



販売者ガイド

AWS Marketplace



AWS Marketplace: 販売者ガイド

Copyright © 2024 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
販売者としての AWS Marketplace の使用	1
の契約構造 AWS Marketplace	3
での製品の料金 AWS Marketplace	4
使用開始	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	9
AWS Marketplace 販売者としての登録	10
ステップ 1: パブリックプロフィールを作成する	10
ステップ 2: 税金情報を提供する	11
ステップ 3: 銀行口座情報を提供する	12
ステップ 4: 顧客を知る (KYC) プロセスを完了する	14
ステップ 5: 銀行口座の検証プロセスを完了する	17
すでに AWS Marketplace 販売者ですか？	18
支払い設定	18
支払い設定の設定	19
AWS Marketplace サブスクリプションの請求	19
販売者が支払いを受ける方法	20
利用可能な通貨	21
出品料金	21
パブリックオファーの出品料金	21
プライベートオファーの出品料金	22
チャンネルパートナーのプライベートオファー (CPPO) のリスティング料金	22
プロフェッショナルサービスの出品料金	22
Amazon Payments Europe の苦情	22
苦情の申し立て	23
Amazon Payments Europe の苦情解決期間	23
苦情のエスカレーション	23
AWS Marketplace 販売者向けの追加ツール	24
AWS Marketplace コマース分析サービス	25
AWS Marketplace フィールドデモンストレーションプログラム	44
製品の準備	46

製品デリバリー	46
製品の料金	50
料金モデル	51
料金モデルの変更	55
料金の変更	55
プライベートオファー	56
製品の返金	56
リージョンと国	59
AWS リージョン	59
Countries (国)	60
標準契約書	60
の標準契約 AWS Marketplace	61
のリセラー契約 AWS Marketplace	63
カテゴリとメタデータ	63
製品の名前付けと説明	64
カテゴリとキーワードの選択	65
AMI およびコンテナ製品の使用方法	66
要件	66
リリースノートを書く	67
使用手順を書く	67
アップグレード手順を書く	68
CloudFormation 配信手順の記述	68
アプリケーション機能のモニタリングと評価	69
プログラムによるシステム認証情報と暗号化キーのローテーション	69
検索エンジンの最適化	69
検索エンジンの最適化	70
AWS Marketplace 検索	71
プライベートオファーの準備	74
プライベートオファーの仕組み	75
プライベートオファーに関する考慮事項	75
購入者向けのプライベートオファーエクスペリエンス	76
プライベートオファーのレポート	77
サポートされる製品タイプ	78
AMI 製品のプライベートオファー	78
コンテナ製品のプライベートオファー	79
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	80

SaaS 製品のプライベートオファー	80
ML 製品のプライベートオファー	80
プライベートオファーの作成と管理	81
新しいプライベートオファーの開始	82
オファーのステータスについて	82
プライベートオファーの起案と公開	83
購入者へのプライベートオファーの送信	85
プライベートオファーの進行状況を保存する	85
プライベートオファーの有効期限の更新	86
プライベートオファーのキャンセル	86
チャンネルパートナーのプライベートオファー	87
追加情報	89
として販売承認を作成する ISV	89
分割プランの作成	91
支払いスケジュールの作成	92
分割プランのレポート	93
プライベートオファーの修正	93
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	94
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	95
アップグレード、更新、および修正の報告	96
将来の日付の契約	97
将来のデータ契約に関する考慮事項	98
将来の日付の契約の作成	98
将来の日付の契約で分割プランを使用する	99
将来の日付の契約の通知を受け取る	99
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	99
AMIベースの製品	101
AMIベースの製品配信方法	101
追加リソース	101
AMIベースの製品について	102
製品のライフサイクル	102
AMI 製品コード	105
変更リクエスト	105
製品ロードフォーム	107
年間契約の変更	107

単一AMI製品の作成	108
前提条件	108
セルフサービスエクスペリエンスを理解する	109
リストを作成する	110
追加リソース	112
単一AMI製品の管理	113
変更リクエストの作成	114
製品の可視化の更新	117
インスタンスの追加と制限	119
バージョンの管理	121
製品情報の更新	127
可用性の管理	128
の更新 EULA	131
返金ポリシーの更新	132
AWS Marketplace へのアクセスの許可 AMI	132
製品を削除する	133
トラブルシューティング	135
の使用 CloudFormation	136
製品の出品の構築	137
CloudFormation テンプレートの準備	137
テンプレートインフラストラクチャの料金見積りの取得	140
アーキテクチャ図	140
送信要件の一致	141
製品リクエストの送信	142
サーバーレスアプリケーションコンポーネントの追加	143
構築のベストプラクティス AMIs	152
再販権の確保	152
の構築 AMI	152
AMI 用の の準備と保護 AWS Marketplace	153
のパブリッシュ要件AMIのスキャン	154
でソフトウェアが実行されていることを確認する AWS Marketplace AMI	155
AMI 製品料金	156
AMI 料金モデル	157
AWS 料金とソフトウェア料金	161
AWS Marketplace Metering Service を使用したカスタム計測	162
AMI 製品の契約料金	178

の使用 AWS License Manager	181
Amazon SNS通知の受信	195
Amazon SNSトピック: aws-mp-subscription-notification	196
Amazon SQSキューを Amazon SNSトピックにサブスクライブする	196
AMI 製品チェックリスト	197
AMIベースの製品要件	199
セキュリティポリシー	199
アクセスポリシー	200
カスタマー情報ポリシー	202
製品使用ポリシー	202
アーキテクチャポリシー	203
AMI 製品の使用方法	204
コンテナベースの製品	205
ヘルプの利用	206
コンテナ製品の開始方法	206
製品のライフサイクル	206
前提条件	207
概要: コンテナ製品を作成する	208
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	208
ステップ 2: 初期リストを作成する	209
ステップ 3: 製品の初期バージョンを追加する	209
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	210
ステップ 5: 製品の可視性を更新する	210
次のステップ	211
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。	212
製品情報の更新	212
製品バージョンの追加	213
製品料金の管理	223
国別の可用性の更新	226
の更新 EULA	227
製品のテストとリリース	227
コンテナベースの製品要件	229
セキュリティ要件	230
アクセスの要件	230
顧客情報の要件	231
製品の使用要件	231

アーキテクチャの要件	232
コンテナ製品の使用手順	233
Amazon EKS アドオン製品の要件	233
コンテナ製品の料金	249
コンテナ料金モデル	250
コンテナ製品の契約料金	254
コンテナ製品の請求、計測、ライセンスの統合	257
AWS Marketplace Metering Serviceによる時間単位計測およびカスタム計測	258
との契約料金 AWS License Manager	261
AWS Marketplace Metering Service を使用した計測の設定	262
AWS Marketplace Metering Service を使用したカスタム計測の設定	274
との契約料金 AWS License Manager	287
コンテナ製品の Amazon SNS通知	321
Amazon SNSトピック: aws-mp-subscription-notification	321
Amazon SQSキューを Amazon SNSトピックにサブスクライブする	322
Machine Learning 製品	323
機械学習製品の使用を開始する	323
SageMaker モデルパッケージ	323
SageMaker アルゴリズム	324
推論モデルのデプロイ	324
セキュリティと知的財産	325
知的財産の保護	325
ネットワークアクセスなし	325
顧客データのセキュリティ	326
機械学習製品の料金	326
インフラストラクチャの料金	327
ソフトウェアの料金	327
で製品を準備する SageMaker	330
コードをイメージにパッケージ化する	330
イメージのアップロード	354
Amazon SageMaker リソースの作成	357
AWS Marketplaceでの製品の公開	364
前提条件	364
公開プロセスの概要	365
必要なアクセス許可	365
製品リストの作成	366

製品のテスト	374
公開のサインオフ	375
製品の更新	375
要件とベストプラクティス	377
必須アセット	377
ML 製品の一般的なベストプラクティス	378
使用状況情報の要件	379
入力と出力の要件	379
Jupyter Notebook の要件	380
ML 製品リストの要件と推奨事項の要約	381
サービスの制限とクォータ	385
ネットワークの隔離	385
イメージのサイズ	386
ストレージサイズ	386
インスタンスサイズ	386
推論のペイロードサイズ	386
推論の処理時間	386
Service Quotas	387
非同期推論	387
サーバーレス推論	387
マネージドスポットトレーニング	387
Docker イメージと AWS アカウント	387
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	387
公開 AWS リージョン でサポート	388
発行エラーのトラブルシューティング	388
での機械学習レポート AWS Marketplace	390
日別ビジネスレポート	391
月別収益レポート	391
支払いレポート	391
その他のレポートと分析	391
SaaS 製品	392
SaaS 製品を開始する	392
前提条件	393
次のステップ	393
SaaS 製品のライフサイクル	393
SaaS 製品の作成	394

初期 SaaS 製品ページの作成	397
SaaS 製品設定の設定	398
SaaS サブスクリプション製品の統合	409
SaaS 契約製品の統合	412
SaaS 契約と pay-as-you-go製品の統合	416
サーバーレス SaaS 統合ソリューションのデプロイ	420
SaaS 製品の計画	421
料金の計画	422
請求の統合の計画	422
Amazon SNS統合を計画する	423
顧客が製品にアクセスする方法の計画	423
SaaS 製品ガイドライン	424
製品設定ガイドライン	424
顧客情報の要件	425
製品使用ガイドライン	425
アーキテクチャガイドライン	426
SaaS 製品の料金	427
SaaS サブスクリプションの料金	428
SaaS 契約の料金	429
SaaS 無料トライアルの作成	434
SaaS 無料トライアルオファターの作成	434
SaaS 無料トライアルオファターのキャンセル	435
顧客オンボーディング	436
新しい購入者を受け入れるための SaaS 製品の設定	437
SaaS 製品の Amazon SNS通知	439
Amazon SNSトピック: aws-mp-entitlement-notification	440
Amazon SNSトピック: aws-mp-subscription-notification	441
SNS トピックへのSQSキューのサブスクライブ	442
Metering and Entitlement Service へのアクセス AWS Marketplace APIs	443
使用量の計測	443
使用権限の確認	450
SaaS 製品統合チェックリスト	451
レポート作成	455
SaaS のコード例	456
ResolveCustomer コード例	456
GetEntitlement コード例	457

BatchMeterUsage コード例	459
使用量割り当てタグ付きの BatchMeterUsage のコード例 (オプション)	460
の使用 AWS PrivateLink	462
序章	463
製品の設定	464
製品の送信先 AWS Marketplace	465
VPC エンドポイントへの購入者アクセス	465
付録: チェックリスト	467
のプロフェッショナルサービス製品 AWS Marketplace	469
ヘルプの利用	470
プロフェッショナルサービス製品の開始方法	470
前提条件	470
プロフェッショナルサービス製品を作成する	471
プライベートオファーを作成する	472
製品情報の編集	474
製品の料金の編集	475
製品の可視性を編集する	476
プロフェッショナルサービス製品を削除する	476
製品の詳細の提供	477
製品の説明	478
追加リソース	480
サポート情報	480
料金ディメンション	480
製品の可視性	480
製品要件	481
製品設定ガイドライン	481
顧客情報の要件	482
製品使用ガイドライン	482
アーキテクチャガイドライン	482
プロフェッショナルサービス製品の料金	483
プライベートオファーの仕組み	483
データ製品	485
製品の送信	486
[製品] タブの使用	488
会社と製品のロゴ要件	489
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	489

ハードウェアコンポーネントを備えた製品の要件	490
AWS CloudFormation- 起動済み製品 (無料または有料) または使用ベースの有料AMI製品	491
製品の更新	495
製品の変更と更新	496
タイミングと心構え	497
AMIs への送信 AWS Marketplace	497
AMI セルフサービススキャン	497
AMI クローン作成と製品コードの割り当て	498
最終チェックリスト	498
製品のマーケティング	501
180 日間のGTMアカデミー	501
製品が入手可能になったことの発表	502
AWS Marketplace メッセージング	502
のレビュー AWS Marketplace	503
へのリンク AWS Marketplace	504
AWS Marketplace ブランディングの使用	504
で製品に直接リンクする AWS Marketplace	505
プレスリリース	506
AWS Marketplace 商標使用ガイドライン	507
通知	509
E メール通知	510
イベントタイプ	510
フィールドの説明	521
通知を管理する	526
Amazon EventBridge 通知	527
新しいオファアのイベント	528
変更セットのイベント	532
セキュリティ概要レポートのイベント	534
出品者レポート、データフィード、ダッシュボード	536
販売者の配信データフィード	537
データフィードの保存と構造	537
データフィードへのアクセス	540
データフィードの使用	545
データフィードテーブルの概要	546
データフィードクエリの例	554
データフィード	677

販売者レポート	719
レポートへのアクセス	720
利用可能な AWS Marketplace 販売者レポート	721
日別ビジネスレポート	721
日次顧客サブスクライバーレポート	731
支払いレポート	734
月別請求済み収益レポート	743
販売補償レポート	752
補足レポート	755
契約詳細レポート	756
販売者ダッシュボード	757
ダッシュボードへのアクセス	720
財務業務用ダッシュボード	760
販売業務用ダッシュボード	788
マーケティング用のダッシュボード	805
AWS Marketplace Vendor Insights	819
AWS Marketplace Vendor Insights について	820
販売者として設定する	821
セキュリティプロファイルを作成する	821
証明書のアップロード	822
自己評価をアップロードする	824
AWS Audit Manager 自動評価を有効にする	825
プロファイルの表示	831
販売者としてのセキュリティプロファイルを確認する	832
スナップショットの管理	833
スナップショットを作成する	834
スナップショットを表示する	834
スナップショットをエクスポートする	835
最新リリースのスナップショットを表示する	835
スナップショットリリースを延期する	835
スナップショットリストの設定を変更する	836
アクセスの制御	836
AWS Marketplace Vendor Insights 販売者のアクセス許可	837
CreateDataSource	838
DeleteDataSource	838
GetDataSource	838

UpdateDataSource	839
ListDataSources	839
CreateSecurityProfile	839
ListSecurityProfiles	839
GetSecurityProfile	840
AssociateDataSource	840
DisassociateDataSource	840
UpdateSecurityProfile	840
ActivateSecurityProfile	840
DeactivateSecurityProfile	841
UpdateSecurityProfileSnapshotCreationConfiguration	841
UpdateSecurityProfileSnapshotReleaseConfiguration	841
ListSecurityProfileSnapshots	842
GetSecurityProfileSnapshot	842
TagResource	842
UntagResource	842
ListTagsForResource	843
追加リソース	843
セキュリティ	199
IAM の AWS Marketplace	845
ユーザーの作成	847
グループの作成または使用	848
ユーザーとしてのサインイン	850
AWS Marketplace 販売者のポリシーとアクセス許可	850
ポリシー	851
アクセス許可	852
AWS マネージドポリシー	859
AWSMarketplaceAmiIngestion	860
AWSMarketplaceFullAccess	861
AWSMarketplaceGetEntitlements	863
AWSMarketplaceMeteringFullAccess	864
AWSMarketplaceMeteringRegisterUsage	864
AWSMarketplaceSellerFullAccess	865
AWSMarketplaceSellerProductsFullAccess	868
AWSMarketplaceSellerProductsReadOnly	870
AWSVendorInsightsVendorFullAccess	871

AWSVendorInsightsVendorReadOnly	873
ポリシーの更新	874
AWS Marketplace Commerce Analytics Service アカウントのアクセス許可	879
Amazon SQS のアクセス許可	880
AWS Marketplace 計測と使用権限 API のアクセス許可	881
SaaS 製品の IAM ポリシー	881
AMI 製品の IAM ポリシー	882
コンテナ製品の IAM ポリシー	882
再販認証のサービスにリンクされたロール	883
のサービスにリンクされたロールのアクセス許可 AWS Marketplace	884
のサービスにリンクされたロールの作成 AWS Marketplace	887
のサービスにリンクされたロールの編集 AWS Marketplace	887
のサービスにリンクされたロールの削除 AWS Marketplace	888
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	888
を使用した通話のログ記録 AWS Marketplace API AWS CloudTrail	888
AWS Marketplace 計測APIログファイルエントリの例	889
ドキュメント履歴	895
AWS 用語集	917
.....	cmxviii

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、ビジネスを実行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、データ製品などの一般的なカテゴリから、医療、金融サービス、通信などの特定の業界にまたがる数千ものソフトウェアリストが含まれています。お客様は、事前設定されたソフトウェアをすばやく起動し、Amazon Machine Images (AMIs)、Software as a Service (SaaS)、およびその他の形式でソフトウェアソリューションを選択できます。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。

を購入者 (サブスクライブ)、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、販売者になるために登録できます。販売者は、独立した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 の資格要件を満たしている必要があります。詳細については、「[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。AWS

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、での の販売について詳しく説明します AWS Marketplace。

[AWS Marketplace の概要](#)

販売者としての AWS Marketplace 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	販売者として、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 パートナーの税務情報が管轄の適格性基準を満たしていることを確認します。支払いを受け取るには、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。 <ul style="list-style-type: none">• AMIベースの製品 AWS Marketplace• で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「Data Exchange とは？」を参照してください。) AWS
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アの両方で 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、1つの価格構造を持つ単一の Amazon マシンイメージ (AMI) など、シンプルな場合があります。または、複数の、テンプレートAMIs、AWS CloudFormation 複雑な料金オプションと支払いスケジュールにより、製品が複雑になる場合があります。

[ステップ]	[アクション]	説明
5	製品を売る	の商品ページへのトラフィックを直接促進 AWS Marketplace することで、 の認識を高め、商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。

