurisdiction_rate,  
tax_amount,  
tax_currency,  
tax_calculation_reason_code,  
date_used_for_tax_calculation,  
customer_exemption_certificate_id,  
customer_exemption_certificate_id_domain,  
customer_exemption_certificate_level,  
customer_exemption_code,  
customer_exemption_domain,  
transaction_reference_id  
from  
(  
    select  
        valid_from,  
        update_date,  
        delete_date,  
        tax_item_id,  
        invoice_id,  
        line_item_id,  
        customer_bill_id,  
        tax_liable_party,  
        transaction_type_code,  
        product_id,  
        product_tax_code,  
        invoice_date,  
        taxed_customer_account_id,  
        taxed_customer_country,  
        taxed_customer_state_or_region,  
        taxed_customer_city,  
        taxed_customer_postal_code,  
        tax_location_code_taxed_jurisdiction,  
        tax_type_code,  
        jurisdiction_level,  
        taxed_jurisdiction,  
        display_price_taxability_type,  
        tax_jurisdiction_rate,  
        tax_amount,  
        tax_currency,
```

```

        tax_calculation_reason_code,
        date_used_for_tax_calculation,
        customer_exemption_certificate_id,
        customer_exemption_certificate_id_domain,
        customer_exemption_certificate_level,
        customer_exemption_code,
        customer_exemption_domain,
        transaction_reference_id,
        row_number() over (partition by tax_item_id, valid_from order by
from_iso8601_timestamp(update_date) desc) as row_num
    from
        taxitemfeed_v1
    )
where
    -- keep latest ...
    row_num = 1
    -- ... and remove the soft-deleted one.
    and (delete_date is null or delete_date = '')
),

taxation as (
    select
        tax_items.invoice_id,
        tax_items.line_item_id,
        tax_items.customer_bill_id,
        tax_items.tax_liable_party,
        tax_items.transaction_type_code,
        tax_items.product_id,
        product_tax_item.title as product_title,
        tax_items.product_tax_code,
        tax_items.invoice_date,
        accounts_with_history.aws_account_id as taxed_customer_account_id,
        tax_items.taxed_customer_country,
        tax_items.taxed_customer_state_or_region,
        tax_items.taxed_customer_city,
        tax_items.taxed_customer_postal_code,
        tax_items.tax_type_code as tax_type,
        tax_items.jurisdiction_level,
        tax_items.taxed_jurisdiction,
        tax_items.display_price_taxability_type,
        tax_items.tax_jurisdiction_rate,
        tax_items.tax_amount,
        tax_items.tax_currency,
        tax_items.tax_calculation_reason_code,

```

```

    tax_items.date_used_for_tax_calculation,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when tax_items.customer_exemption_certificate_id = '' then null else
tax_items.customer_exemption_certificate_id end,
      'Not exempt') customer_exemption_certificate_id,
    coalesce(--empty value in Athena shows as '', change all '' value to null
      case when tax_items.customer_exemption_certificate_id_domain = '' then null else
tax_items.customer_exemption_certificate_id_domain end,
      'Not exempt') customer_exemption_certificate_id_domain,
    coalesce(--empty value in Athena shows as '', change all '' value to null
      case when tax_items.customer_exemption_certificate_level = '' then null else
tax_items.customer_exemption_certificate_level end,
      'Not exempt') customer_exemption_certificate_level,
    coalesce(--empty value in Athena shows as '', change all '' value to null
      case when tax_items.customer_exemption_code = '' then null else
tax_items.customer_exemption_code end,
      'Not exempt') customer_exemption_code,
    tax_items.transaction_reference_id
  from
    tax_items_with_uni_temporal_data as tax_items
  left join products_with_history as product_tax_item on
    tax_items.product_id = product_tax_item.product_id and tax_items.invoice_date
>= product_tax_item.valid_from_adjusted and tax_items.invoice_date <
product_tax_item.valid_to
  left join accounts_with_history as accounts_with_history on
    tax_items.taxed_customer_account_id = accounts_with_history.account_id and
tax_items.invoice_date >= accounts_with_history.valid_from and tax_items.invoice_date
< accounts_with_history.valid_to

)

select *
from taxation
where invoice_date >= date_add('DAY', -90, current_date)
--where invoice_date between cast('2023-01-01' as timestamp) and cast('2024-12-31' as
timestamp)

```

製品別の支払い

製品別に支払われた金額を確認するには、次のようなクエリを実行できます。この例は、[支払いレポート](#) 販売者レポートに相当します。

これらのサンプルクエリは、相互に構築され、支払いを含む製品詳細の最終リストを作成します。特定の時点での製品情報を取得する方法も示します。このサンプルは、図のように使用したり、データやユースケースに合わせてカスタマイズしたりできます。

クエリのコメントは、クエリの動作と変更方法について説明しています。

Note

このクエリを実行するときは、取り込まれたデータが 2 つの時間軸 (valid_from 列と update 列) を使用していることを前提としています。詳細については、「[AWS Marketplace データフィードのストレージと構造](#)」を参照してください。

```
-- Get all the products and keep the latest product_id, valid_from tuple
with products_with_uni_temporal_data as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY product_id, valid_from
          ORDER BY from_iso8601_timestamp(update_date) desc)
          as row_num
      from
        productfeed_v1
    )
  where
    -- A product_id can appear multiple times with the same
    -- valid_from date but with a different update_date column,
    -- making it effectively bi-temporal. By only taking the most
    -- recent tuple, we are converting to a uni-temporal model.
    row_num = 1
),

-- Gets the latest revision of a product
-- A product can have multiple revisions where some of the
```

```

-- columns, like the title, can change.
-- For the purpose of the disbursement report, we want
-- to get the latest revision of a product
products_with_latest_version as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY product_id
          ORDER BY from_iso8601_timestamp(valid_from) desc)
          as row_num_latest_version
      from
        products_with_uni_temporal_data
    )
  where
    row_num_latest_version = 1
),

-- Get all the accounts and keep the latest account_id, valid_from tuple
accounts_with_uni_temporal_data as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY account_id, valid_from ORDER BY
from_iso8601_timestamp(update_date) desc) as row_num
      from
        accountfeed_v1
    )
  where
    -- An account_id can appear multiple times with the same
    -- valid_from date but with a different update_date column,
    -- making it effectively bi-temporal. By only taking the most
    -- recent tuple, we are converting to a uni-temporal model.
    row_num = 1
),

-- Gets the latest revision of an account
-- An account can have multiple revisions where some of the
-- columns, like the mailing_address_id, can change.

```

```
-- For the purpose of the disbursement report, we want
-- to get the latest revision of a product
accounts_with_latest_version as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY account_id
          ORDER BY from_iso8601_timestamp(valid_from) desc)
          as row_num_latest_version
      from
        accounts_with_uni_temporal_data
    )
  where
    row_num_latest_version = 1
),

-- Get all the billing events and keep the
-- latest billing_event_id, valid_from tuple:
billing_events_with_uni_temporal_data as (
  select
    *
  from (
    select
      billing_event_id,
      from_iso8601_timestamp(valid_from) as valid_from,
      from_iso8601_timestamp(update_date) as update_date,
      from_iso8601_timestamp(invoice_date) as invoice_date,
      transaction_type,
      transaction_reference_id,
      product_id,
      disbursement_billing_event_id,
      action,
      from_account_id,
      to_account_id,
      end_user_account_id,
      CAST(amount as decimal(20, 10)) invoice_amount,
      bank_trace_id,
      ROW_NUMBER() OVER (PARTITION BY billing_event_id, valid_from
        ORDER BY from_iso8601_timestamp(update_date) desc)
        as row_num
    from
```



```

        billingeventfeed_v1
    )
    where row_num = 1
),

-- Get all the disbursements
-- The billing events data is immutable.
-- It is not required to use time windows based on the
-- valid_from column to get the most recent billing event
disbursement_events as (
    select
        billing_events_raw.billing_event_id as disbursement_id,
        billing_events_raw.invoice_date as disbursement_date,
        billing_events_raw.bank_trace_id
    from
        billing_events_with_uni_temporal_data billing_events_raw
    where
        -- Only interested in disbursements, so filter out
        -- non-disbursements by selecting transaction type
        -- to be DISBURSEMENT:
        billing_events_raw.transaction_type = 'DISBURSEMENT'
        -- Select a time period, you can adjust the dates
        -- below if need be. For billing events use the
        -- invoice date as the point in time of the
        -- disbursement being initiated:
        and billing_events_raw.invoice_date >=
            from_iso8601_timestamp('2020-10-01T00:00:00Z')
        and billing_events_raw.invoice_date <
            from_iso8601_timestamp('2020-11-01T00:00:00Z')
),

-- Get the invoices along with the line items that
-- are part of the above filtered disbursements
disbursed_line_items as (
    select
        line_items.transaction_reference_id,
        line_items.product_id,
        line_items.transaction_type,
        (case
            -- Get the payer of the invoice from any
            -- transaction type that is not AWS and
            -- not BALANCE_ADJUSTMENT.
            -- For AWS and BALANCE_ADJUSTMENT, the billing
            -- event feed will show the "AWS Marketplace"

```

```

        -- account as the receiver of the funds and the
        -- seller as the payer. Filter those out.
    when line_items.transaction_type
        not like '%AWS%' and transaction_type
        not like 'BALANCE_ADJUSTMENT'
        then line_items.from_account_id
    end) as payer_account_id,
    line_items.end_user_account_id,
    invoice_amount,
    disbursements.disbursement_date,
    disbursements.disbursement_id,
    disbursements.bank_trace_id
from
    billing_events_with_uni_temporal_data line_items
    -- Each disbursed line item is linked to the parent
    -- disbursement via the disbursement_billing_event_id
join disbursement_events disbursements
    on disbursements.disbursement_id
    = line_items.disbursement_billing_event_id
where
    -- we are interested only in the invoice line
    -- items that are DISBURSED
    line_items.action = 'DISBURSED'
),

-- An invoice can contain multiple line items
-- Create a pivot table to calculate the different
-- amounts that are part of an invoice.
-- The new row is aggregated at
-- transaction_reference_id - end_user_account_id level
invoice_amounts_aggregated as (
    select
        transaction_reference_id,
        product_id,
        -- a given disbursement id should have the
        -- same disbursement_date
        max(disbursement_date) as disbursement_date,
        -- Build a pivot table in order to provide all the
        -- data related to a transaction in a single row.
        -- Note that the amounts are negated. This is because
        -- when an invoice is generated, we give you the
        -- positive amounts and the disbursement event
        -- negates the amounts
        sum(case when transaction_type = 'SELLER_REV_SHARE'

```

```

        then -invoice_amount else 0 end) as seller_rev_share,
sum(case when transaction_type = 'AWS_REV_SHARE'
        then -invoice_amount else 0 end) as aws_rev_share,
sum(case when transaction_type = 'SELLER_REV_SHARE_REFUND'
        then -invoice_amount else 0 end) as seller_rev_refund,
sum(case when transaction_type = 'AWS_REV_SHARE_REFUND'
        then -invoice_amount else 0 end) as aws_rev_refund,
sum(case when transaction_type = 'SELLER_REV_SHARE_CREDIT'
        then -invoice_amount else 0 end) as seller_rev_credit,
sum(case when transaction_type = 'AWS_REV_SHARE_CREDIT'
        then -invoice_amount else 0 end) as aws_rev_credit,
sum(case when transaction_type = 'SELLER_TAX_SHARE'
        then -invoice_amount else 0 end) as seller_tax_share,
sum(case when transaction_type = 'SELLER_TAX_SHARE_REFUND'
        then -invoice_amount else 0 end) as seller_tax_refund,
-- This is the account that pays the invoice:
max(payer_account_id) as payer_account_id,
-- This is the account that subscribed to the product:
end_user_account_id as customer_account_id,
bank_trace_id
from
    disbursed_line_items
group by
    transaction_reference_id,
    product_id,
    disbursement_id,
    -- There might be a different end-user for the same
    -- transaction reference id. Distributed licenses
    -- is an example
    end_user_account_id,
    bank_trace_id
),

disbursed_amount_by_product as (
    select
        products.title as ProductTitle,
        products.product_code as ProductCode,
        -- We are rounding the sums using 2 decimal precision
        -- Note that the rounding method might differ
        -- between SQL implementations.
        -- The disbursement seller report is using
        -- RoundingMode.HALF_UP. This might create
        -- discrepancies between this SQL output
        -- and the disbursement seller report

```

```
round(invoice_amounts.seller_rev_share, 2) as SellerRev,
round(invoice_amounts.aws_rev_share, 2) as AWSRefFee,
round(invoice_amounts.seller_rev_refund, 2) as SellerRevRefund,
round(invoice_amounts.aws_rev_refund, 2) as AWSRefFeeRefund,
round(invoice_amounts.seller_rev_credit, 2) as SellerRevCredit,
round(invoice_amounts.aws_rev_credit, 2) as AWSRefFeeCredit,
(
    round(invoice_amounts.seller_rev_share, 2) +
    round(invoice_amounts.aws_rev_share, 2) +
    round(invoice_amounts.seller_rev_refund, 2) +
    round(invoice_amounts.aws_rev_refund, 2) +
    round(invoice_amounts.seller_rev_credit, 2) +
    round(invoice_amounts.aws_rev_credit, 2)
) as NetAmount,
invoice_amounts.transaction_reference_id
    as TransactionReferenceID,
round(invoice_amounts.seller_tax_share, 2)
    as SellerSalesTax,
round(invoice_amounts.seller_tax_refund, 2)
    as SellerSalesTaxRefund,
payer_info.aws_account_id
    as PayerAwsAccountId,
customer_info.aws_account_id
    as EndCustomerAwsAccountId,
invoice_amounts.disbursement_date
    as DisbursementDate,
invoice_amounts.bank_trace_id
    as BankTraceId
from
    invoice_amounts_aggregated invoice_amounts
join products_with_latest_version products
    on products.product_id = invoice_amounts.product_id
left join accounts_with_latest_version payer_info
    on payer_info.account_id = invoice_amounts.payer_account_id
left join accounts_with_latest_version customer_info
    on customer_info.account_id = invoice_amounts.customer_account_id
)

select * from disbursed_amount_by_product;
```

販売補償レポート

顧客別の請求済み収益を確認するには、次のようなクエリを実行します。この例は、[販売補償レポート](#) 販売者レポートに相当します。これらのサンプルクエリは、相互に構築され、各顧客のソフトウェア使用料に対して請求された総額を含む顧客詳細の最終リストを作成します。クエリは図のように使用したり、データやユースケースに合わせてカスタマイズしたりできます。

クエリのコメントは、クエリの動作と変更方法について説明しています。

Note

このクエリを実行するときは、取り込まれたデータが 2 つの時間軸 (valid_from 列と update 列) を使用していることを前提としています。詳細については、「[AWS Marketplace データフィードのストレージと構造](#)」を参照してください。

```
-- Gets all the products and keeps the latest product_id,
-- valid_from tuple.
with products_with_uni_temporal_data as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY product_id, valid_from
                           ORDER BY from_iso8601_timestamp(update_date) desc)
        as row_num
      from
        productfeed_v1
    )
  where
    -- A product_id can appear multiple times with the same
    -- valid_from date but with a different update_date column,
    -- making it effectively bi-temporal. By only taking the most
    -- recent tuple, we are converting to a uni-temporal model.
    row_num = 1
),

-- Gets the latest revision of a product
-- A product can have multiple revisions where some of the
-- columns, like the title, can change.
```

```
-- For the purpose of the sales compensation report, we want
-- to get the latest revision of a product
products_with_latest_revision as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY product_id ORDER BY
from_iso8601_timestamp(valid_from) desc) as row_num_latest_revision
      from
        products_with_uni_temporal_data
    )
  where
    row_num_latest_revision = 1
),

-- Gets all the addresses and keeps the latest address_id,
-- aws_account_id, and valid_from combination.
-- We're transitioning from a bi-temporal data model to an
-- uni-temporal data_model
piifeed_with_uni_temporal_data as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (
          PARTITION BY address_id, aws_account_id, valid_from
          ORDER BY from_iso8601_timestamp(update_date) desc)
          as row_num
        from
          piifeed
      )
  where
    -- An address_id can appear multiple times with the same
    -- valid_from date but with a different update_date column.
    -- We are only interested in the most recent.
    row_num = 1
),

-- Gets the latest revision of an address.
```

```
-- An address_id can have multiple revisions where some of
-- the columns can change.
-- For the purpose of the sales compensation report, we want to
-- get the latest revision of an address + account_id pair.
pii_with_latest_revision as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY address_id, aws_account_id
          ORDER BY from_iso8601_timestamp(valid_from) desc)
          as row_num_latest_revision
      from
        piifeed_with_uni_temporal_data
    )
  where
    row_num_latest_revision = 1
),

-- Gets all the accounts and keeps the latest
-- account_id, valid_from tuple.
-- We're transitioning from a bi-temporal data
-- model to an uni-temporal data_model.
accounts_with_uni_temporal_data as (
  select
    *
  from
    (
      select
        *,
        ROW_NUMBER() OVER (PARTITION BY account_id, valid_from
          ORDER BY from_iso8601_timestamp(update_date) desc)
          as row_num
      from
        accountfeed_v1
    )
  where
    -- An account_id can appear multiple times with the same
    -- valid_from date but with a different update_date column.
    -- We are only interested in the most recent tuple.
    row_num = 1
),
```

```

-- Gets all the historical dates for an account
-- An account can have multiple revisions where some of the
-- columns like the mailing_address_id can change.
accounts_with_history as (
  select
    *,
    -- This interval's begin_date
    case
      when
        -- First record for a given account_id
        lag(valid_from, 1) over (partition by account_id
                                order by from_iso8601_timestamp(valid_from) asc) is null
      then
        -- 'force' begin_date a bit earlier because of different
        -- data propagation times. We'll subtract one day as one
        -- hour is not sufficient
        from_iso8601_timestamp(valid_from) - INTERVAL '1' DAY
      else
        -- not the first line -> return the real date
        from_iso8601_timestamp(valid_from)
    end as begin_date,
    -- This interval's end date.
    COALESCE(
      LEAD(from_iso8601_timestamp(valid_from), 1)
        OVER (partition by account_id
              ORDER BY from_iso8601_timestamp(valid_from)),
      from_iso8601_timestamp('9999-01-01T00:00:00Z')
    ) as end_date
  from
    accounts_with_uni_temporal_data
),

-- Gets all the billing events and keeps the latest
-- billing_event_id, valid_from tuple.
-- We're transitioning from a bi-temporal data
-- model to an uni-temporal data_model.
billing_events_with_uni_temporal_data as (
  select
    *
  from (
    select
      billing_event_id,
      from_iso8601_timestamp(valid_from) as valid_from,

```



```

        from_iso8601_timestamp(update_date) as update_date,
        from_iso8601_timestamp(invoice_date) as invoice_date,
        transaction_type,
        transaction_reference_id,
        product_id,
        disbursement_billing_event_id,
        action,
        currency,
        from_account_id,
        to_account_id,
        end_user_account_id,
        -- convert an empty billing address to null. This will
        -- later be used in a COALESCE call
        case
            when billing_address_id <> '' then billing_address_id else null
        end as billing_address_id,
        CAST(amount as decimal(20, 10)) invoice_amount,
        ROW_NUMBER() OVER (PARTITION BY billing_event_id, valid_from
            ORDER BY from_iso8601_timestamp(update_date) desc)
            as row_num
    from
        billingeventfeed_v1
    where
        -- The Sales Compensation Report does not contain BALANCE
        -- ADJUSTMENTS, so we filter them out here
        transaction_type <> 'BALANCE_ADJUSTMENT'
        -- Keep only the transactions that will affect any
        -- future disbursed amounts.
        and balance_impacting = '1'
    )
    where row_num = 1
),

-- Gets the billing address for all DISBURSED invoices. This
-- will be the address of the payer when the invoice was paid.
-- NOTE: For legal reasons, for CPP0 transactions, the
-- manufacturer will not see the payer's billing address id
billing_addresses_for_disbursed_invoices as (
    select
        billing_events_raw.transaction_reference_id,
        billing_events_raw.billing_address_id,
        billing_events_raw.from_account_id
    from
        billing_events_with_uni_temporal_data billing_events_raw

```

```

where
    -- the disbursed items will contain the billing address id
    billing_events_raw.action = 'DISBURSED'
    -- we only want to get the billing address id for the
    -- transaction line items where the seller is the receiver
    -- of the amount
    and billing_events_raw.transaction_type like 'SELLER_%'
group by
    billing_events_raw.transaction_reference_id,
    billing_events_raw.billing_address_id,
    billing_events_raw.from_account_id
),

-- An invoice can contain multiple line items.
-- We create a pivot table to calculate the different amounts
-- that are part of an invoice.
-- The new row is aggregated at
-- transaction_reference_id - end_user_account_id level
invoiced_and_forgiven_transactions as (
select
    transaction_reference_id,
    product_id,
    -- A transaction will have the same invoice date for all
    -- of its line items (transaction types)
    max(invoice_date) as invoice_date,
    -- A transaction will have the same billing_address_id
    -- for all of its line items. Remember that the billing event
    -- is uni temporal and we retrieved only the latest valid_from item
    max(billing_address_id) as billing_address_id,
    -- A transaction will have the same currency for all
    -- of its line items
    max(currency) as currency,
    -- We're building a pivot table in order to provide all the
    -- data related to a transaction in a single row
    sum(case when transaction_type = 'SELLER_REV_SHARE'
        then invoice_amount else 0 end) as seller_rev_share,
    sum(case when transaction_type = 'AWS_REV_SHARE'
        then invoice_amount else 0 end) as aws_rev_share,
    sum(case when transaction_type = 'SELLER_REV_SHARE_REFUND'
        then invoice_amount else 0 end) as seller_rev_refund,
    sum(case when transaction_type = 'AWS_REV_SHARE_REFUND'
        then invoice_amount else 0 end) as aws_rev_refund,
    sum(case when transaction_type = 'SELLER_REV_SHARE_CREDIT'
        then invoice_amount else 0 end) as seller_rev_credit,

```

```

sum(case when transaction_type = 'AWS_REV_SHARE_CREDIT'
      then invoice_amount else 0 end) as aws_rev_credit,
sum(case when transaction_type = 'SELLER_TAX_SHARE'
      then invoice_amount else 0 end) as seller_tax_share,
sum(case when transaction_type = 'SELLER_TAX_SHARE_REFUND'
      then invoice_amount else 0 end) as seller_tax_refund,
-- this is the account that pays the invoice.
max(case
  -- Get the payer of the invoice from any transaction type
  -- that is not AWS and not BALANCE_ADJUSTMENT.
  -- For AWS and BALANCE_ADJUSTMENT, the billing event feed
  -- will show the "AWS Marketplace" account as the
  -- receiver of the funds and the seller as the payer. We
  -- are not interested in this information here.
  when
    transaction_type not like '%AWS%'
    and transaction_type not like 'BALANCE_ADJUSTMENT'
    then from_account_id
  end) as payer_account_id,
-- this is the account that subscribed to your product
end_user_account_id as customer_account_id
from
  billing_events_with_uni_temporal_data
where
  -- Get invoiced or forgiven items. Disbursements are
  -- not part of the sales compensation report
  action in ('INVOICED', 'FORGIVEN')
group by
  transaction_reference_id,
  product_id,
  -- There might be a different end-user for the same
  -- transaction reference id. Distributed licenses
  -- is an example.
  end_user_account_id
),

invoiced_items_with_product_and_billing_address as (
  select
    invoice_amounts.*,
    products.product_code,
    products.title,
    payer_info.aws_account_id as payer_aws_account_id,
    payer_info.account_id as payer_reference_id,
    customer_info.aws_account_id as end_user_aws_account_id,

```

```

(
    invoice_amounts.seller_rev_share +
    invoice_amounts.aws_rev_share +
    invoice_amounts.seller_rev_refund +
    invoice_amounts.aws_rev_refund +
    invoice_amounts.seller_rev_credit +
    invoice_amounts.aws_rev_credit +
    invoice_amounts.seller_tax_share +
    invoice_amounts.seller_tax_refund
) as seller_net_revenue,
-- Try to get the billing address from the DISBURSED event
-- (if any). If there is no DISBURSEMENT, get the billing
-- address from the INVOICED item. If still no billing address,
-- then default to getting the mailing address of the payer.
coalesce(billing_add.billing_address_id,
         invoice_amounts.billing_address_id,
         payer_info.mailing_address_id)
         as final_billing_address_id
from
    invoiced_and_forgiven_transactions invoice_amounts
    join products_with_latest_revision products
        on products.product_id = invoice_amounts.product_id
    left join accounts_with_history payer_info
        on payer_info.account_id = invoice_amounts.payer_account_id
        -- Get the Payer Information at the time of invoice creation
        and payer_info.begin_date <= invoice_amounts.invoice_date
        and invoice_amounts.invoice_date < payer_info.end_date
    left join accounts_with_history customer_info
        on customer_info.account_id = invoice_amounts.customer_account_id
        -- Get the End User Information at the time of invoice creation
        and customer_info.begin_date <= invoice_amounts.invoice_date
        and invoice_amounts.invoice_date < customer_info.end_date
    left join billing_addresses_for_disbursed_invoices billing_add
        on billing_add.transaction_reference_id =
            invoice_amounts.transaction_reference_id
        and billing_add.from_account_id =
            invoice_amounts.payer_account_id
),

invoices_with_full_address as (
    select
        payer_aws_account_id as "Customer AWS Account Number",
        pii_data.country as "Country",
        pii_data.state_or_region as "State",

```

```
pii_data.city as "City",
pii_data.postal_code as "Zip Code",
pii_data.email_domain as "Email Domain",
product_code as "Product Code",
title as "Product Title",
seller_rev_share as "Gross Revenue",
aws_rev_share as "AWS Revenue Share",
seller_rev_refund as "Gross Refunds",
aws_rev_refund as "AWS Refunds Share",
seller_net_revenue as "Net Revenue",
currency as "Currency",
date_format(invoice_date, '%Y-%m')as "AR Period",
transaction_reference_id as "Transaction Reference ID",
payer_reference_id as "Payer Reference ID",
end_user_aws_account_id as "End Customer AWS Account ID"
from
    invoiced_items_with_product_and_billing_address invoice_amounts
left join pii_with_latest_revision pii_data
    on pii_data.aws_account_id = invoice_amounts.payer_aws_account_id
    and pii_data.address_id = invoice_amounts.final_billing_address_id
-- Filter out FORGIVEN and Field Demonstration Pricing transactions
where seller_net_revenue <> 0
)

select * from invoices_with_full_address;
```

データフィード

AWS Marketplace には、販売者が製品の販売に関する情報を収集および分析するのに役立つ多数のデータフィードが用意されています。データフィードは、登録済みのすべての AWS Marketplace 販売者が利用できます。データフィードは 1 日以内に生成されるため、最新のデータを利用できます。

以降のセクションでは、それぞれのデータフィードについて説明します。

トピック

- [アカウントデータフィード](#)
- [住所データフィード](#)
- [契約データフィード](#)
- [請求イベントデータフィード](#)
- [レガシーマッピングデータフィード](#)

- [オファーデータフィード](#)
- [オファー製品データフィード](#)
- [オファーターゲットデータフィード](#)
- [製品データフィード](#)
- [税品目データフィード](#)

アカウントデータフィード

このデータフィードは、ユーザーが操作するすべてのアカウント (ユーザー自身、使用しているチャネルパートナー、購入者、支払者、およびすべての課税対象アカウント) に関する情報を提供します。

アカウントデータはイミュータブルであり、バージョン番号は関連付けられません。フィールドに対する変更は付け加えられるため、このデータフィードには同じ `account_id` 値と、異なる `valid_from` 値を持つ複数の行が存在する場合があります。データの履歴フィールドの詳細については、「[データの履歴化](#)」を参照してください。

アカウントデータフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

以下の表では、データフィードの列の名前と説明を示しています。

列名	説明
<code>account_id</code>	アカウントのグローバル一意識別子 (GUID)。 Product、Offer_Target 、Billing_Event 、Tax_Item およびデータフィードのフィールドへの結合に使用できます。結合に使用できるフィールドについては、これらのデータフィードを参照してください。
<code>aws_account_id</code>	販売者の AWS アカウントの AWS アカウント番号。AWS パーティションごとに一意です。
<code>encrypted_account_id</code>	アプリケーションの購入者別の一意の暗号化された ID。encrypted_account_id の値は、例えば、 ResolveCustomer アクションによって返される CustomerIdentifier

列名	説明
	の値として、AWS Marketplace メータリングサービスによって使用されます。
mailing_address_id	このアカウントの郵送先住所の参照。
tax_address_id	このアカウントの税宛先住所の参照。
tax_registration_number	米国以外のアカウントの場合、このアカウントの税登録番号。
tax_legal_name	米国以外のアカウントの場合、法人名。これは、税金請求書で使用される名前です。

アカウントデータフィードの例

アカウントデータフィードの例は次のとおりです。見やすいように、データの履歴列は表示していません。データの履歴フィールドの詳細については、「[データの履歴化](#)」を参照してください。

account_id	aws_account_id	encrypted_account_id	mailing_address_id	tax_address_id	tax_registration_number	tax_legal_name
xk0CSmiAm6PQ4QqEog9iiaochlz uPlkMfba7a1oDIZ	444456660000	Zf7oMzheGWpH	25o3k46eN6eViOfFii qtxwX8e3k aOiPalUio fjyFa3			
7nyo5jwTRoPlyX81vx 9ji04eEwTurO1Ff8bi Qi88W8	555567679999	373vuQUqmQ8v	5oJ6vTjSzMrrF2gvh2 Vj9HfqiM800MuLEHmy FY5Lr42s8	5oJ6vTjSzMrrF2gvh2 Vj9HfqiM800MuLEHmy FY5Lr42s8	SE823935083345	
VleGa2t9j3MuxioH9w c8lsndXXC	73739998888	8SPxAYmi8MwX	NLUc5UeiMIGFTTrDWCo ftDPhDUF1	V5NhBYBiYogwy0WMhn dGU4AfMgg	DE469558025	

account_id	aws_account_id	encrypted_account_id	mailing_address_id	tax_address_id	tax_registration_number	tax_legal_name
gGCGUreeX riocM5			oaSd8xgl5 QM8Db7	muoTC2j7P m8ZKKNNyT		

住所データフィード

Important

AWS Marketplace では、[販売者配送データフィードサービス](#)を使用して配信される PIIFeed を 2023 年 12 月に廃止します。必要な住所データには AddressFeed_v1 データフィードを使用してください。質問がある場合やサポートが必要な場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

このデータフィードは、ユーザーが操作するすべてのアカウント (ユーザー自身、使用しているチャネルパートナー、購入者、支払者、およびすべての課税対象アカウント) に関する連絡先情報を提供します。新しい取引が発生するたびに、その取引の顧客住所がスキャンされ、その住所がデータフィードにない場合は、新しいエントリがデータフィードファイルに追加されます。

住所データはイミュータブルです。

住所データフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

以下の表では、データフィードの列の名前と説明を示しています。

列名	説明
address_id	住所の一意のキー。
	billing_address_id フィールドの Billing_Event データフィード、mailing_address_id または tax_address_id フィールドの Account データフィードからの結合に使用できます。

列名	説明
email_domain	このアカウントの登録済み E メールアドレスのドメイン。
company_name	このアカウントの登録済み会社名。
country_code	この住所の登録済み ISO 3166 alpha-2 国コード。
state_or_region	この住所の登録済み都道府県またはリージョン。
city	この住所の登録済み市区町村。
postal_code	この住所の登録済み郵便番号。
address_line_1	この住所の登録済み行 (1 行目)。
address_line_2	この住所の登録済み行 (2 行目)。
address_line_3	この住所の登録済み行 (3 行目)。

住所データフィールドの例

住所データフィールドの例を次に示します。この情報は、データフィールドの単一のテーブルに表示されます。見やすいように、ここではデータを 2 つのテーブルに表示し、データの履歴列は表示していません。データの履歴フィールドの詳細については、「[データの履歴化](#)」を参照してください。

address_id	email_domain	company_name	country_code	state_or_region	city	postal_code
V5NhBYBiYogwy0WMhndGU4AfMggmuoTC2jPm8ZKKNNyT	a.com	Mateo Jackson's Company	DE		Hamburg	67568

address_id	email_domain	company_name	country_code	state_or_region	city	postal_code
G68xdbkZQ DVVHzfBGw 6yf5yos0A 6NiSVWHmT 5ViLjf	b.com	Mary Major's Company	米国	OH	Dayton	57684
NLUc5UeiM IGFTTrDWCo ftDPhDUF1 oaSd8xgl5 QM8Db7	c.com	Our Seller	米国	NY	New York	89475

address_line_1	address_line_2	address_line_3
	19th Floor	

契約データフィード

このデータフィードは、製品の使用を開始するために提案者 (登録販売者) とアクセプタ (AWS 購入者) の間で署名された契約である契約に関する情報を提供します。このデータフィードは、登録販売者として作成したすべての製品オファークの契約に関する情報を提供します。

契約データフィードは 24 時間ごとに更新されます。

次の表は、データフィードの項目を一覧表示し、説明しています。

列	[Description] (説明)
valid_from	主キーの値が他のフィールドの値に関連して有効である最初の日付。
insert_date	レコードがデータフィードに挿入された日付。
update_date	レコードが最後に更新された日付。
delete_date	この列は常に空白です。
ステータス	契約の現在の状態。サポートされているステータスは次のとおりです。 • アクティブ – 契約の条項の一部またはすべてが有効な状態です。 • 終了 – AWS が開始した終了事由により、契約は事前に合意した終了日より前に終了した状態です。最も一般的なイベントは、支払いの不履行です。 • キャンセル済み – 承認者が終了日より前に契約を終了することを選択した状態です。 • 期限切れ – 契約が事前に合意された終了日に終了した状態です。 • 置き換え済み – 契約が代替オファーを使用して置き換えられた状態です。 • 更新済み – 自動更新などの機能を使用して、契約が新しい契約に更新された状態です。 • アーカイブ済み – 契約が終了しましたが、契約の終了、キャンセル、期限切れの正確な理由が指定されていない状態です。 • ロールバック – エラーのため、契約改訂がロールバックされた状態です。以前の改訂が現在有効になっています。非アクティブな契約改訂にのみ適用されます。

列	[Description] (説明)
	優先 – 契約改訂がアクティブでなくなり、別の契約改訂がアクティブな状態です。非アクティブな契約改訂にのみ適用されます。
estimated_agreement_value	契約のライフサイクル全体で顧客による支払いが見込まれる既知の合計金額。
currency_code	契約のライフサイクル全体で顧客による支払いが見込まれる既知の合計金額の通貨。
agreement_id	契約の一意の識別子。
proposer_account_id	この購入契約を提案した販売者は、販売者のアカウントのグローバル一意識別子 (GUID) で表されます。アカウントデータフィールドに参加するために使用できます。
acceptor_account_id	この購入契約を承認した購入者は、購入者のアカウントのグローバル一意識別子 (GUID) で表されます。アカウントデータフィールドに参加するために使用できます。
offer_revision_at_acceptance	本契約に対応するオファーのわかりやすい ID。オファーおよびオファーターゲットデータフィールドに参加するために使用できます。
start_time	契約開始日時。
end_time	契約開始日時。終了日のない従量制料金の場合、このフィールドは、null です。
acceptance_time	オファー承認日時または、契約作成日時。 - 所有しているライセンスの持ち込み (BYOL) 契約の場合、過去の日付を記載可能です。 - 契約が将来の日付の契約機能で作成された場合、start_date とは異なる場合があります

列	[Description] (説明)
intent	契約が最後に変更されたときの購入者の意向。
preceding_agreement_id	前の契約の契約 ID。
status_reason_code	契約ステータスが変更された理由。

契約データフィードの例

valid_m	agreement_id	product_id	accepted_id	offer_id	offer_id_accepted	status	end_date	accepted_date	intent	preceding_agreement_id	ステータス	status_code	estimated_value	currency						
2026年06月21日 20:00	agreement_id	product_id	accepted_id	offer_id	offer_id_accepted	status	2026年06月21日 20:00	2026年06月21日 20:00	新規		AC		1,000	USD						

請求イベントデータフィード

このデータフィードは、請求や支払いなど、請求イベントに関する情報を提供します。

たとえば、このデータフィードを使用して、購入者への請求日や請求内容を確認できます。[サンプルの SQL クエリ](#)を使用して、このデータフィードのデータを分析することもできます。

このデータフィードは、登録販売者としての請求イベントに関連する情報を示します。チャネルパートナーを介して締結された契約の場合、このデータフィードはメーカーと登録販売者との請求イベントに関する情報を示します。

請求イベントデータフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

請求イベントデータはイミュータブルです。

以下の表では、データフィードの列の名前と説明を示しています。

列名	説明
billing_event_id	請求イベントの識別子。この ID は、販売者の環境内で一意です。
from_account_id	請求イベントを開始したアカウント。transaction_type が SELLER_REV_SHARE である場合は、購入者の支払いアカウントです。これは、アカウントデータフィードへの外部キーです。 account_id フィールドの Account データフィードへの結合に使用できます。
to_account_id	製品の取引金額を受け取るアカウント。これは、アカウントデータフィードへの外部キーです。 account_id フィールドの Account データフィードへの結合に使用できます。
end_user_account_id	製品を使用するアカウント。このアカウントは、購入者アカウントや支払者アカウントとは異なる場合があります。 account_id フィールドの Account データフィードへの結合に使用できます。
product_id	製品の識別子。これは 製品データフィードへの外部キーです。 product_id フィールドの Product データフィードへの結合に使用できます。
アクション	このイベントのアクションのタイプ。可能な値は以下のとおりです。

列名	説明
	• INVOICED - 購入者に該当金額を請求しました。 • FORGIVEN - 購入者に該当金額を請求し、AWS がその請求を取り消しました。 • DISBURSED - 販売者に該当金額が支払われました。これは、1 か月分の請求書をまとめるか、オンデマンド支払いとすることができます。

列名	説明
transaction_type	取引のタイプ。例については「課税シナリオ」を参照してください。可能な値は以下のとおりです。 • SELLER_REV_SHARE - 正の金額。販売者が購入者との契約で設定した料金です。 • SELLER_TAX_SHARE - 正の金額。販売者が支払う税金として SELLER_REV_SHARE に追加される金額です。 • AWS_REV_SHARE - 負の金額。出品手数料です。 • AWS_TAX_SHARE - 正の金額。これは、に加えて AWS 徴収される税金の金額です SELLER_REV_SHARE 。この金額は販売者の残高には影響しません。この金額は支払い対象ではなく、購入者に請求される税額が販売者に代わって当局に支払われることを販売者に知らせるために提供されます。 • <i>transaction_type</i> _REFUND - 購入者からリクエストされた返金額。 • <i>transaction_type</i> _CREDIT - 購入者に AWS クレジットする金額。 • BALANCE_ADJUSTMENT - 請求の問題を解決 AWS するために によって行われた調整。 • DISBURSEMENT - action の値が DISBURSED で、balance_impacting の値が 1 の場合、これは販売者に支払われる金額です。action の値が INVOICED である場合、このレコードは parent_billing_event_id レコードを全面的または部分的に無効にします。この場合、関連する支払い disbursement_bill

列名	説明
	ng_event_id が表示され、balance_impacting の値は 0 です。 DISBURSEMENT_FAILURE - 取引を無効にします。

列名	説明
parent_billing_event_id	broker_id の値が AWS_INC で、action の値が DISBURSED または FORGIVEN で、transaction_type の値が DISBURSEMENT の場合、parent_billing_event_id はこの請求イベントを開始した元の billing_event_id の金額を指します。action に別の値がある場合、このフィールドは null です。 broker_id の値が AWS_EUROPE の場合、parent_billing_event_id は次のシナリオで請求イベントを開始した元の billing_event_id を指します。 • action の値は FORGIVEN または INVOICED で、transaction_type の値は、AWS_REV_SHARE、AWS_REV_SHARE_REFUND、または SELLER_REV_SHARE_REFUND です。 • action の値は DISBURSED で、transaction_type の値は ANY (DISBURSEMENT_FAILURE を除く) です。 • transaction_type の値は、AWS_TAX_SHARE、AWS_TAX_SHARE_REFUND、SELLER_TAX_SHARE、または SELLER_TAX_SHARE_REFUND です。 • action の値は DISBURSED で、transaction_type の値は DISBURSEMENT_FAILURE です。 broker_id の値が AWS_EUROPE の場合、parent_billing_event_id は次の

列名	説明
	シナリオで前回成功した支払い請求イベントの元の <code>billing_event_id</code> を指します。 <code>action</code> の値は <code>DISBURSED</code> で、<code>transaction_type</code> の値は <code>DISBURSEMENT_FAILURE</code> です。 <code>broker_id</code> の値が <code>AWS_EUROPE</code> の場合、残りのシナリオではこのフィールドは <code>null</code> になります。
<code>disbursement_billing_event_id</code>	<code>action</code> の値が <code>DISBURSED</code> であり、さらに次のいずれかに該当する場合の関連支払額。 <code>transaction_type like ('SELLER%')</code> <code>transaction_type like ('AWS%')</code> 他のすべてのシナリオでは、この値は <code>null</code> です。
<code>amount</code>	請求イベントの金額。
<code>currency</code>	ISO 639 通貨コード。
<code>balance_impacting</code>	販売者の支払いを計算する際に該当金額を考慮するかどうか。0 の値は、該当金額が情報提供のために表示されており、残高には影響しないことを示します。1 の値は、該当金額が販売者の支払いを決定する際に考慮されることを示します。
<code>invoice_date</code>	請求書が作成された日付。
<code>payment_due_date</code>	<code>action</code> の値が <code>INVOICED</code> である場合は、請求書の支払期限日です。

列名	説明
usage_period_start_date	レコードの期間の開始日。
usage_period_end_date	レコードの期間の終了日。
invoice_id	AWS 請求書 ID。
billing_address_id	住所データフィールドにおける支払人の請求連絡先の参照。 address_id フィールドの Address データフィールドへの結合に使用できます。
transaction_reference_id	以下のレポートのデータを相互参照できる識別子。 • 支払いレポート • 月別請求済み収益レポート • 販売補償レポート
bank_trace_id	支払い取引 (transaction_type = DISBURSEMENT で、アクションはDISBURSED) の場合、銀行によって割り当てられたトレース ID。トレース ID は、銀行が提供する販売者銀行からのレポートと関連付けるために使用できます。

列名	説明
broker_id	取引を促進した事業体の識別子。可能な値は以下のとおりです。 - AWS_INC - AWS Inc. の識別子 (米国を拠点とする)。 - AWS_EUROPE - Amazon Web Services EMEA SARL (ルクセンブルク拠点) の識別子。 - AWS_JAPAN - AWS_AUSTRALIA - AWS_KOREA - NULL - 明示的な broker_id のない以前の取引は、AWS_INC によって円滑に進められました。
buyer_transaction_reference_id	GROUP BY または SQL で作成されたウィンドウ関数を使用して、請求フィード内のすべての関連レコードをグループ化する識別子。これらの関連レコードには、購入者請求書、販売者請求書、および出品料金にかかる付加価値税 (VAT) が含まれる場合があります。

課税シナリオ

購入者と販売者の国および州に適用されている課税モデルによって、税金の徴収と支払の方法が決まります。考えられるシナリオは次のとおりです。

- 税金は によって収集され、送金されます AWS。この場合、transaction_type は AWS_TAX_SHARE です。
- 税金は AWS、販売者が徴収し、販売者に支払い、販売者が税務機関に送金します。この場合、transaction_type は SELLER_TAX_SHARE です。

- 税金は によって徴収されません AWS。販売者が税金を計算して税務当局に納税する必要があります。このような場合、 は税計算を実行 AWS Marketplace したり、税情報を受信したりしません。販売者が収益分配金から納税します。

請求イベントデータフィードの例

このセクションでは、請求時と 1 か月後の請求イベントデータ期間の例を示します。このセクションのすべての表について、次の点に注意してください。

- データフィードの `billing_event_id` 値は 40 文字の英数字文字列です。ここでは、見やすいように 2 文字の文字列として表示しています。
- この情報は、データフィードの単一のテーブルに表示されます。見やすいように、ここではデータを複数のテーブルに表示し、一部の列を省略しています。

このセクションの例では、以下を前提とします。

- Arnav は購入者です。
 - アカウント ID は 737399998888 です。
 - 居住国のフランスでは、マーケットプレイスファシリテーター法が適用されます。詳細については、「[アマゾン ウェブ サービス税務ヘルプ](#)」を参照してください。
 - `prod-o4grxfafcx` を購入し、この製品の月間使用料として 120.60 USD を請求されました。
 - 当月内に請求額を支払いました。
- Jane はメーカーです。
 - アカウント ID は 111122223333 です。
- Paulo は登録販売者です。
 - アカウント ID は 777788889999 です。
 - 居住しているカンザス州は、マーケットプレイスファシリテーター法の対象ではありません。

登録販売者の請求イベントデータフィード

Paulo は、登録販売者として購入者の Arnav に請求を行います。

次の表は、Paulo が Arnav に請求した際のデータフィードの関連情報を示しています。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
I0	737399998888	777788889999	737399998888	prod-o4grxfafcxxxx	INVOICED	SELLER_REV_SHARE
I1	737399998888	AWS	737399998888	prod-o4grxfafcxxxx	INVOICED	AWS_TAX_SHARE
I2	777788889999	111122223333	737399998888	prod-o4grxfafcxxxx	INVOICED	SELLER_REV_SHARE
I3	777788889999	AWS	737399998888	prod-o4grxfafcxxxx	INVOICED	AWS_REV_SHARE

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		100	USD	2018-12-31T00:00:00Z	781216640
		20.6	USD	2018-12-31T00:00:00Z	781216640
		-80	USD	2018-12-31T00:04:07Z	788576665
		-0.2	USD	2018-12-31T00:04:07Z	788576665

次の表は、Arnav が請求額を支払った後の月末における Paulo のデータフィードの関連情報を示しています。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
I10	737399998888	777788889999	737399998888		DISBURSED	SELLER_REV_SHARE
I12	777788889999	111122223333	737399998888		DISBURSED	SELLER_REV_SHARE
I13	777788889999	AWS	737399998888	prod-o4grxfafcxxxx	DISBURSED	AWS_REV_SHARE
I14	AWS	777788889999			DISBURSED	DISBURSEMENT

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
I0	I14	-100	USD	2018-12-31T00:00:00Z	781216640
I2	I14	80	USD	2018-12-31T00:04:07Z	788576665
I3	I14	0.2	USD	2018-12-31T00:04:07Z	788576665
		19.8	USD		

メーカーの請求イベントデータフィード

次の表は、Paulo が Arnav に請求した際の Jane のデータフィードの関連情報を示しています。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
I5	77778888999	11112222333		prod-o4grxfafcxxxx	INVOICED	SELLER_REV_SHARE
I6	77778888999	11112222333		prod-o4grxfafcxxxx	INVOICED	SELLER_TAX_SHARE
I7	11112222333	AWS		prod-o4grxfafcxxxx	INVOICED	AWS_REV_SHARE

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		73.5		2018-12-31T00:04:07Z	788576665
		6.5		2018-12-31T00:04:07Z	788576665
		-7.35		2018-12-31T00:04:07Z	788576665

次の表は、請求額が支払われた後の月末における Jane のデータフィードの関連情報を示しています。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
I30	77778888999	11112222333		prod-o4grxfafcxxxx	DISBURSED	SELLER_REV_SHARE

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
I31	77778888999	11112222333		prod-o4grxfafcxxxx	DISBURSED	SELLER_TAX_SHARE
I32	11112222333	AWS		prod-o4grxfafcxxxx	DISBURSED	AWS_REV_SHARE
I33	AWS	11112222333			DISBURSED	DISBURSEMENT

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
I5	I33	-73.5	USD		
I6	I33	-6.5	USD		
I7	I33	7.35	USD		
		72.65	USD		

クエリの例

[データフィードを使用したデータ収集と分析](#) で説明しているように、[Athena](#) を使用して、マネージド Amazon S3 バケットにデータフィードとして収集されて保存されたデータに対してクエリを実行できます。このセクションでは、これを行う一般的な方法の例をいくつか示します。すべての例で、単一の通貨を使用するものとします。

例 1: 請求金額 (税込み)

購入者への請求金額 (税込み) を確認するには、次の例に示すようなクエリを実行できます。

```
SELECT sum(amount) FROM billing_event
WHERE
```

```
action = 'INVOICED'
AND
(
  (transaction_type in ('SELLER_REV_SHARE', 'SELLER_TAX_SHARE')
    -- to discard SELLER_REV_SHARE from Manufacturer to Channel Partner, aka cost of
    goods
    AND to_account_id='seller-account-id'
  )
OR transaction_type= 'AWS_TAX_SHARE'
);
```

例 2: 販売者に代わって購入者に請求する金額

販売者に代わって購入者に請求した金額を確認するには、次の例に示すようなクエリを実行できます。

```
SELECT sum(amount) FROM billing_event
WHERE
  action = 'INVOICED'
  AND transaction_type in ('SELLER_REV_SHARE', 'SELLER_TAX_SHARE')
  AND to_account_id='seller-account-id'
;
```

例 3: 販売者に代わって回収 AWS できる金額

販売者に代わって が回収 AWS できる金額から返金、クレジット、および許されたアカウントを差し引いた金額を調べるには、次の例に示すようにクエリを実行できます。

```
SELECT sum(amount) FROM billing_event
WHERE
  -- what is invoiced on behalf of SELLER, incl. refunds/ credits and cost of goods
  transaction_type like 'SELLER_%'
  -- FORGIVEN action records will "negate" related INVOICED
  and action in ('INVOICED', 'FORGIVEN')
;
```

例 4: 販売者が徴収できる金額

販売者が徴収できる金額を確認するには、次の例に示すようなクエリを実行できます。この例では、が AWS 徴収する出品料金と税金を削除し、例外的な残高調整を追加します。

```
SELECT sum(amount) FROM billing_event
```

```
WHERE
  (transaction_type like 'SELLER_%' -- what is invoiced on behalf of SELLER
  or transaction_type like 'AWS_REV_%' -- what is owed to AWS
  or transaction_type = 'BALANCE_ADJUSTMENT' -- exceptionnal case
  )
  and action in ('INVOICED','FORGIVEN')
;
```

また、次の例に示すように、次のクエリを使用して同じ情報を収集することもできます。

```
SELECT sum(amount) FROM billing_event
WHERE
  balance_impacting = 1
  and action in ('INVOICED','FORGIVEN')
;
```

次の例は同じ情報を示していますが、2018 年の取引に限定し、すべての購入者が請求額を支払ったと仮定しています。

```
SELECT sum(amount) FROM billing_event
WHERE
  invoice_date between '2018-01-01' and '2018-12-31'
  and balance_impacting = 1
  and action in ('INVOICED','FORGIVEN')
;
```

例 5: 支払い額

支払われた金額を確認するには、次の例に示すようなクエリを実行できます。

```
select sum(amount) FROM billing_event
WHERE
  action = 'DISBURSED'
  and transaction_type like 'DISBURSEMENT%'
;
```

例 6: 保留中の支払い金額

保留中の支払い金額を確認するには、次の例に示すようなクエリを実行できます。このクエリでは、支払い済みの金額を除外します。

```

SELECT sum(amount) FROM billing_event targeted
WHERE
    (transaction_type like 'SELLER_%' -- what is invoiced on behalf of SELLER
    or transaction_type like 'AWS_REV_%' -- what is owed to AWS
    or transaction_type = 'BALANCE_ADJUSTMENT' -- exceptionnal case
    )
-- DISBURSEMENT action records will "negate" 'INVOICED'
-- but do not take into account failed disbursements
AND
    (not exists
        (select 1
            from billing_event disbursement
            join billing_event failed_disbursement
            on disbursement.billing_event_id=failed_disbursement.parent_billing_event_id
            where
                disbursement.transaction_type='DISBURSEMENT'
                and failed_disbursement.transaction_type='DISBURSEMENT_FAILURE'
                and targeted.disbursement_billing_event_id=disbursement.billing_event_id
            )
        )
    )
;

```

同じ情報を取得する別の方法としては、次の例に示すようなクエリを実行して販売者の残高を取得します。

```

SELECT sum(amount) FROM billing_event
WHERE
    balance_impacting = 1
;

```

次のクエリでは、この例を拡張します。クエリ結果を 2018 年の取引に限定し、これらの取引に関する追加の詳細を返します。

```

select sum(residual_amount_per_transaction)
from
    (SELECT
        max(billed_invoices.amount) invoiced_amount,
        sum(nvl(disbursed_invoices.amount,0)) disbursed_amount,
        -- Exercise left to the reader:
        -- use transaction_type to distinguish listing fee vs seller-owed money
        -- still pending collection
        max(transaction_type) transaction_type,

```

```

    max(billed_invoices.amount)
    + sum(nvl(disbursed_invoices.amount,0)) residual_amount_per_transaction
FROM billing_event billed_invoices
-- find related disbursements
left join billing_event disbursed_invoices
    on disbursed_invoices.action='DISBURSED'
    and disbursed_invoices.parent_billing_event_id=billed_invoices.billing_event_id
WHERE
    billed_invoices.invoice_date between '2018-01-01' and '2018-12-31'
    and billed_invoices.transaction_type like 'SELLER_%' -- invoiced on behalf of
SELLER
    and billed_invoices.action in ('INVOICED','FORGIVEN')
    -- do not take into account failed disbursements
AND not exists
    (select 1 from billing_event failed_disbursement
     where disbursed_invoices.disbursement_billing_event_id =
failed_disbursement.parent_billing_event_id
    )
GROUP BY billed_invoices.billing_event_id
);

```

例 7: 一連の請求書の残高

一連の請求書の合計額を確認するには、次の例に示すようなクエリを実行できます。

```

SELECT invoice_id, sum(amount) FROM billing_event targeted
WHERE
-- invoice_id is only not null for invoiced records AND disbursed records
-- linking them to related disbursement -> no need to filter more precisely
invoice_id in ('XXX','YYY')
-- filter out failed disbursements
AND not exists
    (select 1
     from billing_event disbursement
     join billing_event failed_disbursement
     on disbursement.billing_event_id=failed_disbursement.parent_billing_event_id
     where
     disbursement.transaction_type='DISBURSEMENT'
     and failed_disbursement.transaction_type='DISBURSEMENT_FAILURE'
     and targeted.disbursement_billing_event_id=disbursement.billing_event_id
    )
group by invoice_id;

```

レガシーマッピングデータフィード

このデータフィードは、製品 ID とオファー ID をレガシーのグローバル一意識別子 (GUID) にマッピングする方法を示します。レガシー GUID は古いレポートで使用され、新しい ID はデータフィードと AWS Marketplace API で使用されています。

このデータフィードは、登録販売者として作成したすべての製品と、再販が許可されているすべての製品に関する情報を提供します。

レガシーマッピングデータフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

以下の表では、データフィードの列の名前と説明を示しています。

列名	説明
mapping_type	製品 ID とオファー ID のいずれであることを示します。
legacy_id	この製品またはオファーのレガシー ID。
new_id	この製品またはオファーのわかりやすい ID。 この ID は、現在のすべての API アクションで主キーとして使用されます。

レガシーマッピングデータフィードの例

レガシーマッピングデータフィードの例は次のとおりです。見やすいように、データの履歴列は表示していません。データの履歴フィールドの詳細については、「[データの履歴化](#)」を参照してください。

mapping_type	legacy_id	new_id
OFFER	8a806c74-dbd6-403e-9362-bb08f417ff37	offer-dacpxznflfw
PRODUCT	1368541d-890b-4b6c-9bb9-4a55306ab642	prod-o4grxfafcxxy

mapping_type	legacy_id	new_id
OFFER	558d8382-6b3a-4c75-8345-a627b552f5f1	offer-gszhmle5npzip

オファーデータフィード

オファーデータフィードは、登録販売者として作成したすべてのオファーに関する情報を提供します。1つのオファーに複数のリビジョンがある場合、すべてのリビジョンがデータフィードに含まれます。

オファーリビジョンを作成し、公開フィールドのデータが変わると、同じプライマリキー (offer_id プラス offer_revision) データフィードに新しいレコードが作成されます。ただし、valid_from フィールドの値は異なります。データフィードの履歴列の詳細については、「[データの履歴化](#)」を参照してください。

オファーデータフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

以下の表には、データフィードの列の名前と説明が示されています。

列名	説明
offer_id	オファーのわかりやすい識別子。 Offer_Product データフィードの offer_id フィールドへの結合に使用できません。
offer_revision	オファーリビジョン。このフィールドと offer_id フィールドを組み合わせると主キーを形成します。 offer_id では、Target_Offer データフィードの offer_id と offer_revision フィールドの結合に使用できます。
name	販売者が定義したオファー名。

列名	説明
expiration_date	オファーの有効期限が切れる日時。
opportunity_name	このオファーにリンクされた営業案件データ。オファーが AWS の営業案件にバインドされている場合、このフィールドは入力済みです。
opportunity_description	このオファーにリンクされているすべての説明情報。オファーが AWS の営業案件にバインドされている場合、このフィールドは入力済みです。
seller_account_id	販売者のアカウントのグローバル一意識別子 (GUID)。アカウントデータフィードの account_id フィールドと結合するために使用できます。
opportunity_id	機会の ID は、リセラーが製品を販売している場合にのみ入力されます。製品が同じであれば、異なるチャネルパートナー (または販売者) が作成したオファーにはすべて同じ opportunity_id があります。

オファーデータフィードの例

オファーデータフィードの例は次のとおりです。見やすいように、データの履歴列は表示していません。データの履歴フィールドの詳細については、「[データの履歴化](#)」を参照してください。

offer_id	offer_revision	名前	expiration_date	opportunity_name	opportunity_description	seller_account_id	opportunity_id
offer-dac pxznflfw in	1	エンター プライズ 契約プロ	9999-01-0 1T00:00:0 0Z				

offer_id	offer_revision	名前	expiration_date	opportunity_name	opportunity_description	seller_account_id	opportunity_id
		グラム オフ					
offer-gsz hml5npzip	1	販売者 が作成し たプライ ベートオ フ	2020-10-3 1T00:00:0 0Z				
offer-hmz hyle8nphlp	1	エンター プライズ 契約プロ グラムオ フ	9999-01-0 1T00:00:0 0Z				

オファー製品データフィード

1 つのオファーには複数の製品を含めることができ、1 つの製品は複数の異なるオファーに含めることができます。このデータフィードは、オファーと製品との関係に関する情報を示します。

このデータフィードは、登録販売者として作成したすべての製品オファーに関する情報を提供します。

オファーに対して製品を追加または削除した場合は、オファーリビジョンを作成できます。

オファー製品データフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

以下の表では、データフィードの列の名前と説明を示しています。データフィードの履歴列の詳細については、「[データの履歴化](#)」を参照してください。

列名	説明		
offer_id	このオファーのわかりやすい識別子。		

列名	説明		
	Offer データフィールドの offer_id フィールドへの結合に使用できます。		
offer_revision	offer_id フィールドと組み合わせて、オファーリビジョンへの外部キーを形成します。		
product_id	製品のわかりやすい識別子。これは、このオファーで公開する製品への外部キーです。 Product データフィールドの product_id フィールドへの結合に使用できます。		

オファー製品データフィールドの例

オファー製品データフィールドの例は次のとおりです。

offer_id	offer_revision	product_id
offer-dacpxznflfwin	10	prod-o4grxfafcxxxx
offer-gszhmle5npzip	24	prod-o4grxfafcxxxxy

オファーターターゲットデータフィード

このデータフィードは、登録販売者として作成したすべてのオファーに関するオファーのリビジョンのターゲットを示します。1つのオファーに複数のリビジョンがある場合、すべてのリビジョンがデータフィードに含まれます。

オファーリビジョンを作成し、公開フィールドのデータが変わると、別のレコードがデータフィードに作成されます。このレコードの主キー (offer_id プラス offer_revision) は同じですが、valid_from フィールドの値は異なります。

オファーターターゲットデータフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

以下の表では、データフィードの列の名前と説明を示しています。

列名	説明
offer_target_id	フィードの主キー。
offer_id+offer_revision	オファーの識別子とリビジョン。これら 2 つの列は、このターゲットに関連するオファーを参照します。 Targetデータフィードの offer_id および offer_revision フィールドへの結合に使用できます。
target_type	オファーの受取人が BuyerAccounts (プライベートオファーを示す) であるか、ParticipatingPrograms であるかを示します。
polarity	オファーが target_type を対象にしているかどうかを示します。指定できる値は次のとおりです。 PositiveTargeting オファーは、この target_type に適用されます。

列名	説明
	NegativeTargeting オファーは、この target_type に適用されません。
value	ターゲットを表す文字列 (AWS アカウント ID またはオファーで使用するプログラム)。例: AWS Marketplace の標準契約 (SCMP) 、または AWS Marketplace フィールドデモンストレーションプログラム (FDP) 。

オファーターターゲットデータフィールドの例

オファーターターゲットデータフィールドの例を次に示します。見やすいように、データの履歴列は表示していません。データの履歴フィールドの詳細については、「[データの履歴化](#)」を参照してください。

offer_target_id	offer_id	offer_revision	target_type	polarity	value
925ddc73f6a373b7d5544ea3210610803b600	offer-dacpxznflfwin	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract
471ff22ae3165278f1fb960d3e14517bcd601	offer-gszhmle5npzip	1	ParticipatingPrograms	PositiveTargeting	FieldDemonstration
511ff22adfj65278f1fb960d3e14517bcd6e602	offer-gszhmle5npzip	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract

製品データフィード

このデータフィードは、登録販売者として作成したすべての製品と、再販が許可されているすべての製品に関する情報を提供します。

製品データは変更可能です。つまり、次のいずれかのフィールドの値を変更すると、valid_from フィールドの値が異なる別のレコードがデータフィードに作成されます。データフィードの履歴列の詳細については、「[データの履歴化](#)」を参照してください。

製品データフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

以下の表では、データフィードの列の名前と説明を示しています。

列名	説明
product_id	製品のわかりやすい識別子。 Account、Billing_Event 、Offer_Product データフィードの product_id フィールドと結合するために使用できます。
manufacturer_account_id	製品の所有者の識別子。これは、 アカウント データフィードへの外部キーです。 Account データフィードの account_id フィールドへの結合に使用できます。
product_code	製品の計測に使用される、権利が付与された既存の製品コード。この値は、データをレポートに結合する場合や、AWS Marketplace Metering Service で提供される情報を参照する場合にも使用されます。
title	製品のタイトル。

製品データフィードの例

オファーターゲットデータフィードの例を次に示します。見やすいように、データの履歴列は表示していません。データの履歴フィールドの詳細については、「[データの履歴化](#)」を参照してください。

product_id	manufacturer_account_id	product_code	title
prod-o4grxfafcxxxx	555568000000	product_code_1	Product1
prod-t3grxfafcxxxy	444457000000	product_code_2	Product2
prod-x8faxxfafcxy	666678000000	product_code_3	Product3

税品目データフィード

このデータフィードは、顧客請求書の税額計算に関する情報を提供します。

特定の顧客請求書 (invoice_id) の特定の製品 (product_id) に対して複数の明細行品目 (line_item_id) が存在する場合があります。顧客請求書は各税務管轄区域に 1 つ以上存在する場合があります。これは、例えば、異なる AWS エンティティ (米国とアイルランドなど) によって異なる AWS リージョンルールを使用している顧客に対する使用量ベースの請求で発生します。が売上に対する消費税、VAT、または GST をどこで AWS 徴収し、AWS, Inc. の名でそのような税金を現地の税務機関に送金するかの詳細については、[「Amazon Web Service Tax Help」](#)を参照してください。

税品目データフィードは 24 時間ごとに更新されるため、新しいデータを毎日利用できます。

税項目データはイミュータブルです。

以下の表では、データフィードの列の名前と説明を示しています。データの履歴列の詳細については、「[データの履歴化](#)」を参照してください。

列名	説明
tax_item_id	税品目レコードの一意的識別子。
invoice_id	AWS 請求書 ID。この値と product_id の値を組み合わせ、関連する税金請求イベントを検索できます。
line_item_id	顧客請求明細項目の一意的識別子。返金取引の明細項目 ID は先渡し取引税のものと同じです。

列名	説明
customer_bill_id	顧客請求書の一意の識別子。購入者は、この識別子を販売者と共有し、税額計算に関する質問を特定して解決できます。
tax_liable_party	AWS または Seller です。販売者が納税義務者である場合は、税金が徴収されます。AWS が納税義務者である場合、消費税は によって徴収および送金されます AWS。詳細については、AWS Marketplace 「Sellers & Tax Collection」 を参照してください。 徴税しない場合、ここには値が表示されません。販売者は、徴税の義務があるため、請求書ごとに徴税されたかどうかを確認する必要があります。
transaction_type_code	取引のタイプ。可能な値は以下のとおりです。 • AWS - 源泉徴収取引 • REFUND - 全額返金または一部返金 • TAXONLYREFUND - 税のみの返金 返金取引の明細項目 ID は元の先渡し取引のものと同じです
product_id	製品の外部キー。 product_id フィールドの Product データ フィードへの結合に使用できます。
product_tax_code	製品の税プロパティを識別するための標準コード。販売者は、製品の作成時または変更時にプロパティを選択します。
invoice_date	請求書が作成された日付。

列名	説明
taxed_customer_account_id	課税対象の取引先企業エンティティの外部キー。 account_id フィールドの Account データ フィードへの結合に使用できます。
taxed_customer_country	税額計算に使用する住所の ISO 3166 alpha 2 の国別コード。
taxed_customer_state_or_region	税額計算に使用する州、区、郡。
taxed_customer_city	税額計算に使用する市。
taxed_customer_postal_code	税額計算に使用する郵便番号。
tax_location_code_taxed_jurisdiction	課税場所に関連付けられている頂点ジオコード。
tax_type_code	取引に適用される税のタイプ。指定できる値は、None、Sales および SellerUse です。
jurisdiction_level	納税地に使用される住所の管轄区域。指定できる値は、State、County、City、District です。
taxed_jurisdiction	税務管轄区域の名前。
display_price_taxability_type	購入者に料金を税込みで表示するか、税抜きで表示するか。すべての AWS Marketplace サービスには税金は含まれていません。
taxable_amount	該当する管轄区域レベルで課税対象となる取引金額。
nontaxable_amount	該当する管轄区域レベルで非課税となる取引金額。

列名	説明
tax_jurisdiction_rate	該当する管轄区域レベルで適用される税率。
tax_amount	該当する管轄区域レベルで請求される税額。
tax_currency	上記の金額に関する ISO 4217 alpha 3 の通貨コード。
tax_calculation_reason_code	取引が課税対象となるか、非課税となるか、免除とされるか、ゼロ評価済みであるかを管轄区域別に整理したもの。
date_used_for_tax_calculation	取引の税金の計算に使用される日付。
customer_exemption_certificate_id	免除証明書の証明書 ID。
customer_exemption_certificate_id_domain	Amazon システム上の証明書の保存先。
customer_exemption_certificate_level	免除を提供した管轄区域。
customer_exemption_code	免税を指定するコード (RESALE など)。
customer_exemption_domain	顧客の免税情報 (ある場合) をキャプチャするために使用する Amazon システム。
transaction_reference_id	以下のレポートのデータを相互参照できる識別子。 • 支払いレポート • 月別請求済み収益レポート • 販売補償レポート

Note

2021 年 8 月 5 日以降、AWS Marketplace 売上に対する国際マーケットプレイスファシリテーター税は、税項目データフィールドにエントリが含まれます。つまり、2021 年 8 月 5 日以降、請求イベントデータフィールドのすべての AWS_TAX_SHARE レコードと

SELLER_TAX_SHARE レコードは、税項目データフィールドに対応するレコードを持つことが予想されます。

税品目データフィールドの例

税品目データフィールドの例を次に示します。この情報は、データフィールドの単一のテーブルに表示されます。見やすいように、ここではデータを複数のテーブルに表示し、一部の列を省略しています。

tax_item_id	invoice_id	line_item_id	customer_bill_id
6p2ni6tu041xagvhby anbgxl3xameha16txj oav_0001	781216640	710000000 00000000000	221000000 0000000000
6p2ni6tu041xagvhby anbgxl3xameha16txj oav_0002	781216640	530000000 00000000000	221000000 0000000000
flr4jobxjzww8czdsr q4noue2uxd56j39wxw 0k7_0001	250816266	764000000 00000000000	572000000 0000000000
gfkjjobxjzw56jgkrs rqgjt52uxd56j39wg j567d_0002	280336288	764000000 00000000000	572439000 0000000000
wwk1qpvb8ran3geiw8 e3mp6dgs2 qj7wpkuwhgk1_0001	451431024	993000000 00000000000	123000000 0000000000
wwk1qpvb8ran3geiw8 e3mp6dgs2 qj7wpkuwhgk1_0002	451431024	993000000 00000000000	312000000 0000000000
fnohdid8kwgqq9lvii 2k30spn3ftgwihbe8h 75x_0001	229987654	921000000 00000000000	639000000 0000000000

tax_liable_party	transacti on_type_code	product_id	product_t ax_code	invoice_date
販売者	AWS	prod-o4gr xfafcxxxx	AWSMP_SOF TWARE_RA	2018-12-3 1T00:00:00Z
販売者	AWS	prod-o4gr xfafcxxxx	AWSMP_SOF TWARE_RA	2018-12-3 1T00:00:00Z
販売者	AWS	prod-t3gr xfafcxxxxy	AWS_REMOT E_ACCESS_ SOFTWARE	2018-08-3 1T00:00:00Z
販売者	REFUND	prod-t3gr xfafcxxxxy	AWS_REMOT E_ACCESS_ SOFTWARE	2018-08-3 1T00:00:00Z
販売者	AWS	prod-x8fa xxfafcxyy	AWS_REMOT E_ACCESS_ SOFTWARE	2018-08-3 1T00:00:00Z
販売者	TAXONLYRE FUND	prod-x8fa xxfafcxyy	AWS_REMOT E_ACCESS_ SOFTWARE	2018-05-3 1T00:00:00Z
AWS	AWS	prod-wghj 8xfafrhgj	AWS_REMOT E_ACCESS_ SOFTWARE	2019-07-3 1T00:00:00Z

taxed_cus tomer_acc ount_id	taxed_cus tomer_country	taxed_cus tomer_sta te_or_region	taxed_cus tomer_city	taxed_cus tomer_pos tal_code
VleGa2t9j 3MuxioH9w c8lsndXXC	米国	GA	MILTON	48573-4839

taxed_customer_account_id	taxed_customer_country	taxed_customer_state_or_region	taxed_customer_city	taxed_customer_postal_code
gGCGUreeX riocM5				
VleGa2t9j 3MuxioH9w c8lsndXXC gGCGUreeX riocM5	米国	GA	MILTON	48573-4839
7nyo5jwTR oPlyX81vx 9ji04eEwT urO1Ff8bi Qi88W8	米国	NC	DURHAM	27517-4834
7nyo5jwTR oPlyX81vx 9ji04eEwT urO1Ff8bi Qi88W8	米国	NC	DURHAM	27517-4834
7nyo5jwTR oPlyX81vx 9ji04eEwT urO1Ff8bi Qi88W8	米国	TX	NOT APPLICABLE	75844-1235
7nyo5jwTR oPlyX81vx 9ji04eEwT urO1Ff8bi Qi88W8	米国	TX	HOUSTON	75844-1235

taxed_customer_account_id	taxed_customer_country	taxed_customer_state_or_region	taxed_customer_city	taxed_customer_postal_code
192a04213 13e41f069 b52962ed7 babf716291b688	米国	CT	NEW HAVEN	06002-2948

tax_location_code_taxed_jurisdiction	tax_type_code	jurisdiction_level	taxed_jurisdiction	display_price_taxability_type	taxable_amount	nontaxable_amount
460473664	セールス	状態	GA	除外	100	0
66301164	セールス	郡	FULTON	税抜き	0	100
692938178	SellerUse	状態	NC	除外	58.1	523.8
692938178	SellerUse	状態	NC	除外	-58.1	523.8
356794387	セールス	状態	TX	除外	1105.14	0
528887443	セールス	市町村	HOUSTON	除外	-36	0
171248162	セールス	状態	CT	税抜き	0	114.55

tax_jurisdiction_rate	tax_amount	tax_currency	tax_calculation_reason_code	date_used_for_tax_calculation
0.206	20.6	USD	課税対象	2018-10-3 1T00:00:00Z

tax_jurisdiction_rate	tax_amount	tax_currency	tax_calculation_reason_code	date_used_for_tax_calculation
0	0	USD	非課税	2018-10-31T00:00:00Z
0.1	5.8	USD	課税対象	2018-07-31T00:00:00Z
0.1	-5.8	USD	課税対象	2018-07-31T00:00:00Z
0.06	66.3	USD	課税対象	2018-07-31T00:00:00Z
0.01	-0.36	USD	非課税	2018-07-31T00:00:00Z
0	0	USD	免税	2019-06-30T00:00:00Z

販売者が でレポートする AWS Marketplace

Important

2024 年 8 月 30 日、はいくつかのレポートとデータセットを中止 AWS Marketplace します。

- Marketplace レガシーカンマ区切り (csv) 販売者レポート
- The Commerce Analytics Service (CAS) API
- 関連する E メール通知

影響を受けるレポートについてレポートページを確認するか、質問があれば 5/30/2024 にコミュニケーションを発行した AWS 管理者に確認してください。

AWS Marketplace は、製品の使用状況、購入者、請求、支払いに関する情報を含むレポートを提供します。レポートは、登録済みのすべての AWS Marketplace 販売者が利用できます。

レポートの生成に関する重要なポイントは、次のとおりです。

- レポートは、その種類に応じて毎日、毎週、または毎月生成されます。
- レポートは 00:00 UTC に生成され、前日の 24:00 UTC までを対象とします。
- レポートは .csv ファイルとして生成されます。
- 暗号化された Amazon SNS Amazon S3 を設定できます。通知を設定すると、は に登録した AWS アカウントに関連付けられている E メールアドレスに通知 AWS を送信します AWS Marketplace。

通知を設定する方法の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[Amazon SNS の開始方法](#)」を参照してください。

通知 E メールを受信をキャンセルするには、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

- 各レポートの詳細については、[サンプルレポート](#)をダウンロードしてください。

レポートへのアクセス

AWS Marketplace には、レポートを設定する 2 つの方法があります。

- API インターフェイスの使用。[AWS Marketplace Commerce Analytics Service を使用して、製品および顧客データにアクセスする](#)を使用する。この方法では API インターフェイスを介してレポート内のデータに自動的にアクセスできます。情報の取り込みを自動化し、レポート全体ではなくレポートの一部をダウンロードできます。このサービスは、従来の API のように直接ではなく、Amazon Simple Storage Service のファイルに非同期的にデータを返します。データはコンピュータが読み取り可能な形式で配信されるため、システムにデータをインポートしたり組み込んだりすることができます。
- [AWS Marketplace 管理ポータル](#) のレポートダッシュボードを使用する。このダッシュボードは、以前のレポート期間のレポートを提供します。

AWS Identity and Access Management (IAM) アクセス許可を使用して、レポートへのアクセスを制御できます。

利用可能な AWS Marketplace 販売者レポート

次のレポートが で利用できます AWS Marketplace。

- [日別ビジネスレポート](#)
- [日次顧客サブスクライバーレポート](#)
- [支払いレポート](#)
- [月別請求済み収益レポート](#)
- [販売補償レポート](#)

日別ビジネスレポート

日次ビジネスレポートには、以下のデータが記載されています。

- AWS 顧客が製品を毎日使用する方法
- 使用量に基づく予想収益。

該当する情報が使用可能な場合にのみ、このレポートを受け取ります。

受け取るべきレポートが届かない場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。レポートの E メール通知をキャンセルする場合は、そのチームに連絡することもできます。

このレポートは、[AWS Marketplace 管理ポータル](#) でアクセスできます。[the section called “AWS Marketplace Commerce Analytics サービス”](#) に登録している場合は、AWS SDK を使用してレポートにアクセスすることもできます。

各顧客に一意的識別子を使用して、長期にわたってレポート間で顧客を識別できます。この識別子を使用すると、顧客の使用量を追跡して、顧客の出費を見積もり、無料トライアルの使用量、および年間の使用傾向を把握できます。

掲載スケジュール

このレポートは毎日 00:00 UTC に発行され、前日の 00:00 UTC から 23:59 UTC までを対象としています。このスケジュールの例外は日別ビジネスレポートセクションの冒頭に示しています。

トピック

- [セクション 1: インスタンスタイプ別の使用量](#)

- [セクション 2: 料金](#)
- [セクション 3: 無料トライアルのコンバージョン数](#)
- [セクション 4: 新しいインスタンス](#)
- [セクション 5: 新規の製品サブスクライバー](#)
- [セクション 6: キャンセルした製品サブスクライバー](#)

セクション 1: インスタンスタイプ別の使用量

このセクションでは、顧客が使用するインスタンスタイプごとに 1 行ずつデータが一覧表示されます。たとえば、顧客が 1 つのインスタンスタイプで製品を使用し、別のインスタンスタイプで同じ製品を使用している場合、レポートには 2 つのインスタンスタイプそれぞれの行が含まれます。

列名	説明
カスタマーリファレンス ID	アカウント ID ではない一意の識別子。顧客別の使用量、収益、サブスクリプションの追跡に使用されます。 カスタマーリファレンス ID は、2026 年 3 月 31 日に廃止される予定です。代わりに顧客 AWS アカウント ID を使用します。
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
ユーザーの都道府県	製品をサブスクライブしているアカウントに関連付けられている請求先住所の都道府県。
ユーザーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
製品タイトル	製品のタイトル。
製品コード	製品の一意の識別子。
インスタンスタイプ	製品の使用量に関連付けられているインスタンスタイプ (t2.micro など)。

列名	説明
使用量の単位	レポート期間中に顧客が使用した使用量の単位数。
使用量の単位タイプ	顧客の使用量の単位。たとえば、時間や日です。
提供タイプの説明	製品提供の説明。たとえば、製品は時間課金、無料トライアル、または年間サブスクリプションで提供されています。
予想収益	製品の使用量からの予想収益。請求は月末に確定されます。
通貨	取引の通貨。例えば、取引が US ドルの場合、エントリは USD です。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
顧客の国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートにおける顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

セクション 2: 料金

このセクションには、製品に関連付けられている定額制の取引 (年間、月間、SaaS 契約の製品料金、データ製品サブスクリプション料金など) が含まれます。このセクションのデータは、レポートが生成される 72 時間前の 24 時間を対象としています。たとえば、レポートが 5 月 24 日に生成された場合、データは 5 月 21 日の 24 時間を対象としています。

列名	説明
カスタマーリファレンス ID	アカウント ID ではない一意の識別子。顧客別の使用量、収益、サブスクリプションの追跡に使用されます。 カスタマーリファレンス ID は、2026 年 3 月 31 日に廃止される予定です。代わりに顧客 AWS アカウント ID を使用します。
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
ユーザーの都道府県	製品をサブスクライブしているアカウントに関連付けられている請求先住所の都道府県。
ユーザーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
製品タイトル	製品のタイトル。
製品コード	製品の一意的識別子。
量	使用料。返金がある場合、この値はマイナスです。このエントリが AWS Marketplace SaaS 契約の場合、金額は契約全体ではなく、ディメンションの料金を表します。
通貨	取引の通貨。例えば、取引が US ドルの場合、エントリは USD です。

列名	説明
料金の説明	料金の理由 (月額料金、年間料金、返金など)。
顧客の国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
顧客の都道府県	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
顧客の市区町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
顧客の郵便番号	料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
顧客の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
開始日	AWS Marketplace SaaS 契約またはデータ製品サブスクリプションの開始日。
終了日	AWS Marketplace SaaS 契約またはデータ製品サブスクリプションの終了日。
数量	契約で指定されているディメンションの単位数。
ディメンション	契約で指定されているディメンション。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。

列名	説明
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートにおける顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

セクション 3: 無料トライアルのコンバージョン数

このセクションでは、過去 24 時間を対象に、無料トライアルの開始数、コンバージョン数、キャンセル数のデータが一覧表示されます。

列名	説明
製品タイトル	製品のタイトル。
製品コード	製品を表す一意の識別子。
新しい無料トライアル	レポート期間中に開始された新しい無料トライアルの数。
現在の無料トライアルの合計	アクティブな無料トライアルサブスクリプションの合計数。
コンバージョンされた無料トライアル	レポート期間中に無料トライアルから有料使用に移行されたサブスクリプションの合計数。
コンバージョンされなかった無料トライアル	無料トライアルの終了後に有料使用にコンバージョンされなかったサブスクリプションの合計数。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意的識別子。

セクション 4: 新しいインスタンス

このセクションでは、過去 24 時間を対象に、新しい EC2 インスタンスとインスタンスタイプのデータが一覧表示されます。

列名	説明
カスタマーリファレンス ID	アカウント ID ではない一意の識別子。顧客別の使用量、収益、サブスクリプションの追跡に使用されます。 カスタマーリファレンス ID は、2026 年 3 月 31 日に廃止される予定です。代わりに顧客 AWS アカウント ID を使用します。
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
ユーザーの都道府県	製品をサブスクライブしているアカウントに関連付けられている請求先住所の都道府県。
ユーザーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
製品タイトル	製品のタイトル。
製品コード	製品の一意的識別子。
タイプ	Amazon EC2 インスタンスタイプ
カウント	EC2 インスタンスの数。
顧客の国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
顧客の都道府県	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。

列名	説明
顧客の市区町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
顧客の郵便番号	料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
顧客の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートにおける顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

セクション 5: 新規の製品サブスクライバー

このセクションでは、過去 24 時間を対象に、新しい購入者のデータが一覧表示されます。

列名	説明
カスタマーリファレンス ID	アカウント ID ではない一意の識別子。顧客別の使用量、収益、サブスクリプションの追跡に使用されます。 カスタマーリファレンス ID は、2026 年 3 月 31 日に廃止される予定です。代わりに顧客 AWS アカウント ID を使用します。

列名	説明
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
ユーザーの都道府県	製品をサブスクライブしているアカウントに関連付けられている請求先住所の都道府県。
ユーザーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
製品タイトル	製品のタイトル。
製品コード	製品の一意的識別子。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
顧客の国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
顧客の都道府県	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
顧客の市区町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
顧客の郵便番号	料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
顧客の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
ソリューションタイトル	ソリューションの名前。

列名	説明
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートにおける顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

セクション 6: キャンセルした製品サブスクライバー

このセクションでは、過去 24 時間を対象に、購入者のキャンセルに関するデータが一覧表示されます。

列名	説明
カスタマーリファレンス ID	アカウント ID ではない一意の識別子。顧客別の使用量、収益、サブスクリプションの追跡に使用されます。 カスタマーリファレンス ID は、2026 年 3 月 31 日に廃止される予定です。代わりに顧客 AWS アカウント ID を使用します。
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
ユーザーの都道府県	製品をサブスクライブしているアカウントに関連付けられている請求先住所の都道府県。
ユーザーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
製品タイトル	製品のタイトル。

列名	説明
製品コード	製品の一意の識別子。
サブスクライブ日	サブスクリプションが開始された日付。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
顧客の国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
顧客の都道府県	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
顧客の市区町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
顧客の郵便番号	料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
顧客の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートにおける顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

日次顧客サブスクライバーレポート

このレポートには、お客様の製品を購入した購入者のデータが一覧表示されます。このレポートでは、現在または過去の使用量は示されません。顧客が製品をサブスクライブしていることだけわかります。該当する情報が使用可能な場合にのみ、このレポートを受け取ります。受け取るべきレポートが届かない場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。また、AWS Marketplace Seller Operations チームに連絡して、このレポートの E メール通知をキャンセルすることもできます。

このレポートは、[AWS Marketplace 管理ポータル](#) でアクセスできます。[the section called “AWS Marketplace Commerce Analytics サービス”](#) に登録している場合は、AWS SDK を使用してレポートにアクセスすることもできます。

このレポートには 2 つのセクションがあります。1 つは時間課金と月間サブスクリプション用、もう 1 つは年間サブスクリプション用です。レポートには、製品をサブスクライブしているすべての顧客の AWS アカウント IDs のリストが含まれます。

掲載スケジュール

このレポートは毎日 00:00 UTC に発行され、前日の 00:00 UTC から 23:59 UTC までを対象としています。

トピック

- [セクション 1: 時間単位と月単位のサブスクリプション](#)
- [セクション 2: 可変長のサブスクリプション](#)

セクション 1: 時間単位と月単位のサブスクリプション

このセクションでは、前日の 23:59:59 UTC の時点でのすべての従量制サブスクリプションのデータが一覧表示されます。

列名	説明
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
製品タイトル	製品のタイトル。
製品 ID	ソフトウェア製品の一意の識別子。

列名	説明
製品コード	ソフトウェア製品の一意的識別子。
サブスクリプション開始日	サブスクリプションの開始日 (YYYY-MM-DD 形式)。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意的識別子。
支払者リファレンス ID	アカウント ID ではない一意的識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートにおける顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
リセラーアカウント ID	チャネルパートナーリセラー用の一意的識別子。
リセラーアカウント名	チャネルパートナーリセラーの名前。

セクション 2: 可変長のサブスクリプション

このセクションには、前日の 23:59:59 UTC の時点でのすべての定額制サブスクリプションのデータが一覧表示されます。

列名	説明
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
製品タイトル	製品のタイトル。
製品 ID	ソフトウェア製品の一意的識別子。

列名	説明
製品コード	ソフトウェア製品の一意の識別子。この情報は Amazon EC2 インスタンスのメタデータの一部としても使用できます。
サブスクリプション ID	サブスクリプションの ID。
サブスクリプション数	顧客が購入したライセンスの合計数。
サブスクリプションタイプ	サブスクリプションのタイプ。
サブスクリプションの目的	このオファーが以前のオファーのアップグレードや更新であるかどうか。
オファー ID	購入者が署名したオファーの識別子。
サブスクリプション開始日	顧客が製品をサブスクライブした日付 (YYYY-MM-DD 形式)。
以前のオファー ID	アップグレードや更新のオファーに先立つ以前のオファーの ID (存在する場合)。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートにおける顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
リセラーアカウント ID	チャネルパートナーリセラー用の一意の識別子。
リセラーアカウント名	チャネルパートナーリセラーの名前。

支払いレポート

Important

このレポートは、2024 年 8 月 30 日に使用できなくなります。以下の代替手段を使用できます。

- [回収と支払いダッシュボード](#)
- GitHub の [collections_and_disbursements_report](#) SQL クエリ。

支払いレポートでは、前回の支払い以降に、販売者の銀行口座で回収された資金および支払われた資金に関する情報が提供されます。支払いには、販売者の製品のサブスクリプションに対する顧客の支払いまたは返金、顧客に対して徴収または返金される税金が含まれます。資金が顧客から回収されるまで、資金の支払いを受け取りません。顧客ごとに支払い条件が異なるため AWS、未回収の各年齢カテゴリの資金の一部は顧客から支払われていない可能性があります。

返金は販売者による承認後に行われるため、マイナスの金額として表示されます。

このレポートは、レポートタブの AWS Marketplace 管理ポータルで使用できます。このような独自のレポートを作成するには、をベースレポート [製品別の支払い](#) として使用し AWS Marketplace [の販売者配信データフィード AWS Marketplace](#)、ニーズに合わせてカスタマイズできます。

Note

このレポートの通知 E メールをキャンセルするには、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

掲載スケジュール

このレポートは、銀行振込の支払いを開始してから 3~5 日後に発行されます。一般的に、これは支払いを受け取る販売者の月次レポートです。支払いが開始されていない場合、支払いレポートは生成されません。

トピック

- [セクション 1: 製品別の支払い額](#)
- [セクション 2: 顧客地域別の支払い額](#)

- [セクション 3: インスタンス時間単位の支払い額](#)
- [セクション 4: 未回収資金の期間](#)
- [セクション 5: 回収済み資金の期間](#)
- [セクション 6: 支払期限を過ぎた資金の期間](#)
- [セクション 7: 未回収資金の内訳](#)

セクション 1: 製品別の支払い額

このセクションでは、製品別の支払いデータが一覧表示されます。

列名	説明
製品	製品のタイトル。
製品コード	製品の一意的識別子。
SellerRev	顧客に請求される製品の従量課金額または料金。
AWSRefFee	AWS Marketplace 料金の金額。
SellerRevRefund	データカバレッジ期間中に返金処理された場合に顧客に払い戻されるサブスクリプション費用の金額。
AWSRefFeeRefund	データカバレッジ期間中に返金処理された場合に返金される AWS Marketplace 料金の金額。
SellerRevCredit	お客様のアカウントに AWS Marketplace 置いた AWS クレジット。
AWSRefFeeCredit	アカウントに AWS Marketplace 付与した AWS クレジット。
正味金額	販売者に支払われた資金の合計。この列の金額は、SellerRev 列の金額から AWSRefFee 列の金額を引いたものです。顧客への返金時、こ

列名	説明
	の列の金額は、SellerRevRefund 列の金額から AWSRefFeeRefund 列の金額を引いたもの (負の数) です。
取引リファレンス ID	AWS Marketplace レポート間でトランザクションを関連付けるのに役立つトランザクションの一意の識別子。
SellerUSSalesTax	この取引で請求された米国の消費税と使用税の合計額。
SellerUSSalesTaxRefund	返金が処理される場合にこの取引に対して払い戻される米国の消費税と使用税の合計金額。
顧客 AWS アカウント番号	料金の請求先となるアカウントの ID。
顧客の国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
顧客の都道府県	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
顧客の市区町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
顧客の郵便番号	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
顧客の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。

列名	説明
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートの顧客による使用状況、収益、サブスクリプションの追跡に役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

セクション 2: 顧客地域別の支払い額

このセクションには、顧客の地域別の支払いデータが一覧表示されます。

列名	説明
決済 ID	支払いの一意の識別子。
決済期間開始日	支払い期間の開始日時。
決済期間終了日	支払い期間の終了日時。
入金日	支払いの発生日時。
支払い額	支払い合計金額
国コード	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
都道府県または地域	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
市町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
郵便番号	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。

列名	説明
納税地別正味金額	納税地別に販売者に支払われる資金の合計 (AWS Marketplace 料金、返金、米国の消費税と使用税を引いたもの)。
納税地別総額	納税地別に販売者に支払われる資金の合計。
販売者米国消費税	販売者の代理でこの取引で請求された米国の消費税と使用税の合計額。(つまり、米国の売上高および税レポートの関連レコードには「納税義務者」 == 「SELLER」と表示されます)。
販売者米国消費税返金	返金が処理された場合に、このトランザクションに対して返金される米国の売上税および使用税の合計額 (出品者に代わってそのような税金が徴収されたとき)。(つまり、米国の売上高および税レポートの関連レコードには「納税義務者」 == 「SELLER」と表示されます)。

セクション 3: インスタンス時間単位の支払い額

このセクションでは、Amazon EC2 インスタンス時間別に支払いデータが一覧表示されます。

列名	説明
製品	製品のタイトル。
製品コード	製品の一意的識別子。
使用タイプ説明	使用量の説明 (提供タイプ、リージョン、インスタンスタイプなど)。
Rate	この提供タイプ、リージョン、インスタンスタイプの 1 時間あたりのレート。
ユーザーカウント	この提供タイプ、リージョン、インスタンスタイプを使用する一意の顧客の数。

列名	説明
インスタンス時間	この提供タイプ、リージョン、インスタンスタイプの使用時間。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。

セクション 4: 未回収資金の期間

このセクションでは、未回収資金のデータが期間別に一覧表示されます。未回収資金には、期限を過ぎていない金額が含まれる場合があります。

列名	説明
未回収 (31 日未満保留中)	請求されたが 31 日未満に回収されなかった資金の合計。
未回収 (31 ~ 60 日間保留中)	請求されたが 31 ~ 60 日以内に回収されなかった資金の合計。
未回収 (61 ~ 90 日間保留中)	請求されたが 61 ~ 90 日以内に回収されなかった資金の合計。
未回収 (91 ~ 120 日間保留中)	請求されたが 91 ~ 120 日以内に回収されなかった資金の合計。
未回収 (120 日超保留中)	請求されたが 120 日を超えて回収されなかった資金の合計。
未回収 (全体)	請求されたが回収されなかったすべての資金の合計。

セクション 5: 回収済み資金の期間

このセクションでは、支払い以降に回収された資金のデータが一覧表示されます。

列名	説明
回収 (31 日未満保留中)	0～31 日の範囲内に請求されて回収された資金の合計。
回収 (31～60 日間保留中)	31～60 日の範囲に請求されて回収された資金の合計。
回収 (61～90 日間保留中)	61～90 日の範囲で請求されて回収された資金の合計。
回収 (91～120 日間保留中)	91～120 日の範囲で請求されて回収された資金の合計。
回収 (120 日超保留中)	120 日を超えて請求されて回収された資金の合計。
回収 (全体)	回収の資金の合計。

セクション 6: 支払期限を過ぎた資金の期間

このセクションでは、顧客によって支払い可能であるが、AWSとの契約に従って未払いになっている資金のデータが一覧表示されます。

列名	説明
支払期限経過 (31 日未満)	過去 0～31 日間に発生していて支払期限が到来しているが顧客が支払っていない資金の合計。
支払期限経過 (31～60 日)	過去 31～60 日間に発生していて支払期限が到来しているが顧客が支払っていない資金の合計。
支払期限経過 (61～90 日)	過去 61～90 日間に発生していて支払期限が到来しているが顧客が支払っていない資金の合計。

列名	説明
支払期限経過 (91 ~ 120 日)	過去 91 ~ 120 日間に発生していて支払期限が到来しているが顧客が支払っていない資金の合計。
支払期限経過 (120 日超)	過去 121 日以上に発生していて支払期限が到来しているが顧客が支払っていない資金の合計。
支払期限経過 (全体)	発生していて支払期限が到来しているが顧客が支払っていない資金の合計。

セクション 7: 未回収資金の内訳

このセクションには、すべての未回収資金が、支払期日順にソートされて一覧表示されます。

列名	説明
支払者 AWS アカウント番号	ソフトウェア料金の請求先となるアカウント。
製品コード	製品の一意の識別子。
総収益	製品の使用に対して請求された金額、または製品の使用に対する料金。
AWS 収益共有	決済時に請求額から差し引かれる AWS 料金額。
総返金	取引の返金の合計金額。
AWS 返金共有	トランザクションに対して返金される AWS 料金の一部。
正味収益	この取引に対して請求される正味金額から、AWS 手数料、返金、米国の消費税と使用税を差し引いた額。

列名	説明
通貨	取引の通貨。例えば、取引が US ドルの場合、エントリは USD です。
AR 期間	取引の月と年 (YYYY-MM 形式)。
取引リファレンス ID	トランザクションを表す一意の識別子。これを使用して、AWS Marketplace レポート間でトランザクションを関連付けることができます。
オポチュニティの名前	登録された機会の一意の識別子。
オポチュニティの説明	登録されたオポチュニティの任意のメタデータ。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートで顧客による使用状況、収益、サブスクリプションを追跡するのに役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。
支払期日	YYYY-MM-DD 形式の支払期日。

月別請求済み収益レポート

Important

このレポートの 3 つのセクションは、2024 年 8 月 30 日に使用できなくなります。セクション 1~3 では、これらの代替手段を使用できます。

- [請求済み収益ダッシュボード](#)
- GitHub の [billed_revenue_report](#) SQL クエリ。

セクション 4: 柔軟な支払いスケジュールの契約は、2024 年 8 月 30 日以降も引き続き利用できます。

月別請求収益レポートは、会計およびその他の財務報告の目的で、毎月請求収益に関する信頼できる情報を提供します。このレポートには、が製品の時間単位、年単位、または月単位の使用に対して顧客に AWS 請求する合計金額が表示されます。レポートには 4 つのセクションとして、時間課金額と月額料金、任意期間のサブスクリプション数、フィールドデモ使用量、Flexible Payment 額があります。

Important

このレポートの金額には、顧客から回収した金額ではなく、顧客に請求した収益のみが反映されています。

このレポートは、レポートタブの AWS Marketplace 管理ポータルで使用できます。コマース分析サービスに登録している場合は、API AWS Marketplace コールを使用してこのレポートのセクションをプルダウンできます。詳細については、「[the section called “AWS Marketplace Commerce Analytics サービス”](#)」を参照してください。

Note

このレポートの通知 E メールをキャンセルするには、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

掲載スケジュール

このレポートは、毎月 15 日の 00:00 UTC に発行されます。

Billing and Revenue Data は、翌月の 15 日から 14 日までを対象としています。

長期サブスクリプションは、月の最初の日の 00:00 UTC から、前月の最終日の 23:59 UTC までを対象とします。

たとえば、5 月 15 日に発行されたレポートは、4 月 1 日の 00:00 UTC から 4 月 30 日の 23:59 UTC までを対象としています。

トピック

- [セクション 1: 請求および収益データ](#)
- [セクション 2: 可変長のサブスクリプション](#)
- [セクション 3: AWS フィールドデモンストレーションの使用](#)
- [セクション 4: Flexible Payment Schedule による契約](#)

セクション 1: 請求および収益データ

このセクションでは、従量課金額、返金、料金、徴収される米国の消費税と使用税のデータが一覧表示されます。

列名	説明
カスタマーリファレンス ID	アカウント ID ではない一意の識別子。顧客別の使用量、収益、サブスクリプションの追跡に使用されます。 カスタマーリファレンス ID は、2025 年 12 月 31 日に廃止される予定です。代わりに顧客 AWS アカウント ID を使用します。
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
状態	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
市町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
郵便番号	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
製品タイトル	製品のタイトル。

列名	説明
製品コード	製品の一意の識別子。
お客様の請求額	顧客に請求される製品の従量課金額または月額料金。
AWS 出品料金	請求額から差し引かれる AWS Marketplace 料金額。
返金額	データカバレッジ期間中に返金処理された場合に顧客に払い戻されるサブスクリプション費用の合計金額。
AWS 料金の返金	データカバレッジ期間中に返金処理された場合に返金された AWS Marketplace 料金の一部。
コスト	リセラーに対する製品の費用 (リセラーが製品販売時に支払う金額など)。
パートナー収益額	取引に対して請求される合計金額。AWS Marketplace 料金、返金、米国の消費税と使用税を差し引いた額。
通貨	取引の通貨。たとえば、取引が US ドルの場合、エントリは USD です。
取引リファレンス ID	AWS Marketplace レポート間でトランザクションに関連付けるのに役立つトランザクションの一意の識別子。
米国消費税お客様の請求額	販売者の代理でこの取引で請求された米国の消費税と使用税の合計額。(つまり、米国の売上高および税レポートの関連レコードには「納税義務者」 == 「SELLER」と表示されます)。

列名	説明
米国消費税返金額	返金が処理された場合に、このトランザクションに対して返金される米国の売上税および使用税の合計額 (出品者に代わってそのような税金が徴収されたとき)。(つまり、米国の売上高および税レポートの関連レコードには「納税義務者」 == 「SELLER」と表示されます)。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
顧客の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが <i>liu-jie@example.com</i> の場合、エントリは example.com です。
オポチュニティの名前	登録された機会の一意の識別子。
オポチュニティの説明	登録されたオポチュニティのメタデータ。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートで顧客による使用状況、収益、サブスクリプションを追跡するのに役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

セクション 2: 可変長のサブスクリプション

このセクションでは、定額制料金のデータが一覧表示されます。

列名	説明
カスタマーリファレンス ID	アカウント ID ではない一意の識別子。顧客別の使用量、収益、サブスクリプションの追跡に使用されます。 カスタマーリファレンス ID は、2025 年 12 月 31 日に廃止される予定です。代わりに顧客 AWS アカウント ID を使用します。
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
状態	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
市町村	ソフトウェア料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
郵便番号	料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
製品タイトル	製品のタイトル。
製品コード	製品の一意的識別子。
サブスクリプション数	任意期間のサブスクリプション購入の一部として指定されたライセンスの合計数。
サブスクリプション開始日	任意期間のサブスクリプション購入の開始日。
サブスクリプション終了日	任意期間のサブスクリプション購入の終了日。
サブスクリプションインスタンスタイプ	任意期間のサブスクリプション購入に関連付けられているインスタンスタイプ。
お客様の請求額	従量課金額、月額料金、またはその両方。

列名	説明
AWS 出品料金	請求額から差し引かれる AWS Marketplace 料金額。
返金額	データカバレッジ期間中に返金が処理された場合に顧客に払い戻される合計金額。
AWS 料金の返金	データカバレッジ期間中に返金が処理された場合に返金された AWS Marketplace 料金の一部。
コスト	リセラーに対する製品の費用 (リセラーが製品販売時に支払う金額など)。
パートナー収益額	この取引に対して請求される合計金額。AWS Marketplace 料金、返金、および米国の消費税と使用税を差し引いた額。
通貨	取引の通貨。例えば、取引が US ドルの場合、エントリは USD です。
取引リファレンス ID	AWS Marketplace レポート間でトランザクションを関連付けるのに役立つトランザクションの一意の識別子。
米国消費税お客様の請求額	販売者の代理でこの取引で請求された米国の消費税と使用税の合計額。(つまり、米国の売上高および税レポートの関連レコードには「納税義務者」 == 「SELLER」と表示されます)。
米国消費税返金額	返金が処理された場合に、このトランザクションに対して返金される米国の売上税および使用税の合計額 (出品者に代わってそのような税金が徴収されたとき)。(つまり、米国の売上高および税レポートの関連レコードには「納税義務者」 == 「SELLER」と表示されます)。

列名	説明
顧客の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
契約開始日	AWS Marketplace SaaS 契約の開始日。
契約終了日	AWS Marketplace SaaS 契約の終了日。
オポチュニティの名前	登録された機会の一意の識別子。
オポチュニティの説明	登録されたオポチュニティのメタデータ。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートで顧客による使用状況、収益、サブスクリプションを追跡するのに役立ちます。
支払者住所 ID	お客様の住所を表す一意の識別子。

セクション 3: AWS フィールドデモンストレーションの使用

セクションには、製品の AWS [フィールドデモンストレーションの使用](#)に関するデータが一覧表示されます。潜在的な顧客に対して製品デモを実施できるように製品を設定できます。デモからの使用量はここに一覧表示されます。

列名	説明
製品タイトル	製品のタイトル。
製品コード	製品の一意の識別子。
インスタンスタイプ	フィールドデモに関連付けられている Amazon EC2 インスタンスタイプ。
使用量の単位	製品に関連付けられている使用量の単位数。
使用量の単位タイプ	使用量に関連付けられている単位 (時間など)。

セクション 4: Flexible Payment Schedule による契約

このセクションでは、前のレポート期間に Flexible Payment スケジュールを使用して作成したすべての契約のデータが一覧表示されます。

列名	説明
顧客 AWS アカウント番号	料金が請求される AWS アカウント ID。
顧客の国	料金が請求先となる支払いアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
顧客の都道府県	料金の請求先となる支払いアカウントに関連付けられている請求先住所の都道府県。
顧客の市区町村	料金の請求先となる支払いアカウントに関連付けられている請求先住所の市区町村。
顧客の郵便番号	料金の請求先となる支払いアカウントに関連付けられている請求先住所の郵便番号。
顧客の E メールドメイン	料金の請求先となる支払いアカウントに関連付けられている E メールドメイン。たとえば、E

列名	説明
	メールアドレスが <i>liu-jie@example.com</i> の場合、エントリは example.com です。
ユーザー参照 ID	料金の請求先となる支払いアカウントの口座。
AWS ユーザーアカウント番号	製品をサブスクライブしているアカウントの ID。
製品 ID	製品の一意的識別子。
製品タイトル	製品のタイトル。
製品タイプ	製品のタイプ。
AWS Marketplace オファー ID	購入者が署名したオファーの識別子。
契約の作成日付	契約作成日。アカウントがオファーをサブスクライブする日付です。
契約の有効期限	契約が期限切れになる日付。
総契約価値 (USD)	契約の総額 (USD)。
支払い回数	契約でスケジュールされている支払いの回数。
請求書日付	請求書が作成された日付。
請求書金額 (USD)	請求書で請求される金額 (USD)。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。これは、すべての AWS Marketplace 財務レポートで顧客による使用状況、収益、サブスクリプションを追跡するのに役立ちます。

販売補償レポート

Important

このレポートは、2024 年 8 月 30 日に使用できなくなります。以下の代替手段を使用できます。

- [請求済み収益ダッシュボード](#)
- GitHub の [billed_revenue_report](#) SQL クエリ。

レポートには、月別の請求済み収益と、標準の [the section called “月別請求済み収益レポート”](#) にはない追加の顧客情報が一覧表示されます。また、自分の製品の時間別、年別、月別の使用量に対して AWS が顧客に請求する合計金額が一覧表示されます。

Important

このレポートの金額には、顧客から回収された金額ではなく、顧客に請求される収益のみが反映されています。

販売補償レポートとこのプログラムの一部としてお客様と共有されている情報は、お客様との非公開契約の下で Amazon の機密情報を構成するものであり、そのような合意がない場合は AWS Marketplace 販売者の利用規約を構成します。この情報は、AWS Marketplace 収益を会社名、地域、および AWS アカウント ID ごとに担当者にマッピングすることで、セールス担当者に補償することのみに使用できます。この情報は、支払うべき手数料の源泉を理解するために知る必要がある従業員と共有できます。そのような情報の使用と共有は、機密保持契約上の義務と、AWS Marketplace 販売者の利用規約のセクション 3.8 を含むがこれに限定されない AWS Marketplace 販売者の利用規約に準拠する必要があります。

これに似た独自のレポートを作成するには、AWS Marketplace [の販売者配信データフィード](#) [AWS Marketplace](#) を利用することができます。ベースレポートとして [販売補償レポート](#) などを活用すると、ニーズに合わせてカスタマイズできます。

掲載スケジュール

このレポートは、毎月 15 日の 00:00 UTC に発行されます。また、前の暦月の初日 00:00 UTC から末日 23:59 UTC までを対象とします。たとえば、5 月 15 日に発行されたレポートは、4 月 1 日の 00:00 UTC から 4 月 30 日の 23:59 UTC までを対象としています。

請求済み収益

このレポートの請求済み収益セクションには、前の暦月からの従量制および定額制料金が含まれます。以下に示しているのは列名と説明です。

Note

このレポートでは、出品料金は、階層型出品料金に従って決定される、取引収益の割合です (認定済みリセラーによる認定済み転売製品の転売の除く)。

列名	説明
顧客の AWS アカウント番号	料金の請求先となるアカウント。
国	料金の請求先となるアカウントに関連付けられている 2 文字の国コード。このレポートでは、ISO 3166-1 alpha-2 標準を使用します。
都道府県	料金の請求先となるアカウントに関連付けられている請求先住所の都道府県。
市町村	料金の請求先となるアカウントに関連付けられている請求先住所の市区町村。
郵便番号	料金の請求先となるアカウントに関連付けられている請求先住所の郵便番号。
E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。たとえば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
製品コード	製品の一意の識別子。
製品タイトル	製品のタイトル。
総収益	製品の使用に対して請求された金額、または製品の使用に対する月額料金。

列名	説明
AWS 収益分配	決済時に請求額から控除される AWS 料金。 the section called “支払いレポート” に表示されます。
総返金	取引の返金の合計金額。
AWS 返金分配	取引で返金される AWS 料金の一部。
正味収益	この取引で請求された正味額から、AWS 料金、返金、および米国の消費税と使用税を引いた額。
通貨	取引の通貨。例えば、取引が US ドルの場合、エントリは USD です。
AR 期間	取引の月と年 (YYYY-MM 形式)。
取引リファレンス ID	AWS Marketplace レポート全体で取引を関連付けるために使用できる、取引を表す一意の識別子。
オポチュニティの名前	登録された機会の一意の識別子。
オポチュニティの説明	登録されたオポチュニティの任意のメタデータ。
ソリューションタイトル	ソリューションの名前。
ソリューション ID	ソリューションの一意の識別子。
支払者リファレンス ID	アカウント ID ではない一意の識別子。料金が請求されるアカウントと関連付けられています。すべての AWS Marketplace 財務レポート間で顧客別の使用量、収益、サブスクリプションの追跡に使用されます。
支払者住所 ID	お客様の住所を表す一意の識別子。

補足レポート

AWS Marketplace は [の販売者配信データフィード AWS Marketplace](#)、を通じて、販売者の AWS Marketplace 出品に関連付けられた AWS Marketplace 販売者アカウント ID に接続されている販売者所有の Amazon S3 アカウントに補足レポートを提供します。詳細については、[「Amazon Simple Storage Service バケットの作成」](#)を参照してください。

補足レポートは、前日に新しいサブスクライバーがいた場合、毎日 16:00 UTC に発行されます。これらのレポートは、前日の 13:59 UTC から翌日の 16:01 UTC までの期間を対象としています。

契約詳細レポート

契約詳細レポートは、Software as a Service (SaaS) の無償トライアルを利用している顧客をサポートするのに役立ちます。レポートには、サブスクライバー名、サブスクライバー ID、オファー ID、契約開始、契約終了日などの契約の詳細が含まれます。

該当する情報が使用可能な場合にのみ、このレポートを受け取ります。受け取るべきと思われるレポートが届かない場合は、[AWS Marketplace Seller Operations](#) チームまでお問い合わせください。

このレポートには、AWS Marketplace Seller Account ID に関連付けられた Amazon S3 バケットからアクセスできます。

次の表は、契約詳細レポートの列名と説明をまとめたものです。

SaaS 契約無料トライアルレポートデータ

名前	説明
vendor_display_name	製品を販売したベンダーの名前。
vendor_aws_account_id	製品を販売したベンダーの ID。
subscriber_aws_account_id	製品にサブスクライブ AWS アカウント されている に関連付けられた ID。
customer_id	ソフトウェア製品の一意的識別子。
product_title	製品のタイトル。
offer_id	購入者が署名したオファーの識別子。

名前	説明
offer_visibility	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるかの表示。
reseller_name	チャネルパートナーリセラーの名前。
reseller_aws_account_id	チャネルパートナーリセラー用の一意の識別子。
agreement_id	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意的な契約データフィードリファレンス。
agreement_acceptance_date	契約が受諾された日付。
agreement_start_date	契約の開始日。
agreement_end_date	契約の終了日。従量制/従量課金制/サブスクリプションの場合は、9999 年 1 月 1 日に設定されます。
is_free_trial_offer	オファーまたは契約が無料トライアルオファーかどうかを示すフラグ。
is_upgraded_after_free_trial	契約が有料契約にアップグレードされたかどうかを示すフラグ。
total_contract_value	契約の総額。

販売者ダッシュボード

AWS Marketplace は、[Amazon QuickSight](#) を利用したダッシュボードに、財務、営業、マーケティングのデータへのアクセスと分析に役立つグラフ、グラフ、インサイトを提供します。販売者ダッシュボードには以下が含まれます。

the section called “財務業務用ダッシュボード”

- [the section called “請求済み収益ダッシュボード”](#) - 会計やその他の財務報告の目的で請求された収益に関する情報を提供します。
- [the section called “回収と支払いダッシュボード”](#) - 前回の支払い以降に が AWS 回収し、銀行口座に支払った資金に関する情報を提供します。
- [the section called “税ダッシュボード”](#) - 販売者トランザクションの税金に関する情報を提供します。

販売業務用ダッシュボード

- [the section called “契約および更新ダッシュボード”](#) - AWS Marketplaceで契約に署名してから 24 時間以内に、契約と更新に関する情報が表示されます。
- [the section called “使用状況ダッシュボード”](#) - SaaS およびサーバー使用量ベースの製品を使用しているお客様に、視覚化ときめ細かなデータを提供します。
- [マーケティング用ダッシュボード](#) - マーケティングデータの追跡に役立つ複数のダッシュボードを提供します。

マーケティング用ダッシュボード

- [顧客契約ダッシュボード](#) - 契約と製品をサブスクライブする顧客に関するデータを提供します。
- [出品パフォーマンスダッシュボード](#) - AWS Marketplace リスト内のトラフィックとユーザー動作に関するデータを提供します。
- [検索パフォーマンスダッシュボード](#) - AWS Marketplace リストに適用されたキーワードに関するデータを提供します。

ダッシュボードは、適切なアクセス許可を持つ AWS Marketplace 販売者が利用できます。

ダッシュボードへのアクセス

デフォルトでは、販売者アカウントの AWS Marketplace システム管理者は、 のインサイトタブのすべてのダッシュボードにアクセスできます AWS Marketplace 管理ポータル。システム管理者は、AWS Identity and Access Management (IAM) ポリシーを作成して、販売会社の他のユーザーに特定のダッシュボードへのアクセスを提供できます。

Note

2023 年 9 月には、従来の IAM 権限で有効になっていた出品者ダッシュボードへのアクセスはサポートされなくなります。以下のコード例の新しい Amazon リソースネーム (ARN) 形式を使用して IAM 許可を更新してください。

ポリシーの作成の詳細については、「[IAM ポリシーの作成](#)」を参照してください。

ダッシュボードポリシー

ダッシュボードの一部またはすべてへのアクセスを提供するには、次のいずれかのポリシーを使用します。

次の例では、ダッシュボードやレポートなど、現在および将来の AWS Marketplace リソースへのアクセスを提供します。現在および将来のすべてのデータフィードを使用します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

ARNs を含めることで、1 つ以上のダッシュボードへのアクセスを許可することもできます。次の例は、すべてのダッシュボードへのアクセスを提供する方法を示しています。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
  ]
}
```

```
"Resource": [  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/  
BillingEvent_V1/Dashboard/BilledRevenue_V1",  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/  
BillingEvent_V1/Dashboard/CollectionsAndDisbursements_V1",  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/Agreement_V1/  
Dashboard/AgreementsAndRenewals_V1",  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/Usage_V1/  
Dashboard/Usage_V1",  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/TaxItem_V1/  
Dashboard/Tax_V1",  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/Marketing_V1/  
Dashboard/CustomAgreements_V1",  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/Marketing_V1/  
Dashboard/ListingPerformance_V1",  
  "arn:aws:aws-marketplace:::AWSMarketplace/ReportingData/Marketing_V1/  
Dashboard/SearchPerformance_V1"  
]  
}]  
}
```