販売者として、お客様の契約には以下が含まれます。

- 購入者とのエンドユーザーライセンス契約 (EULA)。購入者は、 のパブリックソフトウェアリストの製品リストページにあります AWS Marketplace。多くの販売者は、デフォルトのとして [Standard Contract for AWS Marketplace \(SCMP \)](#) を使用しますEULA。をプライベートオファーでの交渉の基礎SCMPとして使用し、修正テンプレートを使用して を変更することもできます SCMP。プライベートオファーには、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。

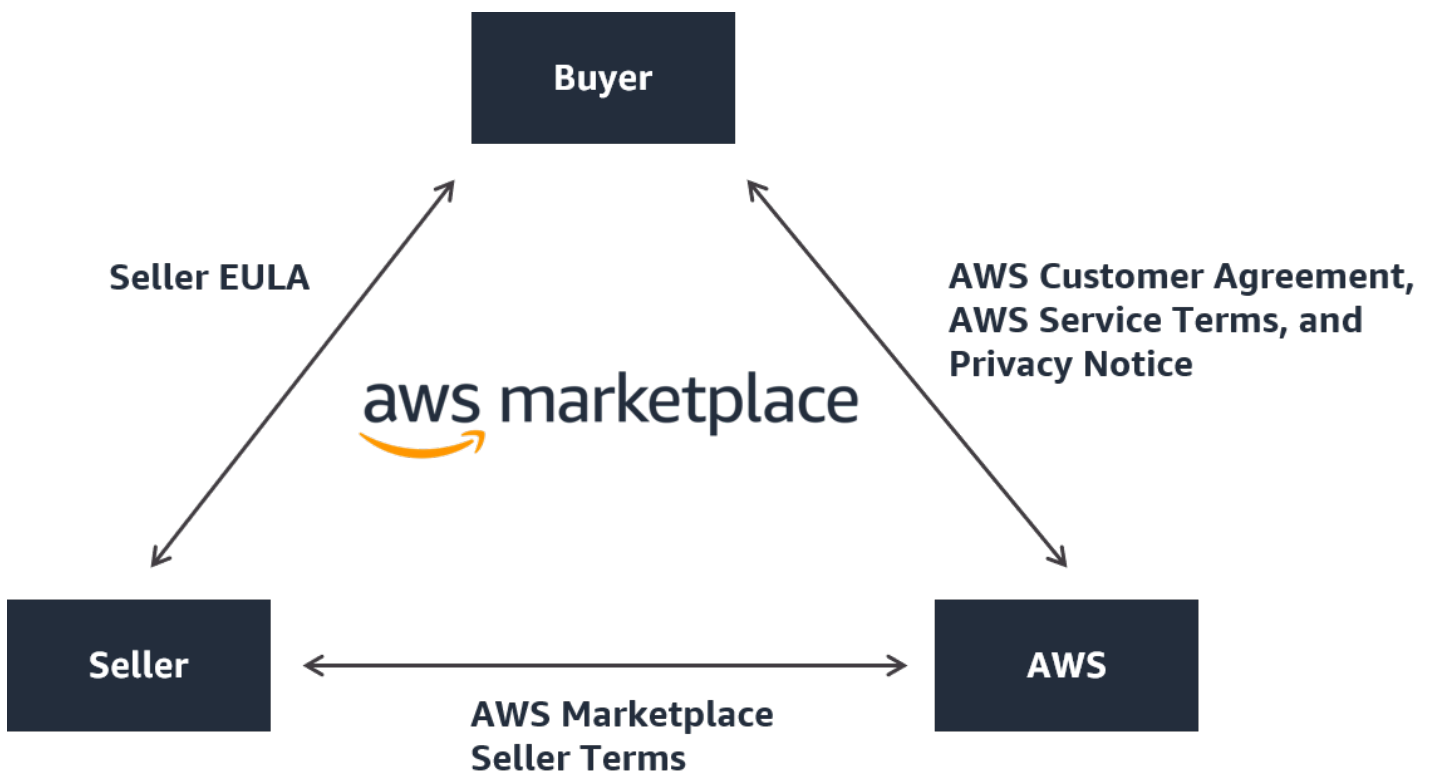
Note

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

- でのお客様のアクティビティに適用される[AWS Marketplace](#)、[販売者のサービス条件 AWS Marketplace](#)。

購入者によるの使用 AWS Marketplace には、[AWSサービス条件](#)、[AWS顧客契約](#)、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、料金が関連付けられている場合があります。料金は購入者の AWS 請求書の一部となり、購入者が支払いを行った後に販売者に 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、購入者のを使用してインスタンス化された Amazon マシンイメージ (AMI) として製品を提供できます AWS アカウント。購入者への配信にテン

プレートを使用する CloudFormation のように製品を設定することもできます。製品は、ウェブアクセスコントロールリスト (ウェブ ACL) ISV、ルールのセット、または の条件からの SaaS オフリングでもあり得ます AWS WAF。製品は、チャンネルパートナー ISV、または からのプロフェッショナルサービスでもあり得ます MSP。

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。請求と支払い AWS を処理し、請求額はお客様の AWS 請求に表示されます。

ソフトウェア製品は、ISV の標準エンドユーザーライセンス契約 () を使用して、表示された価格で購入できます EULA。さらに、ソフトウェア製品は、カスタム料金とプライベートオファー EULA を通じて提供できます。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、購入者の組織内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

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、[AWS Marketplace](#) で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、[AWS Marketplace](#) で販売する製品を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、[AWS Marketplace](#) 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](#)と[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](#)の詳細をご覧ください。

📌 メモ

- [AWS Marketplace](#) 販売者として登録することは、データ製品を [AWS Data Exchange](#) に一覧表示し、[AWS Marketplace](#) で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、[「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](#)
- [AWS Marketplace](#) 販売者が必要とするアクセス許可については、「[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。
- 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

で提供するとき製品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方法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、以下を行う必要があります。

- 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 顧客になり、AWS Marketplace 販売者の利用規約の要件を満たします。

有料製品の販売者の追加要件

製品の料金を請求するか、Bring Your Own License Model (BYOL) 製品を提供する場合は、以下の要件を満たし、この追加情報も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。
- 米国以外の販売者は、(i) W-8 フォーム、付加価値税 (VAT)、または商品サービス税 (GST) 登録番号、および (ii) 対象となる管轄区域のSWIFTコードを持つ銀行口座を提供する必要があります。必要に応じて、[Hyperwallet](#) から仮想米国銀行口座に登録できます。
- データ製品を提供するには、の[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 AWS Support。
- Amazon Web Services 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EMEASARL、[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 を介したトランザクションの場合SARL)。
 - 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](#) のヘルプページを参照してください。リスト料金に対する付加価値税 (VAT) が評価された場合、AWS Marketplace は税準拠の請求書を提供します。
 - Amazon Web Services の詳細についてはSARL、[Amazon Web Services Europe FAQs](#) EMEA ウェブサイトの AWS EMEA Marketplace - Sellers を参照してください。

に販売するには AWS GovCloud (US) Region、販売者は[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 販売者オペレーション](#) 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 で販売するには AWS Marketplace、次のいずれかの国または の永住者または市民であるかSARs、その中に組織または法人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}
- 欧州連合 (EU) 加盟国¹
- 香港 SAR
- イスラエル^{1 2}
- 日本^{1 2 3}
- ニュージーランド¹
- ノルウェー^{1 2}
- カタール
- 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住んでいる場合、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 日本を拠点とする販売者には、日本の消費税 (JCT) の出品手数料を自己計上する義務があります。他の法域に拠点を置く販売者も同様の義務を負っている場合があります。税務顧問にお問い合わせください。

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

事業体がいずれかの国に法人化されていない場合、または上記の国に法人化されていない場合は、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。

AWS Marketplace 管理ポータル

を使用して[AWS Marketplace 管理ポータル](#)、で販売する製品を管理できます AWS Marketplace。登録済みのすべての販売者は、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。
- アクティビティを活用して go-to-market、ソフトウェア製品を増分チャネル収益に管理します。
- 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。
- Amazon マシンイメージ (AMI) の自動スキャンを開始し、脆弱性を検出します。

Note

データ製品は Data Exchange AWS コンソールから発行および管理されます。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 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

独立した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である場合は、の販売者として登録できません AWS Marketplace。販売者として登録することは、で販売する製品を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、で販売者として正常に登録するステップ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ています。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 “IAM の AWS Marketplace”](#)」を参照してください。

販売者として登録するには、次の手順に従います。

トピック

- [ステップ 1: パブリックプロフィールを作成する](#)
- [ステップ 2: 税金情報を提供する](#)
- [ステップ 3: 銀行口座情報を提供する](#)
- [ステップ 4: 顧客を知る \(KYC\) プロセスを完了する](#)
- [ステップ 5: 銀行口座の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: パブリックプロフィールを作成する

登録の最初のステップは、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 AWS Marketplace 潜在的な購入者に表示される情報を提供することです。このアカウントは、での製品の登録販売者 AWS Marketplace となり、からお客様 AWS Marketplace への報告、支払い、および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、に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。新しいアカウントを使用して AWS

Marketplace 販売者として登録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、選択した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: 税金情報を提供する

が製品売上に対する税金 AWS Marketplace を正確に報告および源泉徴収できるように、税金と付加価値税 (VAT) を必要に応じて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。
 - でプロフェッショナルサービスを販売するには AWS Marketplace、 の税金アンケート DAC7を完了する必要があります。
 - 「税務インタビュー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、に入力された情報と一致していることを確認します AWS Marketplace 管理ポータル。税務面接の場所は、ビジネスの場所と一致する必要があります。
4. 税金情報の入力が完了したら、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。この選択は、AWS Billing コンソールの税金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、AWS リージョン をサポートする がある場合にのみ使用できます VAT。

税金ドキュメントにアクセスする

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. Payment Information セクションに移動します。
3. 関連する納税申告書 (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、納税ダッシュボードページでダウンロードできます。

ステップ 3: 銀行口座情報を提供する

で有料製品を販売したい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 AWS Marketplace。

Note

で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](#)。

銀行情報を提供するには

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。1 つ以上の銀行口座を指定することもできます。これらの銀行口座は、米国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 アカウントです。

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USD支払いを受け入れる場合は、Hyperwallet は、米国アカウントを提供できます。

Hyperwallet は、サポートされている通貨で別の銀行口座に資金を送金できる独立したサービスプロバイダーです。期間限定で、特定の Hyperwallet AWS Marketplace 支払いに関連するサービス料金。

- を追加する Hyperwallet アカウントの詳細を AWS Marketplace 販売者アカウントに提供する場合、AWS Marketplace がお客様の名前、Eメールアドレス、およびアカウント番号をと共有することに合意し、承認します。Hyperwallet 販売者としてのステータス AWS Marketplace を確認します。
- の使用には追加料金が適用される場合があります。Hyperwallet サービス (送金手数料、現地通貨への資金の送金に必要な外国為替手数料を含む) と外国為替レート。- Hyperwallet サービス料金は、有料製品からへの収益 AWS Marketplace の支払いに関してのみ、期間限定で免除されます。Hyperwallet アカウント。詳細については、「」の「料金」セクションを参照してください。Hyperwallet サイトまたは問い合わせ Hyperwallet 詳細については、「」を参照してください。また、適用される料金を確認します。サービスの詳細については、「」を参照してください。
[Hyperwallet サポートサイト](#)。

で登録を開始するには Hyperwallet 米国の銀行口座情報を取得する

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. をお持ちでない場合 Hyperwallet アカウント。で使用するには アカウントが必要です。に回答していいえ AWS Marketplace を選択します。米国の銀行口座をお持ちですか？ および に登録されていますか。Hyperwallet？ にサインアップするための個人識別番号 (PIN) とリンクが提供されます。Hyperwallet.
3. をアクティブ化した後 Hyperwallet アカウントで、「」で説明されているステップに従います。Hyperwallet 登録を完了し、預金アカウント情報を受け取るための登録ポータル。
4. アカウントを から取得した場合 Hyperwallet、 を追加します。Hyperwallet にサインイン AWS アカウントして、 のアカウント情報を取得します [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。

ステップ 4: 顧客を知る (KYC) プロセスを完了する

販売者として英国ベースの銀行口座または Amazon Web Services を使用して取引するには EMEASARL、KYCプロセスを完了する必要があります。このプロセスには、会社に関する追加情報、主な連絡先、実質的所有者、補足書類の提供が含まれます。

Know Your Customer (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、銀行やその他の支払い機関などの金融機関に適用される、改訂された Payment Services Directive (PSD 2) および欧州連合のマネーロンダリング防止指令によるものです。

AWS Marketplace Amazon Web Services 経由のトランザクション EMEASARL は、支払いサービスを使用するためにお客様の ID の検証を必要とするルクセンブルクの認可された電子金融機関である Amazon Payments Europe, S.C.A. (APE) を通じて処理されます。

Note

英国に居住する銀行口座は、EUR および 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。さらに、Amazon Web Services EMEA による販売に限定されます SARL。

KYC プロセスを完了するには

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. Go to KYC information を選択するか、Know your customer (KYC) タブを選択し、Start KYC Compliance 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. Know Your Customer (KYC) Overview で、必要な情報とドキュメントのリストを読み、必要なドキュメントを収集します (まだ収集していない場合)。次に、KYCコンプライアンスに進むを選択します。
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、そのアクションは Amazon Payments Europe の利用規約に同意したことを示します。

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

[次へ] を選択して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. 法定代理人が連絡窓口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。
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ステータス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。KYC プロセス全体は通常約 2 週間かかります。

設定タブに戻って、アカウント概要カードでKYCコンプライアンスのステータスを表示できます。KYC ステータスの詳細については、アカウント概要カードの「顧客を知る (KYC)」タブを選択します。レビューが完了するまで [レビュー中] と表示されます。

KYC が検証されたら、を通じて支払いを受け取る前に、支払い情報タブに銀行取引明細書を指定する必要がありますAPE。

(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

ユーザーは、多要素認証 (MFA) を有効にして支払い情報を更新する必要があります。の詳細についてはMFA、[「の多要素認証 \(MFA \) IAM」](#)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、銀行口座の詳細などの財務情報の変更を行える個人です。

KYC 検証され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。

KYC 検証を受けるには、セカンダリユーザーが の手順を完了する必要があります [ステップ 4: 顧客を知る \(KYC\) プロセスを完了する](#)。

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. Know Your Customer (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参照してください。
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ステータス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。KYC プロセス全体は通常約 2 週間かかります。

ステップ 5: 銀行口座の検証プロセスを完了する

Amazon Payments Europe (APE) から支払いを受け取るには、「」の「支払い情報」タブに記載されている支払い銀行口座を確認するための追加情報を入力する必要があります AWS Marketplace 管理ポータル。

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。
5. 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。

を使用する場合 Hyperwallet 仮想銀行口座ソリューションについては、「」を参照してください [the section called “から銀行口座検証ステートメントをダウンロードする Hyperwallet”](#)。

6. 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。

から銀行口座検証ステートメントをダウンロードする Hyperwallet

を使用する販売者の場合 [Hyperwallet 仮想銀行口座ソリューション](#) では、Hyperwallet 次の手順を使用して銀行取引明細書を作成します。その後、[ステップ 5: 銀行口座の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。

から銀行取引明細書をダウンロードするには Hyperwallet

1. にサインインする [Hyperwallet アカウント](#)。
2. [\[預金口座情報\]](#) ページに移動します。
3. [\[銀行口座確認書\]](#) 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 AWS Marketplace と次のステップの詳細を確認してください。

- 支払い – AWS Marketplace 販売者として、選択した通貨で未払い残高を受け取るように支払い設定を設定できます。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。
- 一覧表示料金 – 販売者の一覧表示料金の詳細については、「」を参照してください。[AWS Marketplace 販売者の一覧表示料金](#)。AWS Marketplace
- Amazon Payments Europe (APE) AWS Marketplace での の苦情処理ポリシー – での 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 に対する AWS Marketplace 販売者の苦情の送信 \(APE \)](#)」を参照してください。
- 追加の販売ツール – 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 販売者向けの追加ツール](#)」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、 を通じて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を含む販売者は、未払い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。販売者として、日次または月次の支払いオプションを選択し、支払いを受け取る曜日を選択し、銀行口座に通貨を割り当てることができます。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、支払い設定を行う方法を示します。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、でのみ支払いを受け取ることができます USD。支払い以外の USD 場合は、SWIFT 銀行口座を指定する必要があります。

トピック

- [支払い設定の設定](#)
- [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](#)

支払い設定の設定

支払い設定を設定するには、以下の手順に従います。

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 では、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アカウントを選択します。
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

出品者は、支払い設定オプションにアクセスするためのきめ細かなIAMアクセス許可にオンボーディングされる必要があります。きめ細かなIAMアクセス許可にオンボードするには、「」を参照してください[the section called “AWS Marketplace 販売者のポリシーとアクセス許可”](#)。

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。

の詳細については 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の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](#)、[登録済み銀行口座](#)、および送信された W9 フォームが必要です。
- 有料製品の販売者は、W-8、付加価値税 (VAT)、または物品サービス税 (GST) 登録番号、および[対象管轄区域](#)の銀行口座を提供する必要があります。
- AWS は、以下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。[コレクション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、購入者が請求書を支払った後に、Automated Clearing House (ACH) の移管または 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のプライベートオファーは、USDのみ作成できます。

AWS Marketplace 販売者の一覧表示料金

AWS Marketplace では、製品の次のリスト料金を提供しています。

Note

これらのリスト料金は、2024 年 1 月 5 日午前 0 時に有効になりますUTC。

パブリックオファーの出品料金

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 データ交換 – 3%

プライベートオファーの出品料金

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のプライベートオファー (CPPO) のリスティング料金

CPPO 製品は、オファータイプやデプロイ方法に関係なく、リスト料金が .5% 引き上げられます。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の出品料金