ダッシュボードへのアクセスを削除するには、ポリシーの Resource セクションから削除します。

Note

AWS Identity and Access Management (IAM) ポリシーの作成の詳細については、「AWS Identity and Access Management ユーザーガイド」の [「IAM ポリシーの作成」](#) を参照してください。

財務業務用ダッシュボード

AWS Marketplace には、財務データの追跡に役立つ複数のダッシュボードが用意されています。

トピック

- [請求済み収益ダッシュボード](#)
- [回収と支払いダッシュボード](#)
- [税ダッシュボード](#)

請求済み収益ダッシュボード

請求済み収益ダッシュボードには、AWS Marketplaceのすべての請求済み売上に関する情報が表示されます。請求済み収益ダッシュボードを用いてレポートを作成すると時間の節約となり、[従来の月次請求収益レポート](#)よりも約 45 日早くレポートを入手できます。従来の月次請求収益レポートは毎月 15 日に配信されるため、前月の請求額の確認が遅れます。

このダッシュボードを開くには、にサインインし[AWS Marketplace 管理ポータル](#)、インサイト、財務オペレーションを選択し、請求収益タブを選択します。

トピック

- [請求済み収益ダッシュボードの更新頻度](#)
- [セクション 1: コントロール](#)
- [セクション 2: フィルター](#)
- [セクション 3: 主要なメトリクス](#)
- [セクション 4: トレンド](#)
- [セクション 5: 内訳](#)
- [セクション 6: 詳細なデータ](#)

請求済み収益ダッシュボードの更新頻度

ダッシュボードは、UTC の午前 0 時に毎日更新されます。請求書が、UTC の午前 0 時前の 5 月 10 日に作成された場合、5 月 11 日の更新が請求書に表示されます。請求書が、UTC の午前 0 時後の 5 月 10 日に作成された場合、5 月 12 日の更新が請求書に表示されます。アップストリームシステムから受信した最新の請求データまたは支払いデータが遅延した場合、最新のデータがダッシュボードに反映されるまでに 1~2 日かかる場合があります。

予想される顧客請求書が表示されない場合は、AWS Marketplace ビジネス開発担当者に連絡する前に、以下の手順に従います。

請求済み収益ダッシュボードの顧客請求書が見つからない場合のトラブルシューティング方法

1. まず次の 1 つ以上のツールを使用して、顧客がオファーを受け入れたことを確認します。
 - [日次顧客サブスクライバーレポート](#)
 - [AWS Marketplace イベントの通知](#)
 - [AWS Marketplace 管理ポータル 契約タブ](#)

2. (プライベートオファーの場合) 顧客がオファーを受け入れたことを確認したら、オファーの請求書のスケジュールと金額を確認します。
 - プライベートオファーについては、AWS Marketplace 管理ポータル オフータブを確認してください。
 - チャネルパートナーのプライベートオファーについては、AWS Marketplace 管理ポータル パートナータブをチェックして、パートナーオポチュニティ内のカスタム支払いスケジュールを表示します。
3. 取引において、前月の使用分を毎月 2 日と 3 日に顧客に請求する [使用量料金モデル](#) が指定されているかどうかを確認してください。

セクション 1: コントロール

請求済み収益ダッシュボードの本セクション内には、同ダッシュボードのデータを絞り込むためのフィルターがあります。たとえば、[AWS Marketplace イベントの通知](#) からフィールドのフィルターを選択して、特定の顧客アカウント ID、サブスクライバーの会社名、またはオファー ID の請求を確認できます。また、ビジュアルに含める日付範囲などを設定できるフィルターを分析に追加することも可能です。セクション内で選択済みのフィルターにより、メトリクス、トレンド、内訳、詳細データセクションに表示されるデータが更新されます。

フィルタリングの詳細については、[QuickSight ユーザーガイドの「QuickSight でのデータのフィルタリング」](#)を参照してください。QuickSight

コントロールの記述

コントロール名	説明
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの会社名	製品をサブスクライブしているアカウント名。
製品のタイトル	製品のタイトル。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。

コントロール名	説明
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意的な契約データフィードリファレンス。
AWS 登録販売者	取引を促進した事業体の識別子。可能な値は以下のとおりです: • AWS_INC: AWS Inc. (米国) の識別子。 • AWS_EUROPE: EMEA AWS SARL の識別子 (ルクセンブルクにベース)。 • AWS_AUSTRALIA: AWS Australia Pty Ltd の識別子 • AWS_JAPAN: Japan AWS G.K. の識別子。 • AWS_KOREA: AWS 韓国の識別子。
支払者 AWS アカウント ID	料金の請求先となるアカウントの ID。
支払人の会社名	料金の請求先となるアカウントの商号。
リセラー会社	ソフトウェア製造元の製品の販売を許可されているリセラーアカウントの商号。
リセラー AWS アカウント ID	ISV から卸売で製品またはサービスを購入して顧客に再販したアカウントの ID。
再販承認 ID	登録された機会の一意的な識別子。
再販承認名	登録された機会の一意的な名前。
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。
サブスクライバーの州または地域	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある州または地域

コントロール名	説明
取引リファレンス ID	AWS Marketplace レガシーレポート間でトランザクションを関連付けるのに役立つトランザクションの一意的識別子。

セクション 2: フィルター

ダッシュボードの本セクションには、オファ어의通貨と請求日付範囲に基づいてレコードを絞り込めるフィルターがあります。請求日付範囲を使用すると、請求日が指定された日付より前か後か、日付範囲内かを選択できます。選択したフィルターは、メトリック、傾向、内訳および詳細データセクションで表示されるデータを更新します。デフォルトの請求日付範囲は、過去 6 か月のデータを取り込むように設定されています。

セクション 3: 主要なメトリクス

ダッシュボードのこのセクションでは、主要業績評価メトリクス (KPI) が表示され、選択したフィルターを使用した主な請求済み収益の数値を視覚的に比較できます。指定した請求日付範囲と通貨における総収益、返金総額、出品料金、卸売コスト (該当する場合)、販売者純収益の KPI が表示されます。フィルター条件を更新すると、さまざまな通貨と日付のメトリクスを表示できます。

セクション 4: トレンド

ダッシュボードの本セクションには、指定した日付範囲の請求済み収益のトレンドが表示されます。日ごと、前月比ごと、四半期ごと、前年比ごとなど、指定した日付の集計別にトレンドを表示して、請求済み収益に関するインサイトを得ることができます。請求済み収益のトレンド情報は、集計レベル、またはオファ어의表示タイプ別に確認できます。

- 請求トレンド - 日付集計フィルターから選択した日付範囲における総収益、販売者純収益、卸売コスト (該当する場合) 返金額の推移のスナップショットを表示します。
- オファ어의可視性別の請求トレンド - [プライベートオファ어](#)、パブリックオファ어 (またはセルフサービス)、エンタープライズプログラムにおけるオファ어数と総収益の推移のスナップショットを、オファ어의可視性タイプ別に表示します。

セクション 5: 内訳

ダッシュボードの本セクションには、サブスクライバー、登録販売者、サブスクライバーの地域、製品のタイトルにわたるビジネスに関する主要なメトリクスが表示されます。総収益、支払人数、サブスクライバー数、総返金額、出品手数料、販売者純収益、卸売コストでフィルタリングできます。

セクション 6: 詳細なデータ

ダッシュボードのこのセクションには、時間単位、年単位、または月単位の製品の使用量についてが顧客に AWS 請求する合計金額を含む、すべての請求済み売上が表示されます。は、次の 3 つの頻度で顧客に AWS 請求します。

- サブスクリプションの承認日 (前払い請求)
- カスタム支払いスケジュール (柔軟な支払いスケジュールによって作成されたプライベートオファー)
- 前月の使用量に対する当月の 2 日目と 3 日目の従量制使用量

Note

2021 年 4 月 1 日より前に作成された請求書には、関連する契約 ID、オファー ID、サブスクライバー AWS アカウント ID、サブスクライバー会社名がない場合があります。

顧客への請求は、顧客に請求されてから 24 時間後に、本詳細データセクションに表示されます。例えば、11 月 3 日に顧客に請求が行われた場合、ダッシュボードには 11 月 4 日に同請求書が表示されます。QuickSight テーブルからデータをエクスポートおよびダウンロードする方法の詳細については、QuickSight ユーザーガイドの[「ビジュアルからのデータのエクスポート」](#)を参照してください。

詳細なデータの説明

列	[Description] (説明)
請求書日付	製品サブスクリプションの料金を顧客に請求した日付。
支払期日	YYYY-MM-DD 形式の支払期日。

列	[Description] (説明)
支払い条件	顧客の AWS 請求書の支払い条件。
請求書 ID	料金が請求された AWS 請求書に割り当てられた AWS ID。
出品料金の請求書 ID	AWS EMEA SARL、日本、またはオーストラリアの法人 (登録販売者) を通じて AWS Marketplace サブスクリプションが取引される場合、販売者は販売者出品料金に付加価値税 (VAT) を請求し、税準拠の請求書を提出する必要があります。該当する取引において出品料金に課される VAT の請求書 ID は、ソフトウェアまたは製品のサブスクリプション請求書 ID とは異なります。AWS, Inc. からのトランザクションの値は「MISSING_LISTING_FEE_Invoice_ID」となります。これは、出品料金の請求書 ID が適用されないためです。
卸売請求書 ID	チャネルパートナープライベートオファー (CPPO) で ISV とチャネルパートナー間の販売を表す支払い不可能な請求書に割り当てられた AWS ID。パブリックオファーと Marketplace Private Offer (MPOs 「該当なし」) になります。
サブスクライバーの会社名	製品をサブスクライブしているアカウント名。
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの E メールドメイン	製品をサブスクライブしているアカウントに関連付けられた E メールドメイン。例えば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
サブスクライバーの市町区村	製品をサブスクライブしているアカウントに関連付けられた請求先住所の市町区村。

列	[Description] (説明)
サブスクライバーの州または地域	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある都道府県
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある国
サブスクライバーの郵便番号	製品をサブスクライブしているアカウントに関連付けられた請求先住所の郵便番号。
製品のタイトル	製品のタイトル。
オファー名	販売者が定義したオファー名。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意的な契約データフィードリファレンス。
契約開始日	顧客の製品サブスクリプションの開始日 (MM-DD-YYYY) 形式。将来の日付の契約の場合、この日付は受理日とは異なる場合があります。
契約受理日	顧客が製品をサブスクライブした日付 (MM-DD-YYYY 形式)。
契約終了日	契約の有効期限が終了する日付 (MM-DD-YYYY 形式)。従量制/従量制料金サブスクリプションの場合、この日付は 9999 年 1 月 1 日に設定されます。
使用期間終了日	製品の使用期間終了日。
使用期間開始日	製品の使用期間開始日。

列	[Description] (説明)
支払いステータス	AWS が銀行口座に資金、一部、または全部を支払っていないことを確認する請求書に関連付けられたステータス。考えられるステータスは、支払い済み、一部支払い済み、失敗、未支払いです。
支払日	販売者の銀行への支払い AWS が開始された日付。
銀行への支払いのトレース ID	銀行への支払いの場合、トレース ID は銀行によって割り当てられます。支払い銀行トレース ID を使用して、販売者の銀行提供の預金通知とレポートを AWS Marketplace レポート内の請求書に関連付けることができます。
総収益	顧客に請求される製品の従量課金額または月額料金。
総返金	データカバレッジ期間中に返金が処理された場合に顧客に払い戻されるサブスクリプション費用の合計金額。
出品料金	請求額から差し引かれる AWS Marketplace 料金額。
出品料の払い戻し	データカバレッジ期間中に返金が処理された場合に返金された AWS Marketplace 料金の一部。
出品料金の割合	請求額から差し引かれる AWS Marketplace 料金の割合。
販売者のタックスシェア	この取引で請求された米国の消費税と使用税の合計額
販売者のタックスシェアの払い戻し	返金が処理される場合にこの取引に対して払い戻される米国の消費税と使用税の合計金額。

列	[Description] (説明)
AWS タックスシェア	販売者の代理でこの取引で請求された米国の消費税と使用税の合計額。
AWS タックスシェア返金	返金が処理された場合に、このトランザクションに対して返金される米国の売上税および使用税の合計額 (出品者に代わってそのような税金が徴収されたとき)。
卸売コスト	チャネルパートナーのプライベートオファアのみ。リセラーに請求される製品のコスト。例えば、リセラーが製造元の製品を販売したときに製造元に支払う金額などです。卸売コストは、希望小売り価格に割引率を掛けたものです。
卸売コストの払い戻し	チャネルパートナーのプライベートオファアのみ。リセラーから返金された製品のコスト。
卸売販売者税割引	販売者が課税対象となるチャネルパートナープライベートオファア (CPPO) での ISV とチャネルパートナー間の販売に対する税金。パブリックオファアと Marketplace プライベートオファア (MPPOs 「該当なし」) になります。
卸売販売者のタックスシェアの返金	販売者が責任を負うチャネルパートナープライベートオファア (CPPO) における ISV とチャネルパートナー間の販売に対する税金の返金。パブリックオファアと Marketplace プライベートオファア (MPPOs 「該当なし」) になります。
卸売の他の販売者のタックスシェア	販売者が課税対象となるチャネルパートナープライベートオファア (CPPO) での ISV とチャネルパートナー間の販売に対する税金。このフィールドは、トランザクションに関与した他の販売者が責任を負うときに入力されます。パブリックオファアと Marketplace プライベートオファア (MPPOs 「該当なし」) になります。

列	[Description] (説明)
卸売の他の販売者のタックスシェアの返金	販売者が責任を負うチャネルパートナープライベートオファー (CPPO) における ISV とチャネルパートナー間の販売に対する税金の返金。このフィールドは、トランザクションに關与した他の販売者が責任を負うときに入力されます。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
卸売 AWS 税配分	が課税対象となるチャネルパートナープライベートオファー (CPPO) での ISV とチャネルパートナー間の販売に対する AWS 税金。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
卸売 AWS 税配分の返金	AWS が課税対象となるチャネルパートナープライベートオファー (CPPO) での ISV とチャネルパートナー間の販売に対する税金の返金。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
販売者の純収益	取引に対して請求される合計金額。AWS Marketplace 料金、返金、米国の消費税と使用税を差し引いた額。
通貨	取引の通貨。例えば、取引が US ドルの場合、エントリは USD です。
取引リファレンス ID	トランザクションを表す一意の識別子。これを使用して、AWS Marketplace レポート間でトランザクションを関連付けることができます。

列	[Description] (説明)
AWS 登録販売者	取引を促進した事業体の識別子。可能な値は以下のとおりです。 • AWS_INC: AWS, Inc. の識別子 (米国を拠点とする) • AWS_EUROPE: EMEA AWS SARL の識別子 (ルクセンブルクに基づく) • AWS オーストラリア: AWS Australia Pty Ltd の識別子 • AWS_JAPAN: Japan AWS G.K. の識別子。 • AWS_KOREA: AWS 韓国の識別子
再販承認 ID	登録された機会の一意の識別子。
再販承認名	登録された機会の一意の名前。
再販承認の説明	ISV の定義による登録済み機会の説明。
再販会社名	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウント名。
支払者 AWS アカウント ID	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウントの ID。
支払人の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。例えば、E メールアドレスが liu-jie@example.com の場合、エンタリは example.com です。
支払人の市町区村	ソフトウェア料金の請求先となるアカウントに関連付けられた請求先住所の市区町村。
支払人の都道府県または地域	料金の請求先となるアカウントに関連付けられた請求先住所の都道府県。

列	[Description] (説明)
支払人の国	料金の請求先となるアカウントに関連付けられた 2 文字の国コード。
支払人の郵便番号	ソフトウェア料金の請求先となるアカウントに関連付けられた請求先住所の郵便番号。
ISV アカウント ID	製品またはサービスの所有者の識別子。
ISV の会社名	製品またはサービスの所有者の商号。
製品 ID	ソフトウェア製品のわかりやすい一意の識別子。
支払済み純収益	販売者に支払われた取引の合計額 金額が「販売者純収益」と等しくない場合、これは部分的な支払いです。
未支払いの純収益	販売者に支払われていないトランザクションの合計金額。金額が 0 以外の場合、残高はお客様が負担します。

回収と支払いダッシュボード

回収と支払いダッシュボードには、銀行口座に支払われた資金に関するデータが表示されます。また、部分的な支払い、全額の支払い、または支払いの失敗が発生したすべての請求書のリストも表示されます。このリストには、支払いが実行されていない請求書は含まれません。未払いの請求書を確認するには、請求済み収益ダッシュボードを使用します。

支払いには、顧客への支払い、製品のサブスクリプションの決済済み返金、顧客に徴収または返金された税金が含まれます。返金は販売者による承認後に行われるため、ダッシュボード上の返金はマイナスの金額として表示されます。

回収と支払いダッシュボードを使用すると、顧客の支払いにすばやくアクセスできます。支払いが送信されてから 5 日後に作成される従来の[支払いレポート](#)と比較して、約 4 日間の節約が見込まれます。

Note

顧客の支払い条件は異なるため AWS、未回収の年齢カテゴリの一部の資金は顧客から支払われていない可能性があります。

回収と支払いダッシュボードには、運用プロセスと財務プロセスに関する情報が表示されます。ダッシュボードは毎日更新されます。詳細については、以下のトピックを参照してください。

ダッシュボードを開くには、にサインインし [AWS Marketplace 管理ポータル](#)、インサイト、財務オペレーションを選択し、コレクションと支払いタブを選択します。

トピック

- [回収と支払いダッシュボードの更新頻度](#)
- [セクション 1: コントロール](#)
- [セクション 2: フィルター](#)
- [セクション 3: 主要なメトリクス](#)
- [セクション 4: トレンド](#)
- [セクション 5: 内訳](#)
- [セクション 6: 詳細なデータ](#)

回収と支払いダッシュボードの更新頻度

回収と支払いダッシュボードは、北米の営業日にのみ更新されます。支払い済みの請求書は、銀行に入金されてから 1 日以内に確認できます。

セクション 1: コントロール

ダッシュボードの本セクション内には、ダッシュボードのデータを絞り込むためのフィルターがあります。たとえば、[AWS Marketplace イベントの通知](#)からフィールドのフィルターを選択して、特定の顧客アカウント ID、サブスクライバーの会社名、またはオファー ID の支払いを確認できます。また、支払いステータス別にフィルタリングして、自分に支払われたすべての請求書、または未払いの請求書を確認することもできます。ビジュアルに含める日付範囲などを設定できるフィルターを分析に追加することが可能です。セクション内で選択済みのフィルターにより、メトリクス、トレンド、内訳、詳細データセクションに表示されるデータが更新されます。

フィルタリングの詳細については、[QuickSight ユーザーガイドの「QuickSight でのデータのフィルタリング」](#)を参照してください。QuickSight

コントロールの記述

コントロール名	説明
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの会社名	製品をサブスクライブしているアカウント名。
製品のタイトル	製品のタイトル。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意的な契約データフィードリファレンス。
AWS 登録販売者	取引を促進した事業体の識別子。可能な値は以下のとおりです: • AWS_INC: AWS Inc. の識別子 (米国ベース) • AWS_EUROPE: EMEA AWS SARL の識別子 (ルクセンブルクに基づく) • AWS_AUSTRALIA: AWS Australia Pty Ltd (AWS オーストラリア) の識別子 • AWS_JAPAN: Japan AWS G.K. の識別子。 • AWS_KOREA: AWS 韓国の識別子
支払いステータス	AWS が銀行口座に資金、一部、または全部を支払っていないことを確認する請求書に関連付けられたステータス。考えられるステータスは、支払い済み、一部支払い済み、失敗、未支払いです。
支払者 AWS アカウント ID	料金の請求先となるアカウントの ID。

コントロール名	説明
支払人の会社名	料金の請求先となるアカウントの商号。
リセラー会社	ソフトウェア製造元の製品の販売を許可されているリセラーアカウントの商号。
リセラー AWS アカウント ID	ISV から卸売で製品またはサービスを購入して顧客に再販したアカウントの ID。
再販承認 ID	登録された機会の一意の識別子。
再販承認名	登録された機会の一意の名前。
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。
サブスクライバーの州または地域	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある州または地域
取引リファレンス ID	AWS Marketplace レガシーレポート間でトランザクションに関連付けるのに役立つトランザクションの一意の識別子。
銀行への支払いのトレース ID	銀行への支払いの場合、トレース ID は銀行によって割り当てられます。銀行のトレース ID では、販売者の銀行が提供する入金通知や AWS Marketplace レポートをレポート内の請求書と関連付けることができます。

セクション 2: フィルター

ダッシュボードの本セクションは、オファ어의通貨と 2 つの異なる日付ディメンション (日付フィールドの値が、指定日の前か後か、または日付範囲内か) に基づいてレコードを絞り込むフィルターについて説明しています。日付ディメンションは支払い期日または最終支払い日です。支払い日は、指定された日付範囲内に支払われる請求書に結果を制限します。支払期日には、支払日に関係なく、指定範囲内の支払い期限が記載された請求書が含まれます。日付カテゴリフィルターでは、メトリクス、トレンド、内訳、詳細データセクションに表示されるデータが更新されます。デフォルトの日付カテゴリは最終支払い日で、過去 6 か月のデータを取得します。

セクション 3: 主要なメトリクス

ダッシュボードの本セクションには、主要業績評価メトリクス (KPI) が表示され、支払い済みおよび未支払いの収益を視覚的に比較できます。KPI は、指定されたフィルタ条件に基づいた総収益、純利益、卸売原価 (該当する場合)、支払額、未払額に対して表示されます。

セクション 4: トレンド

ダッシュボードのこのセクションには、指定した日付範囲における支払いと延滞のトレンドが表示されます。日、月、四半期、年など、指定された日付集計で傾向を表示して、AWS Marketplace コレクションのヘルスに関するインサイトを得ることができます。トレンドビューには、次のようなものが含まれます。

- 支払いトレンド - 支払いまでの平均日数とそれに関連する純収益のスナップショットが表示されます。このトレンドでは、請求日から支払い日までの日数を測定して、回収効率をレポートします。日付集計フィルターから日付範囲を選択できます。
- 支払い済み金額の期間 - 標準期間の売掛金バケット (期限前、1～30 日、31～60 日など) で分類された純利益および支払い済み請求書数のスナップショットを提供します。トレンドでは、支払い期日と支払い日の間の日数を測定して、支払いが顧客の支払い条件の範囲内であったかどうかを報告します。
- 未払い金額の期間 - 期日経過バケット (期限前、1～30 日、31～60 日など) で分類された、純収益、未対応請求書数や未支払い請求書数のスナップショットが表示されます。未払い資金には、期限を過ぎていない金額が含まれる場合があります。トレンドでは、本日の日付から支払期日までの日数を測定して、受取売掛金を表示します。

セクション 5: 内訳

ダッシュボードのこのセクションには、オファー ID、製品のタイトル、支払者の会社名、サブスクライバーの会社名、リセラー名 (チャネルパートナーのプライベートオファーに参加している場合)、支払人の地域、サブスクライバーの地域ごとに、売掛金が表示されます。この内訳を使用して、各カテゴリの支払い済み売掛金と未払いの売掛金を比較します。

セクション 6: 詳細なデータ

ダッシュボードのこのセクションには、製品、顧客、オファーの詳細別に、すべての支払いと未回収額が表示されます。

Note

2021 年 4 月 1 日より前に作成された請求書には、関連する契約 ID、オファー ID、サブスクライバー AWS アカウント ID、サブスクライバー会社名がない場合があります。

QuickSight テーブルからデータをエクスポートおよびダウンロードする方法については、QuickSight ユーザーガイドの「[ビジュアルからのデータのエクスポート](#)」を参照してください。

詳細なデータの説明

列	[Description] (説明)
請求書日付	製品サブスクリプションの料金を顧客に請求した日付。
支払期日	YYYY-MM-DD 形式の支払期日。
支払い条件	顧客の AWS 請求書の支払い条件。
請求書 ID	料金が請求された AWS 請求書に割り当てられた AWS ID。
出品料金の請求書 ID	AWS EMEA SARL、日本、またはオーストラリアの法人 (登録販売者) を通じて AWS Marketplace サブスクリプションが取引される場合、販売者の出品料金に VAT を請求するには、販売者のマーケットプレイスオペレーター (AWS EMEA SARL など) が必要です。該当する取引において出品料金に課される VAT の請求書 ID は、ソフトウェアまたは製品のサブスクリプション請求書 ID とは異なります。
卸売請求書 ID	チャネルパートナープライベートオファー (CPPO) で ISV とチャネルパートナー間の販売を表す支払い不可能な請求書に割り当てられた AWS ID。パブリックオファーと Marketplace Private Offer (MPOs 「該当なし」) になります。

列	[Description] (説明)
サブスクライバーの会社名	製品をサブスクライブしているアカウント名。
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの E メールドメイン	製品をサブスクライブしているアカウントに関連付けられた E メールドメイン。例えば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
サブスクライバーの市町区村	製品をサブスクライブしているアカウントに関連付けられた請求先住所の市町区村。
サブスクライバーの州または地域	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある都道府県
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある国
サブスクライバーの郵便番号	製品をサブスクライブしているアカウントに関連付けられた請求先住所の郵便番号。
製品のタイトル	製品のタイトル。
オファー名	販売者が定義したオファー名。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意の契約データフィードリファレンス。
契約開始日	顧客の製品サブスクリプションの開始日 (MM-DD-YYYY) 形式。将来の日付の契約の場合、この日付は受諾日とは異なる場合があります。

列	[Description] (説明)
契約終了日	契約の有効期限が終了する日付 (MM-DD-YYYY 形式)。従量制/従量制料金サブスクリプションの場合、この日付は 9999 年 1 月 1 日に設定されます。
契約受理日	顧客が製品をサブスクライブした日付 (MM-DD-YYYY 形式)。
使用期間終了日	製品の使用期間終了日。
使用期間開始日	製品の使用期間開始日。
支払いステータス	が前回の支払い以降に資金を AWS 回収して銀行口座に支払ったことを確認する請求書に関連付けられたステータス。関連する請求書の支払い資金が回収され、支払いが完了しました。関連する未払請求書の資金は、回収も支払いも行われていません。
支払日	販売者の銀行への支払い AWS が開始された日付。
銀行への支払いのトレース ID	銀行への支払いの場合、トレース ID は銀行によって割り当てられます。銀行トレース ID を使用して、販売者の銀行提供の預金通知とレポートを AWS Marketplace レポートの請求書に関連付けることができます。
総収益	顧客に請求される製品の従量課金額または月額料金。
総返金	データカバレッジ期間中に返金が処理された場合に顧客に払い戻されるサブスクリプション費用の合計金額。

列	[Description] (説明)
出品料金	請求額から差し引かれる AWS Marketplace 料金額。
出品料の払い戻し	データカバレッジ期間中に返金処理された場合に返金された AWS Marketplace 料金の一部。
出品料金の割合	請求額から差し引かれる AWS Marketplace 料金の割合。
販売者のタックスシェア	この取引で請求された米国の消費税と使用税の合計額
販売者のタックスシェアの払い戻し	返金処理される場合にこの取引に対して払い戻される米国の消費税と使用税の合計金額。
AWS タックスシェア出品料金	マーケットプレイスオペレーター AWS として、販売者が行った売上に対して独自の名前で VAT を請求および徴収する必要があります。この金額は、AWS Marketplace 販売者アカウントに請求される VAT の金額に対応します。
AWS タックスシェア返金出品料金	マーケットプレイスオペレーター AWS として、販売者が行った売上に対して独自の名前で VAT を返金する必要があります。この金額は、AWS Marketplace 販売者アカウントに返金される VAT の金額に対応します。
卸売コスト	チャネルパートナーのプライベートオファーマのみ。リセラーに請求される製品のコスト。例えば、リセラーが製造元の製品を販売したときに製造元に支払う金額などです。卸売コストは、希望小売り価格に割引率を掛けたものです。
卸売コストの払い戻し	チャネルパートナーのプライベートオファーマのみ。リセラーから返金された製品のコスト。

列	[Description] (説明)
卸売販売者のタックスシェア	販売者が課税対象となるチャネルパートナープライベートオファー (CPPO) での ISV とチャネルパートナー間の販売に対する税金。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
卸売販売者のタックスシェアの返金	販売者が責任を負うチャネルパートナープライベートオファー (CPPO) における ISV とチャネルパートナー間の販売に対する税金の返金。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
卸売の他の販売者のタックスシェア	販売者が課税対象となるチャネルパートナープライベートオファー (CPPO) での ISV とチャネルパートナー間の販売に対する税金。このフィールドは、トランザクションに参与した他の販売者が責任を負うときに入力されます。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
卸売の他の販売者のタックスシェアの返金	販売者が責任を負うチャネルパートナープライベートオファー (CPPO) における ISV とチャネルパートナー間の販売に対する税金の返金。このフィールドは、トランザクションに参与した他の販売者が責任を負うときに入力されます。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
卸売 AWS 税配分	が課税対象となるチャネルパートナープライベートオファー (CPPO) での ISV とチャネルパートナー間の販売に対する AWS 税金。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。

列	[Description] (説明)
卸売 AWS 税配分の返金	AWS が課税対象となるチャネルパートナープライベートオファー (CPPO) での ISV とチャネルパートナー間の販売に対する税金の返金。パブリックオファーと Marketplace プライベートオファー (MPPOs 「該当なし」) になります。
販売者の純収益	取引に対して請求される合計金額。AWS Marketplace 料金、返金、米国の消費税と使用税を差し引いた額。
通貨	取引の通貨。例えば、取引が US ドルの場合、エントリは USD です。
取引リファレンス ID	トランザクションを表す一意の識別子。これを使用して、AWS Marketplace レポート間でトランザクションを関連付けることができます。
AWS 登録販売者	取引を促進した事業体の識別子。可能な値は以下のとおりです。 • AWS_INC: AWS Inc. の識別子 (米国ベース) • AWS_EUROPE: EMEA AWS SARL の識別子 (ルクセンブルクに基づく) • AWS Australia Pty Ltd (AWS オーストラリア) • AWS 日本株式会社 • AWS_韓国
再販承認 ID	登録された機会の一意の識別子。
再販承認名	登録された機会の一意の名前。
再販承認の説明	ISV の定義による登録済み機会の説明。
リセラーの会社名	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウント名。

列	[Description] (説明)
リセラー AWS アカウント ID	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウントの ID。
支払人の会社名	料金の請求先となるアカウントの名前。
支払者 AWS アカウント ID	料金の請求先となるアカウントの ID。
支払人の E メールドメイン	料金の請求先となるアカウントに関連付けられている E メールドメイン。例えば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
支払人の市町区村	ソフトウェア料金の請求先となるアカウントに関連付けられた請求先住所の市区町村。
支払人の都道府県または地域	料金の請求先となるアカウントに関連付けられた請求先住所の都道府県。
支払人の国	料金の請求先となるアカウントに関連付けられた 2 文字の国コード。
支払人の郵便番号	ソフトウェア料金の請求先となるアカウントに関連付けられた請求先住所の郵便番号。
ISV アカウント ID	製品またはサービスの所有者の識別子。
ISV の会社名	製品またはサービスの所有者の商号。
製品 ID	ソフトウェア製品のわかりやすい一意の識別子。
支払いステータス	AWS が銀行口座に資金、一部、または全部を支払っていないことを確認する請求書に関連付けられたステータス。考えられるステータスは、支払い済み、一部支払い済み、失敗、未支払いです。

列	[Description] (説明)
支払済み純収益	販売者に支払われた取引の合計額 金額が「販売者純収益」と等しくない場合、これは部分的な支払いです。
未支払いの純収益	販売者に支払われていないトランザクションの合計金額。金額が 0 以外の場合、残高はお客様が負担します。
支払い期間	資金が収集される売掛金範囲を説明しているカテゴリー (期限前、1～30 日、31～60 日など)。

税ダッシュボード

税ダッシュボードは、AWS Marketplaceでのトランザクションの米国税と国際税に関する視覚化データときめ細かなデータを提供します。

このダッシュボードを開くには、にサインインし[AWS Marketplace 管理ポータル](#)、インサイト、マーケティングを選択し、パフォーマンスの一覧表示タブを選択します。

トピック

- [税ダッシュボードの更新頻度](#)
- [セクション 1: コントロール](#)
- [セクション 2: 日付フィルター](#)
- [セクション 3: メトリクス](#)
- [セクション 4: トレンド](#)
- [セクション 5: 内訳](#)
- [セクション 6: 詳細なデータ](#)
- [米国の免除コード](#)
- [EU 適用免除コード](#)

税ダッシュボードの更新頻度

ダッシュボードは毎日更新されます。アップストリームシステムからの最新のトランザクションデータが遅延した場合、最新のデータがダッシュボードに反映されるまでに 1~2 日かかる場合があります。

セクション 1: コントロール

ダッシュボードの本セクション内には、税データを絞り込むためのフィルターがあります。また、ビジュアルに含める日付範囲などを設定できるフィルターを分析に追加することも可能です。セクション内で選択済みのフィルターにより、メトリクス、トレンド、内訳、詳細データセクションに表示されるデータが更新されます。

フィルタリングの詳細については、[QuickSight ユーザーガイドの「QuickSight でのデータのフィルタリング」](#)を参照してください。QuickSight

コントロールの記述

コントロール名	説明
請求書 ID	顧客に請求される料金に割り当てられた AWS ID。
課税対象顧客 AWS アカウント ID	製品サブスクリプションに課税されたアカウントの AWS ID。
AWS 登録販売者	取引を促進した事業体の識別子。
課税対象顧客の国	課税対象顧客に関連付けられた 2 文字の国コード
課税対象顧客の州またはリージョン	課税対象顧客に関連付けられた請求先住所の州またはリージョン。
課税対象顧客の都市	課税対象顧客に関連付けられた請求先住所の都市
課税対象	課税対象、非課税、税還付の対象となるソーストランザクション。

コントロール名	説明
納税義務者	AWS または Seller です。販売者が納税義務者である場合は、税金が徴収されます。AWS が納税義務者である場合、消費税は によって徴収および送金されます AWS。詳細については、「AWS Marketplace 出品者の税務ヘルプ」を参照してください。 徴税しない場合、ここには値が表示されません。販売者は、徴税の義務があるため、請求書ごとに徴税されたかどうかを確認する必要があります。
税タイプ	取引に適用される税のタイプ。使用できる値は None、Sales、および SellerUse、Tax registration type、VAT (付加価値税)、CNPJ、IGST、CT です。

セクション 2: 日付フィルター

ダッシュボードの本セクションには、特定の期間中の請求書の日付に基づいてレコードを絞り込むためのフィルターがあります。請求書の日付が選択されたフィルターにより、メトリクス、トレンド、内訳、詳細データセクションに表示されるデータが更新されます。例えば、日付フィルター Past 30 days を選択すると、すべてのセクションに過去 30 日以内に課税された請求書に関連するデータが反映されます。

セクション 3: メトリクス

ダッシュボードのこのセクションには、課税額、非課税額、税額、課税された請求書の数、トランザクションの合計数など、税に関連するメトリクスを視覚化するための主要業績評価指標 (KPI) が表示されます。これらのフィルターセクションの使用日条件を更新して、日付範囲を更新できます。

セクション 4: トレンド

ダッシュボードの本セクションには、指定した日付範囲の税のトレンドが表示されます。日ごと、前月比、前四半期比、前年比など、指定した日付の集計別にトレンドを表示して、税に関するインサイトを得ることができます。税に関する以下のトレンド情報を利用できます。

- 税のトレンド - 金額 - 選択した期間の課税額と税金額の両方を、請求書日付別にスナップショットで表示します。
- 税のトレンド - 請求書の数 - 選択した期間の請求書数のスナップショットを、請求書の日付別に表示します。

セクション 5: 内訳

ダッシュボードのこのセクションには、請求書 ID、税タイプ、製品タイトル、および顧客の地域別に、税メトリクスが表示されます。

セクション 6: 詳細なデータ

ダッシュボードのこのセクションには、AWS Marketplaceでの組織のトランザクションの詳細な税金データが表示されます。

詳細なデータの説明

列	[Description] (説明)
請求書 ID	料金が請求された AWS 請求書に割り当てられた AWS ID。これは、購入者の商用請求書、販売者卸売請求書、または販売者出品料金請求書です。
出品料金の請求書 ID	販売者出品料金と VAT が販売者に請求された請求書に割り当てられた AWS ID。ISVs にのみ適用されます。
明細項目 ID	明細項目の一意的識別子。返金取引の明細項目 ID は先渡し取引税のものと同じです。
顧客請求書 ID	顧客請求書 ID
納税義務者	AWS または Seller です。販売者が納税義務者である場合は、税金が徴収されます。AWS が納税義務者である場合、消費税は によって徴収および送金されます AWS。詳細については、「 AWS Marketplace 出品者の税務ヘルプ 」を参照してください。

列	[Description] (説明)
	徴税しない場合、ここには値が表示されません。販売者は、徴税の義務があるため、請求書ごとに徴税されたかどうかを確認する必要があります。
取引タイプコード	取引のタイプコード。値には以下のものが含まれます。 • AWS - 源泉徴収取引 • REFUND - 全額返金または一部返金 • TAXONLYREFUND - 税のみの返金 返金取引の明細項目 ID は元の先渡し取引のものと同じです
製品 ID	ソフトウェア製品のわかりやすい一意の識別子。
製品のタイトル	購入された製品の名前。
製品税コード	製品の税プロパティを識別するための標準コード。製品を作成または変更するときにプロパティを選択します。
請求書日付	製品サブスクリプションの料金および税を顧客に請求した日付。
課税対象顧客 AWS アカウント ID	課税対象顧客の AWS アカウント ID。
課税対象顧客の国	課税対象顧客に関連付けられた 2 文字の国コード
課税対象顧客の州またはリージョン	課税対象顧客に関連付けられた請求先住所の州またはリージョン。
課税対象顧客の都市	課税対象顧客に関連付けられた請求先住所の都市

列	[Description] (説明)
課税対象顧客の郵便番号	税額計算に使用する郵便番号。
税タイプ	取引に適用される税のタイプ。取り得る値は None、Sales、SellerUse、ax registration type、VAT、CNPJ、IGST、CT です。
管轄区域	納税地に使用される住所の管轄区域。指定できる値は、State、County、City、District です。
課税管轄	課税管轄地域の名前。
表示価格課税タイプ	顧客に表示される価格の課税タイプ。すべての AWS Marketplace サービスは排他的です。
課税管轄地域の税率	管轄区域で適用される税率。
課税額	管轄区域で課される税額。
通貨	課税される通貨。例えば、取引が US ドルの場合、エントリは USD です。
法的な税額	司法レベルで請求される法定通貨で表示される税額。
法的な税通貨	請求された税金の法定通貨で表示される通貨。
税計算理由コード	取引が課税対象となるか、非課税となるか、免除とされるか、ゼロ評価済みであるかを管轄区域別に整理したもの。
税計算に使用された日付	取引の税金の計算に使用される日付。
顧客の免除証明書 ID	免除証明書の証明書 ID。
顧客の免除証明書 ID ドメイン	証明書が Amazon システムに保存されている場所。

列	[Description] (説明)
顧客の免除証明書レベル	免除を提供した管轄区域。
顧客の免除コード	免税を指定するコード。例えば、RESALE。
取引リファレンス ID	AWS Marketplace レポート間でトランザクションを関連付けるのに役立つトランザクションの一意の識別子。
AWS 登録販売者	取引を促進した事業体の識別子。
購入者請求書 ID	購入者に請求された AWS 請求書に割り当てられた AWS ID。販売者は、卸売請求書または出品料金請求書に対応する購入者料金に関連付けることができます。

 Note

合計税額と購入者の合計請求額は、[\[請求済み収益ダッシュボード\]](#) の [AWS 税配分] フィールドと [総収益] フィールドに移動しました。

米国の免除コード

免除コード	説明
501C	501C 組織
AP	農業生産
CO	慈善団体
DIRECT	直接支払いの許可
DISTSBT	ディストリビューター
DP	国連/外交官

免除コード	説明
DV	障害のある退役軍人
EDI	教育機関
FG	連邦政府
HCP	医療プロバイダー
HO	病院 (非営利または州)
INSUR	保険
IPM	IPM
LB	ライブラリ
MPU	複数のポイントオブユースの免除、デジタル製品/ソフトウェアの免除
NA	ネイティブアメリカン
NAI	ネイティブアメリカン個人
NP	非営利組織
NPR	非営利宗教組織
OT	その他
RESALE	リセラー
SBE	中小企業の免除
SLG	州/地方自治体

EU 適用免除コード

免除コード	説明
SpecialOrg	VAT の請求が免除される組織。
SpecialZone	非課税のエリア。
SplitPayment	購入者は、請求書に記載されている付加価値税を税務当局に直接支払います。

販売業務用ダッシュボード

AWS Marketplace には、販売データの追跡に役立つ複数のダッシュボードが用意されています。

トピック

- [契約および更新ダッシュボード](#)
- [使用状況ダッシュボード](#)

契約および更新ダッシュボード

このダッシュボードには、契約に署名してから 24 時間以内に契約と更新に関する情報が表示されます AWS Marketplace。このダッシュボードは、期限切れのサブスクリプションを追跡するのに役立ちます。

ダッシュボードを開くには、 を起動し AWS Marketplace 管理ポータル、インサイトタブを選択し、セールスオペレーションを選択します。

トピック

- [セクション 1: コントロール](#)
- [セクション 2: 日付カテゴリの選択](#)
- [セクション 3: メトリクス](#)
- [セクション 4: サブスクリプションのトレンド](#)
- [セクション 5: 内訳](#)
- [セクション 6: 詳細なデータ](#)

セクション 1: コントロール

このセクションでは、契約と更新データを絞り込むためのフィルターを提供します。以下のフィルターから選択できます。

Note

フィルタリングの詳細については、[QuickSight ユーザーガイドの「QuickSight でのデータのフィルタリング」](#)を参照してください。QuickSight

コントロールの記述

コントロール名	説明
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの会社名	製品をサブスクライブしているアカウント名。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、またはエンタープライズ契約オファーのいずれになるか。
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意の契約データフィードリファレンス。
CPPO フラグ	チャネルパートナーのプライベートオファーを使用して契約が締結されたかどうかを示す「はい/いいえ」のフラグ。「はい」の場合、登録販売者がチャネルパートナーです。「いいえ」の場合、登録販売者はソフトウェアメーカー (独立系ソフトウェアベンダー) です。
ISV AWS アカウント ID	製品またはサービスの所有者のアカウント。

コントロール名	説明
ISV の会社名	製品またはサービスの所有者の商号。
レガシーの製品 ID	ソフトウェア製品のレガシーの一意の識別子。
製品のタイトル	製品のタイトル。
再販承認 ID	ISV がチャネルパートナーに提供した再販承認の ID。
再販承認名	ISV がチャネルパートナーに提供した再販承認の名前。
リセラー AWS アカウント ID	ISV から卸売で製品またはサービスを購入して顧客に再販したアカウントの ID。
リセラーの会社名	ソフトウェア製造元の製品の販売を許可されているリセラーアカウントの商号。
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。
サブスクライバーの州または地域	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある州または地域
サブスクライバーの市町区村	製品をサブスクライブしているアカウントに関連付けられた請求先住所の市町区村。

セクション 2: 日付カテゴリの選択

ダッシュボードのこのセクションには、指定した日付範囲の契約終了日、契約終了期間、または契約開始日に基づいてレコードを絞り込むためのフィルターが用意されています。選択した日付カテゴリ内で選択したフィルターにより、メトリクス、トレンド、内訳、詳細データセクションに表示されるデータが更新されます。

最近署名した契約書に関する情報は、契約開始日で絞り込むことができます。契約の更新について知りたい場合は、契約終了日を使用してフィルタリングできます。

セクション 3: メトリクス

ダッシュボードの本セクションには、主要業績評価メトリクス (KPI) が表示され、主要な契約データを視覚的に比較できます。有効な契約の数と終了した契約の数を示す KPI が表示されます。[日付カテゴリを選択] フィールドの日付条件を更新して、日付範囲を更新できます。

セクション 4: サブスクリプションのトレンド

ダッシュボードの本セクションには、指定した日付範囲のサブスクリプションのトレンドが表示されます。日ごと、前月比ごと、四半期ごと、前年比ごとなど、指定した日付の集計別にトレンドを表示して、契約に関するインサイトを得ることができます。契約と更新のトレンド情報は集計されたレベルで確認できます。

- サブスクリプションのトレンド - 選択した日付集計フィルターと選択した日付カテゴリフィルターに基づいて、有効な契約の数と終了した契約の数のスナップショットが表示されます。契約開始日のサブスクリプションのトレンドを表示するには、[日付カテゴリを選択] フィルターで[契約開始日]を選択します。契約終了日のサブスクリプションのトレンドを表示するには、[日付カテゴリを選択] フィルターで[契約終了日]を選択します。
- 更新内訳 - 特定の日数内に終了した、または終了間近である更新のスナップショットが表示されます。1~30 日前、31~60 日前、または 61 日以上前に終了した契約など、複数の期間の更新内訳を表示できます。また、0~30 日、31~60 日というような期間以内に終了する、終了間近の契約を絞り込むこともできます。

セクション 5: 内訳

ダッシュボードのこのセクションには、アクティブなサブスクライバー、ISV、リセラーの会社名にわたるビジネス指標が表示されます。契約 ID、終了契約 ID、オファー ID、更新契約 ID、サブスクライバーの数を表示できます。

セクション 6: 詳細なデータ

ダッシュボードのこのセクションには、契約、オファー、製品、契約の提案者、サブスクライバー、再販承認、リセラー、ISV に関する詳細なデータが表示されます。

Note

QuickSight テーブルからデータをエクスポートおよびダウンロードする方法の詳細については、QuickSight ユーザーガイドの [「ビジュアルからのデータのエクスポート」](#) を参照してください。

詳細なデータの説明

列	[Description] (説明)
サブスクライバーの会社名	サブスクライバーの会社の名前。
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの E メールドメイン	製品をサブスクライブしているアカウントに関連付けられた E メールドメイン。例えば、E メールアドレスが liu-jie@example.com の場合、エントリは example.com です。
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある国
サブスクライバーの州または地域	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある州または地域
サブスクライバーの市町区村	製品をサブスクライブしているアカウントに関連付けられた請求先住所の市町区村。
サブスクライバーの郵便番号	製品をサブスクライブしているアカウントに関連付けられた請求先住所の郵便番号。
製品のタイトル	製品のタイトル。
製品 ID	ソフトウェア製品のわかりやすい一意の識別子。
レガシーの製品 ID	ソフトウェア製品のレガシーの一意の識別子。
オファー名	販売者が定義したオファー名。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、またはエンタープライズ契約オファーのいずれになるか。

列	[Description] (説明)
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意的な契約データフィードリファレンス。
契約開始日	顧客の製品サブスクリプションの開始日 (MM-DD-YYYY) 形式。将来の日付の契約の場合、この日付は受理日とは異なる場合があります。
契約受理日	顧客が製品をサブスクライブした日付 (MM-DD-YYYY 形式)。
契約終了日	契約の有効期限が終了する日付 (MM-DD-YYYY 形式)。従量制/従量制料金サブスクリプションの場合、この日付は 9999 年 1 月 1 日に設定されます。

列	[Description] (説明)
ステータス	契約の現在の状態。サポートされているステータスは次のとおりです。 • アクティブ – 契約の条項の一部またはすべてが有効な状態です。 • 終了 – AWS が開始した終了事由により、契約は事前に合意した終了日より前に終了した状態です。最も一般的な事由は、支払いの不履行です。 • キャンセル済み – 承認者が終了日より前に契約を終了することを選択した状態です。 • 期限切れ – 契約が事前に合意された終了日に終了した状態です。 • 置き換え済み – 契約が代替オファーを使用して置き換えられた状態です。 • 更新済み – 自動更新などの機能を使用して、契約が新しい契約に更新された状態です。 • アーカイブ済み – 契約が終了しましたが、契約の終了、キャンセル、期限切れの正確な終了理由が指定されていない状態です。 • ロールバック – エラーのため、契約改訂がロールバックされた状態です。以前の改訂が現在有効になっています。非アクティブな契約改訂にのみ適用されます。 • 優先 – 契約改訂がアクティブでなくなり、別の契約改訂がアクティブな状態です。非アクティブな契約改訂にのみ適用されます。 • ロールバック – エラーのため、契約改訂がロールバックされた状態です。以前の改訂が現在有効になっています。非アクティブな契約改訂にのみ適用されます。

列	[Description] (説明)
推定契約額	「契約総額」または「購入額」と呼ばれる契約の推定費用です。これは、SaaS、プロフェッショナルサービス、サーバー、契約または年間料金タイプなどの製品タイプに適用されます。
通貨コード	契約の推定費用に関連付けられたオファー料金の通貨です。
リセラーの会社名	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウント名。
リセラー AWS アカウント ID	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウントの ID。
再販承認 ID	登録された機会の一意の識別子。
再販承認名	登録された機会の一意の名前。
再販承認の説明	ISV の定義による登録済み機会の説明。
CPPO フラグ	QuickSight 内の計算フィールド。

列	[Description] (説明)
契約の意図	現在の契約が、以前の agreement_id 列に入力された以前の契約の更新またはアップグレードである場合に入力されます。可能な値は以下のとおりです: • アップグレード: プライベートオファーの場合、アップグレードとはあらゆる変更 (アップグレード、ダウングレード、修正、拡張、更新) を指す広義の用語です。パブリックオファーの場合、アップグレードとは、条件を変更 (別のオファーを使用) して新しい契約を結ぶという特定の契約であり、その特定の契約はアーカイブされます。 • 更新: パブリックオファーのみが対象です。更新契約の開始日が特定の契約の終了日であり、オファー ID は変更されていないが契約 ID は変更されている場合。例えば、契約 A が契約 B になったとします。 • 自動更新: パブリックオファーのみが対象です。現在の契約が終了する時点で新しい契約が作成される場合。購入者は自動更新を有効にしています。 • 新規: 購入者が (オファーを使用して) 条件に同意し、新しい契約をサブスクライブした場合。
以前の契約 ID	本契約が更新またはアップグレードされた場合の契約リファレンス。
次の契約 ID	現在の契約が更新またはアップグレードされた場合の、次の契約の契約 ID。
以前のオファー ID	現在の契約が更新またはアップグレードされた場合の、以前の契約に関連付けられたオファー ID。

列	[Description] (説明)
次のオファー ID	現在の契約が更新またはアップグレードされた場合の、次の契約に関連付けられたオファー ID。
次の契約 ID の承認日	購入者が次の契約を承諾した日付。
提案者 AWS アカウント ID	製品またはサービスの所有者の識別子。
提案者の会社名	製品またはサービスの所有者の商号。
ISV AWS アカウント ID	販売者の AWS 識別子。
ISV の会社名	販売者のビジネス名。
契約終了 (日単位)	QuickSight 内の計算フィールド。本日から契約が終了するまでの残り日数。
契約終了期間	QuickSight 内の計算フィールド。契約が終了するまでの期間。

使用状況ダッシュボード

使用状況ダッシュボードは、SaaS およびサーバー使用状況ベースの製品を使用しているお客様に可視化ときめ細かなデータを提供します。AWS Marketplace 販売者は、このダッシュボードを使用して、使用状況ベースの製品全体の顧客消費を追跡し、製品サポート、料金設定、パブリックオファーからプライベートオファーへの変換、製品の中止に関する意思決定を行うことができます。このダッシュボードには、過去 6 か月間のデータが表示されます。この期間はローリングウィンドウです。

ダッシュボードを開くには、にサインインし [AWS Marketplace 管理ポータル](#)、インサイトを選択し、セールスオペレーションを選択し、使用状況タブを選択します。

Note

このダッシュボードには、オファー作成時に指定されたすべてのディメンションキーの使用状況が表示されます。例えば、Amazon マシンイメージ (AMI) 製品の場合、ディメンションキーはインスタンスタイプであり、オファーで指定されたすべてのインスタンスタイプは、料金が 0 USD であっても、このダッシュボードに表示されます。オファーの作成時に指定さ

れていなかったディメンションの製品使用状況を確認するには、必要なディメンションを含むように製品を再公開することを検討します。

トピック

- [使用状況ダッシュボードの更新頻度](#)
- [セクション 1: コントロール](#)
- [セクション 2: フィルター](#)
- [セクション 3: メトリクス](#)
- [セクション 4: トレンド](#)
- [セクション 5: 内訳](#)
- [セクション 6: 詳細なデータ](#)

使用状況ダッシュボードの更新頻度

請求済み収益ダッシュボードは毎日午後 4 時 (UTC 時間の午前 0 時) に更新されます。使用状況データは上流のデータソースから受信されるため、遅延が発生する可能性があることに注意してください。使用日と使用状況報告日を参照して、使用がなされた時期を、ダッシュボード上で報告対象期間との対比で明確にすることができます。

セクション 1: コントロール

ダッシュボードの本セクション内には、使用状況データを絞り込むためのフィルターがあります。例えば、以下のフィルターから選択できます。

Note

フィルタリングの詳細については、[QuickSight ユーザーガイドの「QuickSight でのデータのフィルタリング」](#)を参照してください。QuickSight

コントロールの記述

コントロール名	説明
エンドユーザー会社名	製品を使用しているアカウント名。

コントロール名	説明
エンドユーザー AWS アカウント ID	製品を使用しているアカウントの ID。
エンドユーザー国	製品を使用しているアカウントに関連付けられている 2 文字の国コード。
製品のタイトル	製品のタイトル。
製品コード	製品の計測に使用される、権利が付与された既存の製品コード。この値は、データをレポートと結合したり、Metering Service で AWS Marketplace 提供されているものを参照したりするためにも使用されます。
オファー ID	購入者が署名したオファーの識別子。
オファーの可視性	オファーがパブリック、プライベート、エンタープライズ契約オファーのいずれになるか。
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意の契約データフィードリファレンス。
ディメンションキー	製品の使用状況に関連するリソースタイプ。ディメンションキーは SaaS およびサーバー使用量ベースの製品に適用されます。
サブスクライバーの会社名	製品をサブスクライブしているアカウント名。
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられている 2 文字の国コード。
リセラーの会社名	製品製造元の製品の販売を認可されているリセラーアカウントの名前。