すべてのプロフェッショナルサービスには、プライベートオファーの 2.5% のリスティング料金がかかります。

Amazon Payments Europe に対する AWS Marketplace 販売者の苦情の送信 (APE)

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がある場合は、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。以下のセクションでは、Amazon Payments Europe (APE) に関連する苦情を当社の苦情ポリシーに従って送信するために必要な特定のステップについて詳しく説明します。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) の苦情ポリシーについても説明します。

Note

に固有の苦情のみが、以下の手順で対処 AWS Marketplace されます。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. の AWS Marketplace アカウントをお持ちの場合、苦情は Amazon Payments Europe S.C.A. によって処理されます。

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. [コマース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、苦情を受け取った日から 15 営業日以内に苦情の更新を返信します。の管理の及ばない例外的な状況では APE、苦情の解決は、その苦情を APE 最初に受け取った日から最大 35 営業日延長される場合があります。

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)

CSSF は、ルクセンブルクの金融部門の企業の健全性監督を担当する機関です。110 CSSFでに連絡できます。Route d'Arlon L-2991 Luxembourg または <https://www.cssf.lu/contacts/> の問い合わせページを使用します。CSSF と連絡方法の詳細については、CSSFウェブサイトの「[お客様の苦情](#)」を参照してください。

- オンライン紛争解決

EU でオンラインでアカウントを開いた場合は、オンライン紛争解決プラットフォームCSSFを使用して苦情をに照会することもできます。このオプションは、Amazon Payments Europe S.C.A. が金融サービスを提供し、CSSFがそのライセンスを担当する機関であるため、利用できます。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 販売者向けの追加ツール

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して、顧客基盤を把握し、売上をよりよく理解できます。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。で利用できるアクティビティを活用することで、go-to-market製品を増分チャネル収益に管理できます[AWS Marketplace 管理ポータル](#)。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 Resources 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、起動までの 90 日間と起動後 90 日間に製品をマーケティングするためのサポートを受けるには、AWS Marketplace 管理ポータル ホームページの Marketplace Resources でリンク [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 では、を通じて製品データ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、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スできます AWS SDK。詳細については、「[AWS Marketplace Commerce Analytics Service による製品および顧客データへのアクセス](#)」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により、AWS フィールドチーム (社内で承認された 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 – 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 AWS Marketplace 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。

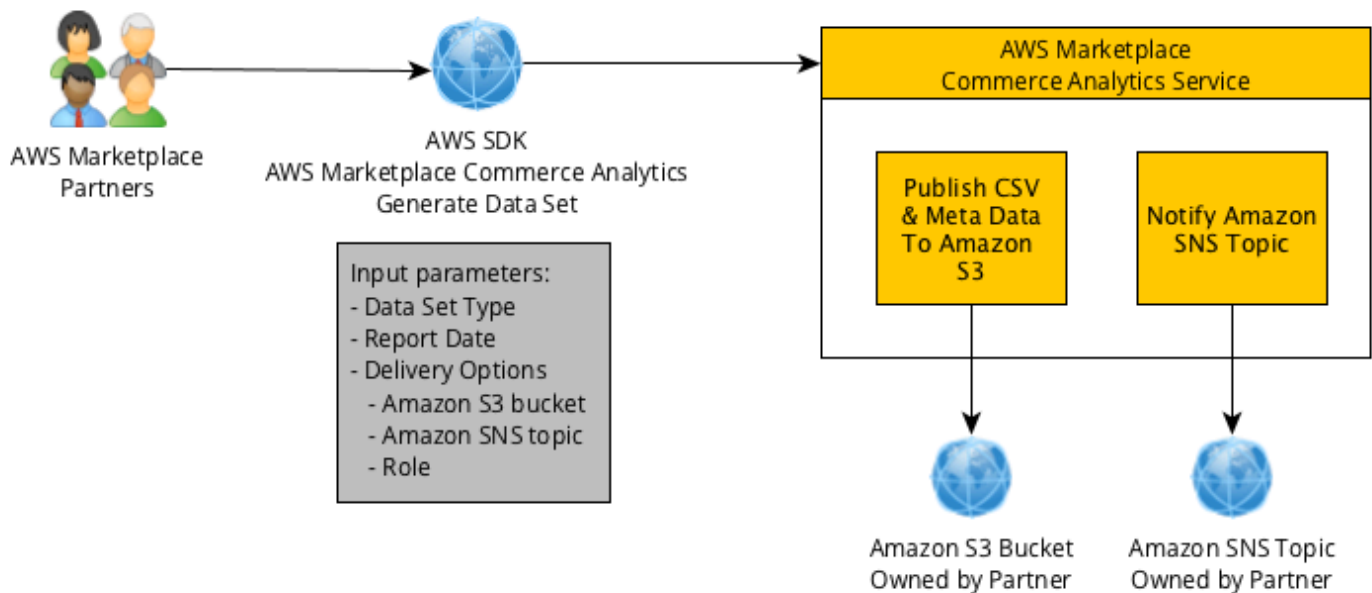
トピック

- [AWS Marketplace Commerce Analytics Service による製品および顧客データへのアクセス](#)
- [AWS Marketplace フィールドデモンストレーションプログラム](#)

AWS Marketplace Commerce Analytics Service による製品および顧客データへのアクセス

AWS Marketplace Commerce Analytics Service を使用すると、を通じて製品データ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、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スできます AWS SDKs。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。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。Amazon Simple Notification Service (Amazon) を通じてデータ配信の通知を受け取ります SNS。このトピックでは、AWS Marketplace Commerce Analytics Service を使用するための利用規約について説明します。

次の視覚化は、Commerce Analytics Service が で製品データと顧客データにアクセスし、データセットとして Amazon S3 バケットに配信 AWS Marketplace し、Amazon 経由で通知を開始する方法を示しています SNS。



利用規約

これらの AWS Marketplace Commerce Analytics Service の利用規約 (CAS 「利用規約」) には、AWS Marketplace 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条件に示されている 「I Accept」 ボタンまたはチェックボックスをクリックした日、または、それより前の場合は CA Service のサービスを使用する日に有効になります。本CAS条件は、お客様と Amazon Web Services, Inc. (「、」 「we AWS」 「us 」 または 「our 」) との間の AWS Marketplace 販売者向け利用規約 (AWS Marketplace 「販売者条件」) の補遺であり、その条件はここに組み込まれます。本CAS条件と AWS Marketplace 販売者条件の間に矛盾がある場合、本CAS条件の条件が適用されますが、かかる矛盾の範囲において、かつ CA サービスの使用に関してのみ適用されます。本書で使用されているが本書で定義されていない大文字で始まる用語は、AWS Marketplace 販売者条件に規定された意味を持つものとしします。

1. CA サービスと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 販売者条件に拘束される AWS Marketplace 販売者である必要があります。CA サービス (CAS 「データ」) に関連して受け取った、またはアクセスできる情報およびデータは、サブスクライバー情報であり、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、機密データとしてCASデータを使用できます。(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データを使用すること。(c) サブスクライバーに連絡して、の外部で代替購入を行うよう影響を与える AWS Marketplace。(d) 侮辱する 関

- 連会社 またはそれらもしくは当社のそれぞれの製品。または (e) 受信者が AWS Marketplace サブスクライバーであることに基づいて、あらゆる種類のターゲット通信。
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るのみ、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。当社は、お客様が本CAS条件に違反している、または CA サービスを悪用している可能性があると判断した場合、CA サービスにアクセスするお客様の権利を制限、停止、または終了する権利を留保します。
 3. CA サービス認証情報の機密性とセキュリティ。CA サービスの認証情報 (パスワード、キー、クライアント などIDs) は、APIクライアントを識別するために使用することを目的としています。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CAS者規約に従ってお客様に通知することで、いつでも本規約を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。本CAS条件の変更の発効日後も CA サービスの使用またはアクセスを継続することにより、お客様は変更された条件に拘束されることに同意するものとします。
 5. ターミネーション。本CAS条件および本書で付与されたCASデータを使用する権利は、何らかの理由で AWS Marketplace 販売条件が終了した時点で、お客様への通知の有無にかかわらず終了します。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。

使用開始

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

- [AWS Marketplace Commerce Analytics Service 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](#)
- [AWS Marketplace Commerce Analytics Service を使用したデータセットの生成](#)

- [AWS Marketplace Commerce Analytics Service のトラブルシューティング](#)

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

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

AWS Marketplace Commerce Analytics Service 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 Amazon S3 バケットを作成する](#)
- [ステップ 3: レスポンス通知用に 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

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 IAM アクセス許可の特定のアクセス許可 [the section called “AWS Marketplace 販売者のポリシーとアクセス許可”](#) については、「 」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 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使用できる場合は、次のステップに進みます。

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

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

Commerce Analytics Service は、Amazon を使用してレスポンス通知を送信します SNS。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。この目的のために Amazon SNS トピックが既にある場合は、次のステップに進みます。

このサービスに Amazon SNS トピックが設定されていない場合は、今すぐ設定してください。手順については、「[トピックの作成](#)」を参照してください。

サービスを呼び出すには ARN が必要なため、作成したトピックのトピック Amazon リソースネーム (ARN) を記録します。

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

Commerce Analytics Service は、バケットの SNS トピックと名前 ARN に を使用してサービスを設定した後、Amazon S3 バケットと Amazon 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. Commerce Analytics Service に登録 AWS Marketplace するために必要な [IAM アクセス許可](#) があることを確認します。
3. [Commerce Analytics Service 登録ページ](#) に移動します。
4. Amazon S3 バケット名 と Amazon SNS トピック を入力し ARN、「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 ARN には、 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。この IAM ロールでは AWS Marketplace 、 が 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}"
```

- の場合 `--data-set-publication-date`、を ISO-8601 形式を使用して現在の日付 {TODAY'S DATE} に置き換えます。ここで YYYY-MM-DDT00:00:00Z、YYYY は 4 桁の年、MM は 2 桁の月、DD は 2 桁の日です。
- の場合 `--role-name-arn`、を の登録プロセスから受け取ったロールARNの {YOUR-ROLE-NAME-ARN} に置き換えます [ステップ 4: Commerce Analytics Service プログラムに登録する](#)。
- `--destination-s3-bucket-name` の場合は、{amzn-s3-demo-bucket} を で作成した Amazon S3 バケツの名前に置き換えます [ステップ 2: 送信先 Amazon S3 バケツを作成する](#)。
- `--sns-topic-arn` の場合、{YOUR-SNS-TOPIC-ARN} を で作成した Amazon SNS トピックに置き換えます [ステップ 3: レスポンス通知用に Amazon SNS トピックを設定する](#)。

サービスから `dataSetRequestId` レスポンスを含むレスポンスを受け取った場合は、オンボーディングプロセスを完了しました。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 SDK for Java での AWS Marketplace Commerce Analytics Service の使用

AWS Marketplace Commerce Analytics Service を使用すると、 を通じて製品データと顧客データにプログラムでアクセスできます AWS Marketplace。AWS Marketplace Commerce Analytics Service 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 [AWS SDK for Java](#) Commerce Analytics Service とやり取りします。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 を実装する方法を示します。

トピック

- [IAM Commerce Analytics Service のポリシー](#)
- [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](#)

IAM Commerce Analytics Service のポリシー

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。

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

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

次のIAMアクセス許可ポリシーを使用して、ユーザーに AWS Marketplace Commerce Analytics Service へのリクエストを許可します。

```
{
  "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 Hourly/Monthly Subscriptions 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 demo-bucket Amazon S3 バケットに発行され、通知メッセージが demo-topic Amazon SNS トピックに配信されます。

```
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、Amazon SNS トピックに発行された通知とリクエストを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、[AWSJava 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 Hourly/Monthly Subscriptions 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 demo-bucket Amazon S3 バケットに発行され、通知メッセージが demo-topic Amazon SNS トピックに配信されます。