コントロール名	説明
リセラー AWS アカウント ID	ISV から卸売で製品またはサービスを購入して顧客に再販したアカウントの ID。
再販承認 ID	ISV から卸売で製品またはサービスを購入して顧客に再販したアカウントの ID。
CPPO フラグ	チャネルパートナーのプライベートオファーを使用して契約が締結されたかどうかを示す「はい/いいえ」のフラグ。「はい」の場合、登録販売者がチャネルパートナーです。「いいえ」の場合、記録上の販売者は製品製造元 (独立系ソフトウェアベンダー) です。

セクション 2: フィルター

ダッシュボードの本セクションには、使用日に基づいてレコードを絞り込むためのフィルターがあります。これらのフィルターで選択された値により、メトリクス、トレンド、内訳、詳細データセクションに表示されるデータが更新されます。デフォルトの選択では、過去 6 か月間の使用量のデータを取得するようになっています。

セクション 3: メトリクス

ダッシュボードの本セクションには、推定使用単位数、使用のある顧客、使用量の多い製品などの、使用に関するメトリクスを視覚化する主要業績評価指標 (KPI) が表示されます。これらのフィルターセクションの使用日条件を更新して、日付範囲を更新できます。主要メトリクスにはすべての単位タイプのデータが表示されることに注意してください。

セクション 4: トレンド

ダッシュボードの本セクションには、指定した日付範囲の使用状況のトレンドが表示されます。日ごと、前月比、前四半期比、前年比など、指定した日付の集計別にトレンドを表示して、使用状況に関するインサイトを得ることができます。また、使用単位タイプを選択して、その使用状況のトレンドをグラフィカルに表示することもできます。

セクション 5: 内訳

ダッシュボードの本セクションには、会社名、製品名、ディメンションキー、選択した単位タイプのオファー ID についてのビジネスの推定使用状況メトリクスが表示されます。また、表示するエントリ数も選択できます。

セクション 6: 詳細なデータ

ダッシュボードの本セクションには、使用状況、オファー、製品、サブスクライバー、支払者、エンドユーザー、再販認可、リセラー、独立系ソフトウェアベンダー (ISV) に関する詳細なデータが表示されます。詳細データテーブルには、すべての単位タイプのデータが表示されることに注意してください。

月末に請求が確定するまでは、収益は推定値とみなす必要があります。使用料ベースの請求書は、当月分の使用量について、翌月の 2 日または 3 日に購入者に提示されます (例えば、11 月 1 日から 11 月 30 日までの使用量を計測した顧客には、12 月 2 日または 12 月 3 日に使用量の請求書が提示されます)。計測された使用量は、実際の使用日の数日後にこのダッシュボードに届く場合があるため、使用日と、使用量の報告日は異なる場合があります。つまり、当月の最新情報を把握するために、1 か月を通して毎日ダッシュボードにアクセスしなければならない場合があります。権威のある顧客請求情報については、[財務業務] タブの「[請求収益ダッシュボード](#)」を参照してください。

Note

フィルタリングの詳細については、QuickSight ユーザーガイドの[ビジュアルからのデータのエクスポート](#)を参照してください。

詳細なデータの説明

列	[Description] (説明)
使用日	顧客が製品を使用した日付。
使用状況レポート日	顧客の製品使用状況がインサイトダッシュボードに表示された日付。
エンドユーザー会社名	製品を使用しているアカウント名。
エンドユーザー AWS アカウント ID	製品を使用しているアカウントの ID。

列	[Description] (説明)
エンドユーザー E メールドメイン	製品を使用しているアカウントに関連付けられた E メールドメイン。例えば、E メールアドレスが abc@example.com の場合、エントリは example.com です。
エンドユーザー市区町村	製品を使用したアカウントに関連付けられている市区町村。
エンドユーザーの都道府県またはリージョン	製品を使用したアカウントに関連付けられていると都道府県または地域。
エンドユーザー国	製品を使用しているアカウントに関連付けられている 2 文字の国コード。
エンドユーザー郵便番号	製品を使用しているアカウントに関連付けられた請求先住所の郵便番号。
製品のタイトル	製品のタイトル。
レガシーの製品 ID	製品のレガシーの一意の識別子。
製品 ID	製品のわかりやすい一意の識別子。
製品コード	製品の計測に使用される、権利が付与された既存の製品コード。この値は、データをレポートに結合する場合や、AWS Marketplace Metering Serviceで提供される情報を参照する場合にも使用されます。
オファー ID	購入者が署名したオファーの識別子。
オファー名	販売者が定義したオファー名。
オファーの可視性	オファーがパブリック、プライベート、またはエンタープライズ契約オファーのいずれになるか。

列	[Description] (説明)
契約 ID	製品の使用を開始するために提案者と承諾者の間で締結された、契約書の一意的な契約データフィードリファレンス。
契約受理日	顧客が製品をサブスクライブした日時 (UTC) のタイムスタンプ。
契約開始日	顧客の製品サブスクリプションが開始された日付 (UTC) のタイムスタンプ。将来の日付の契約の場合、この日付は受理日とは異なる場合があります。
契約終了日	契約が失効する日付 (UTC)。計測制/従量制料金サブスクリプションの場合、この日付は 9999 年 1 月 1 日 12:00 AM に設定されます。
ディメンションキー	製品の使用状況に関連するリソースタイプ。ディメンションキーは SaaS およびサーバー使用量ベースの製品に適用されます。
リージョン	Amazon EC2 インスタンスを購入者がデプロイしたリージョン。
推定使用量	製品について記録された使用量。
使用単位タイプ	使用量が記録される単位タイプ。
予想収益	製品の使用量からの予想収益。月末に請求が確定するまでは、収益は推定値とみなす必要があります。使用量ベースの請求書は、当月分の使用量が、翌月の 2 日または 3 日に購入者に提示されます。
通貨	取引の通貨。例えば、取引が米ドル建ての場合、エントリは USD です。
サブスクライバーの会社名	製品をサブスクライブしているアカウント名。

列	[Description] (説明)
サブスクライバー AWS アカウント ID	製品をサブスクライブしているアカウントの ID。
サブスクライバーの E メールドメイン	製品をサブスクライブしているアカウントに関連付けられた E メールドメイン。例えば、E メールアドレスが abc@example.com の場合、エントリは example.com です。
サブスクライバーの市町区村	製品をサブスクライブしているアカウントに関連付けられた請求先住所の市町区村。
サブスクライバーの州または地域	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある都道府県
サブスクライバーの国	製品をサブスクライブしているアカウントに関連付けられた請求先住所がある国
サブスクライバーの郵便番号	製品をサブスクライブしているアカウントに関連付けられた請求先住所の郵便番号。
支払人の会社名	製品の支払いを行っているアカウント名。
支払者 AWS アカウント ID	製品の支払いを行っているアカウントの ID。
支払人の E メールドメイン	製品の支払いを行っているアカウントに関連付けられた E メールドメイン。例えば、E メールアドレスが abc@example.com の場合、エントリは example.com です。
支払人の市町区村	製品の支払いを行っているアカウントに関連付けられた請求先住所の市町区村。
支払人の都道府県または地域	製品の支払いを行っているアカウントに関連付けられた請求先住所がある都道府県
支払人の国	製品の支払いを行っているアカウントに関連付けられた請求先住所がある国

列	[Description] (説明)
支払人の郵便番号	製品の支払いを行っているアカウントに関連付けられた請求先住所の郵便番号。
リセラーの会社名	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウント名。
リセラー AWS アカウント ID	ISV から卸売コストで製品またはサービスを購入して顧客に再販したアカウントの ID。
再販承認 ID	登録された再販オポチュニティの一意の識別子。
再販承認名	登録された再販オポチュニティの一意の名前。
再販承認の説明	登録された再販オポチュニティの説明。
CPPO フラグ	チャネルパートナーのプライベートオファーを使用して契約が締結されたかどうかを示す「はい/いいえ」のフラグ。「はい」の場合、登録販売者がチャネルパートナーです。「いいえ」の場合、記録上の販売者は製品メーカー (ISV) です。
ISV の会社名	製品またはサービスの所有者の名前。
ISV AWS アカウント ID	製品またはサービスの所有者の識別子。

マーケティング用ダッシュボード

AWS Marketplace には、マーケティングデータの追跡に役立つ複数のダッシュボードが用意されています。

トピック

- [顧客契約ダッシュボード](#)
- [出品パフォーマンスダッシュボード](#)
- [検索パフォーマンスダッシュボード](#)

- [AWS ダッシュボードでの購入について](#)

顧客契約ダッシュボード

顧客契約ダッシュボードには、の契約と製品をサブスクライブする顧客の概要が表示されます AWS Marketplace。ダッシュボードには、新規およびアクティブな契約に関するデータに加えて、顧客の傾向とプロフィールが表示されます。

Note

- このダッシュボードを解除するには、[AWS Marketplace Seller Prime](#) プログラムに登録する必要があります。
- このダッシュボードを開くには、にサインインし[AWS Marketplace 管理ポータル](#)、インサイト、マーケティングを選択し、顧客契約タブを選択します。

AWS Marketplace ダッシュボードの使用の詳細については、このセクションの前半の[販売者ダッシュボード](#)「」を参照してください。

トピック

- [セクション 1: フィルター](#)
- [セクション 2: 日付フィルターの詳細](#)
- [セクション 3: パブリックオファー契約とプライベートオファー契約](#)
- [セクション 4: 顧客メトリクス](#)

セクション 1: フィルター

次のフィルターを使用すると、契約データを絞り込むことができます。

コントロール名	説明
製品のタイトル	製品のタイトル。
日付フィルター	過去 30、60、90 日間のデータ、過去 12 か月 (TTM)、および年初来 (YTD) のデータが含まれ

コントロール名	説明
	ます。[カスタム] を選択して、特定の開始日と終了日を定義します。
顧客業界	販売者製品をサブスクライブした顧客に対してが AWS 定義する業界。
顧客セグメント	販売者製品をサブスクライブした顧客に対してが AWS 定義するセグメント。

フィルタリングの詳細については、「Amazon [QuickSight ユーザーガイド](#)」の「[QuickSight でのデータのフィルタリング](#)」を参照してください。Amazon QuickSight

セクション 2: 日付フィルターの詳細

この日付フィルターは、カスタマー契約ダッシュボードのすべてのメトリクスに適用されます。

例えば、デフォルトの [年の初めから今まで] のフィルター値を使用すると、次のメトリクスが表示されます。

メトリクス	説明
パブリックオファー契約数	オファーの可視性をパブリックオファーとし、YTD 期間中に少なくとも 1 日以上アクティブな契約数。
プライベートオファー契約数	オファーの可視性をプライベートオファーとし、YTD 期間中に少なくとも 1 日以上アクティブな契約数。
新しいパブリックオファー契約数	オファーの可視性をパブリックオファーとし、YTD 期間内に承認日がある契約数。
新しいプライベートオファー契約数	オファーの可視性をプライベートオファーとし、YTD 期間内に承認日がある契約数。
アクティブなお客様	YTD 内で少なくとも 1 つのアクティブな契約を持つ、サブスクライバー AWS アカウント ID

メトリクス	説明
	として識別される一意の顧客の数。アクティブな契約は、特定の期間に 少なくとも 1 日以上アクティブな日がある契約として定義されます。
新規有料顧客	YTD 内に最初の請求月がある、サブスクライバー AWS アカウント ID として識別される一意の顧客の数。

セクション 3: パブリックオファー契約とプライベートオファー契約

ダッシュボードのこのセクションには、契約の概要が表示されます。重要業績評価指標 (KPI) には、パブリックオファー契約数、プライベートオファー契約数、新しいパブリックオファー契約数、新しいプライベートオファー契約数が含まれます。前年比または期間比の変化を数量と割合で確認できます。これらのフィルターセクションの日付フィルタ - を更新すると日付範囲を更新できます。

契約は、提案者 (製品またはサービスの所有者) と承認者 (顧客) の間で署名された契約で、製品の使用を開始します。

Note

新しい契約メトリクスには、アクティブ、期限切れ、キャンセル、終了の各契約が含まれます。

以下の表では、契約の状態の一覧と説明が示されています。

ステータス	説明
アクティブ	アクティブ - 契約内容の一部またはすべてが有効な状態です。
失効	契約が事前に合意された終了日に終了した状態です。

ステータス	説明
キャンセル	承認者が、終了日前に契約を終了することを選択した状態です。
終了	支払不履行などの理由で AWS から終了を催促されたため、契約が終了日前に終了した状態です。
更新済み	自動更新機能などを使用して契約が新しい契約に更新された状態です。
置き換え済み	契約が代替オファーを使用して置き換えられた状態です。

セクション 4: 顧客メトリクス

ダッシュボードの本セクションには、指定した日付範囲の顧客傾向が表示されます。KPI には、アクティブな顧客数、新規有料顧客数が含まれます。

- 月別傾向 – グラフ内のフィルターに基づいて、アクティブな顧客の数と新規有料顧客の数の月別ビューを提供します。
- 顧客メトリクス – 選択したメトリクスフィルターに基づいて、アクティブな顧客数と新規有料顧客の数に関するデータを提供します。顧客セグメントまたは業界を選択して、それぞれが顧客メトリクスの合計にどのように貢献するかを理解できます。

Note

契約にはステータスが指定されます。ステータスの詳細については、[前のセクションの表](#)を参照してください。

以下の表では、メトリックスの一覧と説明を示しています。

メトリクス	説明
パブリックオファー契約	パブリックオファーの可視性を持つ契約 ID として特定される契約数。契約の長さは、選択した日付範囲と少なくとも 1 日重複します。長さは、契約開始日と終了日の間で特定されます。
プライベートオファー契約	プライベートオファーの可視性を持つ契約 ID として特定される契約数。契約の長さは、選択した日付範囲と少なくとも 1 日重複します。長さは、契約開始日と終了日の間で特定されません。
新しいパブリックオファー契約	パブリックオファーの可視性を持つ契約 ID として識別される契約数。契約の承認日は選択した日付範囲内です。
新しいプライベートオファー契約	プライベートオファーの可視性を持つ契約 ID として識別される契約数。契約の承認日は選択した日付範囲内です。
アクティブなお客様	有効な契約が 1 つ以上あるサブスクライバー AWS アカウント IDs として識別された顧客の数。契約期間が、選択した日付範囲と少なくとも 1 日重複している
新規有料顧客	選択した日付範囲内に最初の請求月があるサブスクライバー AWS アカウント IDs として識別された顧客の数。

契約の詳細については、このセクションの「[契約および更新ダッシュボード](#)」および「[請求済み収益ダッシュボード](#)」を参照してください。

出品パフォーマンスダッシュボード

出品パフォーマンスダッシュボードには、AWS Marketplace の出品の概要と詳細なデータが表示されます。ダッシュボードには、トラフィックボリューム、およびお客様が製品をサブスクリプション

登録するために実行する手順に関するデータが表示されます。ダッシュボードには、マーケティングチャネル別のトラフィックの詳細な内訳も表示されます。

 Note

- このダッシュボードを解除するには、[AWS Marketplace Seller Prime](#) プログラムに登録する必要があります。
- このダッシュボードを開くには、にサインインし[AWS Marketplace 管理ポータル](#)、インサイト、マーケティングを選択し、パフォーマンスの一覧表示タブを選択します。

AWS Marketplace ダッシュボードの使用の詳細については、このセクションの前半の[販売者ダッシュボード](#)「」を参照してください。

トピック

- [セクション 1: フィルター](#)
- [セクション 2: 日付フィルターの詳細](#)
- [セクション 3: 概要メトリクス](#)
- [セクション 4: 毎月の傾向とファネル変換](#)
- [セクション 5: チャネル別のトラフィックの傾向](#)
- [セクション 6: 毎月のトラフィックと契約メトリクス](#)
- [セクション 7: ユニークビジターのウェブトラフィックソース](#)
- [セクション 8: ウェブトラフィックソースの内訳](#)

セクション 1: フィルター

次のフィルターを使用してデータを絞り込むことができます。

コントロール名	説明
製品のタイトル	製品のタイトル。
日付	過去 30 日、60 日、90 日、過去 12 か月 (TTM)、および年初来 (YTD) が含まれます。カスタムを選択して、特定の開始日と終了日を定義できます。

フィルタリングの詳細については、[QuickSight ユーザーガイドの「QuickSight でのデータのフィルタリング」](#)を参照してください。QuickSight

セクション 2: 日付フィルターの詳細

この日付フィルターは、出品パフォーマンスダッシュボードのすべてのメトリクスに適用されます。

例えば、デフォルトの年初来フィルター値を使用すると、次のメトリクスが表示されます。

メトリクス	説明
ユニークビジター	YTD のリスト AWS Marketplace ページにアクセスした一意の個人の数。
新しいパブリックオファー契約	オファーの可視性をパブリックオファーとし、YTD 期間中に少なくとも 1 日以上アクティブな契約数。
新しいユニークビジター	YTD で初めて AWS Marketplace 出品ページにアクセスした一意の個人の数。
再訪したユニークビジター	選択した年の初め以前に AWS Marketplace 出品ページを訪問し、YTD で再度訪問した一意の個人の数。

セクション 3: 概要メトリクス

ダッシュボードのこのセクションには、トラフィックとパブリックオファー契約の概要が表示されます。重要業績評価指標 (KPI) には、ユニークビジター数、新しいパブリックオファー契約の数、新しいユニークビジターの数、および再訪したユニークビジターの数が含まれます。前年比または期間比の変化を数量と割合で確認できます。フィルターセクションの日付フィールドの日付条件を更新して、日付範囲を更新できます。

メトリクス	説明
ユニークビジター	AWS Marketplace 出品ページにアクセスした一意の個人の数。

メトリクス	説明
契約	プロポーザー (製品またはサービスの所有者) とアクセプター (お客様) の間で署名された製品の使用を開始するための契約。
新しいパブリックオファー契約	選択された日付範囲内で、アクティブ、期限切れ、キャンセル、終了などのステータスを含む承諾日を持つオファー開始性としてのパブリックオファーを含む契約の数。
ステータス	アクティブ - 契約の条項の一部またはすべてが有効な状態です。期限切れ - 契約が事前に合意された終了日に終了した状態です。キャンセル済み - 承認者が終了日より前に契約を終了することを選択した状態です。終了 - 支払いエラーなど、AWS が開始した終了イベントにより、契約は事前に合意した終了日より前に終了した状態です。更新済み - 自動更新などの機能を使用して、契約が新しい契約に更新された状態です。置き換え済み - 契約が代替オファーを使用して置き換えられた状態です。

セクション 4: 毎月の傾向とファネル変換

ダッシュボードの本セクションには、指定した日付範囲のトラフィックと同意に関するトレンドが表示されます。重要業績評価指標には、新しいパブリックオファー契約を閲覧したユニークビジターと新しい契約の数が含まれます。月別のユニークビジター - 月別のユニークビジター数を示します。1人のユニークビジターが複数の月に訪問できるため、月別のユニークビジターの合計は、サマリーメトリクスセクションのユニークビジターの合計を超える場合があります。月別の新しい契約 - 月別の新しいパブリックオファー契約数を示します。ファネルコンバージョン - ステップバイステップのコンバージョンビューを提供します。これには、出品ページ、調達ページ、設定ページ、フルフィルメントページが含まれます。バーの表示形式として、ユニークビジター、ページビュー、ユニークアカウントを選択できます。各バーにカーソルを合わせると、現在のバーの値として定義されているコンバージョン率を前のバーの割合として表示できます。

メトリクス	説明
ユニークビジター	AWS Marketplace 出品ページにアクセスした一意の個人の数。
ページビュー	AWS Marketplace 出品ページへのアクセス回数。

メトリクス	説明
ユニークアカウント	訪問者に関連付けられた一意の AWS アカウント ID の数。このメトリクスは、ビジターが認証した後にのみ適用されます。

セクション 5: チャンネル別のトラフィックの傾向

ダッシュボードのこのセクションでは、月別のトラフィック傾向について詳しく説明します。バーの表示形式として、ユニークビジターとページビューを選択できます。チャンネルフィルターで 1 つまたは複数のチャンネルを選択すると、選択したチャンネルの月別傾向を確認できます (複数可)。

セクション 6: 毎月のトラフィックと契約メトリクス

ダッシュボードのこのセクションには、ユニークビジター、新しい契約、コンバージョン率、および合計アグリーメントの月次ビューが表示されます。コンバージョン率は、新しい契約を毎月のユニークビジターで割ったものとして定義されます。

テーブル内のデータは、部分的な月の日付範囲を選択した場合でも、月全体を表します。例えば、2024 年 1 月 15 日から 2024 年 3 月 15 日の期間を表示するようにデータをフィルタリングしても、テーブルには 2024 年 1 月 1 日から 2024 年 3 月 31 日の完全な月次データが含まれます。

テーブル内のデータは月でソートされ、直近の月が最初に表示されます。対応する列ヘッダーをクリックして、使用可能な列のいずれかでテーブルをソートすることを選択できます。オプションでは、データを昇順または降順にソートできます。

セクション 7: ユニークビジターのウェブトラフィックソース

ダッシュボードのこのセクションでは、マーケティングソース別のユニークビジターの詳細を説明します。カテゴリには、販売者の有机的検出、販売者主導のマーケティングキャンペーン、AWS 組織的検出、AWS 主導のキャンペーンが含まれます。これらのカテゴリは、ビジターまたはアクセスに関連付けられた追跡コードによって識別されます。

トラフィックソース	説明
販売者経由のオーガニックディスカバリー	ウェブサイトまたはアプリ内通知のコールトゥアクションと検出ポイントを通じて出品にアクセスしたユニークビジター。

トラフィックソース	説明
販売者主導のマーケティングキャンペーン	有料広告またはプロモーションチャネルを通じて出品にアクセスしたユニークビジター。
AWS 器質的検出	オーガニック AWS ウェブサイトトラフィック、AWS コンソール (複数可) 検索、AWS マーケットプレイス検索などの AWS チャンネルを通じて出品したユニークビジター。
AWS マーケティングキャンペーン	AWS 主導のキャンペーンや有料プロモーションを通じて出品にアクセスしたユニークビジター。

セクション 8: ウェブトラフィックソースの内訳

ウェブトラフィックソースの内訳テーブルは、トラフィックをもたらしたマーケティングアクティビティの詳細な内訳を提供します。

- トラフィックソース - AWS主導型と販売者主導型を 2 つの主要なカテゴリとして含めます。
- トラフィックタイプ - 販売者の有機的検出、販売者主導のマーケティングキャンペーン、AWS 組織的検出、主導 AWSのマーケティングキャンペーンが含まれます。
- チャンネル - 有料表示、有料検索、有料ソーシャルメディア、E メール、オーガニック、内部サイトトラフィックの外部リンクなど、さまざまなマーケティング戦術が含まれます。
- Publisher - Facebook、Google、LinkedIn、Seller-led など AWS、キャンペーンが公開されるエンティティを指します。
- プロモーション - 専用追跡コードが生成された特定のキャンペーンを指します。例として、psm_linked_post_free-trial-global-ver-a が挙げられます。

メトリクス	説明
契約への寄与	各プロモーションからのアクセスが寄与した契約の合計。
コンバージョン率	契約数を各プロモーションからのユニークビジターで割ったもの。

検索パフォーマンスダッシュボード

検索パフォーマンスダッシュボードには、AWS Marketplace 検索エクスペリエンスでの出品のパフォーマンスに関するメトリクスが表示されます AWS Marketplace。ダッシュボードを使用して、出品の検出可能性とパフォーマンスを向上させます。ダッシュボードは、インプレッション数、クリック数、クリック率を追跡します。また、指定したキーワードや顧客が使用するキーワードを追跡し、一般的な顧客検索に基づいてキーワードの提案を提供します。

ダッシュボードからのインサイトは、可視性の向上、キーワードと顧客の好みの一致、ビジネスへのより適格なトラフィックの促進のために製品リストを絞り込むのに役立ちます AWS Marketplace。

ダッシュボードを開くには、AWS Marketplace 管理ポータルを起動し、インサイトタブに移動します。

ダッシュボードへのアクセスの制御の詳細については、このセクションの前半にある[販売者ダッシュボード](#)「」を参照してください。

セクション 1: ダッシュボードへのアクセス

このダッシュボードをロック解除するには、[AWS Marketplace Seller Prime Program](#) のメンバーである必要があります。

Seller Prime プログラムに登録すると、販売者アカウントのシステム管理者は の Insights タブを使用して、検索パフォーマンスダッシュボード AWS Marketplace 管理ポータル を開始できます。

システム管理者は、AWS Identity and Access Management (IAM) ポリシーを作成して、販売会社の他のユーザーに特定のダッシュボードへのアクセスを提供することもできます。

Note

2023 年 9 月現在、レガシー IAM アクセス許可で有効になっている販売者ダッシュボードへのアクセスはサポートされなくなりました。次の例に示すように、新しい Amazon リソースネーム (ARN) 形式を使用して、IAM アクセス許可を更新します。

ポリシーの作成の詳細については、「AWS Identity and Access Management ユーザーガイド」の[「IAM ポリシーの作成」](#)を参照してください。

ポリシーを使用してアクセスを制御する

次のいずれかのポリシーを使用して、リストパフォーマンスダッシュボードへのアクセスを提供します。

次の例では、現在または将来のデータフィードに関係なく、ダッシュボードやレポートを含む現在および将来のすべての AWS Marketplace リソースへのアクセスを提供します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

次の例では、ARN を含めることで検索パフォーマンスダッシュボードへのアクセスを提供します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/SearchPerformance_V1"
    ]
  }]
}
```

セクション 2: フィルター

ダッシュボードのこのセクションには、次のフィルターがあります。これらを使用してデータを絞り込みます。

コントロール名	説明
製品のタイトル	製品のタイトル。 このフィルターは、インプレッション、クリック、クリックスルー率、検索キーワードのレコ

コントロール名	説明
	メンデーションに影響します。製品カテゴリ別上位検索キーワードテーブルには影響しません。
日付フィルター	過去 30、60、90 日、過去 12 か月 (TTM)、および年初来 (YTD) でフィルタリングできます。特定の開始日と終了日を使用するようにカスタム を選択できます。 このフィルターは、インプレッション、クリック、クリックスルー率に影響します。AWS Marketplace 検索キーワードのレコメンデーションや製品カテゴリテーブルで検索された上位キーワードには影響しません。

フィルタリングの詳細については、[Amazon QuickSight ユーザーガイド](#)の「[Amazon QuickSight でのデータのフィルタリング](#)」を参照してください。Amazon QuickSight

セクション 3: AWS Marketplace 検索パフォーマンス

ダッシュボードのこのセクションには、AWS Marketplace リストの検出可能性とパフォーマンスに関する主要業績評価指標 (KPIs) が表示されます。ダッシュボードには、インプレッション数、クリック数、クリック率の KPIs が表示されます。

ボリュームと割合に、year-over-yearまたはperiod-over-periodの変化を確認できます。日付範囲を変更することもできます。

メトリクス	説明
インプレッション	上位 20 件の検索結果にリストが表示される回数 AWS Marketplace。
クリック数	ユーザーが AWS Marketplace 検索結果からリストをクリックした回数。
クリックスルー率 (CTR)	クリック数とインプレッション数の比率。検索結果から顧客を引き付けるために出品がどの程度効果的かを示す指標。

セクション 4: インプレッション、クリック、クリックスルー率の月次傾向

ダッシュボードのこのセクションには、特定の日付範囲の月次傾向が表示されます。インプレッション数、クリック数、クリック率KPIs を提供します。

セクション 5: キーワードレコメンデーションAWS Marketplace を検索する

ダッシュボードのこのセクションには、現在のキーワードと、各リストの推奨キーワードを一覧表示する表が表示されます。

列名	説明
製品のタイトル	製品のタイトル。リンクをクリックすると、AWS Marketplace 管理ポータル製品の製品概要ページに移動し、出品を管理および編集できます。
現在のキーワード	AWS Marketplace 管理ポータルの製品情報の更新セクションで指定したキーワード。
上位のキーワード	上位 5 つの顧客が製品を検索するキーワード。
AWS推奨キーワード	顧客が類似製品を検索するときに最もよく使用するキーワード。
推定トラフィック増加率	推奨キーワードを採用すると、トラフィックが増加する可能性があります。日付フィルターの影響を受けません。

Note

キーワードは重要ですが、その他の要因が検索結果に影響を与える可能性があります。

- 出品の最適化
 - 短い説明や長い説明など、リストの他の部分を最適化します。
 - 無料トライアルなどの機能を提供することを検討して、出品者の興味を高めてください。
- キーワードと出品およびターゲット顧客とのマッチング
 - リストを頻繁に使用するキーワードと可能な限り一致させます。
 - 製品とその機能に関連するキーワードのみを使用してください。そうしないと、リストが多数の検索に表示されることがありますが、選択するユーザーが少なくなります。
- キーワードのバリエーションについて

- 「git」や「gitops」などの同様の用語は、同じ検索用語として扱われる場合があることに注意してください。
- バリエーションの関連性を評価し、それに応じてキーワードを調整します。
- キーワードの更新
 1. で AWS Marketplace 管理ポータル、製品メニューを開き、製品カテゴリを選択します。
 2. 製品へのリンクを選択し、製品情報タブを選択して、製品情報の提供ページに移動します。
 3. AWS Marketplace 検出可能性セクションまでスクロールし、AWS Marketplace 検索結果のキーワードを検索して、キーワードを調整します。

セクション 6: 製品カテゴリ別に検索された上位キーワード

類似製品の各製品カテゴリの上位 5 つのカスタマー検索キーワード。このダッシュボードの他の製品および日付フィルターは、このデータには影響しません。

AWS ダッシュボードでの購入について

ダッシュボードでの購入 AWS には、call-to-action ボタンで購入 AWS を選択し、調達での購入 AWS ページにアクセスする顧客が作成したウェブトラフィック、エンゲージメント、契約の概要が表示されます AWS Marketplace。

ダッシュボードへのアクセスの詳細については、このガイドの前半の[販売者ダッシュボード「」](#)を参照してください。

トピック

- [セクション 1: フィルター](#)
- [セクション 2: 概要メトリクス](#)
- [セクション 3: ページビューと契約の合計クリック数と傾向データ](#)
- [セクション 4: AWS サインインページのドロップオフ率](#)
- [セクション 5: 調達で AWS の購入ページのエンゲージメント](#)
- [セクション 6: AWS 調達での購入ページのエンゲージメントメトリクスの傾向データ](#)

セクション 1: フィルター

このセクションでは、以下のデータフィルターについて説明します。

フィルター名	説明
製品のタイトル	製品のタイトル。
製品 ID	ソフトウェア製品のわかりやすい一意の識別子。
日付フィルター	過去 30、60、または 90 日間、過去 12 か月、または過去 1 年でフィルタリングします。Custom を選択して、他の開始日と終了日を入力します。

フィルタリングの詳細については、[QuickSight ユーザーガイドの「QuickSight でのデータのフィルタリング」](#)を参照してください。QuickSight

セクション 2: 概要メトリクス

このセクションでは、Buy with AWS call-to-action ボタンと Buy with AWS procurement ページに関する情報を表示します。

以下の表では、メトリックスの一覧と説明を示しています。日付範囲を更新するには、フィルターセクションの日付条件を変更します。

メトリクス	説明
ボタンの合計クリック数	販売者ウェブサイトの AWS での購入など、call-to-action ボタンのクリックの合計数。
AWS ページビューでの購入数	調達での購入 AWS ページへの訪問の合計数。これには、繰り返しの訪問が含まれます。
Buy with AWS page からの契約の合計	調達での購入 AWS ページで作成された契約の合計数。

セクション 3: ページビューと契約の合計クリック数と傾向データ

このセクションでは、指定した日付範囲の毎月または毎週の傾向を示します。次の表に、KPIs。

メトリクス	説明
ボタンの合計クリック数	販売者ウェブサイトのcall-to-actionボタンの合計クリック数。
ページビュー	合計ページは、AWS 調達での購入ページに移動します。リストでデータを表示を使用して、月次ビューと週次ビューのどちらかを選択します。
契約の合計	AWS 調達での購入ページで作成された契約の合計。リストでデータを表示を使用して、月次ビューと週次ビューのいずれかを選択します。

セクション 4: AWS サインインページのドロップオフ率

このセクションには、AWS サインインページのドロップオフ率が表示されます。レートは、サインインし、契約を作成したユーザーと、サインインのみを行ったユーザーの数の比率です。リストでデータを表示を使用して、週または月でフィルタリングします。

セクション 5: 調達で AWS の購入ページのエンゲージメント

このセクションでは、指定した日付範囲の概要メトリクスと傾向データを示します。次の表に、メトリクスの一覧と説明を示します。フィルターセクションの日付フィールドを使用して、日付範囲を更新します。

メトリクス	説明
ユニークビジターの総数	一意の調達ページのユーザーの総数。
バウンス率	1 つのサイトページを表示した後にサインインして退出するユーザーと、複数のページを訪問するユーザーの比率。

メトリクス	説明
分単位の平均ユーザードウェル時間	ユーザーがサインインしてから、そのユーザーがページを離れるまでの時間。

セクション 6: AWS 調達での購入ページのエンゲージメントメトリクスの傾向データ

グラフには、[セクション 5](#) のメトリクスの毎月または毎週の傾向が表示されます。リストでデータを表示を使用して、毎月のビューと毎週のビューのいずれかを選択します。

AWS Marketplace Vendor Insights

AWS Marketplace Vendor Insights は、組織が実行するソフトウェアリスク評価を簡素化して、信頼できるソフトウェアを調達し、基準を満たすようにする機能です。AWS Marketplace Vendor Insights を使用すると、購入者は単一のコンソールから製品のセキュリティプロファイルをほぼリアルタイムでモニタリングできます。AWS Marketplace Vendor Insights を使用すると、購入者の調達プロセスが簡単になり、販売者の売上が増加する可能性があります。ソフトウェア製品のセキュリティとコンプライアンス情報のダッシュボードを提供することにより、購入者の評価にかかる労力を低減できます。

AWS Marketplace Vendor Insights ダッシュボードのすべてのセキュリティおよびコンプライアンス情報は、次のソースから収集された証拠に基づいています。

- AWS Marketplace Vendor Insights のセキュリティ自己評価とコンセンサス評価イニシアチブアンケート (CAIQ) を含む、販売者の自己認証
- 業界標準の監査報告書 (例えば、国際標準化機構 ISO 27001)
- 販売者の本番環境からの証拠収集を自動化する、AWS Audit Manager

AWS Marketplace Vendor Insights は、製品に関するコンプライアンスアーティファクトとセキュリティコントロール情報を収集し、ダッシュボードに表示します。このダッシュボードは、販売者の自己評価から得たデータ、監査レポートからの証拠、および AWS アカウントから得たライブエビデンスを取得します。このデータはセキュリティコントロールにフィードされ、次に購入者が確認できるようにダッシュボードにフィードされます。ライブエビデンスとは、複数のソースからのデータを常に更新して最新の情報を表示する方法です。AWS Config は、販売者の環境では有効になっています。設定、有効になっているバックアップ、その他の情報に関するデータは自動的に更新されます。例えば、製品のアクセスコントロールが [準拠] していて、Amazon Simple Storage Service (Amazon S3) バケットが公開されたとします。ダッシュボードには、コントロールのステータスが [準拠] から [未定] に変わったことが表示されます。

AWS Marketplace Vendor Insights を使用する AWS アカウント 前に、[ベースラインリソースとインフラストラクチャ](#)を設定する必要があります。セットアップが完了すると、AWS Marketplace Vendor Insights は情報を収集し、Software as a Service (SaaS) 製品のセキュリティプロファイルを生成できます AWS Marketplace。

内容

- [AWS Marketplace Vendor Insights について](#)

- [AWS Marketplace Vendor Insights のセットアップ](#)
- [AWS Marketplace Vendor Insights プロファイルの表示](#)
- [AWS Marketplace Vendor Insights でのスナップショットの管理](#)
- [AWS Marketplace Vendor Insights でのアクセスの制御](#)

AWS Marketplace Vendor Insights について

AWS Marketplace Vendor Insights は、製品のコンプライアンスアーティファクトとセキュリティコントロール情報を収集し、ダッシュボードに表示します。このダッシュボードは、製品所有者の自己評価から得たデータ、監査レポートからの証拠、および AWS アカウントから得たライブエビデンスを取得します。このデータはセキュリティコントロールにフィードされ、次に購入者が確認できるようにダッシュボードにフィードされます。

ダッシュボードには、複数のセキュリティコントロールカテゴリから AWS Marketplace Vendor Insights によって収集された証拠ベースの情報が表示されます。これにより、セキュリティプロファイルをほぼリアルタイムで確認できるため、購入者と販売者の間の話し合いが減ります。購入者は、評価を完了する販売者の情報を数時間以内に検証できます。AWS Marketplace Vendor Insights は、販売者がセキュリティおよびコンプライアンス体制の情報を自動的に up-to-date 状態に保つためのメカニズムを提供します。購入者とオンデマンドで情報を共有できるため、ランダムにアンケートに回答する必要がなくなります。

AWS Marketplace Vendor Insights は、次の 3 つのソースから証拠ベースの情報を収集します。

- ベンダーの自己評価 – サポートされている自己評価には、AWS Marketplace Vendor Insights のセキュリティ自己評価とコンセンサス評価イニシアチブアンケート (CAIQ) が含まれます。
- 本番環境アカウント - 複数のコントロールのうち、25 のコントロールが本番環境アカウントからのライブエビデンスの収集をサポートしています。各コントロールのライブ証拠は、1 つ以上の AWS Config ルールを使用して AWS リソースの設定を評価することによって生成されます。は証拠を AWS Audit Manager キャプチャし、AWS Marketplace Vendor Insights が使用できるように準備します。オンボーディング AWS CloudFormation テンプレートは、ライブ証拠収集を有効にするために必要な前提条件のステップを自動化します。AWS Config は販売者の環境で有効になっています。設定、有効になっているバックアップ、その他の情報に関するデータは自動的に更新されます。例えば、製品のアクセスコントロールが [準拠] していて、Amazon S3 バケットが公開されたとします。ダッシュボードには、コントロールのステータスが [準拠] から [未定] に変わったことが表示されます。
- AWS Config と AWS Audit Manager サービスをオンにします。

- AWS Config ルールと AWS Audit Manager 自動評価の作成。
- AWS Marketplace Vendor Insights が評価結果を取得できるように AWS Identity and Access Management (IAM) ロールをプロビジョニングします。
- [ISO 27001 および SOC2 Type II レポート] - コントロールカテゴリは、国際標準化機構 (ISO) レポートおよび System and Organization Controls (SOC2) レポート内のコントロールにマッピングされます。これらのレポートを AWS Marketplace Vendor Insights と共有すると、Vendor Insights はこれらのレポートから関連する証拠を抽出してダッシュボードに表示できます。

AWS Marketplace Vendor Insights のセットアップ

次の手順では、Software as a Service (SaaS) リストで AWS Marketplace Vendor Insights を設定するための大まかな手順について説明します。

SaaS 出品で AWS Marketplace Vendor Insights を設定するには

1. [the section called “セキュリティプロファイルを作成する”](#).
2. (オプション) [the section called “証明書のアップロード”](#)。
3. [the section called “自己評価をアップロードする”](#).
4. (オプション) [the section called “AWS Audit Manager 自動評価を有効にする”](#)。

セキュリティプロファイルを作成する

セキュリティプロファイルにより、購入者はソフトウェア製品のセキュリティ状況に関する詳細な情報を得ることができます。セキュリティプロファイルは、自己評価、認定、AWS Audit Manager 自動評価など、関連するデータソースを使用します。

Note

作成できるセキュリティプロファイルは数に限りがあります。より多くのセキュリティプロファイルを作成するには、クォータの引き上げをリクエストします。詳細については、「AWS 全般のリファレンス」の「[AWS の Service Quotas](#)」を参照してください。

セキュリティプロファイルを作成するには

1. AWS Marketplace 販売者アカウントにアクセスできる IAM ユーザーまたはロールを使用してサインインします。
2. [製品] を選択し、[SaaS] を選択して [SaaS 製品] ページに移動します。
3. [製品] を選択します。
4. [ベンダーインサイト] タブを選択し、[セキュリティプロファイルの追加をサポートに問い合わせ] を選択します。
5. フォームに入力し、[送信] を選択します。

Seller AWS Marketplace Operations チームがセキュリティプロファイルを作成します。セキュリティプロファイルの作成が完了すると、フォームに記載されている受信者に通知メールが送信されます。

証明書のアップロード

証明書は、複数のディメンションにわたる製品のセキュリティ体制の証拠を提供するデータソースです。AWS Marketplace Vendor Insights は、次の証明書をサポートしています。

- FedRAMP 認定 - 米国政府のクラウドセキュリティ標準への準拠を検証します
- GDPR コンプライアンスレポート - 一般データ保護規則 (GDPR) の要件の遵守、個人データおよび個人のプライバシー権の保護を実証します。
- HIPAA コンプライアンスレポート - 保護対象の医療情報を保護する、「医療保険の相互運用性と説明責任に関する法令 (HIPAA)」の規制に準拠していることを証明します。
- ISO/IEC 27001 監査報告書 - 情報セキュリティ基準に重点を置いた国際標準化機構 (ISO)/国際電気標準会議 (IEC) 27001 への準拠を確認しています
- PCI DSS 監査報告書 - PCI セキュリティ標準審議会が設定したセキュリティ基準への準拠を証明します。
- SOC 2 タイプ 2 監査報告書 - サービス組織統制 (SOC) のデータプライバシーとセキュリティ統制への準拠を確認します。

証明書のアップロードするには

1. [ベンダーインサイト] タブで、[データソース] セクションに移動します。
2. [証明書] で [認定資格をアップロード] を選択します。

3. [認定資格の詳細] で、必要な情報を入力し、証明書をアップロードします。
4. (オプション) [タグ] で、新しいタグを追加します。

 Note

タグの詳細については、[「AWS リソースのタグ付けユーザーガイド」](#)の「AWS リソースのタグ付け」を参照してください。

5. [証明書をアップロード] を選択します。

 Note

証明書は現在のセキュリティプロファイルに自動的に関連付けられます。アップロード済みの証明書を関連付けることもできます。製品詳細ページで、[認定] の下の [認定を関連付ける] を選択し、リストから認定を選択して、[認定を関連付ける] を選択します。証明書をアップロードしたら、製品詳細ページの [証明書のダウンロード] ボタンを使用してダウンロードできます。[認定書の更新] ボタンを使用して認定書の詳細を更新することもできます。

認証の詳細が検証されるまで、認証ステータスは [ValidationPending] に変わります。データソースの処理中と処理後に、別のステータスが表示されます。

- 使用可能 - データソースがアップロードされ、システム検証が正常に完了しました。
- AccessDenied - データソースの外部ソースリファレンスは、AWS Marketplace Vendor Insights が読み取るためにアクセスできなくなりました。
- ResourceNotFound - データソースの外部ソース参照は、ベンダーインサイトが読み取れなくなりました。
- ResourceNotSupported - データソースはアップロードされましたが、提供されたソースはまだサポートされていません。検証エラーの詳細については、ステータスメッセージを参照してください。
- ValidationPending - データソースはアップロードされましたが、システム検証はまだ実行中です。この段階では実行できるアクション項目はありません。ステータスが [利用可能]、[ResourceNotSupported]、または [ValidationFailed] に更新されます。
- ValidationFailed - データソースはアップロードされましたが、1 つ以上の理由でシステム検証が失敗しました。検証エラーの詳細については、ステータスメッセージを参照してください。

自己評価をアップロードする

自己評価は、製品のセキュリティ体制の証拠を提供するデータソースの一種です。AWS Marketplace Vendor Insights は、次の自己評価をサポートしています。

- AWS Marketplace Vendor Insights の自己評価
- コンセンサス評価イニシアティブアンケート (CAIQ) 詳細については、Cloud Security Alliance ウェブサイトの「[What is CAIQ](#)」を参照してください。

自己評価をアップロードするには

1. <https://console.aws.amazon.com/marketplace> で AWS Marketplace コンソールを開きます。
2. [ベンダーインサイト] タブで、[データソース] セクションに移動します。
3. [自己評価] で [自己評価をアップロード] を選択します。
4. [自己評価の詳細] で、次の情報を入力します。
 - a. [名前] - 自己評価の名前を入力します。
 - b. [タイプ] - リストから評価タイプを選択します。

Note

[ベンダーインサイトのセキュリティ自己評価] を選択した場合は、[テンプレートをダウンロード] を選択して自己評価をダウンロードします。スプレッドシートの回答ごとに [はい]、[いいえ]、または [N/A] を選択します。

5. 完了した評価をアップロードするには、[自己評価をアップロード] を選択します。
6. (オプション) [タグ] で、新しいタグを追加します。

Note

タグの詳細については、「[AWS リソースのタグ付け](#)ユーザーガイド」の「AWS リソースのタグ付け」を参照してください。

7. [自己評価をアップロード] を選択します。

 Note

自己評価は現在のセキュリティプロファイルに自動的に関連付けられます。アップロード済みの自己評価を関連付けることもできます。製品詳細ページで、[自己評価] の下の [自己評価を関連付ける] を選択し、リストから自己評価を選択して、[自己評価を関連付ける] を選択します。

自己評価をアップロードしたら、製品詳細ページの [自己評価をダウンロード] ボタンを使用してダウンロードできます。[自己評価の更新] ボタンを使用して自己評価の詳細を更新することもできます。

ステータスは、次のいずれかの値に更新されます。

- 使用可能 - データソースがアップロードされ、システム検証が正常に完了しました。
- AccessDenied - データソースの外部ソース参照は、ベンダーインサイトが読み取れなくなりました。
- ResourceNotFound - データソースの外部ソース参照は、ベンダーインサイトが読み取れなくなりました。
- ResourceNotSupported - データソースはアップロードされましたが、提供されたソースはまだサポートされていません。検証エラーの詳細については、ステータスメッセージを参照してください。
- ValidationPending - データソースはアップロードされましたが、システム検証はまだ実行中です。この段階では実行できるアクション項目はありません。ステータスが [利用可能]、[ResourceNotSupported]、または [ValidationFailed] に更新されます。
- ValidationFailed - データソースはアップロードされましたが、1 つ以上の理由でシステム検証が失敗しました。検証エラーの詳細については、ステータスメッセージを参照してください。

AWS Audit Manager 自動評価を有効にする

AWS Marketplace Vendor Insights は、複数の AWS のサービスを使用して、セキュリティプロファイルの証拠を自動的に収集します。

自動評価には、次の AWS のサービス および リソースが必要です。

- AWS Audit Manager – AWS Marketplace Vendor Insights のセットアップを簡素化するために、必要なリソースのプロビジョニングと設定を行う AWS CloudFormation スタックと StackSets を使

用します。スタックセットは、AWS Configによって自動的に入力されるコントロールを含む自動評価を作成します。

詳細については AWS Audit Manager、[AWS Audit Manager 「ユーザーガイド」](#) を参照してください。

- AWS Config – スタックセットはコン AWS Config フォーマンスパックをデプロイして、必要な AWS Config ルールを設定します。これらのルールにより、Audit Manager の自動評価は、そのに AWS のサービス デプロイされた他の のライブ証拠を収集できます AWS アカウント。AWS Config 機能の詳細については、「[AWS Config デベロッパーガイド](#)」を参照してください。

Note

記録の最初の月は、その後の月と比較して AWS Config 、アカウントでのアクティビティが増加することがあります。最初のブートストラッププロセス中に、 は、 が記録するために選択した AWS Config アカウント内のすべてのリソース AWS Config を確認します。エフェメラルワークロードを実行すると、これらの一時リソースの作成と削除に関連する設定変更が記録され AWS Config するため、 からのアクティビティが増加することがあります。一時的なワークロードとは、必要なときにロードされて実行されるコンピューティングリソースを一時的に使用することです。

エフェメラルワークロードの例としては、Amazon Elastic Compute Cloud (Amazon EC2) スポットインスタンス、Amazon EMR ジョブ AWS Auto Scaling、 などがあります AWS Lambda。エフェメラルワークロードの実行によるアクティビティの増加を回避するには、これらのタイプのワークロードを AWS Config をオフにして別のアカウントで実行できます。このアプローチでは、設定の記録やルール評価が増えるのを防ぐことができます。

- Amazon S3 – スタックセットは、次の 2 つの Amazon Simple Storage Service バケットを作成します。
 - vendor-insights-stack-set-output-bucket-{アカウント番号} - このバケットには、実行されたスタックセットからの出力が含まれています。Seller AWS Marketplace Operations チームは、出力を使用して自動データソース作成プロセスを完了します。
 - vendor-insights-assessment-reports-bucket-{account number} – この Amazon S3 バケットに評価レポート AWS Audit Manager を発行します。評価レポートの公開に関する詳細については、「AWS Audit Manager ユーザーガイド」の「[評価レポート](#)」を参照してください。

Amazon S3 機能でのファイルの保存に関する詳細については、「[Amazon S3 ユーザーガイド](#)」を参照してください。

- IAM – オンボーディングスタックセットは、アカウントで次の AWS Identity and Access Management (IAM) ロールをプロビジョニングします。
- VendorInsightsPrerequisiteCFT.yml テンプレートをデプロイすると、管理者ロール AWSVendorInsightsOnboardingStackSetsAdmin と実行ロール AWSVendorInsightsOnboardingStackSetsExecution が作成されます。スタックセットは管理者ロールを使用して、必要なスタックを複数の AWS リージョン スタックに同時にデプロイします。管理者ロールは、AWS Marketplace Vendor Insights のセットアッププロセスの一環として、必要な親スタックとネストされたスタックをデプロイする実行ロールを引き受けまます。セルフマネージド型アクセス許可の詳細については、「[AWS CloudFormation ユーザーガイド](#)」の「セルフマネージド型のアクセス許可を付与する」を参照してください。
- このAWSVendorInsightsRoleロールは、AWS Marketplace Vendor Insights に AWS Audit Manager リソースの評価を読み取るためのアクセスを提供します。AWS Marketplace Vendor Insights は、AWS Marketplace Vendor Insights プロファイルの評価で見つかった証拠を表示します。
- AWSVendorInsightsOnboardingDelegationRole は、AWS Marketplace Vendor Insights にvendor-insights-stack-set-output-bucketバケット内のオブジェクトを一覧表示および読み取るアクセスを提供します。この機能を使用すると、AWS Marketplace カタログオペレーションチームは AWS Marketplace Vendor Insights プロファイルの設定を支援できます。
- このAWSAuditManagerAdministratorAccessロールは、評価 AWS Audit Manager、コントロール、フレームワークを有効または無効にし、設定を更新し、管理するための管理アクセスを提供します。ユーザーまたはチームがこの役割を引き受け、AWS Audit Managerで自動評価のためのアクションを実行できます。

AWS Audit Manager 自動評価を有効にするには、オンボーディングスタックをデプロイする必要があります。

オンボーディングスタックをデプロイする

AWS Marketplace Vendor Insights のセットアップを簡素化するために、必要なリソースのプロビジョニングと設定を行う AWS CloudFormation Stacks と StackSets を使用します。複数のアカウントまたは複数の AWS リージョン SaaSソリューションがある場合、StackSets を使用すると、一元管理アカウントからオンボーディングスタックをデプロイできます。

CloudFormation StackSets の詳細については、「AWS CloudFormation ユーザーガイド」の「[AWS CloudFormation StackSets の操作](#)」を参照してください。

AWS Marketplace Vendor Insights のセットアップでは、次の CloudFormation テンプレートを使用する必要があります。

- VendorInsightsPrerequisiteCFT - アカウントで CloudFormation StackSets を実行するために必要な管理者ロールとアクセス許可を設定します。このスタックは販売者アカウントで作成してください。
- VendorInsightsOnboardingCFT - 必要な AWS のサービスを設定し、適切な IAM アクセス許可を設定します。これらのアクセス許可により、AWS Marketplace Vendor Insights はで実行されている SaaS 製品のデータを収集 AWS アカウントし、そのデータを AWS Marketplace Vendor Insights プロファイルに表示できます。StackSets を通じて SaaS ソリューションをホストしている販売者アカウントと本番稼働用アカウントの両方にこの StackSets を作成します。

VendorInsightsPrerequisiteCFT スタックを作成します。

VendorInsightsPrerequisiteCFT CloudFormation スタックを実行することにより、スタックセットのオンボーディングを開始するための IAM アクセス許可を設定します。

VendorInsightsPrerequisiteCFT スタックを作成するには

1. GitHub ウェブサイトの [Vendor Insights テンプレートフォルダ用AWS サンプルリポジトリ](#) から最新の VendorInsightsPrerequisiteCFT.yml ファイルを確認してダウンロードしてください。
2. AWS Marketplace 販売者アカウント AWS Management Console を使用して にサインインし、<https://console.aws.amazon.com/cloudformation> で AWS CloudFormation コンソールを開きます。
3. CloudFormation コンソールのナビゲーションペインで、[スタック] を選択し、ドロップダウンから [スタックを作成] と [新しいリソースを使用 (標準)] を選択します (ナビゲーションペインが表示されていない場合は、左上隅にあるナビゲーションペインを選択して展開します)。
4. [テンプレートの指定] で、[テンプレートファイルのアップロード] を選択します。ダウンロードした VendorInsightsPrerequisiteCFT.yml ファイルをアップロードするには、[ファイルを選択] を使用します。次いで、[次へ] を選択します。
5. スタックの名前を入力し、[次へ] を選択します。
6. (オプション) 必要に応じてスタックオプションを設定します。

[次へ] を選択します。

7. [確認] ページで選択内容を確認します。変更するには、変更する領域で [編集] を選択します。スタックを作成する前に、[能力] エリアの確認チェックボックスを選択する必要があります。

[Submit] を選択してください。

8. スタックが作成されたら、[リソース] タブを選択し、作成された以下のロールを書き留めます。

- AWSVendorInsightsOnboardingStackSetsAdmin
- AWSVendorInsightsOnboardingStackSetsExecution

VendorInsightsOnboardingCFT スタックセットを作成する

VendorInsightsOnboardingCFT CloudFormation スタックセットを実行することで、必要な を設定し AWS のサービス、適切な IAM アクセス許可を設定します。これにより、AWS Marketplace Vendor Insights は で実行されている SaaS 製品のデータを収集 AWS アカウント し、それを AWS Marketplace Vendor Insights プロファイルに表示できます。

マルチアカウントソリューションを使用している場合や、販売者アカウントと本番稼働用アカウントを別々に使用している場合は、このスタックを複数のアカウントにデプロイする必要があります。StackSets では、前提条件スタックを作成した管理アカウントからこれを行うことができます。

スタックセットは、セルフマネージド型のアクセス許可を使用してデプロイされます。詳細については、AWS CloudFormation ユーザーガイドの[セルフマネージド型のアクセス許可を持つスタックセットの作成](#)を参照してください。

VendorInsightsOnboardingCFT スタックセットを作成するには

1. GitHub ウェブサイトの[ベンダーインサイトテンプレートフォルダ用AWS サンプルリポジトリ](#)から最新の VendorInsightsOnboardingCFT.yml ファイルを確認してダウンロードしてください。
2. AWS Marketplace 販売者アカウント AWS Management Console を使用して にサインインし、<https://console.aws.amazon.com/cloudformation> で AWS CloudFormation コンソールを開きます。
3. CloudFormation コンソールのナビゲーションペインで、[StackSet の作成] を選択します。(ナビゲーションペインが表示されていない場合は、左上隅にあるナビゲーションペインを選択して展開します)。
4. [アクセス許可] で、管理者ロールとして [IAM ロール名] を選択し、ドロップダウンからロール名として [AWSVendorInsightsOnBoardingStackSetsAdmin] を選択します。

5. [IAM 実行ロール名] として **AWSVendorInsightsOnboardingStackSetsExecution** を入力します。
6. [テンプレートの指定] で、[テンプレートファイルのアップロード] を選択します。ダウンロードした VendorInsightsOnboardingCFT.yml ファイルをアップロードするには、[ファイルを選択] を使用し、[次へ] を選択します。
7. 次の StackSet パラメータを指定し、[次へ] を選択します。
 - CreateVendorInsightsAutomatedAssessment – このパラメータは、で AWS Audit Manager 自動評価を設定します AWS アカウント。管理アカウントと本稼働アカウントが別々にある場合、このオプションは運用アカウントでのみ選択し、管理アカウントには選択しないでください。
 - CreateVendorInsightsIAMRoles – このパラメータは、AWS Marketplace Vendor Insights が の評価データを読み取ることを許可する IAM ロールをプロビジョニングします AWS アカウント。
 - PrimaryRegion - このパラメータは、SaaS デプロイメントのプライマリ AWS リージョンを設定します。これは、Amazon S3 バケットが に作成されるリージョンです AWS アカウント。SaaS 製品が 1 つのリージョンのみにデプロイされている場合、そのリージョンがプライマリリージョンです。
8. StackSet オプションを必要に応じて設定します。[実行] 設定を [非アクティブ] のままにして、[次へ] を選択します。
9. デプロイオプションを設定します。マルチアカウントソリューションを使用している場合は、1 回の操作で複数のアカウントとリージョンにスタックセットをデプロイするように設定できます。[次へ] を選択します。

 Note

マルチアカウントソリューションを使用している場合は、1 つのスタックセットとしてすべてのアカウントにデプロイすることはお勧めしません。ステップ 7 で定義したパラメータには細心の注意を払ってください。デプロイ先のアカウントの種類によっては、一部のパラメータを有効または無効にする必要がある場合があります。StackSets は、1 回のデプロイで指定されたすべてのアカウントに同じパラメータを適用します。アカウントをスタックセットにグループ化することによりデプロイ時間を短縮できますが、マルチアカウントソリューションではやはり複数回デプロイする必要があります。

⚠ Important

複数のリージョンにデプロイする場合は、最初にリストするリージョンが PrimaryRegion である必要があります。[リージョンの同時実行] オプションはデフォルト設定の [シーケンシャル] のままにしておきます。

10. [確認] ページで選択内容を確認します。変更するには、変更する領域で [編集] を選択します。スタックセットを作成する前に、[能力] エリアの確認チェックボックスを選択する必要があります。

[Submit] を選択してください。

スタックセットはリージョンごとに約 5 分で完了します。

AWS Marketplace Vendor Insights プロファイルの表示

AWS Marketplace Vendor Insights のプロフィールは、購入者が製品を評価する際に使用する重要な情報を提供します。データ保護の目的で、認証情報を保護し AWS アカウント、AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。このアプローチでは、それぞれのジョブを遂行するために必要なアクセス許可のみを各ユーザーに付与できます。IAM でのユーザーの作成方法の詳細については、「[the section called “グループの作成または使用”](#)」を参照してください。

購入者は AWS Marketplace Vendor Insights ダッシュボードを使用して製品を評価できます。そこでは、購入者はプロフィールに追加したデータソースによって定義された製品概要を確認できます。セキュリティプロフィールは、10 のカテゴリの複数のセキュリティコントロールによって定義されます。

データの定義に使用される 10 種類のセキュリティカテゴリは次のとおりです。

- アクセス管理
- アプリケーションセキュリティ
- 監査、コンプライアンス、セキュリティポリシー
- ビジネスの回復力
- データセキュリティとプライバシー
- エンドユーザー、デバイス、モバイルセキュリティ

- 人事
- インフラストラクチャセキュリティ
- リスク管理とインシデント対応
- セキュリティと設定に関するポリシー

詳細については、「AWS Marketplace購入者ガイド」の「[コントロールカテゴリを理解する](#)」を参照してください。

AWS Marketplace Vendor Insights を設定して使用することで、ユーザー情報を非公開かつ安全に保つための AWS のサービス 条件とデータプライバシールールを遵守することに同意したものとみなされます。AWS データプライバシー条件の詳細については、「[データプライバシーに関するよくある質問](#)」を参照してください。サービス条件の詳細については、「[AWS のサービス 条件](#)」を参照してください。

販売者としてのセキュリティプロフィールを確認する

自己評価を完了し、その他の実在する証拠を追加したら、販売者としてのプロフィールを確認することが重要です。プロフィールに追加された情報の確認が必要になります。

Note

Vendor AWS Marketplace Insights サポートチームに可視性の更新をリクエストするまで、このプロフィールは購入者に表示されません。サポートチームが更新を完了すると、秘密保持契約 (NDA) に署名した購入者はセキュリティプロフィールにアクセスできるようになります。

AWS Marketplace Vendor Insights プロファイルからサブスクライバーの個人を特定できる情報 (PII) データを削除する場合は、に連絡してサポートケースを開始します[サポート](#)。

購入者が表示するようにセキュリティプロフィールを表示するには

1. AWS Management Consoleにサインインします。
2. ポータルの [SaaS 製品](#) ページに移動します。
3. セキュリティプロフィールが関連付けられている製品を選択します。
4. [ベンダーインサイト] タブを選択し、[最新リリースのスナップショットを表示] を選択します。
5. [概要] タブには、アップロードしたすべての証明書バッジが表示されます。

6. [セキュリティとコンプライアンス] タブを選択すると、複数のコントロールから収集されたデータを表示できます。詳細を表示するには、各コントロールセットを選択します。

AWS Marketplace Vendor Insights でのスナップショットの管理

スナップショットは、セキュリティプロファイルの特定の時点の状態です。AWS Marketplace Vendor Insights では、スナップショットを使用していつでも販売者の製品を評価できます。販売者は、さまざまな時点でのプロファイルのセキュリティ対策を比較したり、さまざまなセキュリティプロファイルの最新のスナップショットを比較したりして、意思決定の参考にすることができます。スナップショットは、データの鮮度やソースに関する透明性を提供するだけでなく、必要なセキュリティ情報を提供します。

AWS Marketplace コンソールの AWS Marketplace Vendor Insights の [スナップショットの概要] セクションでは、作成およびリリーススケジュールに関する以下のスナップショットの詳細を確認できます。

- [最終作成スナップショット] - このプロファイル用に最後に作成されたスナップショット。
- [次回作成予定] - 次に作成される予定のスナップショット。
- [作成頻度] - スナップショットを作成する間隔またはスナップショットを作成する頻度。
- [回次の予定リリース] - 次にリリースされる予定のスナップショット。
- [ステージング時間] - スナップショットは少なくともこの期間ステージングされ、スナップショットリリースイベント中にリリース可能になります。
- [リリース頻度] - リリースイベント間の時間。

[スナップショットリスト] セクションでは、スナップショットのステータスは次のとおりです。

- [リリース済み] - スナップショットは公開されており、この製品に対する権限を持つユーザーが閲覧できます。
- [リリース待ち] - スナップショットは完成しているか、必須の最小ステージング期間に入っており、次のリリースが予定されています。
- [プライベート] - セキュリティプロファイルの有効化前に作成されたスナップショット、または検証エラーがあり、一般には公開されていないスナップショット。非公開のスナップショットは販売者にのみ表示されます。

トピック

- [スナップショットを作成する](#)
- [スナップショットを表示する](#)
- [スナップショットをエクスポートする](#)
- [最新リリースのスナップショットを表示する](#)
- [スナップショットリリースを延期する](#)
- [スナップショットリストの設定を変更する](#)

スナップショットを作成する

プロフィールのスナップショットを作成するには、以下の手順に従ってください。1 日に最大 20 件のスナップショットを作成できます。

1. AWS Management Console にサインインして、[AWS Marketplace コンソール](#) を開きます。
2. [ベンダーインサイト] を選択します。
3. [ベンダーインサイト] から製品を選択します。
4. 製品プロファイルページで、[スナップショット] リストに移動し、[新規スナップショットの作成] を選択します。
5. スナップショットスケジュールが変更されることを通知するメッセージが表示されます。
[Create] (作成) を選択します。

Note

スナップショットスケジュールは、新しいスナップショットが作成されると変更されます。新しいスナップショットは、手動で作成したスナップショットと同じ時間にスケジュールされます。このメッセージには新しいスケジュールが含まれます。

新しいスナップショットは 30 分以内に作成され、スナップショットリストに追加されます。新しいスナップショットは [リリース保留中] ステータスで作成されます。ステータスが [リリース済み] に変わるまで、新しいスナップショットは表示できません。

スナップショットを表示する

プロフィールのスナップショットを表示するには、以下の手順に従ってください。

1. AWS Management Console にサインインして、[AWS Marketplace コンソール](#) を開きます。

2. [ベンダーインサイト] を選択します。
3. [ベンダーインサイト] から製品を選択します。
4. 製品プロファイルページで、[スナップショットリスト] に移動し、表示するスナップショットの [スナップショット ID] を選択します。
5. 完了したら、[戻る] を選択してスナップショットビューを終了します。

スナップショットをエクスポートする

JSON 形式または CSV 形式にエクスポートできます。スナップショットをエクスポートするには、次の手順を実行します。

1. AWS Management Console にサインインして、[AWS Marketplace コンソール](#) を開きます。
2. [ベンダーインサイト] を選択します。
3. [ベンダーインサイト] から製品を選択します。
4. 製品プロファイルページで、[スナップショットリスト] に移動し、エクスポートするスナップショットの [スナップショット ID] を選択します。
5. [エクスポート] をクリックします。
6. ドロップダウンリストから [ダウンロード (JSON)] または [ダウンロード (CSV)] を選択します。

最新リリースのスナップショットを表示する

最新リリースのスナップショットは、ユーザーが製品の状態を確認したり評価したりするために使用するものです。製品を正確な情報で表現するには、最新リリースのスナップショットの内容を知ることが重要です。プロフィールの最新のスナップショットを表示するには、次の手順を実行します。

1. AWS Management Console にサインインして、[AWS Marketplace コンソール](#) を開きます。
2. [ベンダーインサイト] を選択します。
3. [ベンダーインサイト] から製品を選択します。
4. 製品プロファイルページで、[スナップショットリスト] に移動し、表示するスナップショットの [スナップショット ID] を選択します。
5. [最新リリースのスナップショットを表示] を選択します。
6. 完了したら、[戻る] を選択してスナップショットビューを終了します。

スナップショットリリースを延期する

プロファイルへのスナップショットのリリースを遅らせるには、特定の [スナップショット ID] のスナップショットリリースを延期できます。

1. AWS Management Console にサインインして、[AWS Marketplace コンソール](#) を開きます。
2. [ベンダーインサイト] を選択します。
3. [ベンダーインサイト] から製品を選択します。
4. 製品プロファイルページで、[スナップショットリスト] に移動し、リリースを延期するスナップショットの [スナップショット ID] を選択します。
5. [スナップショットの概要] から [スナップショットリリースを延期] を選択します。
6. スナップショットスケジュールが変更されることを通知するメッセージが表示されます。[延期] を選択します。

この製品のスナップショットリリースを正常に延期したことを示す成功メッセージが表示されます。

スナップショットリストの設定を変更する

スナップショットを作成したら、[スナップショットリスト] でのスナップショットの表示方法の設定を変更できます。

1. AWS Management Console にサインインして、[AWS Marketplace コンソール](#) を開きます。
2. [ベンダーインサイト] を選択します。
3. [ベンダーインサイト] から製品を選択します。
4. 製品プロファイルページで、[スナップショットリスト] に移動し、変更するスナップショットの [スナップショット ID] を選択します。
5. 設定アイコンを選択します。スナップショットに関する以下の設定をカスタマイズできます。
 - [ページサイズ] - 各ページに表示するスナップショットの数を、1 ページあたり [10 リソース]、[20 リソース]、または [50 リソース] の中から選択します。
 - [行を折り返す] - 行を折り返してレコード全体を表示するオプションを選択します。
 - [時間フォーマット] - [絶対]、[相対]、[ISO] のいずれかを選択します。
 - [表示列] - スナップショットの詳細に表示するオプション ([スナップショット ID]、[ステータス]、[作成日]) を選択します。

AWS Marketplace Vendor Insights でのアクセスの制御

AWS Identity and Access Management (IAM) は AWS のサービス、AWS リソースへのアクセスを制御するのに役立つです。IAM は、追加料金なしで AWS のサービス 使用できる です。管理者は、誰を認証 (サインイン) し、誰に リソースの使用 AWS Marketplace を承認する (アクセス許可を付与する) かを制御します。AWS Marketplace Vendor Insights は IAM を使用して、販売者データ、評価、販売者の自己認証、業界標準の監査レポートへのアクセスを制御します。

で誰が何をできるかを制御するための推奨方法は AWS Marketplace 管理ポータル、IAM を使用してユーザーとグループを作成することです。次に、そのユーザーをグループに追加してグループを管理します。読み取り専用のアクセス許可を付与するポリシーまたはアクセス許可をグループに割り当てます。読み取り専用アクセスが必要なユーザーが他にもいる場合は、ユーザーのアクセス許可を追加するのではなく、作成したグループにそれらのユーザーを追加します。

ポリシーは、ユーザー、グループ、またはロールに適用されるアクセス許可を定義したドキュメントです。AWSでユーザーができることを決定するアクセス許可。ポリシーでは通常、特定のアクションへのアクセスを許可し、オプションで、それらのアクションを Amazon EC2 インスタンスや Amazon S3 バケットなどの特定のリソースで実行することを許可することができます。また、ポリシーにより、アクセスを明示的に拒否することもできます。アクセス許可は、特定のリソースへのアクセスを許可または拒否するポリシー内のステートメントです。

Important

作成したすべてのユーザーは、自分の認証情報を使用して認証します。ただし、同じ AWS アカウントを使用します。ユーザーが行った変更はアカウント全体に影響を及ぼす可能性があります。

AWS Marketplace には、それらのアクセス許可を持つユーザーが で実行できるアクションを制御するためのアクセス許可が定義されています AWS Marketplace 管理ポータル。また、が AWS Marketplace 作成および管理し、複数のアクセス許可を組み合わせたポリシーもあります。AWSMarketplaceSellerProductsFullAccess ポリシーでは、ユーザーに AWS Marketplace 管理ポータルの製品へのフルアクセスを許可します。

使用可能なアクション、リソース、および条件キーの詳細については、「サービス認可リファレンス」の[AWS Marketplace 「Vendor Insights のアクション、リソース、および条件キー」](#)を参照してください。

AWS Marketplace Vendor Insights 販売者のアクセス許可

AWS Marketplace Vendor Insights の IAM ポリシーでは、次のアクセス許可を使用できます。アクセス許可を 1 つの IAM ポリシーにまとめて、必要なアクセス許可を付与できます。

CreateDataSource

CreateDataSource では、ユーザーは新しいデータソースリソースを作成できます。サポートされているデータソースは以下の通りです。

- SOC2Type2AuditReport
- ISO27001AuditReport
- AWSAuditManagerSecurityAutomatedAssessment
- FedRAMPCertification
- GDPRComplianceReport
- HIPAAComplianceReport
- PCIDSSAuditReport
- SecuritySelfAssessment

アクショングループ: 読み取り/書き込み

必要なリソース: なし

リソースの作成: DataSource

DeleteDataSource

DeleteDataSource では、ユーザーは所有するデータソースを削除できます。削除するには、データソースとプロファイルの関連付けを解除する必要があります。詳細については、「[the section called “AssociateDataSource”](#)」を参照してください。

アクショングループ: 読み取り/書き込み

必要なリソース: DataSource

GetDataSource

GetDataSource では、ユーザーがデータソースの詳細を取得できます。データソースの詳細には、関連するタイムスタンプ、元の作成パラメータ、処理情報 (ある場合) などのメタデータ情報が含まれます。

アクショングループ: 読み取り専用、読み取り/書き込み

必要なリソース: DataSource

UpdateDataSource

UpdateDataSource では、ユーザーがデータソースの詳細を更新できます。詳細には、名前やソース情報 (ロール、ソース Amazon リソースネーム (ARN)、ソースコンテンツなど) などのメタデータ情報が含まれます。

アクショングループ: 読み取り専用、読み取り/書き込み

必要なリソース: DataSource

ListDataSources

ListDataSources では、ユーザーが所有するデータソースを一覧表示できます。

アクショングループ: 読み取り専用、読み取り/書き込み、リスト専用

必要なリソース: なし

CreateSecurityProfile

CreateSecurityProfile では、ユーザーが新しいセキュリティプロファイルを作成できます。セキュリティプロファイルは、スナップショットをいつどのように生成するかを管理するためのリソースです。また、ユーザーはプロファイルのステータスや適用条件を管理して、購入者がスナップショットにアクセスする方法を制御することもできます。

アクショングループ: 読み取り専用、読み取り/書き込み、リスト専用

必要なリソース: なし

リソースの作成: SecurityProfile

ListSecurityProfiles

ListSecurityProfiles では、ユーザーが所有するセキュリティプロファイルを一覧表示できます。

アクショングループ: 読み取り専用、読み取り/書き込み、リスト専用

必要なリソース: なし

GetSecurityProfile

CreateSecurityProfile では、ユーザーはセキュリティプロファイルの詳細を取得できます。

アクショングループ: 読み取り専用、読み取り/書き込み

必要なリソース: SecurityProfile

AssociateDataSource

AssociateDataSource では、ユーザーは既存の を AWS Marketplace Vendor Insights DataSource プロファイルに関連付けることができます。ユーザーは、データソースをプロファイルに関連付けたり関連付けを解除したりすることにより、スナップショットの内容を制御できます。

アクショングループ: 読み取り/書き込み

必要なリソース: SecurityProfile および DataSource

DisassociateDataSource

DisassociateDataSource では、ユーザーは既存の DataSource と AWS Marketplace Vendor Insights プロファイルの関連付けを解除できます。ユーザーは、データソースをプロファイルに関連付けたり関連付けを解除したりすることにより、スナップショットの内容を制御できます。

アクショングループ: 読み取り/書き込み

必要なリソース: SecurityProfile および DataSource

UpdateSecurityProfile

UpdateSecurityProfile では、ユーザーは名前や説明などのセキュリティプロファイル属性を変更できます。

アクショングループ: 読み取り/書き込み

必要なリソース: SecurityProfile

ActivateSecurityProfile

ActivateSecurityProfile では、ユーザーがセキュリティプロファイルの Active ステータスを設定できます。セキュリティプロファイルを有効にすると、他の条件が満たされた場合に公開できる Staged 状態で新しいスナップショットを作成できます。詳細については、「[the section called “UpdateSecurityProfileSnapshotReleaseConfiguration”](#)」を参照してください。

少なくとも 1 つの Released スナップショットを持つ Active セキュリティプロファイルは、エンドユーザーの AWS Marketplace Vendor Insights 検出の対象となります。

アクショングループ: 読み取り/書き込み

必要なリソース: SecurityProfile

DeactivateSecurityProfile

DeactivateSecurityProfile では、ユーザーがセキュリティプロファイルの Inactive ステータスを設定できます。セキュリティプロファイルのこの終了状態は、プロファイルを共有状態から削除することと同じです。ユーザーがセキュリティプロファイルを非アクティブ化できるのは、プロファイルにアクティブなサブスクライバーがない場合のみです。

アクショングループ: 読み取り/書き込み

必要なリソース: SecurityProfile

UpdateSecurityProfileSnapshotCreationConfiguration

UpdateSecurityProfileSnapshotCreationConfiguration では、ユーザーはスナップショット作成設定のカスタムスケジュールを定義できます。週次作成のデフォルト作成設定は、このアクションで上書きできます。

ユーザーはこのアクションを使用して、スケジュールをキャンセルしたり、スケジュールを将来の日付に延期したり、新しいスナップショット作成を以前に開始したりするなど、スケジュールを変更できます。

アクショングループ: 読み取り/書き込み

必要なリソース: SecurityProfile

UpdateSecurityProfileSnapshotReleaseConfiguration

UpdateSecurityProfileSnapshotReleaseConfiguration では、ユーザーはスナップショットリリース設定のカスタムスケジュールを定義できます。毎週リリースされ、ステージング期間が 2 日でレビュー対象のデフォルト作成設定は、このアクションで上書きできます。

ユーザーはこのアクションを使用して、スケジュールをキャンセルしたり、スケジュールを将来の日付に延期したりするなど、スケジュールを変更できます。

アクショングループ: 読み取り/書き込み

必要なリソース: SecurityProfile

ListSecurityProfileSnapshots

ListSecurityProfileSnapshots では、ユーザーが所有するセキュリティプロファイルのスナップショットを一覧表示できます。

アクショングループ: 読み取り専用、リスト専用、読み取り/書き込み。

必要なリソース: SecurityProfile

GetSecurityProfileSnapshot

GetSecurityProfileSnapshot では、ユーザーが所有するセキュリティプロファイルのスナップショットを取得できます。

アクショングループ: 読み取り専用、読み取り/書き込み

必要なリソース: SecurityProfile

TagResource

TagResource では、ユーザーがリソースに新しいタグを追加できます。サポートされているリソースは SecurityProfile と DataSource です。

アクショングループ: タグ付け

オプションリソース: SecurityProfile と DataSource

UntagResource

UntagResource では、ユーザーはリソースからタグを削除できます。サポートされているリソースは SecurityProfile と DataSource です。

アクショングループ: タグ付け

オプションリソース: SecurityProfile と DataSource

ListTagsForResource

ListTagsForResource では、ユーザーはリソースのリソースタグを一覧表示できます。サポートされているリソースは SecurityProfile と DataSource です。

アクショングループ: 読み取り専用

オプションリソース: SecurityProfile と DataSource

追加リソース

「IAM ユーザーガイド」の以下のリソースには、IAM の始め方と使用に関する詳細情報が記載されています。

- [IAM でのセキュリティのベストプラクティス](#)
- [IAM ポリシーを管理する](#)
- [IAM ユーザーグループにポリシーをアタッチする](#)
- [IAM ID \(ユーザー、ユーザーグループ、ロール\)](#)
- [AWS アカウントに IAM ユーザーを作成する](#)
- [IAM ユーザーグループを作成する](#)
- [ポリシーを使用した AWS リソースへのアクセスの制御](#)

AWS Marketplace セキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。

セキュリティは、AWS お客様とお客様の間の責任共有です。[責任共有モデル](#)では、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ – AWS クラウドで AWS サービスを実行するインフラストラクチャを保護する AWS 責任があります。AWS また、では、安全に使用できるサービスも提供しています。セキュリティの有効性は、[AWS コンプライアンスプログラム](#)の一環として、サードパーティーの審査機関によって定期的にテストおよび検証されています。が適用されるコンプライアンスプログラムの詳細については AWS Marketplace、[AWS 「コンプライアンスプログラムによる対象範囲内のサービス」](#)を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、お客様のデータの機密性、組織の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

このドキュメントは、を使用する際の責任共有モデルの適用方法を理解するのに役立ちます AWS Marketplace。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために AWS Marketplace、へのアクセス AWS Identity and Access Management を管理するようにを設定する方法を示します。また、リソースのモニタリングや保護 AWS Marketplace に役立つ他の AWS サービスの使用方法についても説明します。

が提供する製品に関するセキュリティおよびその他のポリシーの詳細については AWS Marketplace、以下のトピックを参照してください。

- [の AMI ベースの製品要件 AWS Marketplace](#)
- [のコンテナベースの製品要件 AWS Marketplace](#)
- [機械学習製品を作成するための要件とベストプラクティス](#)
- [の SaaS 製品ガイドライン AWS Marketplace](#)
- [でのプロフェッショナルサービス製品の要件 AWS Marketplace](#)

Note

データ製品の AWS Data Exchange のセキュリティについては、「AWS Data Exchange ユーザーガイド」の「[セキュリティ](#)」を参照してください。

の購入者のセキュリティの詳細については AWS Marketplace、「AWS Marketplace 購入者ガイド」の「[のセキュリティ AWS Marketplace](#)」を参照してください。

トピック

- [AWS Marketplace 管理ポータルへのアクセスのコントロール](#)
- [AWS Marketplace 販売者のポリシーとアクセス許可](#)
- [AWS Marketplace 販売者の 管理ポリシー](#)
- [AWS Marketplace Commerce Analytics Service アカウントのアクセス許可](#)
- [Amazon SQS のアクセス許可](#)
- [AWS Marketplace 計測と使用権限 API アクセス許可](#)
- [での再販認可のためのサービスにリンクされたロールの使用 AWS Marketplace](#)
- [を使用した AWS Marketplace Metering API コールのログ記録 AWS CloudTrail](#)

AWS Marketplace 管理ポータルへのアクセスのコントロール

AWS Identity and Access Management (IAM) は AWS のサービス、AWS リソースへのアクセスを制御するのに役立ちます。管理者は、誰を認証 (サインイン) し、誰に AWS Marketplace リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで AWS のサービス 使用できる です。

で誰が何をできるかを制御するための推奨方法は、IAM を使用してユーザーとグループを作成する AWS Marketplace 管理ポータル ことです。次に、そのユーザーをグループに追加してグループを管理します。例えば、製品の表示を John に許可する場合は、専用のユーザーを作成し、そのユーザーを読み取り専用アクセス用に作成したグループに追加します。読み取り専用のアクセス許可を付与するポリシーまたはアクセス許可をグループに割り当てます。読み取り専用アクセスが必要なユーザーが他にもいる場合は、アクセス許可をユーザーに許可するのではなく、作成したグループにそれらのユーザーを追加します。例えば、John のロールが変更になり、読み取り専用アクセスが不要になった場合は、John をそのグループから削除します。

ポリシーは、ユーザー、グループ、またはロールに適用されるアクセス許可を定義したドキュメントです。また、AWSでユーザーに許可するアクションをアクセス許可で指定します。ポリシーでは通

常、特定のアクションへのアクセスを許可し、オプションで、それらのアクションを Amazon EC2 インスタンスや Amazon S3 バケットなどの特定のリソースで実行することを許可することができます。また、ポリシーにより、アクセスを明示的に拒否することもできます。アクセス許可は、特定のリソースへのアクセスを許可または拒否するポリシー内のステートメントです。許可は、「A は C に対して B を実行する許可を持つ」という形で提示することができます。例えば、Jane (A) は John の Amazon Simple Queue Service キュー (C) からメッセージ (B) を読む許可がある、などです。Jane が John のキューを使用するリクエストを Amazon SQS に送信するたびに、サービスは Jane にアクセス許可があるかどうか確認します。さらに、アクセス許可に指定された条件をリクエストが満たしているかどうか確認します。

 Important

作成したすべてのユーザーは、自分の認証情報を使用して認証します。ただし、同じ AWS アカウントを使用します。ユーザーが行った変更はアカウント全体に影響を及ぼす可能性があります。

AWS Marketplace には、これらのアクセス許可を持つユーザーが実行できるアクションを制御するためのアクセス許可が定義されています AWS Marketplace 管理ポータル。また、複数のアクセス許可を組み合わせた AWS Marketplace 作成および管理するポリシーもあります。

IAM の使用開始と使用の詳細については、以下のリソースを参照してください。

- [管理ユーザーを作成する](#)
- [IAM でのセキュリティのベストプラクティス](#)
- [IAM ポリシーを管理する](#)
- [IAM ユーザーグループへのポリシーのアタッチ](#)
- [IAM ID \(ユーザー、グループ、ロール\)](#)
- [ポリシーを使用した AWS リソースへのアクセスの制御](#)

ユーザーとグループを作成し、ユーザーとしてサインインするためのいくつかのガイダンスを以下のトピックに示します。

トピック

- [ユーザーの作成](#)
- [グループの作成または使用](#)

• ユーザーとしてのサインイン

ユーザーの作成

社内のユーザーがサインインできるようにするには AWS Marketplace 管理ポータル、アクセスが必要なユーザーごとにユーザーを作成します。

ユーザーを作成するには

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/iam/> で IAM コンソールを開きます。
2. ナビゲーションペインの [アクセス管理] で、[ユーザー] を選択し、[ユーザーの作成] を選択します。
3. 番号付きのテキストボックスに、作成する各ユーザーの名前を入力します。
4. [ユーザーごとにアクセスキーを作成] のチェックボックスをオフにし、[作成] を選択します。

作成したばかりの各ユーザーにパスワードを割り当てるには

1. ユーザーのリストで、新しいユーザーの名前を選択します。
2. [セキュリティ認証情報] タブを選択し、[パスワードの管理] を選択します。
3. 自動生成パスワードまたはカスタムパスワードのいずれかのオプションを選択します。次回のサインインで新しいパスワードの選択をユーザーに要求する場合は、[次回のサインインで新しいパスワードの作成をユーザーに要求する] チェックボックスをオンにします。[Apply] を選択します。
4. [認証情報をダウンロード] を選択して、サインイン認証情報とアカウント固有のサインイン URL をコンピュータ上のカンマ区切り値 (CSV) ファイルに保存します。次に、[Close] (閉じる) を選択します。

Note

作成したばかりのサインイン認証情報を使用してサインインするには、ユーザーはアカウント固有のサインイン URL に移動する必要があります。この URL はダウンロードしたばかりの認証情報ファイルにあり、IAM コンソールでも確認できます。詳細については、「IAM ユーザーガイド」の「[IAM ユーザーが AWS アカウントにサインインする方法](#)」を参照してください。

i Tip

自分が AWS アカウント 所有者であっても、自分用のサインイン認証情報も作成してください。これは、アカウント所有者であっても、ユーザーとして AWS Marketplace で操作を行うために全員に推奨されるベストプラクティスです。管理者権限を持つユーザーを自分で作成する方法については、「IAM ユーザーガイド」の「[管理ユーザーを作成する](#)」を参照してください。

グループの作成または使用

ユーザーを作成したら、グループを作成し、AWS Marketplace 管理ポータルでページにアクセスするアクセス許可を作成します。次に、そのアクセス許可をグループに追加し、ユーザーをグループに追加します。

グループにアクセス許可を割り当てると、そのグループのすべてのメンバーに、特定のアクションの実行が許可されます。新しいユーザーをグループに追加すると、そのユーザーはグループに割り当てられたアクセス許可を自動的に取得します。グループは複数のアクションのアクセス許可を持つことができます。独自のポリシーを作成せずに [AWS Marketplace 管理ポリシー](#) を使用することをお勧めします。

の 管理ポリシー AWS Marketplace をグループに割り当てるには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [グループ] を選択後、ポリシーをアタッチするグループを選択します。
3. グループの概要ページの [アクセス許可] タブで、[Attach Policy (ポリシーのアタッチ)] を選択します。
4. [ポリシーのアタッチ] ページで、[Filter:] の横に、「awsmarketplace」と入力します。
5. アタッチするポリシーを選択し、[ポリシーのアタッチ] を選択します。

アクセス AWS Marketplace 管理ポータル 許可を持つポリシーを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで、[Policies (ポリシー)] を選択し、次に [Create Policy (ポリシーの作成)] を選択します。

3. [Policy Generator] の横の [Select] をクリックします。
4. [Edit Permissions] ページで、以下を実行します。
 - a. [Effect] で、[Allow] を選択します。
 - b. [AWS Service (サービス)] で、[AWS Marketplace 管理ポータル] を選択します。
 - c. [Actions (アクション)] で、付与するアクセス許可を選択します。
 - d. [Add Statement (ステートメントの追加)] を選択します。
 - e. [Next Step] (次のステップ) をクリックします。
5. [Review Policy (ポリシーのレビュー)] ページで、以下の作業を行います。
 - a. [Policy Name (ポリシー名)] にこのポリシーの名前を入力します。ポリシー名をメモします。後のステップで必要になります。
 - b. (オプション) [Description (説明)] に、ポリシーの説明を入力します。
 - c. [ポリシーの作成] を選択します。

適切なアクセス許可を持つ IAM グループを作成し、そのグループにユーザーを追加するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで、[Groups]、[Create New Group] の順に選択します。
3. [Group Name: (グループ名:)] に、グループの名前を入力します。その後、[Next Step] (次のステップ) を選択します。
4. [Attach Policy (ポリシーのアタッチ)] ページで、次の操作を実行します。
 - a. [Filter: (フィルター:)] で、[Customer Managed Policies (顧客が管理するポリシー)] を選択します。
 - b. このグループにアタッチするポリシーの名前の横にあるチェックボックスをオンにします。通常、これは先ほど作成したポリシーになります。
 - c. [Next Step] (次のステップ) をクリックします。
5. [グループを作成] を選択します。
6. [Groups (グループ)] リストで新しいグループを見つけ、その横にあるチェックボックスをオンにします。[Group Actions (グループアクション)] で、[Add Users to Group (ユーザーをグループに追加)] を選択します。
7. グループに追加する各ユーザーの横にあるチェックボックスをオンにして、[Add Users (ユーザーの追加)] を選択します。

ユーザーとしてのサインイン

IAM でユーザーを作成すると、ユーザーは自分のサインイン認証情報を使用してサインインできます。これを行うには、AWS アカウントに関連付けられている一意の URL を使用する必要があります。サインイン URL を取得し、ユーザーに配信できます。

アカウント固有のサインイン URL を取得するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで、[ダッシュボード] を選択します。
3. コンテンツペインの上部にある [IAM ユーザーのサインインリンク:] を探し、次のような形式のサインインリンクをメモします。