```
/*
 * 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

```

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

AWS Marketplace Commerce Analytics Service に登録すると、AWS CLI および 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 による製品および顧客データへのアクセス](#)」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、レスポンス、出力に関する情報を提供します。

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公開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。

フィールド	説明
	<p>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。</p> <p>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。</p>
ロール名 ARN	サービスにリソースへのアクセスを提供するアクセス許可ポリシーがアタッチされたロール ARN の。
Destination Amazon S3 Bucket Name	送信先 Amazon S3 バケットの名称 (ではなくわかりやすい名称 ARN)。データセットはこの場所に発行されます。
Destination Amazon S3 Prefix	<p>(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。</p> <p>たとえば、バケットの名称が 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。</p> <p>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。</p> <p>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。</p>
SNS トピック ARN	データセットが公開されたとき、またはエラーが発生したときに通知される Amazon SNS トピック ARN の。

レスポンス

AWS Marketplace Commerce Analytics Service は 2 つのレスポンスを返します。1 つ目は同期で、すぐに返され、2 つ目は非同期で、Amazon を使用して返されます SNS。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。この識別子は、Amazon SNSトピックの通知とリクエストを関連付けるために使用できます。

非同期レスポンスは JSON形式のドキュメントとして Amazon SNSトピックに投稿され、この例に似ています。

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。この識別子は、Amazon SNSトピックの通知とリクエストを関連付けるために使用できます。
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。フォーマットJSONされた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公開されます。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。この識別子は、Amazon SNSトピックの通知とリクエストを関連付けるために使用できます。
Data Set Coverage Range	データカバレッジ範囲date/time and end date/timeの開始を定義します。これらの日付は8601 ISO形式です。

フィールド	説明
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。

フォーマットJSONされた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"
  },
  "dataSetMetaDataSetS3Location": {
    "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、[AWSSDKドキュメント](#) を参照してください。

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

AWS Marketplace Commerce Analytics Service に関する問題はトラブルシューティングできます。このサービスでは、 から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されていない場合は AWS Marketplace、 [AWS Marketplace 管理ポータル](#) にアクセスして登録してください。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。

SDKドキュメントで、この日付で使用できるようにする必要があると示されているにもかかわらず、過去の日付のデータセットをリクエストすることはできません。

過去の特定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。

サービスを呼び出すと、「エンドポイントに接続できませんでしたURL: https://marketplacecommerceanalytics.eu-central-1.amazonaws.com/”

AWS Marketplace Commerce Analytics Service 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、`--region` フラグを各呼び出しに追加し 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を実行したときに選択したトピックとは異なる Amazon S3 バケットまたは Amazon SNS トピックを使用します。

AWS Marketplace Commerce Analytics Service 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、これらの特定のリソースのみへのサービスアクセスを許可するアクセスIAM許可を設定します。異なる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、 でIAMコンソールを開きます <https://console.aws.amazon.com/iam/>。
2. IAM コンソールの左側にあるロールを選択します。
3. を選択します MarketplaceCommerceAnalyticsRole。
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. で始まる名前ポリシーを見つけます。 oneClick_MarketplaceCommerceAnalyticsRole ポリシーの編集 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、次のガイドを参照してください。 https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

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。次の手順では、IAM コンソールを使用してこれらのアクセス許可を持つ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ます。

JSON ポリシーエディタを使用してポリシーを作成するには

1. にサインイン AWS Management Console し、 でIAMコンソールを開きます <https://console.aws.amazon.com/iam/>。
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. [Next (次へ)] を選択します。

Note

ビジュアルオプションとJSONエディタオプションはいつでも切り替えることができます。ただし、ビジュアルエディタで変更を加えるか、Next を選択すると、IAMはポリシーを再構築してビジュアルエディタ用に最適化する場合があります。詳細については、IAM「ユーザーガイド」の[「ポリシーの再構築」](#)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。

- 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。

- ID プロバイダーIAMを介して で管理されるユーザー :

ID フェデレーションのロールを作成します。IAM ユーザーガイドの [「サードパーティー ID プロバイダーのロールの作成 \(フェデレーション\)」](#) の指示に従います。

- IAM ユーザー：
 - ユーザーが担当できるロールを作成します。 [「ユーザーガイド」のIAM「ユーザーのロールを作成する」](#) の手順に従います。 IAM
 - (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する。IAM ユーザーガイドの [「ユーザー \(コンソール\) へのアクセス許可を追加する」](#) の手順に従います。

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) では、AWS フィールドチーム (社内で承認された 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。

- [Amazon 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 \(SageMaker \)](#)
- [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。

用の製品の準備 AWS Marketplace

を通じて販売されるソフトウェア製品を準備して公開できます AWS Marketplace。製品の準備には、パッケージの設定、料金設定スキームの設定、製品を一覧表示する関連カテゴリの決定、関連検索に製品が表示されるようにキーワードの追加が含まれます。独立したソフトウェアベンダー (ISV)、チャンネルパートナー、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である場合は、このプロセスを使用して、で製品を公開する準備をすることができます AWS Marketplace。以下のトピックでは、で製品を正常に準備して公開する手順について説明します AWS Marketplace。

トピック

- [の製品配信 AWS Marketplace](#)
- [の製品料金 AWS Marketplace](#)
- [AWS Marketplace 製品のリージョンと国](#)
- [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](#)
- [の製品の使用手順の作成AMIとコンテナ AWS Marketplace](#)
- [検索のための AWS Marketplace 製品の最適化](#)

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。

標準定価とエンドユーザーライセンス契約 (EULA) を使用して製品を作成できます EULA。また、カスタム料金と を使用して、個々の顧客向けのプライベートオファーを作成することもできます EULAs。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 で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。カスタマーエンゲージメント (APN) の対象となる AWS パート

ナーネットワーク () パートナーで、このオプションを購入者に提供したい場合は、AWS 担当者に連絡して、さらなる適格性を判断してください。 [APN ACE](#)

次の表に、ソフトウェア製品の配信に使用できる方法と、AWS Marketplace 購入者が AWS Marketplace コンソールで各タイプの成果物を見つける方法を示します。

製品配信方法

製品配信方法	コンソールの配送方法フィルター	説明
シングル AMI	Amazon マシンイメージ (AMI)	<p>製品用に 1 つのカスタム 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。</p> <p>購入者は 1 つの を使用して AMI、製品がすでにインストールされ、すぐに使用できる状態で Amazon EC2 インスタンスを作成できます。</p> <p>詳細については、「AMI ベースの製品 AWS Marketplace」を参照してください。</p>
AMI AWS CloudFormation テンプレートを使用して配信	CloudFormation テンプレート	<p>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。</p> <p>購入者は、AMIs その製品のすべての に権限を付与する単一</p>

製品配信方法	コンソールの配送方法フィルター	説明
		<p>のソリューションを購入できません。</p> <p>テンプレートAMIsとしての配信の詳細については CloudFormation、「AMIを使用したベースの配信 AWS CloudFormation」 を参照してください。</p> <p>CloudFormation テンプレートの詳細については、「ユーザーガイド」のAWS CloudFormation「概念」 を参照してください。AWS CloudFormation</p>
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]	<p>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。</p> <p>詳細については、「でのコンテナベースの製品 AWS Marketplace」 を参照してください。</p>

製品配信方法	コンソールの配送方法フィルター	説明
データ製品	AWS Data Exchange	<p>AWS Data Exchange を使用してデータ製品を作成します。</p> <p>AWS Data Exchange を介したデータ製品とオファターの発行と管理については、AWS「Data Exchange ユーザーガイド」の「Data Exchange でのデータ製品の提供」を参照してください。 AWS</p>
機械学習アルゴリズムとモデルパッケージ	SageMaker モデル	<p>Amazon を使用してアルゴリズムまたはモデルパッケージ SageMaker を作成し、に公開します AWS Marketplace。</p> <p>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、「Machine Learning 製品」を参照してください。</p> <p>の詳細については SageMaker、「Amazon SageMaker デベロッパーガイド」の「とは SageMaker」を参照してください。</p>

製品配信方法	コンソールの配送方法フィルター	説明
Software as a service (SaaS)	SaaS	サブスクリプションベース、契約ベース、または従量制の契約料金モデルで SaaS 製品を提供できます。 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]	他の製品をサポートしたり、他の AWS Marketplace 製品と連携したりするプロフェッショナルサービスを提供できます。

の製品料金 AWS Marketplace

AWS Marketplace 販売者として、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 (\$) に基づいています USD。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ス料はかかりません。このトピックでは、のソフトウェア製品の料金モデルと料金について説明します AWS Marketplace。

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。

- [AMI の製品料金 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をお客様に提示する方法の詳細については、[「有料コンテナ製品のAMIサブスクリプションまたは料金モデル」](#)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でも、インスタンスタイプごとに異なるものでも構いません。
- 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の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、前払いAMIで複数年 (最大 3 年) またはカスタム期間、または柔軟な支払いスケジュールを提供できます。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 [the section called “分割プランの作成”](#) 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ものとします [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

お客様が年間サブスクリプションで 12 か月間ソフトウェア AWS を使用する場合、 の製品の詳細ページで EULA 指定した が適用されます 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せん。

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、APIがソフトウェアによって呼び出されたときに測定され、報告されます。ベストプラクティスとして、販売者はユースケースに応じて 1 時間に 1 回呼び出すAPIのように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 を使用すると、いくつかの新しい料金シナリオを処理できます。

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

Product Load Form (PLF) では、関連する列の前に「Flexible Consumption PricingFCP」が付きます。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。

- ソフトウェアが既に 上にある場合は AWS Marketplace、代替の使用ディメンションを有効にするために製品を作成する必要があります。Marketplace AWS Metering Service を使用するように標準製品を変換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、その時間のお客様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 を使用する製品は、1-Clickをサポートしていません。購入者は、特定のアクセス許可と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。

- 単一AMIベースの製品とAMI AWS CloudFormation テンプレートベースの製品。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。既存の製品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、新しいリストを作成してから、Product Load Form (PLF) を使用して契約料金モデルを適用し、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、ライセンスマネージャー を使用してソフトウェア AWS アカウント が確認できるライセンスが によって顧客 AWS Marketplace 内に作成されますAPI。お客様は、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ためのIAM ロールが必要です。

Bring Your Own License の料金

の Bring Your Own License (BYOL) 製品にはサービス料金はかかりません AWS Marketplace。

お客様の選択の約束を果たすには、すべてのBYOL製品に有料オプションも必要です。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、ソフトウェアのオンライン購入は、一部の企業がビジネスを行う方法から逸脱していることがわかります。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。Marketplace [AWS Seller Operations](#) チームに連絡して、料金モデルの変更について説明します。

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。

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したり、変更をリクエストできます。

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。

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 販売者はカスタム料金と個々の AWS Marketplace 顧客 (購入者) EULAs と交渉できます。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

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。この返金ポリシーには、返金の条件だけでなく、販売者に連絡して返金をリクエストする方法を盛り込む必要があります。返金ポリシーの詳細を決定するのは販売者です。ただし、製品の使用に対して何らかの方法で購入者に返金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、および顧客への返金リクエストの送信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](#)
- [AWS Marketplace 製品の返金プロセス](#)

AWS Marketplace 製品の返金リクエストタイプ

お客様は 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。お客様が 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指示します。AWS インフラストラクチャ料金の返金は、の裁量に委ね AWS られ、ソフトウェアの返金とは無関係に処理されます。

サードパーティーによって製品が販売されている場合は、購入者は製品の詳細ページの返金ポリシーを確認するよう指示されます。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の概要を示しています。

- 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一覧表示した場合、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 日以内に発生したソフトウェア料金に対して、はユーザーに代わって返金を発行 AWS 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。製品の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ものとみなされます。

- 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、AWS が処理する前に、お客様の承認を受ける必要があります。

- 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計測する場合 AWS Marketplace Metering Service、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発行 AWS できます。これらのエラーが複数のお客様に共通している場合、は各お客様に適切な返金を決定し、各お客様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。AWS Marketplace Metering Service を製品で使用すると、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

プライベートオファー以外の購入から 48 時間以内に購入者がサブスクリプションをキャンセルした場合、AWS から全額返金されます (100% 返金されるキャンセル)。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。AWS Marketplace に製品を出品することにより、本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、はお客様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、AWS が処理する前にお客様の承認を受ける必要があります。

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

AWS Marketplace 製品の返金プロセス

[返金申請フォーム](#)を送信すると、製品ソフトウェアの使用に対する返金を開始できます。AWS Marketplace 購入者サポートチームが受け取ると、関連するサポートケースが[AWS Support センターコンソール](#)に作成され、返金ステータスが件名に表示されます。返金関連のサポートは、これらのケースを通じて直接行われます。詳細については、「[AWS Supportへのアクセス](#)」を参照してください。

以下の手順は、外部の顧客または内部のテスト用アカウントへの返金をリクエストする方法の概要です。

顧客にソフトウェアの返金を開始するには

1. 顧客についての以下の情報を収集します。
 - に関連付けられているお客様の E メールアドレス AWS アカウント。
 - 製品のサブスクリプションに使用されるアカウントの顧客 AWS アカウント 番号。顧客が組織の支払者である場合は、製品にサブスクリプションしているリンクされたアカウントの AWS アカウント ID を提供する必要がありますを顧客に再認識させます。
 - 顧客が返金を求める請求期間。
2. にサインインし AWS アカウント、[返金リクエストフォーム](#)に移動します。
3. フォームに顧客の情報を入力します。
4. 顧客が返金をリクエストしている製品の製品 ID を入力します。製品 ID は、[日次カスタマーサブスクリプターレポート](#)で確認できます。
5. 顧客が年間製品に対して返金、アップグレード、ダウングレードをリクエストしている場合は、次のタスクを完了する必要があります。
 - a. サブスクリプターレポート (反映が 24 時間後になる場合がある) を使用して、顧客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. コメントフィールドにサブスクリプションのキャンセルの日付を入力します。
 - c. コメントフィールドに、承認中 (返金、アップグレード、ダウングレード) の変更の説明を入力します。
6. フォームを送信します。当社に通知され、返金を処理した後、顧客に返金します。
 7. 返金リクエストのステータス情報を含むアウトバウンドケースが [AWS Support Center コンソール](#) で作成されます。件名には、次のいずれかが含まれます。
 - 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。
 8. 返金が正常に処理されると、24～48 時間以内に顧客のアカウントに反映されます。ただし、顧客の金融口座に資金が表示されるまでに最大 5 営業日かかります。

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、使用可能な 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。例えば、購入者が米国に所在し、米国から購入している場合でも、欧州 (フランクフルト) リージョンに製品をインストールする予定がある場合があります。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、製品を、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 リージョンに制限できます。

Amazon マシンイメージ (AMI)、コンテナベース、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 が製品を使用できます。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、[「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。

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。

での標準化された契約の使用 AWS Marketplace

製品を AWS Marketplace 販売者として[準備](#)するときは、製品の使用に適用されるエンドユーザーライセンス契約 (EULA) を決定する必要があります。以下を使用できます。

- 独自の EULA。
- (AWS Marketplace SCMP) の標準契約は、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。
- (AWS Marketplace RCMP) のリセラー契約。ISVs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。

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、購入者および販売者の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を管理し、購入者と販売者の義務を定義します。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。

は、使用、保証、補償、準拠法など、主要な契約条項に共通する根拠を SCMP 事前に定義します。販売者は、セルフサービストラランザクシオン EULA のとして SCMP 用語を提供できます。購入者はソリューションを検索、購入、迅速にデプロイできます。[プライベートオファー](#) の場合、購入者は販売者から SCMP テンプレートをリクエストでき、条件は当事者間で合意されたカスタムランザクシオン要件に対応するために修正できます。

セルフサービスまたはプライベートオファー SCMP の では、次のオプションの追加も使用できます。

- [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](#) - 1996 年医療保険の相互運用性と説明責任に関する法律 (HIPAA) コンプライアンス要件の取引をサポートします。

免責事項

EULA は、お客様と購入者の間にあります。を SCMP として使用するかどうか EULA は、お客様の自由です。SCMP を製品リストに適用することで、SCMP プログラムにオプトインします。このプログラムでは、AWS が SCMP テンプレートを定期的に更新し、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 に置き換えることで、いつでも SCMP プログラムから退会できます EULA。

SCMP の使用開始

EULA をに更新SCMPし、新規および既存の単一の Amazon マシンインスタンス (AMI) および Software as a Service (SaaS) 製品の購入者に提供できます。更新をリクエストするために使用する手順は、製品がセルフサービスリスト () を通じてリストされているかどうかによって異なります SSL。

質問がある場合は、 の AWS Marketplace Standard Contracts チームに E メールメッセージを送信します。aws-mp-standardcontract@amazon.com.

を通じてリストされている SCMP for AMIおよび SaaS 製品の EULAに を更新するには SSL

1. マーケット [AWSプレイスの標準契約](#) の条件を確認します。
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EULAされていない場合は、 の Standard Contract for 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。

を使用してリストされていない SCMP for AMIおよび SaaS 製品の EULAに を更新するには SSL

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。
 - 説明で、 **Please update these products to SCMP V2** で更新IDsする製品タイトルと を入力して指定しますSCMP。

のリセラー契約 AWS Marketplace

AWS Marketplace (RCMP) のリセラー契約は、チャンネルパートナーが AWS Marketplace 購入者に ISV 製品を再販することを許可する際に ISVs 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。この契約は、法的契約レビューの冗長性を軽減し、ISVs および チャンネルパートナーが再販業者の関係を締結したり、[チャンネルパートナーのプライベートオファー \(CPPO\)](#) に再販業者の条件を使用したりするとき市場投入までの時間を短縮するのに役立ちます。を作成すると CPPO、ISVs は契約をオポチュニティ (再販認証) にアップロードでき、チャン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 を表示できません RCMP。

免責事項

RCMP は、 のオプション契約です ISVs。チャンネルパートナーを通じて製品を再販すること ISVs を決定した場合、オポチュニティを作成するとき、既存の契約条件または事前交渉済みの契約条件を RCMP または独自のカスタマイズ契約条件をアタッチできます。

RCMP の使用開始

このセクションでは、用語を確認し、 を使用する方法について説明します RCMP。

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

1. [リセラー契約 for AWS Marketplace](#) の条件を確認してください。
2. [RCMP ガイド](#) を参照しながら、[「チャンネルパートナーの再販機会を作成する」](#) のステップに従います。

AWS Marketplace 製品のメタデータの提供

販売者として、 に製品を追加するときは AWS Marketplace、製品メタデータを指定します。製品メタデータには、お客様の製品を記述するために使用する名前、説明、カテゴリ、キーワードが含まれます。 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 AWS Marketplace 改訂します。このトピックでは、AWS Marketplace 販売者として製品に効果的に名前を付け、説明し、分類するため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。

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。

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
- 記述データやハイパーボールを避ける

製品ソリューションの例: Smart Solution は、受信アプリケーション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。スマートソリューションは、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。

サポート情報の書き込み

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、製品を参照または検索する際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 事前に定義されており AWS Marketplace、製品リクエストプロセス中にリストから選択して、製品に適用され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。

の製品の使用手順の作成AMIとコンテナ AWS Marketplace

AWS Marketplace 販売者として、製品を購入者に配信する方法の 1 つは、[Amazon Machine Images \(AMIs\)](#) を使用することです。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。AMI 製品のカスタムを作成すると、購入者はそれを使用して、製品がすでにインストールされ、すぐに使用できる状態で Amazon EC2 インスタンスを作成できます。以下のセクションでは、使用方法の記述と AMIs の管理について説明します。

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

トピック

- [要件](#)
- [リリースノートを書く](#)
- [使用手順を書く](#)
- [アップグレード手順を書く](#)
- [CloudFormation 配信手順の記述](#)
- [アプリケーション機能のモニタリングと評価](#)
- [プログラムによるシステム認証情報と暗号化キーのローテーション](#)

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保存場所
- すべてのデータ暗号化設定の説明
- Step-by-step プログラムによるシステム認証情報と暗号化キーをローテーションするための手順。[the section called “AMIベースの製品要件”](#) は、認証情報と暗号化キーを使用するリストの基本要件について説明します。
- アプリケーションが何らかの暗号化技術を使用している場合、ユーザーがアプリケーションをどのように操作して必要なデータを復号化するかについて詳細に説明してください。
- Step-by-step アプリケーションの正常性と適切な機能を評価してモニタリングする方法の手順。
例:
 - [Amazon EC2コンソール](#)に移動し、正しいリージョンにあ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。

リリースノートを書く

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。

使用手順を書く

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。

- ユーザーが関心はあるが情報がないと仮定します。
- 設定や特別なステップなど、製品の起動と使用に必要なすべての情報をユーザーに提供します。

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、https://<EC2_Instance_Public_DNS>/index.html でアプリケーションにアクセスします。

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: インスタンスのインスタンス ID

アップグレード手順を書く

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 配信を使用する場合は、以下も含める必要があります。

- テンプレートによって AWS CloudFormation 作成された各 AWS Identity and Access Management (IAM) ロールと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 統合キーセットアップ (LUKS) など)

アプリケーション機能のモニタリングと評価

アプリケーション機能をモニタリングおよび評価するには

1. [Amazon EC2コンソール](#)に移動し、正しいリージョンにあることを確認します。
2. [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。
3. 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。

Note

いずれかのデータストアが専有データである場合は、設定、バックアップ、復旧の手順を提供します step-by-step。

プログラムによるシステム認証情報と暗号化キーのローテーション

[the section called “AMIベースの製品要件”](#) は、認証情報と暗号化キーを使用するリストの基本要件について説明します。

プログラムによるシステム認証情報と暗号化キーのローテーションには、以下を含めてください。

- 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。

検索のための AWS Marketplace 製品の最適化

AWS Marketplace 販売者として、購入者がより簡単に検索できるように、検索のため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、顧客が独自のニー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) との検索関数の 2 つの主要な場所で行われます AWS Marketplace。このトピックでは、両方のローテーションのリスト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製品詳細ページへのよりユニークな訪問者を誘導するために不可欠です。

製品の詳細ページに対する有機的な検索を改善す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があります。キーワード、タイトルタグ、H1 見出しタグです。

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あり、トピックを検索エンジンの結果を推進する集中クエリに抽出します。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識別できます。

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

タイトルタグは、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、リーダーと検索エンジンのウェブクローラーの両方にページの内容を通知します。AWS Marketplace 製品の詳細ページでは、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、および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 つのルールがあります。

- 訪問者がページの内容をスキャンして、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検索最適化手法と同様の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 管理ポータルで [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、[「製品の変更と更新」](#)を参照してください。

AWS Marketplace 販売者オペレーションチームは、類似した発音の単語または同様の意味の単語を使用して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、お客様は製品を参照または検索する際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、用に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]

- コストの予測: AnyCompanyの製品では、使用した分に対してのみ料金が発生します。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

- スペルと文法を確認する
- 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、受信アプリケーション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、受信するアプリケーショントラフィックに対応するために必要な負荷分散容量をシームレスに提供することで、アプリケーションの耐障害性を向上させることができます。AnyCompanyの製品は、プール内の異常なインスタンスを検出し、異常なインスタンスが復元されるまでトラフィックを自動的に正常なインスタンスにルーティングします。お客様は、1つのAWS アベイラビリティゾーンで、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。