```
https://AWS_account_ID.signin.aws.amazon.com/console/
```

Note

サインインページの URL に AWS アカウント ID ではなく会社名 (またはその他のわかりやすい識別子) を含める場合は、カスタマイズ を選択してアカウントのエイリアスを作成できます。詳細については、「IAM [ユーザーガイド](#)」の「[AWS アカウント ID とそのエイリアス](#)」を参照してください。

4. この URL を、それぞれに作成したサインイン認証情報とともに AWS Marketplace、作業できる社内のユーザーに配布します。AWS Marketplace にアクセスする前に、アカウント固有のサインイン URL を使用してサインインするように指示してください。

AWS Marketplace 販売者のポリシーとアクセス許可

AWS Marketplace には、で使用する一連の管理ポリシーが用意されています AWS Marketplace 管理ポータル。さらに、個々のアクセス許可を使用して独自の AWS Identity and Access Management (IAM) ポリシーを作成できます。

設定、お問い合わせ、ファイルのアップロード、インサイトタブ AWS Marketplace 管理ポータル へのきめ細かなアクセスを提供することもできます。きめ細かなアクセス許可を使用することで、以下が可能になります。

- パスワードやアクセスキーを共有しなくても、お客様の AWS アカウント アカウントのリソースを管理および使用するためのアクセス許可を他の人に付与します。
- さまざまなリソースに対する細やかな権限を複数のユーザーに付与します。例えば、一部のユーザーに AWS Marketplace 管理ポータル の [設定] タブを表示することを許可し、他のユーザーには [設定] タブと [お問い合わせ] タブで編集することを許可できます。

Note

データ製品のための AWS Data Exchange のポリシーと権限の詳細については、「AWS Data Exchange ユーザーガイド」の「[Identity and Access Management in AWS Data Exchange](#)」を参照してください。

AWS Marketplace 購入者のポリシーとアクセス許可の詳細については、AWS Marketplace 「購入者ガイド」の[AWS Marketplace 「サブスクリプションへのアクセスの制御」](#)を参照してください。

AWS Marketplace 販売者のポリシー

次の管理ポリシーを使用して、AWS Marketplace 管理ポータルへの制御されたアクセスをユーザーに付与することができます。

AWSMarketplaceSellerFullAccess

Amazon マシンイメージ (AMI) 管理など、AWS Marketplace 管理ポータル および他の AWS サービス内のすべてのページへのフルアクセスを許可します。

AWSMarketplaceSellerProductsFullAccess

AWS Marketplace 管理ポータルの[製品](#)ページへのフルアクセスを許可します。

AWSMarketplaceSellerProductsReadOnly

AWS Marketplace 管理ポータルの[製品](#)ページへの読み取り専用アクセスを許可します。

Important

AWS Marketplace 購入者は、管理ポリシーを使用して、購入するサブスクリプションを管理できます。で使用する管理ポリシーの名前は、で AWS Marketplace 管理ポータル 始まりますAWSMarketplaceSeller。IAM でポリシーを検索するには、必ず先頭が

AWSMarketplaceSeller となっているポリシー名を検索してください。これらのポリシーの詳細については、AWS「マネージドポリシーリファレンス」を参照してください。

AWS Marketplace は、特定のシナリオに特化した マネージドポリシーも提供します。AWS Marketplace 販売者の AWS 管理ポリシーの完全なリストと、それらが提供するアクセス許可の説明については、「」を参照してください [AWS Marketplace 販売者の 管理ポリシー](#)。

AWS Marketplace 販売者のアクセス許可

以下のアクセス許可は、AWS Marketplace 管理ポータル の IAM ポリシーで使用できます。

aws-marketplace-management:PutSellerVerificationDetails

本人確認 (KYC) プロセスを開始するためのアクセスを許可します。

aws-marketplace-management:GetSellerVerificationDetails

AWS Marketplace 管理ポータル の KYC ステータスを表示するためのアクセスを許可します。

aws-marketplace-management:PutBankAccountVerificationDetails

[銀行口座検証](#) プロセスを開始するためのアクセスを許可します。

aws-marketplace-management:GetBankAccountVerificationDetails

AWS Marketplace 管理ポータル に銀行口座検証ステータスを表示するためのアクセスを許可します。

aws-marketplace-management:PutSecondaryUserVerificationDetails

でセカンダリユーザーを追加するアクセスを許可します AWS Marketplace 管理ポータル。

aws-marketplace-management:GetSecondaryUserVerificationDetails

AWS Marketplace 管理ポータル にセカンダリユーザーステータスを表示するアクセスを許可します。

aws-marketplace-management:GetAdditionalSellerNotificationRecipients

AWS Marketplace 通知の E メール連絡先を表示するアクセスを許可します。

aws-marketplace-management:PutAdditionalSellerNotificationRecipients

AWS Marketplace 通知の E メール連絡先を更新するためのアクセスを許可します。

tax:PutTaxInterview

AWS Marketplace 管理ポータルで [Tax Interview](#) を受けるアクセスを許可します。

tax:GetTaxInterview

AWS Marketplace 管理ポータルで Tax Interview ステータスを表示するアクセスを許可します。

tax:GetTaxInfoReportingDocument

AWS Marketplace 販売者が Tax ダッシュボードから税ドキュメント (1099-K フォームなど) を表示およびダウンロードすることを許可する

payments:CreatePaymentInstrument

に銀行口座を追加するアクセスを許可します AWS Marketplace 管理ポータル。

payments:GetPaymentInstrument

で既存の銀行口座を表示するためのアクセスを許可します AWS Marketplace 管理ポータル。

support:CreateCase

内で AWS Marketplace ケースを作成するためのアクセスを許可します AWS Marketplace 管理ポータル。

aws-marketplace-management:viewSupport

AWS Marketplace 管理ポータルの [\[カスタマーサポート利用資格\]](#) ページへのアクセスを許可します。

aws-marketplace-management:viewReports

AWS Marketplace 管理ポータルの [\[レポート\]](#) ページへのアクセスを許可します。

aws-marketplace:ListEntities

でオブジェクトを一覧表示するためのアクセスを許可します AWS Marketplace 管理ポータル。
AWS Marketplace 管理ポータルの [\[ファイルのアップロード\]](#)、[\[オファー\]](#)、[\[パートナー\]](#) ページにアクセスするために必要です。

Note

[設定] タブを表示するためのアクセスを許可するには、この権限、ListEntity 権限、および Amazon リソースネーム (ARN) `arn:{partition}:{aws-marketplace}:`

`{region}:{account-id}:AWSMarketplace/Seller/{entity-id}` を使用できません。

aws-marketplace:DescribeEntity

でオブジェクトの詳細を表示するためのアクセスを許可します AWS Marketplace 管理ポータル。AWS Marketplace 管理ポータルの[\[ファイルのアップロード\]](#)、[\[オファー\]](#)、[\[パートナー\]](#)、[\[契約\]](#) ページにアクセスするために必要です。

Note

[設定] タブを表示するためのアクセスを許可するには、この権限、DescribeEntity 権限、および ARN `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/*` を使用できます。

aws-marketplace:StartChangeSet

で製品変更を作成するためのアクセスを許可します AWS Marketplace 管理ポータル。AWS Marketplace 管理ポータルの[\[ファイルのアップロード\]](#)、[\[オファー\]](#)、[\[パートナー\]](#)、[\[契約\]](#) ページに変更を加えるために必要です。

Note

で販売者として登録するためのアクセスを許可するには AWS Marketplace、このアクセス許可、`catalog:ChangeType: "CreateSeller"` 条件キー、および次の ARN を使用できます: `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`。

で販売者プロフィールを更新するためのアクセスを許可するには AWS Marketplace、このアクセス許可、`catalog:ChangeType: "UpdateInformation"` 条件キー、および次の ARN を使用できます。 `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`

Amazon Web Services の支払い設定を更新するためのアクセスを許可するには、この権限、`catalog:ChangeType: "UpdateDisbursementPreferences"` 条件キー、および ARN `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}` を使用できます。

aws-marketplace:SearchAgreements

[\[契約\]](#) ページでの契約の概要リストの表示、[\[パートナー\]](#) ページでの ISV とチャネルパートナー間の機会の表示を許可します。

aws-marketplace:DescribeAgreement

[\[契約\]](#) ページでの契約詳細の概要の表示、[\[パートナー\]](#) ページでの ISV とチャネルパートナー間の機会の表示を許可します。

aws-marketplace:GetAgreementTerms

[\[契約\]](#) ページでのすべての契約条件の詳細の表示、[\[パートナー\]](#) ページでの ISV とチャネルパートナー間の機会の表示を許可します。

aws-marketplace:GetSellerDashboard

AWS Marketplace 管理ポータル [\[インサイト\]](#) ページのダッシュボードへのアクセスを許可します。

aws-marketplace:ListAssessments

販売者のアクションが保留中の評価のリストを表示するためのアクセスを許可します。

aws-marketplace:DescribeAssessment

販売者のアクションが保留中の評価の詳細を表示するためのアクセスを許可します。

Note

[\[製品の管理\]](#) ページへのアクセスをユーザーに許可するには、マネージドアクセス許可 `AWSMarketplaceSellerProductsFullAccess` または `AWSMarketplaceSellerProductsReadOnly` のいずれかを使用する必要があります。

前のアクセス許可を 1 つの IAM ポリシーにまとめて、必要なアクセス許可を付与することができます。次の例を参照してください。

例 1: KYC ステータスを表示するアクセス許可

で KYC ステータスを表示するアクセス許可を付与するには AWS Marketplace 管理ポータル、次の例のようなポリシーを使用します。

で KYC ステータスを表示するアクセス許可を付与するには AWS Marketplace 管理ポータル、次の例のようなポリシーを使用します。

JSON

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

例 2: プライベートオファ어의アップグレードと更新を作成するためのアクセス許可

[契約] ページを表示および使用してプライベートオファ어의アップグレードと更新を作成するためのアクセス許可を付与するには、次の例に示すようなポリシーを使用します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aws-marketplace:SearchAgreements",
        "aws-marketplace:DescribeAgreement",
        "aws-marketplace:GetAgreementTerms",
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:StartChangeSet"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {

```

```

        "aws-marketplace:PartyType": "Proposer"
      },
      "ForAllValues:StringEquals": {
        "aws-marketplace:AgreementType": [
          "PurchaseAgreement"
        ]
      }
    }
  ]
}

```

例 3: オファーページにアクセスして新しいプライベートオファーを作成する権限

[オファー] ページを表示および使用して既存のプライベートオファーを表示しプライベートオファーを作成するためのアクセス許可を付与するには、次の例に示すようなポリシーを使用します。

JSON

```

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

```

例 4: 設定ページにアクセスする権限

[設定] ページを表示して使用するアクセス許可を付与するには、次の例のようなポリシーを使用します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aws-marketplace:ListEntities",
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:StartChangeSet"
      ],
      "Effect": "Allow",
      "Resource": "arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/*",
    }
  ]
}
```

例 5: ファイルのアップロードページにアクセスする権限

[ファイルをアップロード] ページを表示して使用するアクセス許可を付与するには、次の例のようなポリシーを使用します。

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

IAM グループの使用

または、AWS Marketplace 管理ポータル内の各個別のページへのアクセス権限を付与する、個別の IAM グループを作成できます。ユーザーは複数のグループに属することができます。そのため、ユーザーが複数のページにアクセスする必要がある場合、適切なすべてのグループにユーザーを追

加できます。例えば、1 つの IAM グループを作成し、そのグループに [インサイト] ページにアクセスする権限を付与します。次に、別のグループを作成し、そのグループに [ファイルをアップロード] ページにアクセスする権限を付与するなどの操作ができます。ユーザーが [インサイト] ページと [ファイルをアップロード] ページの両方にアクセスするための権限が必要な場合は、ユーザーを両方のグループに追加します。

ユーザーとグループの詳細については、「IAM ユーザーガイド」の「[IAM アイデンティティ \(ユーザー、グループ、ロール\)](#)」を参照してください。

AWSAWS Marketplace 販売者の 管理ポリシー

AWS 管理ポリシーは、によって作成および管理されるスタンドアロンポリシーです AWS。AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有の[カスタマー管理ポリシー](#)を定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使えるようになったときに、AWS マネージドポリシーを更新する可能性が高くなります。

詳細については「IAM ユーザーガイド」の「[AWS マネージドポリシー](#)」を参照してください。

このセクションには、販売者の AWS Marketplace へのアクセスを管理するために使用される各ポリシーが記載されています。購入者ポリシーの詳細については、「[AWSAWS Marketplace 購入者ガイド](#)」の「[購入者の 管理ポリシー](#)」を参照してください。AWS Marketplace

トピック

- [AWS 管理ポリシー: AWSMarketplaceAmiIngestion](#)
- [AWS 管理ポリシー: AWSMarketplaceFullAccess](#)
- [AWS 管理ポリシー: AWSMarketplaceGetEntitlements](#)
- [AWS 管理ポリシー: AWSMarketplaceMeteringFullAccess](#)
- [AWS マネージドポリシー: AWSMarketplaceMeteringRegisterUsage](#)

- [AWS 管理ポリシー: AWSMarketplaceSellerFullAccess](#)
- [AWS マネージドポリシー: AWSMarketplaceSellerProductsFullAccess](#)
- [AWS マネージドポリシー: AWSMarketplaceSellerProductsReadOnly](#)
- [AWS マネージドポリシー: AWSMarketplaceSellerOfferManagement](#)
- [AWS 管理ポリシー: AWSVendorInsightsVendorFullAccess](#)
- [AWS 管理ポリシー: AWSVendorInsightsVendorReadOnly](#)
- [AWS MarketplaceAWS 管理ポリシーの更新](#)

AWS 管理ポリシー: AWSMarketplaceAmiIngestion

このポリシーを使用してサービスロールを作成し、がユーザーに代わってアクションを実行 AWS Marketplace するために使用できます。AWSMarketplaceAmiIngestion の使用の詳細については、「[AMI AWS Marketplace へのアクセスの許可](#)」を参照してください。

このポリシーは、Amazon マシンイメージ (AMIs) を一覧表示するために AWS Marketplace がコピーできるようにする寄稿者アクセス許可を付与します AWS Marketplace。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceAmiIngestion](#)」を参照してください。

AWS 管理ポリシー: AWSMarketplaceFullAccess

AWSMarketplaceFullAccess ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、販売者と購入者の両方として、AWS Marketplace および関連サービスへのフルアクセスを許可する管理アクセス許可を付与します。これらの権限には以下の権限が含まれます。

- AWS Marketplace ソフトウェアをサブスクライブおよびサブスクライブ解除します。
- から AWS Marketplace ソフトウェアインスタンスを管理します AWS Marketplace。
- アカウントでプライベートマーケットプレイスを作成して管理する。
- Amazon EC2、AWS CloudFormationおよび Amazon EC2 Systems Manager へのアクセスを提供します。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceFullAccess](#)」を参照してください。

AWS 管理ポリシー: AWSMarketplaceGetEntitlements

AWSMarketplaceGetEntitlements ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、Software as a Service (SaaS) 製品販売者が顧客が AWS Marketplace SaaS 製品をサブスクライブしているかどうかをチェックできるようにする読み取り専用アクセス許可を付与します。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceGetEntitlements](#)」を参照してください。

AWS 管理ポリシー: AWSMarketplaceMeteringFullAccess

AWSMarketplaceMeteringFullAccess ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、AWS Marketplaceで柔軟な使用料金が設定されている AMI およびコンテナ製品に対応する従量制使用量をレポートできるようにする権限を寄稿者に付与します。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceMeteringFullAccess](#)」を参照してください。

AWS マネージドポリシー: AWSMarketplaceMeteringRegisterUsage

AWSMarketplaceMeteringRegisterUsage ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、時間単位の料金が設定されたコンテナ製品に対応する従量制使用量をレポートできるようにするアクセス許可を寄稿者に付与します AWS Marketplace。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceMeteringRegisterUsage](#)」を参照してください。

AWS 管理ポリシー: AWSMarketplaceSellerFullAccess

AWSMarketplaceSellerFullAccess ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは AWS Marketplace、AMI ベースの製品で使用される Amazon EC2 AMI の管理など AWS Marketplace 管理ポータル、のすべての販売者オペレーションへのフルアクセスを許可する管理アクセス許可を付与します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MarketplaceManagement",
      "Effect": "Allow",
      "Action": [
        "aws-marketplace-management:uploadFiles",
        "aws-marketplace-management:viewReports",
        "aws-marketplace-management:viewSupport",
        "aws-marketplace:ListChangeSets",
        "aws-marketplace:DescribeChangeSet",
        "aws-marketplace:StartChangeSet",
        "aws-marketplace:CancelChangeSet",
        "aws-marketplace:ListEntities",
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:GetSellerDashboard",
        "aws-marketplace:ListAssessments",
        "aws-marketplace:DescribeAssessment",
        "ec2:DescribeImages",
        "ec2:DescribeSnapshots",
        "ec2:ModifyImageAttribute",
        "ec2:ModifySnapshotAttribute"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AgreementAccess",
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:SearchAgreements",
        "aws-marketplace:DescribeAgreement",
        "aws-marketplace:GetAgreementTerms"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws-marketplace:PartyType": "Proposer"
        },
        "ForAllValues:StringEquals": {
          "aws-marketplace:AgreementType": [

```

```

        "PurchaseAgreement"
    ]
}
},
{
    "Sid": "IAMGetRole",
    "Effect": "Allow",
    "Action": [
        "iam:GetRole"
    ],
    "Resource": "arn:aws:iam::*:role/*"
},
{
    "Sid": "AssetScanning",
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::*:role/*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": "assets.marketplace.amazonaws.com"
        }
    }
},
{
    "Sid": "VendorInsights",
    "Effect": "Allow",
    "Action": [
        "vendor-insights:GetDataSource",
        "vendor-insights:ListDataSources",
        "vendor-insights:ListSecurityProfiles",
        "vendor-insights:GetSecurityProfile",
        "vendor-insights:GetSecurityProfileSnapshot",
        "vendor-insights:ListSecurityProfileSnapshots"
    ],
    "Resource": "*"
},
{
    "Sid": "TagManagement",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:TagResource",

```



```

        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
    ],
    "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
},
{
    "Sid": "SellerSettings",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace-management:GetSellerVerificationDetails",
        "aws-marketplace-management:PutSellerVerificationDetails",
        "aws-marketplace-management:GetBankAccountVerificationDetails",
        "aws-marketplace-management:PutBankAccountVerificationDetails",
        "aws-marketplace-management:GetSecondaryUserVerificationDetails",
        "aws-marketplace-management:PutSecondaryUserVerificationDetails",
        "aws-marketplace-
management:GetAdditionalSellerNotificationRecipients",
        "aws-marketplace-
management:PutAdditionalSellerNotificationRecipients",
        "payments:GetPaymentInstrument",
        "payments:CreatePaymentInstrument",
        "tax:GetTaxInterview",
        "tax:PutTaxInterview",
        "tax:GetTaxInfoReportingDocument",
        "tax:ListSupplementalTaxRegistrations",
        "tax:PutSupplementalTaxRegistration",
        "tax>DeleteSupplementalTaxRegistration",
        "tax:GetTaxRegistration"
    ],
    "Resource": "*"
},
{
    "Sid": "Support",
    "Effect": "Allow",
    "Action": [
        "support:CreateCase"
    ],
    "Resource": "*"
},
{
    "Sid": "ResourcePolicyManagement",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:GetResourcePolicy",

```

```

        "aws-marketplace:PutResourcePolicy",
        "aws-marketplace>DeleteResourcePolicy"
    ],
    "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
},
{
    "Sid": "CreateServiceLinkedRole",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:AWSServiceName": "resale-
authorization.marketplace.amazonaws.com"
        }
    }
}
]
}

```

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceSellerFullAccess](#)」を参照してください。

AWS マネージドポリシー: AWSMarketplaceSellerProductsFullAccess

AWSMarketplaceSellerProductsFullAccess ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、製品と へのフルアクセス AWS Marketplace 管理ポータル、および AMI ベースの製品で使用する Amazon EC2 AMI の管理を許可する寄稿者アクセス許可を付与します。

アクセス許可の詳細

JSON

```

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

```

```

        "aws-marketplace:ListChangeSets",
        "aws-marketplace:DescribeChangeSet",
        "aws-marketplace:StartChangeSet",
        "aws-marketplace:CancelChangeSet",
        "aws-marketplace:ListEntities",
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:ListAssessments",
        "aws-marketplace:DescribeAssessment",
        "ec2:DescribeImages",
        "ec2:DescribeSnapshots",
        "ec2:ModifyImageAttribute",
        "ec2:ModifySnapshotAttribute"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iam:GetRole"
    ],
    "Resource": "arn:aws:iam::*:role/*"
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::*:role/*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": "assets.marketplace.amazonaws.com"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "vendor-insights:GetDataSource",
        "vendor-insights:ListDataSources",
        "vendor-insights:ListSecurityProfiles",
        "vendor-insights:GetSecurityProfile",
        "vendor-insights:GetSecurityProfileSnapshot",
        "vendor-insights:ListSecurityProfileSnapshots"
    ],

```

```

        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "aws-marketplace:TagResource",
            "aws-marketplace:UntagResource",
            "aws-marketplace:ListTagsForResource"
        ],
        "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "aws-marketplace:GetResourcePolicy",
            "aws-marketplace:PutResourcePolicy",
            "aws-marketplace>DeleteResourcePolicy"
        ],
        "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
    }
]
}

```

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceSellerProductsFullAccess](#)」を参照してください。

AWS マネージドポリシー: AWSMarketplaceSellerProductsReadOnly

AWSMarketplaceSellerProductsReadOnly ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、AWS Marketplace 管理ポータルに製品を表示したり、AMI ベースの製品で使用されている Amazon EC2 AMI を表示したりするためのアクセスを許可する読み取り専用権限を付与します。

アクセス許可の詳細

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:ListChangeSets",
      "aws-marketplace:DescribeChangeSet",
      "aws-marketplace:ListEntities",
      "aws-marketplace:DescribeEntity",
      "aws-marketplace:ListAssessments",
      "aws-marketplace:DescribeAssessment",
      "ec2:DescribeImages",
      "ec2:DescribeSnapshots"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:ListTagsForResource"
    ],
    "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetResourcePolicy"
    ],
    "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
  }
]
```

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceSellerProductsReadOnly](#)」を参照してください。

AWS マネージドポリシー: AWSMarketplaceSellerOfferManagement

AWSMarketplaceSellerOfferManagement ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、オファーと契約の管理アクティビティへのアクセス権を販売者に付与します。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSMarketplaceSellerOfferManagement](#)」を参照してください。

AWS 管理ポリシー: AWSVendorInsightsVendorFullAccess

AWSVendorInsightsVendorFullAccess ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、AWS Marketplace Vendor Insights 上のすべてのリソースを作成および管理するためのフルアクセスを許可します。AWS Marketplace Vendor Insights では、このガイドの目的上、評価者は購入者と等しく、ベンダーは販売者と等しくなります。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSVendorInsightsVendorFullAccess](#)」を参照してください。

AWS 管理ポリシー: AWSVendorInsightsVendorReadOnly

AWSVendorInsightsVendorReadOnly ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、AWS Marketplace Vendor Insights プロファイルと関連リソースを表示するための読み取り専用アクセスを許可します。AWS Marketplace Vendor Insights では、このガイドの目的上、評価者は購入者と等しく、ベンダーは販売者と等しくなります。

このポリシーのアクセス許可を確認するにはAWS マネージドポリシーリファレンスの「[AWSVendorInsightsVendorReadOnly](#)」を参照してください。

AWS MarketplaceAWS 管理ポリシーの更新

このサービスがこれらの変更の追跡を開始 AWS Marketplace してからの AWS の管理ポリシーの更新に関する詳細を表示します。このページへの変更に関する自動アラートについては、AWS Marketplace [ドキュメント履歴](#) ページの RSS フィードを購読してください。

変更	説明	日付
AWSMarketplaceSellerFullAccess - ポリシーの更新	AWS Marketplace は、補足税プロファイル機能に、ListSupplementalTaxRegistrations、PutSupplementalTaxRegistrat	2024 年 12 月 20 日

変更	説明	日付
	ion 、 の DeleteSupplementalTaxRegistration 4 つの新しいSellerSettings アクセス許可を追加しましたGetTaxRegistration。	
AWSMarketplaceSellerFullAccess - ポリシーの更新 AWSMarketplaceSellerProductsFullAccess - ポリシーの更新 AWSMarketplaceSellerFullAccess - ポリシーの更新 AWSMarketplaceSellerProductsReadOnly - ポリシーの更新	AWS Marketplace は、ListTasks 、 、 UpdateTasks 、 および DescribeTask アクセスCompleteTasks 許可を削除しました。	2024 年 12 月 10 日
AWSMarketplaceSellerOfferManagement – 新しいポリシーを追加	AWS Marketplace に新しいポリシーが追加されました。AWSMarketplaceSellerOfferManagement	2024 年 11 月 18 日
AWSMarketplaceSellerFullAccess - ポリシーの更新	AWS Marketplace に アクセスUploadFiles 許可が追加されました。この変更により、販売者はで廃止されたページを使用できます AWS Marketplace 管理ポータル。	2024 年 11 月 6 日

変更	説明	日付
AWSMarketplaceSellerFullAccess - ポリシーの更新	AWS Marketplace に ListAssessments および アクセスDescribeAssessments 許可が追加されました。この変更により、SSLv2 ユーザーは評価データにアクセスできます。	2024 年 10 月 22 日
AWSMarketplaceSellerProductsFullAccess - ポリシーの更新	AWS Marketplace に ListAssessments および アクセスDescribeAssessments 許可が追加されました。この変更により、SSLv2 ユーザーは評価データにアクセスできます。	2024 年 10 月 22 日
AWSMarketplaceSellerProductsReadOnly - ポリシーの更新	AWS Marketplace に ListAssessments および アクセスDescribeAssessments 許可が追加されました。この変更により、SSLv2 ユーザーは評価データにアクセスできます。	2024 年 10 月 22 日

変更	説明	日付
AWSMarketplaceSellerFullAccess - ポリシーを更新	aws-marketplace-management:viewMarketing、aws-marketplace-management:viewSettings、および aws-marketplace-management:uploadFiles のアクションの削除を反映して AWSMarketplaceSellerFullAccess ドキュメントを更新しました。この更新には、きめ細かなアクセス許可の使用セクションの削除も含まれます。	2024 年 6 月 4 日
AWSMarketplaceGetEntitlements - ポリシーを更新	AWS Marketplace が更新されAWSMarketplaceGetEntitlements、ポリシーステートメントsidに が追加されました。	2024 年 3 月 22 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace が更新されAWSMarketplaceSellerFullAccess、サービスにリンクされたロールを作成するためのアクセス許可が追加されました。	2024 年 3 月 15 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace が更新されAWSMarketplaceSellerFullAccess、税金情報にアクセスするためのアクセス許可が追加されました。	2024 年 2 月 8 日

変更	説明	日付
AWSVendorInsightsVendorFullAccess - ポリシーを更新	AWS Marketplace が更新されAWSVendorInsightsVendorFullAccess 、データソースを更新するアクセス許可が追加されました。	2023 年 10 月 18 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace が更新されAWSMarketplaceSellerFullAccess 、エンティティを共有するためのアクセス許可が追加されました。	2023 年 6 月 1 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace が更新されAWSMarketplaceSellerFullAccess 、アカウント検証、銀行口座検証、ケース管理、販売者通知の詳細に関連するアクセス許可が追加されました。	2023 年 6 月 1 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace が更新されAWSMarketplaceSellerFullAccess 、販売者ダッシュボードにアクセスするためのアクセス許可が追加されました。	2022 年 12 月 23 日
AWSMarketplaceSellerFullAccess 、 AWSMarketplaceSellerProductsFullAccess 、 AWSMarketplaceSellerProductsReadOnly - 既存のポリシーの更新	AWS Marketplace により新しいタグベースの認可機能のポリシーが更新されました。	2022 年 12 月 9 日

変更	説明	日付
AWS Marketplace 更新済み AWSVendorInsightsVendorFullAccess	AWS Marketplace を更新AWSMarketplaceSellerProductsFullAccess して、契約検索、プロフィールスナップショットの更新、ベンダーのタグ付けを追加し、AWS Artifact サードパーティーのレポート (プレビュー) への読み取り専用アクセスを許可しました。	2022 年 11 月 30 日
AWS Marketplace 更新済み AWSVendorInsightsVendorReadOnly	AWS Marketplace タグを一覧表示するアクセス許可を追加し、サードパーティーレポート (プレビュー) へのAWS Artifact 読み取り専用アクセスを許可するAWSVendorInsightsVendorReadOnly ように を更新しました。	2022 年 11 月 30 日
AWSVendorInsightsVendorFullAccess と AWSVendorInsightsVendorReadOnly - ポリシーを追加	AWS Marketplace AWS Marketplace Vendor Insights の新機能である AWSMarketplaceSellerProductsFullAccess およびのポリシーが追加されましたAWSVendorInsightsVendorReadOnly 。	2022 年 7 月 26 日

変更	説明	日付
AWSMarketplaceSellerProductsFullAccess および AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace AWS Marketplace Vendor Insights の新機能のポリシーが更新されました: AWSMarketplaceSellerProductsFullAccess および AWSMarketplaceSellerFullAccess。	2022 年 7 月 26 日
AWSMarketplaceSellerFullAccess と AWSMarketplaceSellerProductsFullAccess - 既存のポリシーに対する更新	AWS Marketplace は、iam:PassedToService 条件が にのみ適用されるようにポリシーを更新しました iam:PassRole。	2021 年 11 月 22 日
AWSMarketplaceFullAccess - 既存ポリシーへの更新。	AWS Marketplace はAWSMarketplaceFullAccess ポリシーから重複するec2:DescribeAccountAttributes アクセス許可を削除しました。	2021 年 7 月 20 日
AWS Marketplace が変更の追跡を開始しました	AWS Marketplace は、AWS マネージドポリシーの変更の追跡を開始しました。	2021 年 4 月 20 日

AWS Marketplace Commerce Analytics Service アカウントのアクセス許可

Commerce Analytics Service に登録するには、次の AWS Marketplace IAM アクセス許可ポリシーを使用します。

登録方法については、[オンボーディングガイド](#)を参照してください。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles",
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "aws-marketplace-management:viewReports"
      ],
      "Resource": "*"
    }
  ]
}
```

以下の IAM アクセス許可ポリシーを使用して、ユーザーが AWS Marketplace Commerce Analytics Service にリクエストを送信できるようにします。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "marketplacecommerceanalytics:GenerateDataSet",
      "Resource": "*"
    }
  ]
}
```

この機能の詳細については、「[AWS Marketplace Commerce Analytics Service を使用して、製品および顧客データにアクセスする](#)」を参照してください。

Amazon SQS のアクセス許可

SaaS 製品の公開プロセスの一環として、は、顧客のサブスクリプションまたは使用権限のステータスに変更された場合に通知を受け取るために使用できる Amazon SNS トピック AWS Marketplace を提供します。キューが通知に応じてアクションを実行できるよう、1 つまたは複数の Amazon SQS キューをトピックに設定できます。例えば、顧客が SaaS 製品のサブスクリプションにストレージを追加した場合、Amazon SNS トピックは Amazon SQS キューにメッセージを送信し、その顧客が使用可能なストレージ容量を自動的に増加するプロセスを開始できます。

提供された Amazon SNS トピックへの Amazon Simple Queue Service (Amazon SQS) キューをサブスクライブすると、そのトピックがそのキューにメッセージを発行するためのアクセス許可が自動的に追加されます。ただし、AWS Marketplace Metering and Entitlement Service API ユーザーにキューへのアクセス権を付与するには、IAM ポリシーが必要です。そのポリシーは、サービスが同じ認証情報で実行されている場合は、同じユーザーに適用できます。次の内容のポリシーを作成して、販売者の ユーザーまたはロールにアタッチします。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sqs:ReceiveMessage", "sqs:DeleteMessage",
        "sqs:GetQueueAttributes", "sqs:GetQueueUrl"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:sqs:REGION_HERE:XXXXXXXXXXXX:NAME_HERE"
    }
  ]
}
```

Note

Resource フィールドは Amazon SQS キューの Amazon リソースネーム (ARN) です。

SaaS 製品のメッセージ通知およびキューに関する詳細情報については、[the section called “SNS トピックへの SQS キューのサブスクライブ”](#) および [the section called “AWS Marketplace Metering Service API および使用権限管理サービス API へのアクセス”](#) を参照してください。

AWS Marketplace 計測と使用権限 API アクセス許可

Software as a Service (SaaS) 製品、Amazon Machine Image (AMI) 製品、およびコンテナ製品は、AWS Marketplace Metering Service および AWS Marketplace Entitlement Service APIsを使用できます。タイプ別に異なる AWS Identity and Access Management (IAM) アクセス許可が必要です。製品については、すべての使用量を計測し、顧客は指定した計測レコード AWS に基づいて から請求されます。AWS Marketplace 計測レコードの提供に必要な統合を有効にするには、統合が使用しているサービスアカウントでアクセスを有効にするために制約された IAM ポリシーが必要です。計測情報を送信する対象の製品タイプのポリシーを、統合で使用しているユーザーまたはロールにアタッチします。

トピック

- [SaaS 製品の IAM ポリシー](#)
- [AMI 製品の IAM ポリシー](#)
- [コンテナ製品の IAM ポリシー](#)

SaaS 製品の IAM ポリシー

以下のポリシーでは、すべての SaaS 統合に最初のアクセス許可、aws-marketplace:ResolveCustomer が必要です。2 つ目のアクセス許可、aws-marketplace:BatchMeterUsage は AWS Marketplace Metering Service API に必要です。3 つ目のアクセス許可、aws-marketplace:GetEntitlements は AWS Marketplace Entitlement Service API に必要です。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aws-marketplace:ResolveCustomer",
        "aws-marketplace:BatchMeterUsage",
```

```
        "aws-marketplace:GetEntitlements"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

SaaS 製品の詳細については、「[の SaaS ベースの製品 AWS Marketplace](#)」を参照してください。

AMI 製品の IAM ポリシー

AMI 製品には以下の IAM ポリシーを使用してください。

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

AMI 製品の詳細については、「[の AMI ベースの製品 AWS Marketplace](#)」を参照してください。

コンテナ製品の IAM ポリシー

コンテナ製品には以下の IAM ポリシーを使用してください。

JSON

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



```
        "aws-marketplace:RegisterUsage"  
      ],  
      "Effect": "Allow",  
      "Resource": "*"   
    }  
  ]  
}
```

コンテナ製品の詳細については、「[のコンテナベースの製品 AWS Marketplace](#)」を参照してください。

ユーザーの作成の詳細については、「IAM ユーザーガイド」の「[AWS アカウントでの IAM ユーザーの作成](#)」を参照してください。ポリシーの作成および割り当ての詳細については、「[IAM ユーザーのアクセス許可の変更](#)」を参照してください。

このポリシーでは、ポリシーがアタッチされる IAM ロールまたはユーザーに API へのアクセス許可を付与しています。これらの API コールで別のアカウントによるロールの引き受けを有効にする方法の詳細については、AWS Partner Network (APN) ブログの「[How to Best Architect Your AWS Marketplace SaaS Subscription Across Multiple AWS アカウント](#)」を参照してください。

での再販認可のためのサービスにリンクされたロールの使用 AWS Marketplace

AWS Marketplace は AWS Identity and Access Management (IAM) [サービスにリンクされたロール](#)を使用します。サービスにリンクされたロールは、直接リンクされた一意のタイプの IAM ロールです AWS Marketplace。サービスにリンクされたロールは、によって事前定義 AWS Marketplace されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、の設定 AWS Marketplace が簡単になります。は、サービスにリンクされたロールのアクセス許可 AWS Marketplace を定義し、特に定義されている場合を除き、のみがそのロールを引き受け AWS Marketplace ることができます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、最初に関連リソースを削除する必要があります。これにより、AWS Marketplace リソースへのアクセス許可が誤って削除されないため、リソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスの詳細については、[AWS「IAM と連携するサービス」](#)を参照し、「サービスにリンクされたロール」列で「はい」があるサービスを探します。サービスリンクロールに関するドキュメントをサービスで表示するには、リンクで [はい] を選択します。

トピック

- [のサービスにリンクされたロールのアクセス許可 AWS Marketplace](#)
- [のサービスにリンクされたロールの作成 AWS Marketplace](#)
- [のサービスにリンクされたロールの編集 AWS Marketplace](#)
- [のサービスにリンクされたロールの削除 AWS Marketplace](#)
- [AWS Marketplace サービスにリンクされたロールでサポートされているリージョン](#)

のサービスにリンクされたロールのアクセス許可 AWS Marketplace

AWS Marketplace は、AWSServiceRoleForMarketplaceResaleAuthorization という名前のサービスにリンクされたロールを使用します。これにより、が再販認可 AWS Marketplace に使用または管理する AWS サービスやリソースにアクセスできます。

AWSServiceRoleForMarketplaceResaleAuthorization サービスリンクロールは、以下のサービスを信頼してロールを引き受けます。

- `resale-authorization.marketplace.amazonaws.com`

AWSMarketplaceResaleAuthorizationServiceRolePolicy という名前のロールアクセス許可ポリシーを使用すると、AWS Marketplace は指定されたリソースに対して次のアクションを完了できます。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowResaleAuthorizationShareActionsRAMCreate",
    "Effect": "Allow",
    "Action": [
      "ram:CreateResourceShare"
    ],
    "Resource": [
```