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、 から製品を購入するために使用されるネゴシエートされた用語です 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、公開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を拡張して、プライベートオファーとパブリックオファーのどちらを選択するかを選択できる購入者。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、購入者に の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 AWS Marketplace を提供し、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。カスタマーエンゲージメント (APN) の対象となる AWS パートナーネットワーク () パートナーで、このオプションを購入者に提供したい場合は、AWS 担当者に連絡して、さらなる適格性を判断してください。 [APN 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](#)
- [購入者向けのプライベートオファーエクスペリエンス](#)
- [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の作成](#)
- [でのプライベートオファーの修正 AWS Marketplace](#)
- [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の仕組み

プライベートオファーは、 のオファーページから作成および管理できます [AWS Marketplace 管理ポータル](#)。オファーの製品を指定して、一意の ID と を生成します URL。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に拡張します AWS アカウント。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、潜在的な購入者に通知した後、購入者はオファーを表示して承諾できます。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

購入者は、リンクされたアカウントまたは管理アカウントにオファーを拡張しない限り、オファーを表示できません。オファーでサービス制限を指定することはできません。そのため、製品に制限がない限り、購入者は交渉済みの価格で製品を必要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、 [「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。

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、 [「プライベートオファーのレポート」](#) および [「販売者レポートガイド」](#) 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使用する場合は、次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、製品のプライベートオファーを既に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、 [「プライベートオファーのアップグレード、更新、および修正」](#) を参照してください。

- セカンドパーティ、Amazon Machine Image (AMI) の毎月のプライベートオファー、または multi-AMI-based AWS CloudFormation 製品を使用した配信、または顧客の使用制限のために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、[「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) モデルをサポートしていません。
- プライベートオファーで一意的なネゴシエートされた契約条件を持つプライベートオファーを作成するときは、カスタムEULAオプションを使用します。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。例えば、これを行うと、新しい権限を付与したり、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、エンドユーザーライセンス契約 (EULA) を変更して標準化されたライセンス条件を使用したりできます。詳細については、[「プライベートオファーのアップグレード、更新、および修正」](#)を参照してください。

購入者向けのプライベートオファーエクスペリエンス

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受け入れると、すべての AWS Marketplace トランザクションで使用されている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。購入者はこ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確認し、AWS Management Console、販売者はこ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ファイナンスにアクセスできます。詳細については、[「でカスタマーファイナンスが利用可能になりました 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も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、承諾済みおよび期限切れのオファータブに移動します。

<p>プライベートオファーを表示して受け入れるには</p>	<p>購入者は次のことができます。</p>
<p>AWS Marketplace コンソールから</p>	<p>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。</p> <p>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。</p>
<p>販売者が提供したリンクを使用する</p>	<p>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。</p> <p>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。</p>
<p>製品ページから</p>	<p>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。</p> <p>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。</p>

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。詳細については、「[出品者ダッシュボード](#)」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어의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 販売者として、プライベートオファーを使用して、Amazon Machine Images (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、 から製品を購入するために使用されるネゴシエートされた用語です AWS Marketplace。このトピックでは、 、コンテナAMI、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 年間プライベートオファーでは、時間単位またはAMI時間単位: 最大 3 年間 (1,095 日)。年間プライベートオファーまたはAMI契約では、1 AMI時間ごとにのみ柔軟な支払いスケジュールがサポートされます。
- AMI 契約プライベートオファー: 最大 5 年間 (60 か月)

AMI 契約の場合、プライベートオファーは使用状況をモニタリングしません。

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、サービスの制限を適用し、購入者により多くのユニットで契約を手動でアップグレードするよ

う助言するのは、独立したソフトウェアベンダー (ISV) 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ターの契約期間は、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タ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のAMIプライベートオファターでは、契約期間中に契約で合意された年間インスタンスタイプを設定できます。

Note

プライベートオファタ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ター

コンテナベースの製品契約には、プライベートオファター価格を提供できます。

オファターは、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ター付きのコンテナ (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ター付きのコンテナ (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ター: 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ターによって、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 (ISV) は契約階層を定義し、サービス制限を適用し、購入者により多くのユニットで契約を手動でアップグレードするようアドバイスします。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ターの契約期間は、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タ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ターでは、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、独立したソフトウェアベンダー (ISV) は契約階層を定義し、サービスの制限を適用し、必要に応じて、購入者に手動で上位の契約階層にアップグレードするよう助言します。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の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。製品の既存の実行中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 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ターの推論レートは、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ターは、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ターで設定した時間単位の料金で請求されます。例えば、固定前払い料金が 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タ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ターを受け入れると、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ターを作成して、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ターが適用された購入者は、プライベートオファターとパブリックオファタ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ターは 1 つのみです。プライベートオファターとパブリックオファタ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にプライベートオファターを作成するには、[AWSマーケットプレイス販売者オペレーション](#)チームにお問い合わせください。

プライベートオファターの作成をリクエストするときは、次の情報を提供する必要があります。ProductId、ターゲット購入者AWSアカウント、日時 (オファターを承諾する必要がある時期を示す)、カスタムEULAファイル (オプション)、返金ポリシー、ライセンス期間 (オプション)、ライセンス料金 (オプション)、柔軟な支払いスケジュール (オプション)、およびインスタンスタイプごとの料金。オファターが作成されたら、購入者は指定された日時にオファターを受け入れる必要があります。

プライベートオファターの作成と管理

AWS Marketplace 販売者として、プライベートオファターを作成および管理できます。プライベートオファターは、から製品を購入するために使用されるネゴシエートされた用語です 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ターを作成および管理する方法を説明します。

トピック

- [新しいプライベートオファターの開始](#)

- [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起案と公開](#)
- [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のキャンセル](#)

新しいプライベートオファーの開始

以下のプロセスを使用して、Create Offer 変更リクエストを使用してオファーを作成し、オファー ID を生成します。API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には

1. にサインインし[AWS Marketplace 管理ポータル](#)、オファー を選択します。
2. オファーページで、オファーの作成 を選択します。
3. 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。このプロセス中に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、再販認証から、独自の製品または AWS Marketplace チャンネルパートナーのプライベートオファー (CPPO) のオファーを作成するかどうかを選択します。の場合はCPPO、独立したソフトウェアベンダー (ISV)、製品、および認可を選択します。

4. 続行 を選択して詳細を提供します。step-by-step エクスペリエンスが開き、プライベートオファーの作成を続行できます。

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あります。

- ドラフト – オファーは不完全であり、まだ準備中です。オファーを公開して購入者に拡張するには、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、購入者に拡張されます。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、購入者に拡張されます。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受け入れる時間を増やすことができます。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

オファーが承諾されると、契約タブに契約として表示されます。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起案と公開

以下のプロセスを使用して、プライベートオファーをドラフトおよび公開します。

プライベートオファーをドラフトして公開するには

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。これが更新オファーである場合は、 で作成された既存の AWS Marketplace 契約を更新する場合は を、既存の顧客を AWS Marketplace に移行させる場合は を AWS Marketplace、既存の顧客を に移行させる場合は 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。UTC この日付の 23:59:59 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諾できなくなります。

2. [Next (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

非USDプライベートオファーは、契約料金商品に限定されます。さらに、支払い以外のUSD設定が設定されていることを確認してください。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のプライベートオファーは、でのみ作成できませんUSD。

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。選択した各購入者は、選択したオファー通貨がサポートされ AWS リージョンの AWS アカウント に持っている必要があります。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. [Next (次へ)] を選択します。
6. 法的条件とオファードキュメントの設定ページで、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EULAの 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、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マージされます。
7. レビューと作成ページで、プライベートオファーの詳細を確認します。確認したら、オファーの作成を選択してオファーを公開し、選択した購入者に拡張します。オファーの発行には AWS Marketplace カタログ へのリクエストが含まれているためAPI、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延長されます。リクエストが失敗した場合、顧客には拡張されません。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。

以下のガイドでは、特定の製品のプライベートオファーの作成に関する詳細情報を提供します。

- [AWS Marketplace – SaaS PAYG プライベートオファターの作成](#)
- [AWS Marketplace – AMI 時間単位/PAYGプライベートオファターを作成する](#)
- [AWS Marketplace – 年間プライベートオファターを使用して 1 AMI 時間ごとに作成する](#)
- [AWS Marketplace – SaaS 契約プライベートオファターの作成](#)

次のビデオでは、SaaS 契約プライベートオファターの作成について詳しく説明します。

購入者へのプライベートオファターの送信

プライベートオファターが公開されると、購入者は のプライベートオファターページの 利用可能なプライベートオファター タブに移動して、プライベートオファターを表示できます AWS Marketplace 管理ポータル。利用可能なプライベートオファタータブでは、購入者は、チャンネル AWS Marketplace パートナーによって拡張されたオファターを記録販売者列に表示できます。独立したソフトウェアベンダー (ISV) が Publisher 列に表示されます。購入者はオファターリストで適切な [オファター ID] を選択すると、プライベートオファターに移動できます。

購入者は、承諾IDsされたオファターまたは期限切れのオファターを、承諾されたオファターまたは期限切れのオファタータブで表示できます。

プライベートオファターが公開されたら、購入者にオファターのフルフィルメントページURLに を送信できます。

購入者にプライベートオファターを送信するには

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

プライベートオファタ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. 完了したステップで、保存を選択して を終了します。ダイアログボックスで、コンテンツをドラフト状態に保存し、検証エラーを確認します。検証エラーや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、問題を解決できます。準備ができたら、保存を選択して終了し、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。ステップを続行したり、リクエストが成功するまで変更したりすることはできません。リクエストが成功したら、保存を完了します。リクエストが失敗した場合、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ありました。

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拡張するかを選択できます。

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の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ーの有効期限を更新するには

1. にサインインし[AWS Marketplace 管理ポータル](#)、オファー を選択します。
2. オファーページで、更新するオファーを選択します。
3. [編集] を選択します。
4. 新しいオファーの有効期限 を指定します。
5. [送信] を選択します。

更新が完了すると、オファーはアクティブステータスに変更され、購入者は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のキャンセル

プライベートオファーをキャンセルするには、次のプロセスを使用します。

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

Note

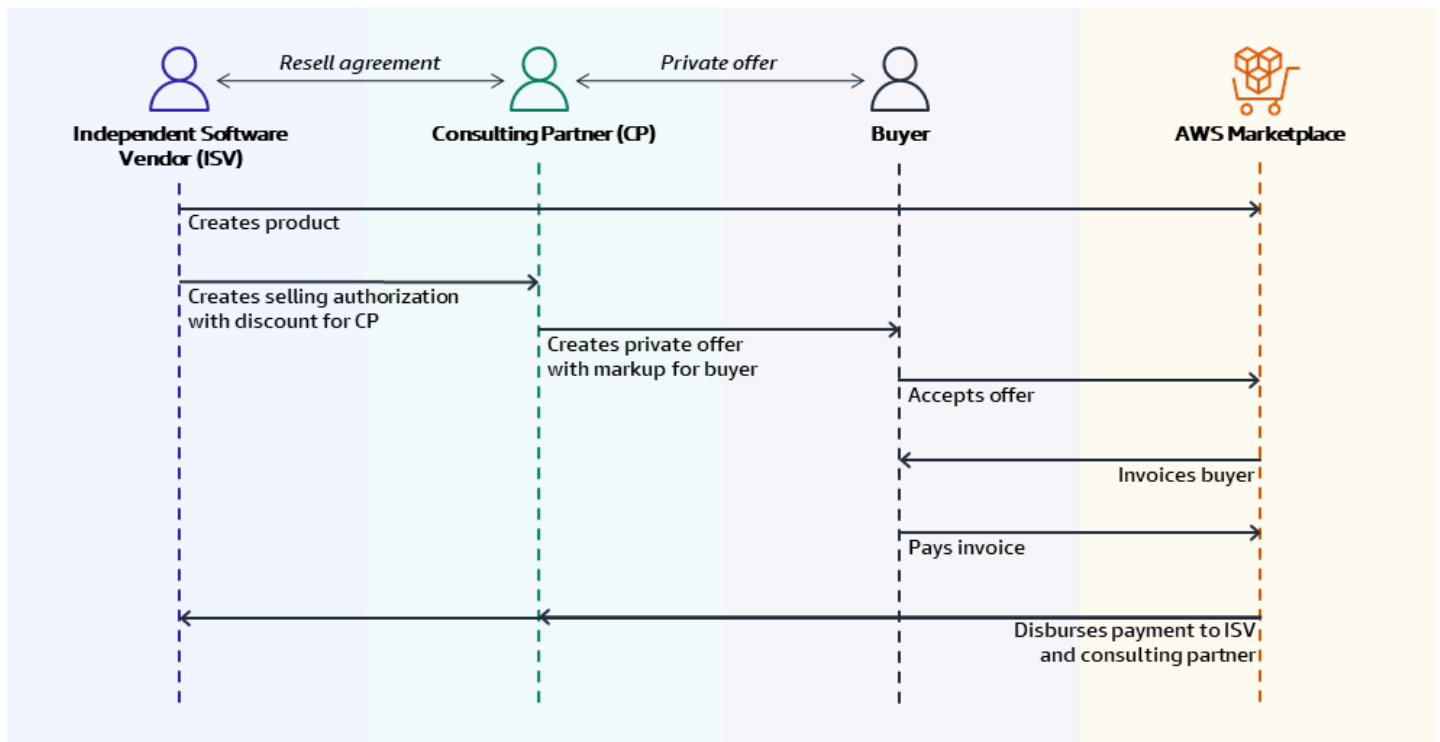
オファーをキャンセルすると、オファーの有効期限が変更されるため、このオファーを延長した購入者にはオファーの有効期限が切れたものとして表示されます。

3. アクション を選択し、オファーのキャンセル を選択します。

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

AWS Marketplace チャンネルパートナーのプライベートオファーは、チャンネルパートナーに、で独立したソフトウェアベンダーの (ISVs) 製品を再販する機会を提供します AWS Marketplace。Channel AWS Marketplace Partner とは、 の 1 つ以上の ISV 製品を再販する契約 ISV を結び、その製品の購入者にプライベートオファーを延長します。

次の図は、 、チャンネルパートナー ISV、購入者間のこの関係を示しています。



Note

チャンネルパートナーの販売承認の作成の詳細については、[ISV「」を参照してください](#)としての[チャンネルパートナーの販売承認 AWS Marketplace の作成 ISV](#)。

各 AWS Marketplace チャン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あります。プライベートオファーを作成するときは、によって設定された卸売コストから開始しますISV。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。卸売コストは、次の2つの方法のいずれかで決定されます。

- 定期的な割引 - は、AWS Marketplace チャンネルパートナーが、定期的な販売ISV承認により、販売価格から割引に同意した価格で製品を再販売することを許可します。AWS Marketplace チャンネルパートナーは、この割引を活用して、との価格交渉を行わずに製品の再販を継続できますISV。この割引は、ISVまたはチャンネルパートナーのいずれかによって終了されるまで、指定された日付まで、または無期限に継続するように設定できます。
- 非定期割引 - が AWS Marketplace チャンネルパートナーISVに付与する販売認可は、特定の購入者のみに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ISVに基づいて AWS Marketplace、チャンネルパートナーとに資金を配分します。

Tip

ISV またはチャンネルパートナーとして、のパートナーメニューから、付与または受け取った販売承認を表示できます[AWS Marketplace 管理ポータル](#)。

プライベートオファーの作成方法の詳細については、「[AWS Marketplace チャンネルパートナーによる作成](#)」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およびチャネルパートナーがAWS Marketplace チャンネルチームと接続することをお勧めします。連絡先がわからない場合は、aws-mp-channel@amazon.com に Eメールメッセージを送信してください。チームの担当者が 1 営業日以内に返信します。

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

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に製品の再販を許可できます。割引率または製品ディメンションあたりのカスタム料金を指定して、チャネルパートナーの卸売価格 AWS Marketplace を作成できます。パートナーは、購入者の AWS Marketplace チャンネルパートナーのプライベートオファーを作成するときに卸売価格をマークアップできます。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

以下の手順では、ISVsがチャ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 AWS Marketplace。この機能を使用するには、AWS Marketplace 管理ポータル の [パートナー] タブを使用する権限が必要です。詳細については、「[AWS Marketplace 販売者向けポリシー](#)」を参照してください。

として AWS Marketplace チャンネルパートナーの販売承認を作成するには ISV

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。

Tip

AWS Marketplace 販売者アカウントでサインイン AWS アカウント する前に、必ず別のアカウントからサインアウトしてください。

2. Partners タブを選択し、Create selling authorization を選択します。
3. 販売承認の詳細ページで、販売承認名 と販売承認の説明 を入力します。

Note

販売承認名と販売承認の説明に入力した情報は、販売者の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、ドロップダウンリストから承認する AWS Marketplace チャンネルパートナー (再販業者) 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。
- この販売承認の一部となる製品のいずれかを選択します。
- 適用する [割引] を選択します。

[割引タイプ] は以下の複数の方法で公表できます。

- [割引率] - 選択したすべての製品に 1 つの割引率 (パーセンテージ) を適用します。
- [個別の料金] - 特定の製品に特定の割引を適用します。
- 柔軟な支払いスケジュール - AWS Marketplace チャンネルパートナーの販売承認の柔軟な支払いスケジュールを設定します。

Note

プロフェッショナルサービス販売者では、個別の料金と柔軟な支払いスケジュールのみがサポートされています。

- 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。

販売承認期間の長さは、複数の方法で発行できます。

- 単一使用 - 1 つの販売承認に適用され、AWS Marketplace チャンネルパートナーがプライベートオファーを作成した後は適用されなくなります。
- 特定の期間 - によって選択された日付より後に適用されなくなった特定の期間にわたって持続します ISV。

- [期間設定なし] - 関係者のいずれかが終了するまで続きます。
8. (オプション) SaaS 契約製品の場合は、カスタム製品ディメンションを追加または削除し、追加の使用料を変更して販売承認をカスタマイズします。
 9. (オプション) 1 つ以上の購入者アカウントIDsを設定して、販売承認がそれらの購入者専用であることを指定します。
 10. (オプション) エンドユーザーライセンス契約 (EULA) バージョンを選択するかEULA、販売承認に含める 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では、カスタムのみがサポートEULAsされています。

11. (オプション) リセラー契約 for AWS Marketplace (RCMP) 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。
12. 販売承認の確認 を選択し、情報が正しいことを確認します。
13. 販売承認の作成を選択して販売承認を完了し、チャネルパートナーを承認します。

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、割引、作成日、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は、その日付を延長することはできません。ただし、販売承認を取り消し、いつでも再作成できます。販売承認を取り消すと、新しいオファーはその割引を使用できません。既存のオファーは影響を受けず、販売承認割引を保持します。

販売承認を選択し、クローン を選択することで、販売承認のクローンを作成することもできます。これにより、すべてが事前入力され、フィールドを編集できます。

プライベートオファーの分割プランの作成

AWS Marketplace 販売者として、分割払いプラン (フレキシブル支払いスケジュール と呼ばれます) 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。支払いスケジュールは、承認された契約期間に分散でき、顧客は定期的な分割払いで支払いを行うことができます。顧客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。以下のセクションでは、支払いスケジュールの作成と、分割払いプランの販売者レポートの管理について詳しく説明します。

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の作成

分割払いプランを使用してカスタム支払いスケジュールを作成するプロセスは、プライベートオファーを作成するプロセスの一部です。プライベートオファーの分割プランオプションを有効にするには、プライベートオファーを作成するときに、購入者支払いオプションセクションの分割プランによる契約料金を選択します。購入者支払いオプションを選択したら、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、「[プライベートオファー](#)」を参照してください。

Note

分割プラン付きのプライベートオファーの場合、複数年およびカスタム期間の Amazon Machine Image (AMI) 製品の場合は、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、追加の起動されたインスタンスの時間単位の料金を設定します。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、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。

購入者分割払いプランで、契約合計、支払い数、初回請求日を指定し、分割払いプランの生成を選択します。最大 60 件の支払いを追加できます。また、各支払い明細項目を調整することもできます。支払い明細項目を調整するたびに、購入者から支払うべき合計金額が更新されます。

Note

最初の分割払いの請求書日付は、プライベートオファーに対して顧客に初回請求書が送付される日付です。顧客から支払いを受け取った後 AWS Marketplace、各請求書の支払いを受け取ります。

Flexible Payment Scheduler 機能によって、請求書日付が契約期間内に収まっていることが検証されます。請求書の日付が契約期間より後である場合は、エラーメッセージが表示されます。

すべての請求書の金額と日付を確認したら、購入者から支払うべき合計金額が、プライベートオファーの過程で顧客に支払うべき合計料金と一致していることを確認します。プライベートオファーの作成を完了するには、残りのステップを完了します。

Note

顧客に対して拡張しているオファー受諾日付前に設定できる請求書日付は 1 つだけです。

販売者が定義しているスケジュールに基づいて顧客に請求され、顧客がオファーを受諾した後に請求が開始されます。オファーが承諾される前に最初の請求日がスケジュールされている場合、オファーが承諾された直後に処理されます。

Note

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。支払いスケジュールを変更するには、新しいオファーを作成する必要があります。

分割プランのレポート

柔軟な支払いスケジュールを持つプライベートオファーのレポートは、月次請求収益レポートの [セクション 4: Flexible Payment Schedule による契約](#) にあります。

でのプライベートオファーの修正 AWS Marketplace

AWS Marketplace 販売者は、購入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受け入れたときに最初に作成したアクティブな契約を置き換えるために、アップグレード、更新、修正を提供できます。例えば、新しい権限の付与、料金の割引の提供、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) を変更して、[標準化されたライセンス条件](#) を使用することができます。ユニット数と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、プライベートオファーの修正をサポートします。すべての AWS Marketplace 販売者は、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、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、SaaS 製品のプライベートオファーを修正、更新、アップグレードする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購入済み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 の [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

- [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](#)
- [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、および修正の報告](#)

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

以下の製品タイプのみがプライベートオファーの修正をサポートしています。

- SaaS 契約
- SaaS 契約 (追加使用量あり)

の「契約」タブには、次の追加の製品タイプが表示されます AWS Marketplace 管理ポータル。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。

- SaaS 使用量ベースの製品
- 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

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

次の手順 AWS Marketplace 管理ポータル を使用して、 からプライベートオファーのアップグレード、更新、および修正を作成できます。Channel Partner のプライベートオファー (CPPO) では、Channel Partner は再販承認で定義された通貨を使用するのみ契約ベースのオファーを作成できます。

プライベートオファーのアップグレード、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、詳細の表示 を選択します。
3. 契約を表示ページで、契約ベースのオファーの作成 を選択します。
4. 契約内容の修正ページで、販売者はプライベートオファーが更新用であるかどうかを示す必要があります。プライベートオファーが更新用であるかどうかを尋ねられた場合は、はいまたはいいえを選択します。

Note

また、契約内容の修正ページで、サービス日、製品ディメンション、オファー通貨、支払いスケジュール、更新ステータス、オファーの有効期限を変更することもできます。

5. 完了したら、オファーの作成を選択し、 を送信します。

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指定することをお勧めします。**upgrade** や **renewal** などの概要、カスタム社名などの使用もお勧めします。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。変更EULAsする日付、ディメンション、支払いスケジュール、および の情報を編集し、次へ を選択します。[Review and create (確認と作成)] ページで、情報を確認します。準備ができたなら、「契約ベースのオファーの作成」を選択します。

新しいプライベートオファーは、約 45 分で [Manage Private Offer (プライベートオファーの管理)] 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、オファーを選択してプライベートオファーの管理ページを開きます。

これにより、購入者は新しいオファーを承諾するか、引き続き元の契約に従うかを選択できます。

- 購入者が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、新しい契約は直ちに有効になり、契約は AWS Marketplace 管理ポータルの [契約] ページに表示されます。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承諾するのと同じ方法で、契約ベースのプライベートオファーを承諾します。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。

- 購入者がプライベートオファーのアップグレードまたは更新を期限前に承諾しない場合、元の契約が変更なしで有効に存続します。

Note

[SaaS 製品の Amazon SNS通知の場合](#)、購入者が契約ベースのオファーを受け入れる offer-identifier と、新しいとともに subscribe-success メッセージが表示されません。

アップグレード、更新、および修正の報告

アップグレードと更新の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。[日次顧客サブスクライバーレポート](#) レポートと [日別ビジネスレポート](#) 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](#) 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。[Previous offer ID (以前のオファー ID)] フィールドは、新しいオファーに先立つ以前のオファーの ID (存在する場合) を表示します。すべてのプライベートオファーでは、エントリはプライベートとマークされます。

⚠ Important

契約ベースのオファーは、購入者の現在のサブスクリプションを置き換えます。既存の請求書は変更されません。ただし、契約ベースのオファーの支払いスケジュールは、以前のサブスクリプションからの保留中の請求書を置き換えます。

プライベートオファーの将来の日付の契約の作成

AWS Marketplace 販売者として、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。これとは対照的に、将来の日付の契約 (FDA) では、購入者は事前に定義された将来の日付に製品ライセンスまたは権利を受け取ります。FDA を使用して、購入者との既存のトランザクションの更新を設定できます。FDA は、Software as a Service (SaaS) 製品でサポートされています。この製品は、柔軟な支払いの有無にかかわらず、契約および消費 (CCP) 料金で契約できます。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。

を使用すると FDA、購入者が製品の使用を開始するタイミングではなく、購入者が選択したタイミングで取引を終了できます。FDA を使用して、でのトランザクションに対して以下のアクションを依存的に実行できます AWS Marketplace。

- 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。

トピック

- [将来のデータ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分割プランを使用する](#)
- [将来の日付の契約の通知を受け取る](#)
- [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](#)

将来のデータ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。

トピック

- [将来の日付の契約の作成](#)
- [将来の日付の契約で分割プランを使用する](#)
- [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ときは、**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FDAに を使用するには、サービス開始日を更新したい契約の終了日に合わせます。

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分割プランを使用する

で分割プランを使用するとFDA、契約の署名日から契約終了日までの間に購入に対する支払いをいつでも設定できます。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、「」を参照してください[the section called “支払いスケジュールの作成”](#)。

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 の既存の Amazon Simple Notification Service (Amazon SNS) 通知はすべて、でも機能しますFDA。の場合FDAs、両方の Amazon SNS トピックは契約開始日 (契約署名日ではなく) に開始されます。詳細については、「[the section called “SaaS 製品の 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ーの再販に関する将来の日付の契約を使用する

メーカーとリセラーは、AWS Marketplace チャネルパートナーのプライベートオファー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 Channel Partner Private Offers (CPPOs) と同様に、メーカーは再販承認を延長して AWS Marketplace、Channel Partner が将来の開始日 CPPOs で作成することを承認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 “として販売承認を作成する ISV”](#) 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 Channel Partner プライベートオファーを作成するときに AWS Marketplace Channel Partner が指定できる最大サービス開始日 AWS Marketplace になります。

Note

メーカー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 年先の将来のサービス日を指定できます。

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ーを作成する手順と、将来の日付の通常のプライベートオファー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ー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ー”](#)」を参照してください。

AMIベースの製品 AWS Marketplace