```

        "arn:aws:ram:*:*:*"
    ],
    "Condition": {
        "StringEquals": {
            "ram:RequestedResourceType": "aws-marketplace:Entity"
        },
        "ArnLike": {
            "ram:ResourceArn": "arn:aws:aws-
marketplace:*:*:AWSMarketplace/ResaleAuthorization/*"
        },
        "Null": {
            "ram:Principal": "true"
        }
    }
},
{
    "Sid": "AllowResaleAuthorizationShareActionsRAMAssociate",
    "Effect": "Allow",
    "Action": [
        "ram:AssociateResourceShare"
    ],
    "Resource": [
        "arn:aws:ram:*:*:*"
    ],
    "Condition": {
        "Null": {
            "ram:Principal": "false"
        },
        "StringEquals": {
            "ram:ResourceShareName": "AWSMarketplaceResaleAuthorization"
        }
    }
},
{
    "Sid": "AllowResaleAuthorizationShareActionsRAMAccept",
    "Effect": "Allow",
    "Action": [
        "ram:AcceptResourceShareInvitation"
    ],
    "Resource": [
        "arn:aws:ram:*:*:*"
    ],
    "Condition": {
        "StringEquals": {

```

```

        "ram:ResourceShareName": "AWSMarketplaceResaleAuthorization"
    }
}
},
{
    "Sid": "AllowResaleAuthorizationShareActionsRAMGet",
    "Effect": "Allow",
    "Action": [
        "ram:GetResourceShareInvitations",
        "ram:GetResourceShareAssociations"
    ],
    "Resource": [
        "arn:aws:ram:*:*:*"
    ]
},
{
    "Sid": "AllowResaleAuthorizationShareActionsMarketplace",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:PutResourcePolicy",
        "aws-marketplace:GetResourcePolicy"
    ],
    "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/ResaleAuthorization/*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": ["ram.amazonaws.com"]
        }
    }
},
{
    "Sid": "AllowResaleAuthorizationShareActionsMarketplaceDescribe",
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:DescribeEntity"
    ],
    "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/ResaleAuthorization/*"
}
]
}

```

ユーザー、グループ、またはロールにサービスリンクロールの作成、編集、または削除を許可するには、アクセス許可を設定する必要があります。詳細についてはIAM ユーザーガイドの「[サービスにリンクされた役割のアクセス許可](#)」を参照してください。

のサービスにリンクされたロールの作成 AWS Marketplace

サービスリンクロールを手動で作成する必要はありません。でサービスにリンクされたロールを選択すると AWS Marketplace 管理ポータル、によってサービスにリンクされたロールが自動的に AWS Marketplace 作成されます。

サービスリンクロールを作成するには

1. [AWS Marketplace 管理ポータル](#) で管理アカウントにサインインし、[設定] を選択します。
2. [設定] セクションで、[サービスリンクロール] タブを選択します。
3. [サービスリンクロール] ページで、[再販売認証用のサービスリンクロール] または [再販売認証統合] を選択して、[サービスリンクロールの作成] または [統合の設定] を選択します。
4. [再販売認証用のサービスリンクロール] または [再販売認証統合の作成] ページで情報を確認して、[サービスリンクロールの作成] または [統合の作成] を選択して確認します。

[サービスリンクロール] ページにメッセージが表示され、再販売認証サービスにリンクされたロールが正常に作成されたことを示します。

サービスにリンクされたロールを削除する場合は、以下の手順に従ってロールを再作成できます。

のサービスにリンクされたロールの編集 AWS Marketplace

AWS Marketplace では、AWSServiceRoleForMarketplaceResaleAuthorization サービスにリンクされたロールを編集することはできません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「[サービスリンクロールの編集](#)」を参照してください。

のサービスにリンクされたロールの削除 AWS Marketplace

サービスにリンクされたロールが必要な機能またはサービスが不要になった場合には、そのロールを削除することをお勧めします。そうすることで、モニタリングや保守が積極的に行われていない未使用のエンティティを排除できます。

Note

独立系ソフトウェアベンダー (ISVs) に ロールがない場合、ターゲットのチャネルパートナーと新しい再販認可を自動的に共有 AWS Resource Access Manager しません。チャネルパートナーに ロールがない場合、チャネルパートナーを対象とする再販承認は自動的に承認 AWS Resource Access Manager されません。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、AWS CLI、または AWS API を使用し

て、AWSServiceRoleForMarketplaceResaleAuthorization サービスにリンクされたロールを削除します。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの削除](#)」を参照してください。

AWS Marketplace サービスにリンクされたロールでサポートされているリージョン

AWS Marketplace は、サービスが利用可能なすべてのリージョンでサービスにリンクされたロールの使用をサポートしています。詳細については、「[AWS リージョンとエンドポイント](#)」を参照してください。

を使用した AWS Marketplace Metering API コールのログ記録 AWS CloudTrail

AWS Marketplace は、ユーザー AWS CloudTrail、ロール、または によって実行されたアクションを記録するサービスであると統合 AWS のサービスされています AWS Marketplace。CloudTrail は、の API コールをイベント AWS Marketplace としてキャプチャします。キャプチャされた呼び出しには、AWS Marketplace コンソールからの呼び出しと AWS Marketplace API オペレーションへのコード呼び出しが含まれます。

CloudTrail は、アカウントの作成 AWS アカウント 時に 有効になります。でサポートされているイベントアクティビティが発生すると AWS Marketplace、そのアクティビティはイベント履歴の他の AWS のサービス イベントとともに CloudTrail イベントに記録されます。 アカウントで最近のイベントを表示、検索、ダウンロードできます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストがルート認証情報または AWS Identity and Access Management ユーザー認証情報を使用して行われたかどうか。
- リクエストが、ロールとフェデレーティッドユーザーのどちらかの一時的なセキュリティ認証情報を使用して送信されたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

AWS Marketplace は、CloudTrail ログファイルのイベントとして BatchMeterUsage オペレーションのログ記録をサポートします。

AWS Marketplace 計測 API ログファイルエントリの例

例: BatchMeterUsage

次の例は、AWS Marketplace Metering Service からの BatchMeterUsage アクションを示す CloudTrail ログエントリの例です。販売者が [にリストされている Software as a Service \(SaaS\) 製品の顧客使用状況を報告するために計測レコードを送信する](#) と AWS Marketplace、この CloudTrail ログエントリは販売者の に記録されます AWS アカウント。 SaaS

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

例: コンテナ用 **RegisterUsage**

次の例は、AWS Marketplace Metering Serviceからの RegisterUsage アクションを示す CloudTrail ログエントリの例です。からの時間単位の料金のコンテナ製品が購入者の にデプロイされると AWS アカウント、コンテナ内のソフトウェア AWS Marketplace は購入者の RegisterUsage内で AWS アカウント を呼び出して、その Amazon Elastic Container Service (Amazon ECS) タスクまたは Amazon Elastic Kubernetes Service (Amazon EKS) ポッドの時間単位の計測を開始します。この CloudTrail ログエントリは購入者の AWS アカウントに記録されます。

```

{
  "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"

```

}

例: Amazon EKS 上のコンテナの **MeterUsage**

次の例は、Amazon EKS 上のコンテナ AWS Marketplace Metering Service のからのMeterUsageアクションを示す CloudTrail ログエントリを示しています。からの[カスタム計測](#)を使用するコンテナ製品が購入者の にデプロイされると AWS アカウント、コンテナ内のソフトウェア AWS Marketplace は購入者の MeterUsage内で AWS アカウント を呼び出して 1 時間ごとにレポートします。この CloudTrail ログエントリは購入者の AWS アカウントに記録されます。

```
{
  "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
    boto/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"
}

```

例: AMI の MeterUsage

次の例は、Amazon マシンイメージ (AMIs) AWS Marketplace Metering Service の から
 の MeterUsage アクションを示す CloudTrail ログエントリを示しています。からのカスタム計測
 を使用する AMI 製品が購入者の にデプロイされると AWS アカウント、AMI のソフトウェア AWS
 Marketplace は購入者の MeterUsage 内で を呼び出し AWS アカウント、1 時間ごとに使用状況を
 報告します。この CloudTrail ログエントリは購入者の AWS アカウントに記録されます。

```

{
  "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"
}
```

トラブルシューティング

一般的な販売者登録エラーメッセージ

次の表に、一般的な販売者登録エラーメッセージとその解決策を示します。

エラーメッセージ	ソリューション
コンプライアンス要件が満たされていない	すべての日本の販売者は、有料製品を公開する前に、規制上の目的で追加情報を収集する必要があります。日本を拠点とする販売者で、出品に関心がある場合は AWS Marketplace、<aws-jp-marketplace@amazon.co.jp> の日本語販売者オンボーディングチームにお問い合わせください。詳細については、 「JP SOR に関するよくある質問」 を参照してください。
AWS Marketplace は、内部の問題のため、現在リクエストを処理できません。後でもう一度お試しください。問題が解決しない場合は、お問い合わせください。	ブラウザのキャッシュをクリアして、もう一度試してください。問題が解決しない場合は、 AWS Marketplace Seller Operations チーム にお問い合わせください。
製品をパブリックに更新する前に、販売者アカウントにパブリックプロファイルを追加する必要があります。	製品をパブリックに更新する前に、販売者アカウントにパブリックプロファイルを追加します。 AWS Marketplace 管理ポータル で、設定、次にパブリックプロファイルに移動します。
支払い設定を更新して、CurrencyCode と互換性を持たせます。	通貨コードと互換性があるように支払い設定を更新します。現地通貨は使用料金でサポートされていないため、必ず USD を追加してください。詳細については、 「利用可能な通貨」 を参照してください。
VAT/GST 情報は、有料販売者にとどまるために必要です。	これには 2 つのオプションがあります。

エラーメッセージ	ソリューション
	1. メインアカウントの税設定を更新します。税継承ドキュメントを確認して、これがすべての連結アカウントに与える影響を理解します。 2. このアカウントを組織から削除し、スタンドアロンアカウントとして使用します。つまり、Billing の税設定ページで VAT/GST 設定を手動で追加する必要があります。
税務調査の場所がビジネスの場所と一致しません	現在の場所設定を確認するには： 1. 税務調査の場所： • AWS Marketplace 管理ポータルで、「設定」に移動し、「税金情報」に移動します。 • 「税情報の更新」を選択し、「再処理」を選択します。 • 税プロファイルの更新を選択したら、下にスクロールして、指定した国を確認します。 2. 事業所： • AWS Marketplace 管理ポータルで、「設定」タブの「アカウント概要」で提供されているビジネスロケーションを確認します。 ロケーションが一致しない場合： • 請求での税金情報の更新: 詳細については、「アカウントの場所」を参照してください。 • AMMP 設定で税務調査を更新する: Tax 情報から税務調査を再取得する 支払い処理の問題を避けるため、両方の場所が同じであることを確認します。
の使用を開始するには AWS Marketplace、AWS アカウント登録を完了します。	AWS アカウント登録を完了します。 販売者登録の詳細については、「」を参照してくださいAWS Marketplace 販売者としての登録。

エラーメッセージ	ソリューション
パブリックプロファイルがレビュー中です。パブリックプロファイルが確認されるまで、支払い方法を追加することはできません。2 営業日以内にもう一度確認してください。現在公開できるのは無料製品のみです。	パブリックプロファイルがレビュー中です。パブリックプロファイルが確認されるまで、支払い方法を追加して KYC を完了することはできません。承認には 2~3 営業日かかります。現在公開できるのは無料製品のみです。プロファイルが承認されると、ルート E メールアドレスに E メールが送信されます。 3 営業日が経過しても支払い情報を追加できない場合は、AWS Marketplace Seller Operations チームにお問い合わせください。
事業所がサポート対象外の管轄区域にある。有料製品を公開することはできません。	事業所がサポート対象外の管轄区域にある。有料製品を公開することはできません。対象となる管轄区域の詳細については、「」を参照してください 有料製品の対象となる管轄区域 。
AWS アカウントが販売者として登録されていません AWS Marketplace。	販売者として登録されていません AWS Marketplace。 AWS Marketplace Management Portal から登録を完了します。
販売者が有料販売者になるには、パブリックプロファイルが必要です。	販売者のパブリックプロファイルを完了して承認する必要があります。これを完了し、 AWS Marketplace 管理ポータル でアカウントのステータスを確認します。

ドキュメント履歴

以下の表は、本リリースの AWS Marketplace 販売者ガイドのドキュメントを説明するものです。

このドキュメントの更新に関する通知については、RSS フィードにサブスクライブできます。

変更	説明	日付
支払い設定を販売者登録に設定するステップを追加しました	販売者の登録手順 https://docs.aws.amazon.com/marketplace/latest/userguide/seller-registration-process.html については、「」を参照してください。	2025 年 5 月 21 日
Marketplace の標準契約に連邦追加契約を追加	この付録は、米国政府調達合理化に役立ちます。詳細については、「 での標準化された契約 AWS Marketplace の使用」を参照してください。	2025 年 4 月 29 日
SaaS ガイドラインの更新	「デプロイ日」の AWS 指定に関する情報を追加しました。詳細については、「 https://docs.aws.amazon.com/marketplace/latest/userguide/saas-guidelines.html 」を参照してください。	2025 年 4 月 8 日
認可書	会社に代わって AWS アカウントを開くアクセス許可を誰かに付与するために使用される letter-of-authorization テンプレートを追加しました。詳細については、 AWS Marketplace 「販売者としての登録」 を参照してください。	2025 年 4 月 8 日

[セルフサービス出品情報を ML に追加しました](#)

セルフサービスのリスト情報<https://docs.aws.amazon.com/marketplace/latest/userguide/machine-learning-products>については、「」を参照してください。

2025 年 3 月 26 日

[更新されたマネージドポリシー](#)

[AWSMarketplaceSellerFullAccess](#) ポリシーに、ListSupplementalTaxRegistrations 、PutSupplementalTaxRegistration 、および のアクセス許可が追加されましたDeleteSupplementalTaxRegistration GetTaxRegistration 。

2024 年 12 月 20 日

[プロモーションメディアで AWS Marketplace 製品を強化する](#)

プロモーションメディアの使用方法に関するガイドを追加しました。

2024 年 12 月 17 日

[更新されたマネージドポリシー](#)

ListTasks 、DescribeTask 、UpdateTasks 、および のCompleteTasks アクセス許可は、次の AWS 管理ポリシーから削除されました。

2024 年 12 月 10 日

- [AWSMarketplaceSellerFullAccess](#)
- [AWSMarketplaceSellerProductsFullAccess](#)
- [AWSMarketplaceSellerProductsReadOnly](#)

新しい Buy with AWS 機能のドキュメントを追加しました

で購入 AWSすると、AWS Marketplace の販売者は独自のウェブサイトで AWS Marketplace 製品を表示および販売できます。AWS Marketplace の販売者は AWS Marketplace call-to-action (CTA) ボタンおよび APIs と統合して、販売者所有のウェブサイト で AWS を活用したソフトウェア検出および調達エクスペリエンスを構築できます。

2024 年 12 月 4 日

EC2 Image Builder コンポーネントベースの製品

製品配信方法として追加された EC2 Image Builder コンポーネントベースの製品。

2024 年 12 月 1 日

新しい AWS パートナーアシスタント

AWS Partner Assistant は、AWS パートナー向けの生成 AI を活用したチャットボットです。Partner Central と AWS Marketplace Portal (AMMP) の両方からアクセスできます。

2024 年 11 月 25 日

更新されたマネージドポリシー

次のポリシーに アップロードFiles 許可が付与されました。

2024 年 11 月 18 日

- [awsmarketplacefullaccess](#)
- [AWSMarketplaceSellerFullAccess](#)

新しい管理ポリシーを追加しました

AWSMarketplaceSellerOfferManagement ドキュメントを追加しました。

2024 年 11 月 18 日

[更新されたマネージドポリシー](#)

次の AWS 管理ポリシーに DescribeAssessment および アクセス ListAssessments 許可が付与されました。

2024 年 10 月 9 日

- [AWSMarketplaceSellerFullAccess](#)
- [AWSMarketplaceSellerProductsFullAccess](#)
- [AWSMarketplaceSellerProductsReadOnly](#)

[廃止されたアクセス許可の削除](#)

AWSMarketplaceSellerFullAccess ドキュメントを更新して 3 つのアクションを削除しました。

2024 年 6 月 4 日

[Amazon マシンイメージ \(AMI\) の年間契約変更オプションの更新](#)

販売者は、既存のサブスクリプションのインスタンスタイプを追加または切り替えできるようになりました。

2024 年 5 月 30 日

[本人確認 \(KYC\) プロセスの更新](#)

販売者の KYC プロセスに手順を追加しました。

2024 年 5 月 21 日

[AWS Marketplace 販売者のプライベートオファーエクスペリエンスの更新](#)

プライベートオファーを作成および管理するための拡張エクスペリエンスのためのコンテンツを追加しました。

2024 年 5 月 20 日

[Amazon EKS アドオン製品の要件の更新](#)

AWS Marketplace 「アドオンとしてコンテナ製品を準備する」セクションを更新し、「アドオン設定要件とアドオンプロバイダーのベストプラクティス」を追加しました。

2024 年 5 月 8 日

[AWS Marketplace 販売者のアクセス許可を更新しました](#)

構文エラーを修正するためのアクセス許可の例を更新しました。

2024 年 4 月 2 日

[SaaS 契約料金の更新](#)

従量制料金モデルを使用した SaaS 契約の超過の報告に関連するコンテンツを更新しました。

2024 年 4 月 2 日

[の新しいデモおよびプライベートオフアープシオン AWS Marketplace](#)

AWS Marketplace では、一部の販売者の製品詳細ページで[デモ](#)および[プライベートオフアープシオン](#)のリクエストオプションがサポートされるようになりました。

2024 年 4 月 1 日

[税の新しいダッシュボード](#)

AWS Marketplace は、での取引の米国税および国際税に関する可視化およびきめ細かなデータを提供する課税ダッシュボードをサポートするようになりました AWS Marketplace。

2024 年 3 月 29 日

[AWSMarketplaceGetEntitlements の更新](#)

AWSMarketplaceGetEntitlements マネージドポリシーに sid を追加しました。

2024 年 3 月 22 日

[プロフェッショナルサービス手順の更新](#)

製品の可視性の編集とプロフェッショナルサービス製品の手順の削除を更新しました。

2024 年 3 月 19 日

AMI アクセスポリシーの更新	Linux 固有の AMI アクセスポリシーと Unix のような AMI アクセスポリシーを明確にするために、セクションを更新しました。	2024 年 3 月 19 日
の検索エンジンの最適化情報を追加しました AWS Marketplace	の検索エンジンの最適化に関連するコンテンツを追加しました AWS Marketplace。	2024 年 3 月 19 日
の販売者の管理ポリシーを更新しました AWS Marketplace	サービスリンクロールの作成に関連するアクセス許可を追加するために AWSMarketplaceSellerFullAccess を更新しました。	2024 年 3 月 15 日
での製品の新しいサービスにリンクされたロール AWS Marketplace	AWS Marketplace は、が再販認可 AWS Marketplace のために使用または管理する AWS サービスおよびリソースへのアクセスを可能にするサービスにリンクされたロールを提供するようになりました。	2024 年 3 月 15 日
の販売者の管理ポリシーを更新しました AWS Marketplace	税情報へのアクセスに関連するアクセス許可を追加するために AWSMarketplaceSellerFullAccess を更新しました。	2024 年 2 月 8 日

[セキュリティレポートの Amazon EventBridge イベント](#)

AWS Marketplace は、販売者の製品でセキュリティ脆弱性レポートが利用可能になったときに、以前は Amazon CloudWatch Events と呼ばれていた Amazon EventBridge イベントをサポートするようになりました。

2024 年 1 月 31 日

[Amazon EKS アドオンのサ ポート](#)

AWS Marketplace コンテナベースの製品から Amazon EKS アドオンへの公開に関連するコンテンツと手順を追加しました。

2024 年 1 月 29 日

[プロフェッショナルサービス 販売者が製品を再販するた めのサポートの追加](#)

販売者は、チャネルパートナーに独立系ソフトウェアベンダー (ISV) としての再販機会を作成できるようになりました。

2024 年 1 月 18 日

[での将来の日付の契約の一般 提供 AWS Marketplace](#)

すべての ISVs と AWS Marketplace チャネルパートナーは、前払い料金でプライベートオファーを公開する一環として、将来の開始日を指定できます。

2024 年 1 月 16 日

[新しい使用状況ダッシュボー ド](#)

AWS Marketplace は、SaaS およびサーバー使用量ベースの製品を使用しているお客様に視覚化ときめ細かなデータを提供する使用状況ダッシュボードをサポートするようになりました。

2024 年 1 月 10 日

販売者向けの新しいクイック起動デプロイオプション	Software as a Service (SaaS) 製品の新しいクイック起動デプロイオプションに関連する内容と手順を追加しました。	2023 年 11 月 29 日
プライベートオファー用の Flexible Payment Schedule が利用可能に	プライベートオファー用の Flexible Payment Schedule (FPS) が AWS Marketplace のすべてのお客様に利用可能になりました。	2023 年 11 月 17 日
コンテナ製品のセルフサービスコンテンツを追加しました	コンテナ製品のセルフサービスアクションに関連するコンテンツと手順を追加しました。	2023 年 11 月 3 日
契約および更新用の新しいダッシュボード	AWS Marketplace は、販売者の契約と更新ダッシュボードを提供するようになりました。	2023 年 10 月 31 日
変更セットの Amazon EventBridge イベント	AWS Marketplace は、変更セットが成功、失敗、またはキャンセルのステータスで完了すると、以前は Amazon CloudWatch Events と呼ばれていた Amazon EventBridge イベントをサポートするようになりました。	2023 年 10 月 31 日
AWS Marketplace Vendor Insights で販売者の管理ポリシーを更新	AWS Marketplace が更新されAWSVendorInsightsVendorFullAccess 、データソースを更新するアクセス許可が追加されました。	2023 年 10 月 18 日

<u>AWS Marketplace for Desktop Applications (AMDA) のサポート終了</u>	AWS Marketplace は、2023 年 10 月 2 日に AMDA のサポートを終了しました。AMDA に関するすべての内容と手順がガイドから削除されました。	2023 年 10 月 2 日
<u>SaaS 製品のセルフサービスコンテンツを追加しました</u>	Software as a Service (SaaS) 製品のセルフサービスアクションに関連するコンテンツと手順を追加しました。	2023 年 9 月 12 日
<u>AWS Marketplace 販売者が日次支払いを利用できるようになりました</u>	の販売者は、毎日または毎月の支払いを受け取ることが AWS Marketplace 管理ポータル できるようになりました。	2023 年 9 月 7 日
<u>Amazon EventBridge のサポート</u>	AWS Marketplace は、販売者が新しいオファーを受け取ったときに EventBridge イベントをサポートするようになりました。	2023 年 9 月 6 日
<u>AWS Marketplace Vendor Insights の新しいセルフサービスオンボーディングエクスペリエンス</u>	AWS Marketplace Vendor Insights がセルフサービスのオンボーディングエクスペリエンスをサポートするようになりました。	2023 年 8 月 17 日
<u>の販売者の 管理ポリシーを更新しました AWS Marketplace</u>	エンティティの共有に関連するアクセス許可を追加するように AWSMarketplaceSellerFullAccess を更新しました。	2023 年 6 月 1 日

[の販売者の管理ポリシーを更新しました AWS Marketplace](#)

AWSMarketplaceSellerFullAccess が更新され、アカウント検証、銀行口座検証、ケース管理、販売者への通知の詳細に関するアクセス許可が追加されました。

2023 年 6 月 1 日

[のきめ細かなアクセスに関するコンテンツを追加しました AWS Marketplace](#)

に、きめ細かなアクセスの概要情報、アクセス許可、手順を追加しました AWS Marketplace 管理ポータル。

2023 年 6 月 1 日

[AMI 製品の手順を追加しました](#)

AMI 製品のセルフサービス手順を追加しました。

2023 年 5 月 12 日

[での Amazon Tax Exemption Program と米国税計算サービスのサポートの終了 AWS Marketplace](#)

Amazon 免税プログラムと米国税計算サービスは、2023 年 3 月 6 日にサポートが終了したため、このガイドから削除されました。

2023 年 3 月 6 日

[コンテナ製品の手順を追加しました](#)

コンテナ製品の設定を変更する新しい手順を追加しました。

2023 年 2 月 13 日

[AWS Marketplace 製品サポート接続のサポート終了](#)

AWS Marketplace Product Support Connection および Commerce Analytics Service を使用したお客様の連絡先情報の共有は、2022 年 11 月 30 日をもってサポートされなくなりました。ガイドから AWS Marketplace Product Support Connection コンテンツ AWS Marketplace を削除しました。

2023 年 1 月 27 日

[コンサルティングパートナー 名の変更](#)

AWS Marketplace は、コンサルティングパートナーをチャネルパートナーとして参照するようになりました。ガイドが更新され、名前の変更のみが反映されました。

2023 年 1 月 26 日

[カスタマーサービスチームの 名前が変更されました](#)

Managed Catalog Operations (MCO) チームは、名前を AWS Marketplace Seller Operations チームに変更しました。ガイドが更新され、名前の変更のみが反映されました。

2023 年 1 月 24 日

[プライベートオファーページ](#)

認証された購入者は、AWS Marketplace プライベートオファーページでに拡張された AWS アカウント プライベートオファーを表示できるようになりました。

2023 年 1 月 19 日

[SaaS 製品設定に関するトピック を追加しました](#)

SaaS 製品設定を変更する手順を含む新しいトピックを追加しました。

2023 年 1 月 6 日

[の販売者の管理ポリシーを更新 しました AWS Marketplace](#)

AWS Marketplace が更新されAWSMarketplaceSellerFullAccess 、販売者ダッシュボードにアクセスするためのアクセス許可が追加されました。

2022 年 12 月 23 日

[販売者へのメール通知が更新 されました](#)

プライベートオファーが公開されると販売者に通知されるようになりました。

2022 年 12 月 22 日

[AWS Marketplaceの販売者は、サブスクリプションのSaaS 無料トライアルを利用できるようにになりました](#)

販売者はサブスクリプション商品の無料トライアルを作成できるようになりました。

2022 年 12 月 16 日

[AWS Marketplaceでの AMI セルフサービス出品 \(バージョン 2\) の販売者体験の更新](#)

の販売者は AWS Marketplace、単一 AMI 製品のセルフサービス出品を作成できるようになりました。販売者は、処理に余計な時間を費やすことなく、自分で更新を行うことができます。

2022 年 12 月 14 日

[タグベースの承認の 3 つのポリシーが更新されました](#)

AWS Marketplace タグベースの認可機能の 3 つのポリシー (AWSMarketplaceSellerFullAccess、AWSMarketplaceSellerProductsFullAccess、) を更新AWSMarketplaceSellerProductsReadOnly しました。

2022 年 12 月 9 日

[AWS Marketplace Vendor Insights で販売者のポリシーを更新](#)

AWS Marketplace Vendor Insights 販売AWSVendorInsightsVendorReadOnly 者の 管理ポリシーAWSVendorInsightsAssessorFullAccess とを更新しました。

2022 年 11 月 30 日

[AWS Marketplace Vendor Insights で販売者のアクセスを制御する](#)

AWS Marketplace Vendor Insights に、販売者が利用できるアクションとアクセス許可を説明する新しいトピックを追加しました。

2022 年 11 月 30 日

[AWS Marketplace Vendor Insights の 4 つの管理ポリシーを更新](#)

AWS Marketplace Vendor Insights の AWSVendor InsightsVendorFull Access および AWSVendor InsightsVendorRead Only マネージドポリシーを更新しました。

2022 年 11 月 28 日

[販売者は Amazon EKS でアドオン製品を公開できます](#)

AWS Marketplace と Amazon EKS の統合により、販売者は購入者のために Amazon EKS コンソールで製品を提示できます。

2022 年 11 月 28 日

[AWS Marketplace Vendor Insights の設定を更新しました](#)

AWS Marketplace Vendor Insights のセットアップ手順を更新しました。

2022 年 11 月 18 日

[AWS Marketplace Vendor Insights の 2 つのポリシーを更新しました](#)

AWS Marketplace Vendor Insights AWSMarketplaceSellerFullAccess の 2 つのポリシーAWSMarketplaceSellerProductsFullAccess とを更新しました。

2022 年 7 月 26 日

[AWS Marketplace Vendor Insights にソフトウェアリスク評価を提供する機能に関する 2 つのポリシーを追加しました。](#)

Vendor AWS Marketplace Insights AWSVendor InsightsVendorRead Only にソフトウェアリスク評価を提供する 2 つのポリシーAWSVendorInsightsVendorFullAccess とを追加しました。

2022 年 7 月 26 日

[AWS Marketplace Vendor Insights は、に追加された新機能です。 AWS Marketplace](#)

AWS Marketplace Vendor Insights はソフトウェアリスク評価を提供する機能です。AWS Marketplace Vendor Insights Vendor Insights はソフトウェアリスク評価を提供する機能です。

2022 年 7 月 26 日

[AWS Marketplace Commerce Analytics Service のアクセス許可の更新](#)

Commerce Analytics Service AWS Marketplace には追加の IAM アクセス許可があります。

2022 年 7 月 21 日

[「販売者配送データフィードサービス」セクションが追加されました](#)

ドキュメントのみを更新し、「販売者配送データフィードサービス」セクションを追加し、データフィードに関連するセクションを再編成しました。

2022 年 6 月 15 日

[補足レポートのセクションの追加](#)

最近の機能起動用に AWS Marketplace を提供する補足レポート用の新しいセクションを追加しました。

2022 年 6 月 14 日

[契約用の SaaS 無料トライアルがで販売者が利用できるようになりました AWS Marketplace](#)

販売者は、無料トライアルの期間、トライアル期間中に利用できるディメンション、顧客が受け取る無料使用容量を定義することにより、追加の開発作業なしで無料トライアルを作成できるようになりました。

2022 年 5 月 31 日

購入者と販売者の取引にメール通知が追加されました

で行われたオファーと契約を検証する購入者と販売者への E メール通知を可能にする新機能 AWS Marketplace。

2022 年 5 月 23 日

機械学習製品リストの作成に例が追加されました

機械学習セクションのドキュメントのみを更新し、機械学習製品を作成する際の販売者と購入者の見解を比較する例を追加しました。

2022 年 4 月 22 日

機械学習セクションが更新されました

機械学習セクションの更新は、手順を明確にするため、ドキュメントのみの更新が行われました。

2022 年 4 月 15 日

現在、イスラエルは適格な管轄区域となっています

イスラエルの居住者が、AWS Marketplaceの販売者としての登録対象になりました。

2022 年 4 月 13 日

オファーデータフィードの新しい項目について更新されました

メーカーは、チャネルパートナーが作成したオファーのオファー情報を受け取るようになりました。これには、該当するアカウントが登録販売者であり、オファーのメーカーでもある場合のオファーも含まれます。

2022 年 3 月 29 日

再販オポチュニティの通知

販売者は再販商談の通知を受け取ることができるようになりました。

2022 年 3 月 28 日

プロフェッショナルサービス製品に動画を追加しました

プロフェッショナルサービス製品ページを更新し、サービス製品の管理方法の詳細を動画にまとめました。

2022 年 2 月 24 日

[サーバーレス SaaS 統合ソリューションのデプロイに関する新しいトピック](#)

サーバーレス SaaS デプロイの統合に関する新しい情報が追加されました。これには、デプロイ手順に関する参照用の[AWS クイックスタート](#)へのリンクが含まれます。

2022 年 2 月 15 日

[コンテナベースの要件と AMI セクションへの最小限の更新](#)

コンテナベースの要件のポリシーの更新を最小限に抑え、AMI 価格契約に関する誤った情報を削除しました

2022 年 2 月 14 日

[コンテナバージョニングの更新](#)

コンテナイメージやその他のアーティファクトをリポジトリにプッシュする方法を明確にするための、ドキュメントのみの更新です。

2022 年 2 月 10 日

[SaaS 製品の ResolveCustomer コード例の更新](#)

SaaS 製品の ResolveCustomer コード例が更新され、CustomerAWSAccount ID が含まれるようになりました。

2022 年 2 月 3 日

[Containers Anywhere 製品の AWS Marketplace AWS License Manager との統合に関するドキュメントを追加](#)

License Manager と統合することで、AWS Marketplace Containers Anywhere 製品の契約料金を追加するための詳細なガイダンスを追加するためのドキュメントのみの更新。

2022 年 2 月 1 日

[SaaS SNS 通知の更新](#)

SaaS 通知メッセージを明確にするためのドキュメントのみの更新です。

2022 年 1 月 25 日

[販売者が Amazon Web Services EMEA SARL を通じて EMEA に拠点を置く購入者と取り引きする機能](#)

対象となる AWS Marketplace 販売者は、Amazon Web Services EMEA SARL を通じて、欧州、中東、アフリカ (EMEA) の国と地域を拠点とする AWS アカウント 顧客と取引できるようになりました。

2022 年 1 月 7 日

[Helm チャートの配送方法を使用してコンテナベースの商品の配送オプションを作成する方法に関するドキュメントを追加しました](#)

販売者は Helm チャートの配送方法で配送オプションを提供できるようになりました。購入者は、販売者が提供する Helm チャートを起動環境にインストールすることにより、これらの配送オプションを使用してコンテナベースのアプリケーションを起動できます。Helm チャート配信方法を提供する場合、販売者は購入者のクイック起動を有効にできます。クイック起動は、購入者が新しい Amazon EKS クラスター AWS CloudFormation をすばやく作成してコンテナベースのアプリケーションを起動するために使用できる機能です。

2021 年 11 月 29 日

[既存のポリシーに対する更新](#)

AWS Marketplace 販売者のセキュリティポリシーが更新されました。

2021 年 11 月 22 日

[AMI およびコンテナベースの製品の契約料金](#)

独立系ソフトウェアベンダー (ISV) は、AMI ベースの新製品またはコンテナベースの製品を出品し、購入者に前払いの契約料金を提示できるようになりました。

2021 年 11 月 17 日

[ベンダー計測タグ付け](#)

コード例を含む、ベンダー計測タグ付けに関するドキュメントのみの更新です。

2021 年 11 月 11 日

[AMI またはコンテナ製品用の Amazon Simple Notification Service](#)

独立系ソフトウェアベンダー (ISV) は、顧客が Amazon 簡易通知サービスを通じて AMI またはコンテナ製品をサブスクライブまたはサブスクライブ解除したときに通知を受け取ることができます。

2021 年 11 月 10 日

[新しい販売者アクセス許可](#)

AWS Marketplace に、のオファーとパートナータブにアクセスするための新しいアクセス許可が追加されました
AWS Marketplace 管理ポータル。

2021 年 11 月 9 日

[機械学習製品の非同期推論用に設定されたエンドポイントをデプロイできるようになりました](#)

最大よりも高いペイロード推論を期待する機械学習ソフトウェアの場合、または呼び出しあたりの最大処理時間を超える処理時間を必要とする場合、購入者は Amazon SageMaker AI 非同期推論用に設定されたエンドポイントをデプロイできます。

2021 年 11 月 8 日

返金ポリシーと承認	返金ポリシーを明確にし、すべての返金情報をAWS Marketplace 販売者ガイドの 1 か所にまとめるためのドキュメントのみの更新です。	2021 年 8 月 20 日
コンサルティングパートナーのプライベートオフアーの EULA を選択またはアップロードする	独立系ソフトウェアベンダーは、コンサルティングパートナーに再販の機会を創出する際に、エンドユーザー使用許諾契約 (EULA) を選択またはアップロードできるようになりました。	2021 年 8 月 17 日
SaaS 契約製品のカスタム製品ディメンション	独立系ソフトウェアベンダー (ISV) は、コンサルティングパートナーに再販機会を創出する際に、SaaS 契約製品の規模をカスタマイズできるようになりました。	2021 年 8 月 17 日
AWS Marketplace フィールドデモンストレーションプログラム	AWS Marketplace フィールドデモンストレーションプログラムの AWS Data Exchange データセット製品の要件を明確にするためのドキュメントのみの更新。	2021 年 8 月 3 日
SaaS 製品ガイドラインの更新	SaaS 製品の製品ガイドラインが更新されました。	2021 年 7 月 29 日
コンテナベースの製品要件の更新	コンテナベースの製品要件が更新されました。	2021 年 7 月 29 日
AMI セキュリティポリシーの更新	AMI 製品のセキュリティポリシーが更新されました。	2021 年 7 月 29 日

対象管轄区域の追加	現在、AWS Marketplace香港特別行政区とカタールで販売者になる資格があります。	2021 年 6 月 23 日
データフィードの概要	販売者が利用できるデータフィードの構造の概要を説明するドキュメントのみの更新。	2021 年 6 月 23 日
「機械学習」の章を更新しました	機械学習製品の作成と保守に関する情報のドキュメントのみの更新。	2021 年 5 月 27 日
コンテナ製品のセルフサービスアップデート	販売者は、AWS Marketplace管理ポータルを使用して、コンテナベースの商品をより簡単かつ迅速に更新できるようになりました。	2020 年 12 月 17 日
プロフェッショナルサービス	販売者は AWS Marketplace、購入者にプロフェッショナルサービスを提供できるようになりました。ドキュメントにプロフェッショナルサービスのセクションを追加しました。	2020 年 12 月 3 日
AMI 製品のセルフサービスアップデート	販売者は、AWS Marketplace管理ポータルを使用して Amazon マシンイメージ (AMI) ベースの商品をより簡単かつ迅速に更新できるようになりました。	2020 年 11 月 23 日

対象管轄区域の追加

AWS Marketplaceバーレーン、ノルウェー、スイス、アラブ首長国連邦 (アラブ首長国連邦) で販売される資格を得ました。

2020 年 6 月 17 日

承諾済みのプライベートオファ어의アップグレードや更新を提供できます

SaaS 契約製品および従量課金制の SaaS 契約製品については、承諾済みのプライベートオファ어に対するアップグレードや更新をプライベートオファ어として提供できます。

2020 年 5 月 28 日

詳細はデータフィードで利用できます。

データの検索と分析を簡素化するために、レポートの詳細情報を小さいデータフィードに分割します。

2020 年 5 月 21 日

標準化されたライセンス条項が利用可能になりました

カスタム EULA の代わりに標準化されたライセンス条項を提供して契約プロセスを簡素化できます。

2020 年 4 月 28 日

オーストラリアとニュージーランドは対象の管轄区域です

販売者は、AWS Marketplace(i) オーストラリア (AU) ニュージーランド (NZ) の永住者と市民、または (ii) いずれかの地域で組織化または法人化された事業体です。

2020 年 4 月 2 日

コンテナ製品でカスタム計測と料金設定の拡張がサポートされるようになりました

独自の料金単位を定義し、その請求使用量を計測する場合は、AWS Marketplace Metering Service の meterUsage アクションと統合します。

2019 年 11 月 14 日

AWS Marketplace が AWS Data Exchange を通じてデータ製品をサポート	AWS Marketplaceでデータ製品を提供できるようになりました。	2019 年 11 月 13 日
AWS Marketplace Catalog API サービスの紹介	AWS Marketplace Catalog API サービスは、承認された販売者がプログラムで製品を管理するための API インターフェイスを提供します。	2019 年 11 月 12 日
AWS Marketplace が有料の時間単位コンテナをサポート	AWS Marketplace は、Amazon Elastic Kubernetes Service (Amazon EKS) で実行されている有料の時間単位コンテナをサポートするようになりました。	2019 年 9 月 25 日
更新された AMI 製品の機能。	を使用して AMIs と Lambda 関数を一緒にデプロイできるようになりました AWS CloudFormation。	2019 年 9 月 11 日
セキュリティセクションが追加されました	新しいセキュリティセクションの下にセキュリティコンテンツを統合しました。	2019 年 5 月 7 日
AMI Security Policies を更新しました	AMI 製品のセキュリティポリシーを更新しました。	2019 年 4 月 11 日
バージョニング情報を機械学習製品セクションに追加しました	機械学習製品の製品バージョンの機能について説明するコンテンツを追加しました。	2019 年 3 月 21 日
機械学習製品セクションを追加しました	機械学習製品の公開に関するコンテンツを追加しました。	2018 年 11 月 28 日

<u>コンテナベースの製品セクションを追加しました</u>	コンテナベース製品の公開に関するコンテンツを追加しました。	2018 年 11 月 27 日
<u>販売者のヘルプリクエスト送信のリンクを更新しました</u>	E メールアドレスをウェブフォームアドレスに変更しました。	2018 年 10 月 22 日
<u>消費コンテンツに関する SaaS 契約を追加しました</u>	SaaS コンテンツを再構築し、消費機能付きの SaaS 契約のリリースをサポートするためのコンテンツを追加しました。	2018 年 10 月 18 日
<u>プライベートオファー用の Flexible Payment Schedule に関するコンテンツを追加しました</u>	プライベートオファーコンテンツ用の Flexible Payment Scheduler のリリースをサポートするためのコンテンツを追加しました。	2018 年 10 月 15 日
<u>IAM アクセス許可のコンテンツを更新しました</u>	AMMP 読み取り専用アクセス権のための新しい IAM のアクセス許可をサポートするためのコンテンツを追加しました。	2018 年 10 月 9 日
<u>コンサルティングパートナーのプライベートオファーに関するコンテンツを追加しました</u>	コンサルティングパートナーのプライベートオファー機能をサポートするためのコンテンツを追加しました。	2018 年 10 月 9 日
<u>プライベートイメージビルドに関するコンテンツを追加しました</u>	AMI 機能用の Private Image Build のリリースをサポートするためのコンテンツを追加しました。	2018 年 8 月 13 日

[販売者の検索エンジン最適化のガイダンスを追加しました。](#)

検索のために製品の最適化を望む販売者向けのガイダンスを追加しました。

2018 年 7 月 3 日

[AWS Marketplace ロゴへのリンクを更新しました](#)

新しい AWS Marketplace ロゴを指すようにリンクを更新しました。

2018 年 12 月 6 日

[販売者向けガイドを追加しました](#)

PDF 版の販売者向けガイドをすべてオンラインコンテンツに変換しました。

2018 年 5 月 9 日

AWS 用語集

最新の AWS 用語については、AWS の用語集 リファレンスの[AWS 用語集](#)を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。