AWS Marketplace 販売者として、[Amazon Machine Images \(AMIs\)](#) 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。AMI 製品のカスタムを作成すると、購入者はそれを使用して、製品がすでにインストールされ、すぐに使用できる状態でEC2インスタンスを作成できます。このトピックでは、を使用して AWS Marketplace 製品を配信AMIsする方法について説明します。

購入者AMIが指定したを使用する場合、製品用に作成した料金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課金されます。購入者はAMI、の新しいカスタムバージョンの作成などAWS、AMIsで他のを使用するのと同じ方法で製品を使用できますAMI。EC2 から作成されたインスタンスAMIは、製品AMI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、次のいずれかの方法で配信できます。

- シングル AMI – 購入者はを選択し、EC2インスタンスのテンプレートAMIとして使用します。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、「[単一AMI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1つ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、「[を使用したAMIベースの製品の配信 AWS CloudFormation](#)」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、以下のトピックを参照してください。

AWS Marketplace

- [の製品料金 AWS Marketplace](#)
- [を使用したAMI製品のカスタム計測の設定 AWS Marketplace Metering Service](#)

AMIベースの製品

- [AMIでの ベースの製品について AWS Marketplace](#)
- [単一AMI製品の作成](#)
- [単一AMI製品を AWS Marketplace 販売者として管理する](#)
- [を使用した AMIベースの製品の配信 AWS CloudFormation](#)
- [でAMIs使用するために を構築するためのベストプラクティス AWS Marketplace](#)
- [AMI の製品料金 AWS Marketplace](#)
- [でAMI製品の Amazon SNS通知を受信する AWS Marketplace](#)
- [AMI の製品チェックリスト AWS Marketplace](#)
- [AMIの ベースの製品要件 AWS Marketplace](#)

AMIでの ベースの製品について AWS Marketplace

AWS Marketplace 販売者として、[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ベースの製品を使用するための主要な概念について説明します。

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](#)
- [年間契約の変更](#)

製品のライフサイクル

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、製品の名前、説明、料金 AWS Marketplace を含むプロパティを で設定します。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。ソフトウェアの配信方法によっては、購入者が独自の の作成に使用する単一の AMI、テンプレート AMIs を含む AWS CloudFormation 1 つ以上の のセット、またはソフトウェアパッケージになる場合があります AMIs。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、「[を使用したAMI製品のカスタム計測の設定 AWS Marketplace Metering Service](#)」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 カタログに発行して、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。


Server [製品](#) ページを使用して、製品のリストを表示します。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] 変更リクエストを使用します。これにより、AWS Marketplace 販売者オペレーションチームが当社の [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。

AMIの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も終了しません。ソフトウェアの新しいバージョン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適用して、製品を最新の状態に保つ必要があります。

AMIベースの製品ライフサイクルの例として、販売者が で AMIベースの製品を販売したいとします AWS Marketplace。以下は、販売者が長期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 に発行します 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されている場合にのみ、お客様が確認 AWS Marketplace せずに 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、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 – 製品が不要になった場合、販売者は から製品を削除します AWS Marketplace。

この例では、販売者は製品AMIに 3 つの異なるバージョンの を作成しましたが、新しい購入者が利用できるのは 2 つだけでした (製品を削除する前)。

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

AMIベースの製品を作成および管理するための手順の詳細については、「」を参照してください[単一AMI製品の作成](#)。

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。この製品コードは、製品の AMIs に関連付けられており、製品の使用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。例えば、顧客が をサブスクライブして起動しAMI、設定して、新しい を生成しますAMI。新しい には元の製品コードAMIがまだ含まれているため、正しい使用状況の追跡とアクセス許可はそのまま残ります。

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、インスタンスメタデータAMIとして から作成されたインスタンスにアタッチされます。その製品コードAMIを持つ を使用してインスタンスを作成すると、顧客は関連する製品 ID を示す請求書を受け取ります。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。

販売者として、ソフトウェアは実行中の Amazon Elastic Compute Cloud (Amazon EC2) インスタンスの製品コードをインスタンスメタデータから実行時に取得できます。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。製品がテストのために制限付き状態で公開されるまで、AMIの製品コードをAPI呼び出すことはできません。製品コードの検証の詳細については、「[でソフトウェアが実行されていることを確認する AWS Marketplace AMI](#)」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 カタログ 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。
- AWS リージョン 製品が利用可能な 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。次に、既存の詳細を上書きする更新を行います。

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、AMIアセットが一度追加されるとインスタンスとリージョンにプロビジョニングされた場合、完全に削除されるのではなく、制限することができます。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、製品を作成または編集するときは、AWS Marketplace 管理ポータル ユーザーインターフェイス内で必要な変更を行います。ただし、いくつかのオペレーションでは、Product Load Form () を使用するように指示しますPLF。

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。を取得する方法はいくつかありますPLF。

- PLF 既存の製品のは、の製品の詳細ページからダウンロードできます AWS Marketplace 管理ポータル。
- 必要なアクションのメニュー項目PLFを選択すると、をダウンロードするように求められます。例えば、毎月請求される新しいサーバー製品を作成する場合は、適切なをダウンロードするよう求められますPLF。

アクションが既存の製品に対する編集である場合、PLF にはその製品の情報が事前に入力されているため、更新する詳細のみを変更する必要があります。

- 新しい空白のが必要な場合はPLF、AWS Marketplace 管理ポータル [ファイルアップロード](#) ページにPLFs、作成する製品のタイプに基づいて、へのリンクがあります。

を完了したらPLF、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 それ自体には、「指示」タブに詳細な指示があります。

年間契約の変更

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のサイズを切り替える
- 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、変更の比例配分コストが 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インスタンスの按分コストは、契約期間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間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日割

りコストは、新しく追加された Amazon EC2 インスタンスタイプの日割りコスト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、パブリック 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。

購入者によって行われた修正は、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](#) – 支払い。

単一AMI製品の作成

Amazon Machine Image (AMI) セルフサービスエクスペリエンスは、製品リストを作成し、変更 リクエストを行う際のガイドとなります。セルフサービスエクスペリエンスを使用すると、AWS Marketplace 販売者オペレーションチームが処理するの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、 のカタログシステムと連携しているため AWS Marketplace、AWS Marketplace 販売者オペレーション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の の製品リストを作成する方法について説明しますAMI。お客様はAMIs、を使用して、製品がすでにインストールおよび設定されている Amazon EC2 インスタンスを作成します。

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前に、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できます。これは、販売者として登録し、 で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](#)」を参照してください。
3. 売りたい製品がある。AMIベースの製品の場合、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を作成していることを意味します。で使用するAMIためのの準備の詳細についてはAWS Marketplace、「」を参照してください[でAMIs使用するために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。ステップを進めるときは、、、AWS リージョンインスタンスタイプ、AMI詳細などの製品情報とAMIデプロイ設定を指定します。また、料金、国の可用性、、返金ポリシーなどEULA、トランザクションの詳細も設定します。オプションとして、のAWS アカウント IDs許可リストを指定して、制限付きステータスの状態で製品にアクセスしてテストできます。

開始する前に、セルフサービスエクスペリエンスの以下の重要な側面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、保存を選択して終了し、完了したステップをステージングエリアに送信します。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. [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、単一AMI製品のライセンスタイプのいずれかを選択します。
 - Bring your own license (BYOL) – ユーザーが の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) インスタンスまたはその他の AWS リソースに対して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支払い – 購入者が時間単位または時間単位で年間契約で支払う製品。 は、 の製品コードに基づいて計測 AWS を行いますAMI。
 - [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。
 - AMI 契約料金 – 購入者が前払い料金を支払う AWS CloudFormation スタックを持つ単一AMI 製品または単一AMI製品。
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詳細)、公開オファーの詳細(料金、国別の可用性EULA、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。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) と AWS Marketplace 販売者オペレーションチームのテストアカウントのみがアクセスできます。Server 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 販売者オペレーションチームは、パブリック への移行を承認する前に、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 販売者オペレーションチームが処理します。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の準備と送信の詳細については、「」を参照してください [追加リソース](#)。

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](#)
- [で公開するための製品の送信 AWS Marketplace](#)

単一AMI製品を に送信する準備の詳細については AWS Marketplace、次のリソースを参照してください。

- [でAMIs使用するために を構築するためのベストプラクティス AWS Marketplace](#)
- [AMI の製品チェックリスト AWS Marketplace](#)
- [AMIの ベースの製品要件 AWS Marketplace](#)

単一AMI製品を AWS Marketplace 販売者として管理する

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、製品全体に関するソフトウェアとメタデータの 1 つ以上のバージョンのセットが含まれます。製品を作成するときは、製品名 AWS Marketplace、説明、料金など、 で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して、製品またはバージョンに変更を加えることができます。

AMIベースの製品に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。
- AWS リージョン 製品が利用可能な 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、単一の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 1 つのAMI製品の変更リクエストを作成する AWS Marketplace](#)
- [単一AMI製品の可視性の更新](#)
- [の単一AMIインスタンスの追加と制限 AWS Marketplace](#)
- [での単一AMI製品のバージョン管理 AWS Marketplace](#)
- [での単一AMI製品情報の更新 AWS Marketplace](#)
- [AWS リージョン および国による単一AMI製品の可用性の管理](#)
- [単一AMI AWS Marketplace 製品のエンドユーザーライセンス契約 \(EULA\) の更新](#)

- [での単一AMI製品の返金ポリシーの更新 AWS Marketplace](#)
- [AWS Marketplace へのアクセスの許可 AMI](#)
- [から製品を削除する AWS Marketplace](#)
- [で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 1 つのAMI製品の変更リクエストを作成する 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。

次の状況で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AMI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。AMIベースの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 カタログ 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、[リクエスト](#) 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、特定の変更については、Product Load Form () をダウンロード、入力、アップロードする必要がありますPLF。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1つを選択すると、作成しようとしているリクエストPLFに対して正しい をダウンロードするよう求められます。PLF には、既存の製品の詳細に関する情報があらかじめ入力されています。完了した をPLF AWS Marketplace 管理ポータル [ファイルアップロード](#) ページにアップロードできます。

Note

最新の をダウンロードして使用することを強くお勧めしますPLF。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。PLF 製品を選択し、製品ロードフォームのダウンロード を選択すると、サーバーの製品ページから製品の最新の情報を確認できます。

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、「[で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](#)」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内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。

- [での単一AMI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](#)

単一AMI製品の可視性の更新

AWS Marketplace 販売者として、製品の可視性を更新して、で単一の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パブリック 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。製品の可視性が制限 に設定されている場合、製品は許可リストの AWS アカウント IDs にのみ表示されます。製品を表示できる の AWS アカウント IDs 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、製品の可視性と制限付き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

で製品を表示できる購入者を変更するには AWS Marketplace、可視性の更新 を使用します。

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、変更リクエストをパブリック に移動するには、AWS Marketplace 販売者オペレーションチームの承認プロセスを経る必要があります。

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

制限された状態で製品を表示できる AWS アカウント IDs のリストを変更するには、許可リストの更新 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. 可視性に望ましい AWS アカウント IDs を追加し、 をカンマIDs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。

の単一AMIインスタンスの追加と制限 AWS Marketplace

AWS Marketplace 販売者として、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。購入者が使用できる単一のAMI製品に新しいインスタンスを追加できます。同様に、新しい購入者が特定のインスタンスから単一のAMI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。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手順を説明します。

トピック

- [インスタンスを追加する](#)
- [インスタンスの制限](#)

インスタンスを追加する

購入者が単一 を使用できる新しいインスタンスを追加できますAMI。

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. 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

現在の料金モデルが無料ではない、または Bring Your Own License (BYOL) モデルを使用している場合は、料金を追加する追加のステップがあります。
新しいインスタンスの料金を指定して [インスタンスを追加] を作成した場合、または料金を引き上げるために [料金を更新] した場合は、変更を行った日から 90 日間はセルフ

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お客様は、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。お客様がアクセスする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、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、「[プライベートオファーのアップグレード、更新、および修正](#)」を参照してください。

インスタンスの制限

新しい購入者が特定のインスタンスから単一のAMI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。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。制限付きインスタンスAMI上の1つの既存のユーザーは、サブスクライブされている限り、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。

での単一AMI製品のバージョン管理 AWS Marketplace

に基づいて Amazon マシンイメージ (AMI) ベースの製品を作成する場合 AWS Marketplace、ソフトウェアの特定のバージョンを含めます。AMIの 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。製品は up-to-date、ソフトウェアの新しいバージョンで保管する必要があります。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、またはバージョンAMIに関連付けられた 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

製品の新しいバージョンは、製品、ベースイメージ、またはAMI製品の を変更する必要がある場合に追加できます。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

AMI 用の の作成については AWS Marketplace、「」を参照してください [AMIを使用するために 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。[新しいバージョンを追加] フォームが表示され、最新バージョンの情報が事前入力されます。
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 – 有効な文字列を入力します (例 : *1.1* または *Version 2.0*)。製品全体で一貫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. 新しい Amazon マシンイメージ (AMI) セクションで、次の情報を指定します。
- Amazon マシンイメージ ID – このバージョンAMIで使用する の AMI ID を入力します。ID は、コンソールの AMI のリストから確認できます。 [AMIs](#) 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AMI必要があります。これに関連付けられたスナップショットは暗号化AMIできません。
 - IAM アクセスロール ARN – AWS Marketplace が へのアクセスを許可する (ARN) ロールの Amazon リソースネーム AWS Identity and Access Management (IAM) を入力しますAMI。IAM ロールを作成する方法については、「」を参照してください [AWS Marketplace へのアクセスの許可 AMI](#)。の標準形式を使用します。IAMARN 例 : *arn:aws:iam::123456789012:role/RoleName*。は AWS Marketplace 、販売者アカウントに存在するARN必要があります。
 - OS ユーザー名 – 用 Linuxベースの ではAMIs、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。
 - ポートのスキャン – 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を入力します。SSHLinux AMI または のRDPポート Windows AMI。
6. まだ公開されていない場合は、設定を展開して を AWS Marketplace カスタマーウェブサイトセクションAMIに公開し、次の情報を指定します。
- 使用手順 – を使用する手順、AMIまたは の使用に関する詳細情報へのリンクを入力しますAMI。例: *To get started with the product, navigate to 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関する情報を提供します。プロトコル (https または http)、相対 URL (例 : */index.html*)、およびポート (例 : *443*) 購入者が製品にアクセスするために使用できる。(ホスト名はEC2インスタンスに依存するため、相対パスのみを指定する必要があります)。
 - オペレーティングシステム (OS) – で使用される OS の名前を入力します AMI (例 : *Amazon Linux*)。)
 - OS バージョン – に OS の特定のバージョンを入力しますAMI。
 - [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。

- セキュリティグループのレコメンデーション – プロトコル (TCP または UDP)、許可するポートの範囲、および のリスト IPv4 CIDR IPs (例えば、xxx.xxx.xxx.xxx/nn の形式で **192.0.2.0/24**).
7. [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。
 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。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 ルートユーザーとして、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 <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>

Amazon Web Services, 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 Amazon Web Services 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、是非推奨としてマークされます。詳細については、の Amazon Elastic Compute Cloud ユーザーガイドの「[の非推奨AMI](#)」を参照してください。Windows インスタンス。

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と、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 ルートユーザーとして、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>

Amazon Web Services, 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 Amazon Web Services Inc., 410 Terry Ave. North, Seattle, WA 98109-5210

での単一AMI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 で変更できます 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、次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
- 製品ビデオ 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。

AWS リージョン および国による単一AMI製品の可用性の管理

で製品を作成するときは AWS Marketplace、使用可能な 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。例えば、購入者が米国に所在し、米国から購入している場合でも、欧州 (フランクフルト) リージョンに製品をインストールする予定がある場合があります。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、リージョンと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。顧客にアクセスさせたいリージョンで、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、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、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、[「プライベートオファーのアップグレード、更新、および修正」](#)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

Amazon Web Services, 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、今後のリージョンサポートの更新 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、 が AWS リージョン ユーザーに代わって新しく起動 AWS Marketplace 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ます。
 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。

単一AMI 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 (EULA) 製品の使用を管理するエンドユーザーライセンス契約 (AMI) を更新できます。EULA は、 のパブリックソフトウェアリストの製品リストページにあります AWS Marketplace。独自の を適用するEULAが、 [\(SCMP\) の AWS Marketplace 標準契約](#) を使用できます。以下の手順は、EULA単一AMI製品の を更新する方法を示しています。

の詳細についてはEULA、「 」を参照してください [での標準化された契約の使用 AWS Marketplace](#)。

を更新するには EULA

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. [Standard Contract for AWS Marketplace \(SCMP\)](#) を選択するか、独自のカスタム を送信できますEULA。カスタム の場合EULA、Amazon S3 バケットからURLカスタム契約の を指定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。

での単一AMI製品の返金ポリシーの更新 AWS Marketplace

AWS Marketplace 販売者として、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。次の手順は、返金ポリシーを更新する方法を示しています。

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 へのアクセスの許可 AMI

に新しい Amazon マシンイメージ (AMI) を追加するリクエストを作成する場合 AWS Marketplace、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AMIする必要があります。に対して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AMIと、 がロールを引き受けることができる信頼ポリシーAMIを作成し

て AWS Marketplace、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 1 回だけ作成する必要があります。次の手順では、AWS Marketplace へのアクセスを許可する AWS Marketplace アセット取り込みのロールを作成する方法を示しますAMI。

アセット取り込みの AWS Marketplace AMI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 – AWS Marketplace を選択します。
 - ユースケースを選択する – Marketplace を選択する – AMIアセットの取り込み。
 - 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. を選択します。AWSMarketplaceAmiIngestion ポリシー。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. 「ロール」が表示されます。*rolename* ページの上部に「」が作成され、ロールがロールのリストに表示されるはずですが。

このページでは、作成したロールを選択すると、そのロールが `arn:aws:iam::123456789012:role/exampleRole` ARNの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、ARN IAM アクセスロールARNの を使用します。

から製品を削除する AWS Marketplace

製品を公開したら、 から削除できます AWS Marketplace。これはサンセットとも呼ばれます。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

エラストを送信します。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した後、新しいお客様は製品に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。このトピックでは、AWS Marketplace 削除条件の詳細と、 から製品を削除する方法について説明します AWS Marketplace。

からの製品削除のリクエストは、次の条件で処理 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。製品の詳細ページには引き続き を使用してアクセスURLでき、パブリック検索エンジンでインデックス作成され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 Sellers 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。リクエストの変更ド롭ダウンリストから、製品の可視性の更新 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. 情報が正しいかどうかを確認し、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。AWS Marketplace 販売者オペレーションは、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。

で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

で製品の情報を変更すると AWS Marketplace、エラーが発生する可能性があります。このトピックでは、いくつかの一般的なエラーについて説明し、それらの修正方法を提案します。

- のスキャン AMI — のスキャン時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 アクセス許可を付与していませんAMI。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、「[AWS Marketplace へのアクセスの許可 AMI](#)」を参照してください。
 - スキャンでセキュリティの問題または一般的な脆弱性と露出 (CVEs) が見つかった場合は AMI、イメージ内のオペレーティングシステムに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、「[AMIの ベースの製品要件 AWS Marketplace](#)」を参照してください。

の構築に関する一般的なガイドラインについてはAMI、「」を参照してください [でAMIs使用するために を構築するためのベストプラクティス AWS Marketplace](#)。

- AWS Marketplace 管理ポータル フィールド – のいくつかのフィールドには、非常に具体的な情報 AWS Marketplace 管理ポータル 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、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。
 - の提供を求められた場合はARN、通常、コンソールの他の場所にあります。例えば、AWS Marketplace へのアクセスを許可するために作成したIAMロールARNの AMIは、IAMコンソールの [ロールページ](#) にあります。ARNs すべてののに同様の形式があります。例えば、IAMロールARNは `arn:aws:iam::123456789012:role/exampleRole` の形式です。

- ロゴと動画は、としてコンテンツURLに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、「[で公開するための製品の送信 AWS Marketplace](#)」を参照してください。

- Product Load Form (PLF) の問題 – スプレッドシートに含まれる手順PLFs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 販売者オペレーション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。

を使用した AMIベースの製品の配信 AWS CloudFormation

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。テンプレートを使用して、製品のクラスターまたは分散アーキテクチャを定義したり、さまざまなAMI組み合わせや製品設定を選択したりできます。CloudFormation テンプレートは、関連する設定ファイルと Lambda 関数を含む単一の Amazon マシンイメージ (AMI) を配信するように設定できます。購入者は、ソリューションの選択を参照し AWS Marketplace、ワンクリックで購入し、指定した CloudFormation テンプレートを使用してデプロイできます。

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。

また、を使用してサーバーレスアプリケーションに Lambda 関数を含めるAMIことで、購入者が を通じてデプロイできるようにすることもできます CloudFormation。Lambda 関数とサーバーレスアプリケーションを に含める方法についてはAMI、このガイド[サーバーレスアプリケーションコンポーネントの追加](#)の「」を参照してください。

トピック

- [製品の出品の構築](#)
- [CloudFormation テンプレートの準備](#)
- [テンプレートインフラストラクチャの料金見積りの取得](#)
- [アーキテクチャ図](#)
- [送信要件の一致](#)
- [製品リクエストの送信](#)
- [サーバーレスアプリケーションコンポーネントの追加](#)

製品の出品の構築

製品を送信するには、 を準備して検証しAMIs、テンプレートを作成し AWS CloudFormation 、アーキテクチャ図を作成し、製品ロードフォームに記入し、 にマテリアルを送信する必要があります AWS Marketplace。の作成と検証から開始しAMIs、 CloudFormation テンプレートを完了して検証することをお勧めします。これらの手順が完了したら、アーキテクチャ図を作成し、ソフトウェアとインフラストラクチャの料金を見積もる必要があります。AWS Marketplace は、送信内容を検証し、販売者と連携して製品を公開します。[AWS Pricing Calculator](#) を使用すると、テンプレートのインフラストラクチャのコスト見積もりに役立ちます。

CloudFormation テンプレートの準備

CloudFormation テンプレートを構築するには、テンプレートの前提条件を満たし、必要な入力パラメータとセキュリティパラメータを指定する必要があります。CloudFormation テンプレートを送信するときは、以下のセクションのガイドラインを使用してください。

テンプレートの前提条件

- 製品 で AWS リージョン 有効なすべての で AWS CloudFormation 、コンソールを介してテンプレートが正常に起動されていることを確認します。ツールを使用してテンプレートを[TaskCat](#) テストできます。

- 単一AMI製品を作成する場合、テンプレートには 1 つの のみを含める必要がありますAMI。
- AMIs は、各リージョンの [マッピングテーブル](#) にある必要があります。チームはクローンを作成 AMIIDs した後、AWS Marketplace を更新します。ソースAMIは にありus-east-1、他のリージョンはプレースホルダーを使用できます。次の YAML の例を参照してください。

```
Mappings:
  RegionMap:
    us-east-1:
      ImageId: ami-0123456789abcdef0
    us-west-1:
      ImageId: ami-xxxxxxxxxxxxxxxxxxxxx
    eu-west-1:
      ImageId: ami-xxxxxxxxxxxxxxxxxxxxx
    ap-southeast-1:
      ImageId: ami-xxxxxxxxxxxxxxxxxxxxx
```

- AMIs CloudFormation テンプレートに含まれる は、発行する製品の であるか、最新の Amazon Linux 2 AMIなどの AWS管理AMIの である必要があります。コミュニティや、ユーザーAMIまたはその他のサードパーティーがAMI所有および共有しているコミュニティを含めないでください。AWSマネージド を使用するにはAMI、ハードコーディング の代わりに [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AMIIDs。例えば、AMIID を指定する CloudFormation テンプレート内では、動的リファレンス を使用しますImageId: `'{{resolve:ssm:/aws/service/ecs/optimized-ami/amazon-linux-2/recommended/image_id}}'`。
- 特定のアベイラビリティゾーン (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。すべての顧客がすべての にアクセスできるわけではなくAZs、アカウントごとに異なる方法でマッピングAZsされます。
- Lambda 関数、設定ファイル、スクリプトなどの依存関係を に含めることができますAMI。詳細については、「[ステップ 1: サーバーレスアプリケーションを作成する](#)」を参照してください。
- 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、プレースメントグループを使用することを検討してください。

- ソリューションに Docker コンテナが含まれている場合は、Docker イメージを に組み込む必要がありますAMI。
- AWS Marketplace チームによるレビューを容易にし、お客様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。

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) を含めないでください。
- 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。
- 利用可能な入力には 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の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CIDR 範囲
- パブリックインターネットからの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 に開いていないことを確認します。
- デフォルトの仮想プライベートクラウド (VPC) を使用する代わりに、適切なアクセスコントロールリスト (ACLs) とセキュリティグループVPCを使用して を構築することをお勧めします。
- AWS Identity and Access Management (IAM) ロールを使用して [AssumeRole](#)から を呼び出すことで、お客様の AWS 環境へのアクセスを有効にします AWS Security Token Service。

- **最小権限を付与**するようにIAMロールとポリシーを設定し、絶対に必要な場合にのみ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。

テンプレートインフラストラクチャの料金見積りの取得

顧客に提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 Pricing Calculator](#) を使用して指定した見積もりに基づくものです。この見積もりには、一般的なデプロイ向けのデフォルト値ととも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が含まれている必要があります。

テンプレートの推定月額コストを計算したら、米国東部 (バージニア北部) リージョンの Save and Share リンク AWS Marketplace を指定します。これは、送信プロセスの一部です。

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。

- で標準デプロイを図解します 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 テンプレートを介して AWS のサービス デプロイされた各 の最新 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やオンプレミスAPIsの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。

- 図は 1100 x 700 ピクセルのサイズである必要があります。ストレッチやクロップを行わずに、元の図の比率を維持します。

送信要件の一致

AWS CloudFormation テンプレートを使用して配信される製品を送信するには、次のリソースを提供する必要があります。

- CloudFormation テンプレート
 - 単一AMI製品には 1 つから 3 つの CloudFormation テンプレートを含めることができます。
- 各テンプレートのデフォルト設定に対するインフラストラクチャの見積もり価格。
- アーキテクチャ図とアーキテクチャメタデータ
- 入力済みの Product Form ([AWS Marketplace 管理ポータル](#) から利用可能)
 - 単一AMI製品の場合は、[商用製品](#) フォームを使用します。

製品フォームには、リファレンスとして送信情報の例が含まれます。

各製品について、必要な製品データとメタデータのほとんどは、従来の単一AMI製品と同じです。したがって、テンプレートを使用して CloudFormation 配信AMIされる各は、で説明されている標準と要件を満たし続ける必要があります AWS Marketplace。

テンプレートごとに CloudFormation、次の情報も指定する必要があります。

フィールド	説明	制限事項
タイトル	アーキテクチャのタイトル。これは詳細ページとフルフィルメントページ、およびアーキテクチャ詳細を表示するポップアップに表示されます。	50 文字
短い説明	詳細およびフルフィルメントのページに表示されます。	200 文字

フィールド	説明	制限事項
長い説明	これはアーキテクチャ詳細ポップアップに表示されません。	2000 文字

CloudFormation 製品を使用した AMIベースの配信には、次のフィールドが必要です。

- ソリューションのタイトル
- ソリューションの短い説明
- ソリューションの長い説明
- CloudFormation テンプレートの場合 (ソリューションあたり最大 20)
 - デプロイメントタイトル (テンプレートごと)
 - 短い説明 (テンプレートごと)
 - 長い説明 (テンプレートごと)
 - アーキテクチャの図 (テンプレートごと)
 - インフラストラクチャの料金表見積もり (テンプレートごと)
 - この CloudFormation テンプレートに含まれる製品/コンポーネントのリスト
 - この CloudFormation テンプレートでサポートされているリージョンのリスト

製品リクエストの送信

[AWS Marketplace 管理ポータル](#) を使用して、製品を送信します。[アセット] タブで、[ファイルのアップロード] を選択します。送信するファイルをアップロードし、簡単な説明を入力します。YAML 形式と JSON 形式の両方がサポートされています。リクエストの処理には 3~5 週間かかります。これには、以下の処理が含まれます。

- および CloudFormation テンプレートのAMIテンプレートAMI、 、メタデータ CloudFormationの確認
- AWS Marketplace 製品への CloudFormation テンプレートの発行

サーバーレスアプリケーションコンポーネントの追加

1 つ以上の AWS CloudFormation テンプレートを使用して配信される 1 つ以上の Amazon マシンイメージ (AMIs) 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。例えば、1 つはコントローラーサーバーとしてAMI設定され、もう 1 つはワーカーサーバーとしてAMI設定され、AWS CloudFormation スタックとして配信される製品を作成できます。スタックの作成に使用される AWS CloudFormation テンプレートには、いずれかのサーバーで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。これらのセクションでは、このタイプの製品を作成して提供する方法について説明します。

製品の作成については、AMIs「」を参照してください[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[を使用した AMIベースの製品の配信 AWS CloudFormation](#)。

サーバーレスアプリケーションを定義するときは、に保存している 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、リスティング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遂行の一環として、AMIsサーバーレスアプリケーション、および AWS CloudFormation テンプレートを各の 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発行する](#)
- [ステップ 3: テンプレートを作成する CloudFormation](#)
- [ステップ 4: CloudFormation テンプレートと設定ファイルを送信する](#)
- [ステップ 5: アプリケーションアクセス許可を更新する AWS Serverless Application Repository](#)
- [ステップ 6: を共有する AMI](#)
- [ステップ 7: AMIおよびサーバーレスアプリケーションを使用して 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説明については、の AWS Labs の [AWS「::Serverless::Application」](#) を参照してください GitHub。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。

```
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発行する

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、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へのアクセス AWS Serverless Application Repository 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. で Amazon S3 コンソールを開きます <https://console.aws.amazon.com/s3/>。
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。

```
{
  "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. 置換 `123456789012` AWS アカウント ID を持つ Condition 要素。Condition 要素は、AWS Serverless Application Repository のみが指定された からアプリケーションにアクセスするアクセス許可を持っていることを確認します AWS アカウント。
6. [Save] 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 テンプレート

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、[「を使用したサーバーレスアプリケーションの発行 AWS SAM CLI」](#)を参照してください。

ステップ 3: テンプレートを作成する CloudFormation

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、必要な入力パラメータとセキュリティパラメータを指定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 テンプレートでは、サーバーレスアプリケーションとを参照できますAMI。ネストされた CloudFormation テンプレートとリファレンスサーバーレスアプリケーションは、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も使用できます。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 テンプレートには次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、によってクローンされると更新されます AWS Marketplace。

テンプレートで CloudFormation設定ファイルとスクリプトファイルを参照する場合は、次の形式を使用します。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://${CFRefFilesBucket}.${S3Region}amazonaws.com/
                    ${CFRefFilesBucketKeyPrefix}/sampleScript.sh
                  - S3Region:
                      !If
                        - GovCloudCondition
                        - s3-us-gov-west-1
                        - s3
                      owner: root
                      mode: '000700'
                      authentication: S3AccessCreds
          ..
          ..
          ..
  SampleNestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: 'https://sellerbucket.s3.amazon.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を送信する

CloudFormation テンプレート、設定、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アクセス許可を付与 AWS Marketplace 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。

```
{
  "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: アプリケーションアクセス許可を更新する AWS Serverless Application Repository

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 AWS Marketplace アクセス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、[Save] (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: を共有する AMI

に AMIs 構築および送信されるすべての AWS Marketplace は、すべての製品ポリシーに従う必要があります。セルフサービス AMI スキャンは、で利用できません AWS Marketplace 管理ポータル。この機能を使用すると、の スキャンを開始できます AMIs。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。が正常にスキャン AMI されたら、製品ロードフォームをアップロードして、AWS Marketplace 販売者オペレーションチームによる処理 AMI のために を送信します。

ステップ 7: AMI およびサーバーレスアプリケーションを使用して 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 テンプレートを通じ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。単一の製品と複数の AMI 製品には、異なる AMI 製品ロードフォームが必要です。製品ロードフォームでは、テンプレート URL に CloudFormation パブリックを指定します。CloudFormation テンプレートはパブリックの形式で送信する必要があります URL。
- AWS Marketplace 管理ポータルを使用してリストを送信します。[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け取ったら、は リクエストの処理 AWS を開始します。

リストを送信すると、は製品ロードフォーム AWS Marketplace を確認して検証します。さらに、は AMIs および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 販売者オペレーションチームが E メールで連絡します。

でAMIs使用するために 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。AMIs に構築され、送信される 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](#)
- [の構築 AMI](#)
- [AMI 用の の準備と保護 AWS Marketplace](#)
- [のパブリッシュ要件AMIのスキャン](#)
- [でソフトウェアが実行されていることを確認する AWS Marketplace AMI](#)

再販権の確保

お客様は、AWSが提供する Amazon Linux、 、RHEL、SUSEおよび Windows を除き、無料ではない Linux ディストリビューションの再販権を保護する責任がありますAMIs。

の構築 AMI

の構築には、次のガイドラインを使用しますAMIs。

- ガルトログインの無効化を含むすべての AWS Marketplace ポリシーAMI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AMIで を作成します。
- Amazon Elastic Block Store (Amazon EBS) がAMIsサポートする、メンテナンスの行き届いた既存の製品から、 などの信頼できる信頼できるソースが提供するライフサイクルを明確に定義して製品を作成します AWS Marketplace。
- ほとんどの up-to-dateオペレーティングシステム、パッケージ、ソフトウェアAMIsを使用して構築します。

- 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AMIを使用するパブリックからすべてのを開始AMIsする必要があります。
- の構築、更新、再発行の反復可能なプロセスを開発しますAMI。
- 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。ec2-user をお勧めします。
- 実行中のインスタンスをファイナルから必要なエンドユーザーエクスペリエンスAMIに設定し、に送信する前に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 AWS Marketplace。
- 以下のようにポート設定を確認してください。
 - Linux ベース AMIs – 有効なSSHポートが開いていることを確認します。デフォルトのSSHポートは 22 です。
 - Windows ベース AMIs – RDP ポートが開いていることを確認します。デフォルトのRDPポートは 3389 です。また、WinRM ポート (デフォルトは 5985) は 10.0.0.0/16 と 10.2.0.0/16 に関する必要があります。

の作成の詳細についてはAMI、次のリソースを参照してください。

- Amazon EC2ユーザーガイドでの[独自の作成 AMI](#)
- Amazon EC2 ユーザーガイドでの[カスタム Windows の作成 AMI](#)
- [EBS-backed インスタンスから Amazon マシンイメージ \(AMI\) を作成するにはどうすればよいですか？](#)
- [Amazon Linux AMI](#)
- [Amazon EC2 インスタンスタイプとインスタンスタイプ](#)

AMI 用の の準備と保護 AWS Marketplace

安全な を作成するために、次のガイドラインをお勧めしますAMI。

- Amazon EC2ユーザーガイドの[共有 Linux のガイドラインAMI](#)を使用する
- アタックサーフェスを減らすAMIために、最小限のインストールとしてデプロイするようにをアーキテクトします。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は暗号化を使用します end-to-end。例えば、Secure Sockets Layer (SSL) を使用して、お客様と購入者間のHTTPセッションを保護します。サービスが有効な証明書と up-to-date証明書ののみを使用していることを確認します。

- AMI 製品に新しいバージョンを追加するときは、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ようにセキュリティグループを設定します。セキュリティグループが、サービスに必要な機能を提供するために必要な最小ポートセットにのみアクセスできるように設定されていることを確認します。最小セットのポートと必要な送信元 IP アドレス範囲に対してのみ管理アクセスを許可します。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、「」を参照してください[新しいバージョンを追加する](#)。
- 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第三者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ら、に付属しているウェブアプリケーションを迅速に更新しますAMI。この情報を含むリソースの例は、[SecurityFocus](#)および [NIST National Vulnerability Database](#) です。

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 クラウドのセキュリティ](#)
- [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](#)
- [Open Web Application Security Project \(OWASP\): Secure Coding Practices - クイックリファレンスガイド](#)
- [OWASP ウェブアプリケーションセキュリティリスクの上位 10 件](#)
- [SANS \(SysAdmin、監査、ネットワーク、セキュリティ\) 一般的な弱点の列挙 \(CWE\) 最も危険なソフトウェアエラーの上位 25 件](#)
- [セキュリティフォーカス](#)
- [NIST National Vulnerability Database](#)

のパブリッシュ要件AMIのスキャン

を新しい製品またはバージョンとして送信AMIする前に を検証するには、セルフサービススキャンを使用できます。セルフサービススキャナーは、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、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、「[the section called “AMI 用の の準備と保護 AWS Marketplace”](#)」を参照してください

から AWS Marketplace 管理ポータル、アセットメニューから Amazon マシンイメージを選択します。追加AMIを選択してスキャンプロセスを開始します。このページに戻るAMIsと、 のスキャンステータスを確認できます。

Note

AWS Marketplace へのアクセス許可についてはAMI、「」を参照してください[AWS Marketplace へのアクセスの許可 AMI](#)。

でソフトウェアが実行されていることを確認する AWS Marketplace AMI

実行時に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ソフトウェアに確認してもらうこともできます。

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 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EC2。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。

```
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 最初に製品を公開用に送信すると、製品には [製品コード](#) が割り当てられます AWS Marketplace。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。

```
curl http://169.254.169.254/latest/meta-data/product-codes
0vg00000000000000000000000000000000
```

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ました。

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。

AMI の製品料金 AWS Marketplace

AWS Marketplace には、Amazon Machine Image (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の作成](#)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ついて説明します。

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 を使用したAMI製品のカスタム計測料金](#)
- [でのAMI製品の契約料金 AWS Marketplace](#)
- [を使用した AMIベースの製品へのライセンスの関連付け AWS License Manager](#)



AMI 料金モデル

次の表は、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ています。

料金モデル	説明
無料	<p>お客様は、Amazon Elastic Compute Cloud (Amazon EC2) がサポートするインスタンスを、追加のソフトウェア料金なしで実行できます。</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>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。</p> </div>
独自のライセンスの持ち込み (BYOL)	<p>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。</p>
時間単位料金または時間あたりの年間料金	<p>[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。</p> <p>[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、</p>

料金モデル	説明
	<p>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。NOTE: これは Amazon の AWS 無料EC2利用枠とは別のモデルで、お客様は毎月 750 時間の無料利用を受けることができます。</p> <p>[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。</p> <p>1 時間単位の年間使用 – お客様は、1 つのインスタンスタイプの 1 つの Amazon EC2 インスタンスについて 1 年分の使用を前払いで購入できます。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。</p> <p>[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 the section called “分割プランの作成”を参照してください。</p>

料金モデル	説明
	<p>[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。</p> <p>[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年分の使用量を前払いで購入できます。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。</p> <p>[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ターの準備 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ターを使用して、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、オファタ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ターの場合、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タ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ターの準備 および the section called “分割プランの作成” を参照してください。</p>

料金モデル	説明
	<p> Note</p> <p>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。</p>
月額	<p>[月額] - ソフトウェア料金は、顧客が実行するインスタンスの数に関係なく、固定月額料金で請求されます。月額料金はサインアップおよびキャンセル時に按分されます。例：その月の1日分をサブスクライブした顧客には、その月の1/30の料金が請求されます。</p> <p>[時間単価を伴う月額料金] - 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。</p> <p> Note</p> <p>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。</p>
有料使用料金	<p>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。料金はすべて、お客様によって時間料金で請求されます。</p> <p>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。使用料はAWS Marketplace Metering Serviceとも呼ばれます。</p>
契約料金モデル	AMI 契約料金 – 購入者が前払い料金を支払う単一AMI製品または単一AMI AWS CloudFormation スタック。

AWS 料金とソフトウェア料金

Amazon Machine Image (AMI) ベースの製品料金は、次の 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 – 使用されるサービスや AWS インフラストラクチャに応じて、すべての AMI ベースの製品に関連するインフラストラクチャ料金が発生します。これらの料金と料金は、[AWS](#)、[AWS リージョン](#)、[AWS リージョン](#)によって異なる場合があります。詳細については、「[Amazon EC2料金](#)」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、購入者が製品を使用する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、およびアンサブ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](#) に記載されている特定の返金ポリシーに同意するものとします。

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 を使用したAMI製品のカスタム計測料金](#)
- [でのAMI製品の契約料金 AWS Marketplace](#)

AMI ベースの製品の一括請求、計測、ライセンス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。

AMI 製品料金については、以下のトピックを参照してください。

- [を使用したAMI製品のカスタム計測の設定 AWS Marketplace Metering Service](#)
- [を使用したAMIベースの製品へのライセンスの関連付け AWS License Manager](#)

AWS Marketplace Metering Service を使用したAMI製品のカスタム計測料金

AWS Marketplace Metering Service を使用すると、ソフトウェアを変更して、計測レコードをエンドポイントに送信して使用状況をキャプチャできます。使用カテゴリを選択し、その1つのカテゴリに最大 24 個のディメンションを定義できます。これらのディメンションは 1 時間に 1 回計測され、集計され、定義した料金プランに対して課金されます。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 が公開されると、変更できなくなります。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 \(オプション\)](#)
- [カスタム計測の設定](#)
- [を使用したAMI製品のカスタム計測の設定 AWS Marketplace Metering Service](#)

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。その例を以下に示します。
 - 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。ソフトウェアは、適切なディメンションと使用量を AWS Marketplace Metering Service に送信するように設定する必要があります。
- 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 を使用してソフトウェアを料金設定する場合は、まず使用カテゴリと使用方法を決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 インスタンスをモニタリングするソフトウェアを提供しているとします。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、一覧表示 AWS Marketplace する用の新しい製品を作成する必要があります。製品がすでにある場合は 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、新しい購入者が使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。新製品は、「」で説明されているように、AWS Marketplace Metering Service の使用量を計測する必要があります [Metering Service を使用するためのソフトウェアの変更](#)。

を取得したらAMI、標準プロセスに従って、AMIセルフサービスツールを使用して 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。

Marketplace AWS Metering Service の製品ロードフォームのフィールドを完了するには、次の定義を使用します。Product Load Form では、これらのフィールドは Flexible Consumption Pricing (FCP) としてラベル付けされ、時間単位および月単位の料金商品と区別されます。

- タイトル – に既に製品があり AWS Marketplace 、 で同じ製品を追加する場合は AWS Marketplace Metering Service、FCPカテゴリとディメンションを括弧内に含めて区別します (例: PRODUCTTITLE 「(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使用して有料製品に課金されるカテゴリ。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使用して有料製品に課金される測定単位。オプションは、選択したFCPカテゴリに基づいてドロップダウンリストに表示されます。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ときに使用される名前。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。名前を変更するには、新しいが必要でAMI。
- FCP デイメンションの説明 – 製品のデイメンションを説明する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 Rate – この製品の単位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 3 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。
- 複数の AMIs および クラスターと 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 を使用する製品は、時間単位、月単位、Bring Your Own License () などの他の料金モデルに変換することはできませんBYOL。
- 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。
- 最大 24 個のFCPディ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、既存のディメンションを変更することはできませんが、新しいディ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、ディ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ィ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。

複数のディメンションの料金については、複数の値を測定し、ディメンションごとに 1 つずつ Metering Service に送信する必要があります。そのためには、を指定するときに、既知のディメンションセットを使用してソフトウェアをプログラムまたは設定する必要がありますAMI。製品が作成された後は、一連のディ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	<p>現在の GB に格納されたデータ (サンプリングされた)。</p> <p>-または-</p> <p>格納されたデータの最大 GB (その時間を参照)</p> <p>-または-</p> <p>その時間に追加または処理されたデータの合計 (GB)。</p> <p>-または-</p> <p>その時間に処理されたデータの合計 (GB)。</p>

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 (ISVs) が購入者にソフトウェア使用状況をより詳細に把握し、コスト配分を実行するのに役立ちます。

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 () 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 <https://console.aws.amazon.com/billing/>。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。

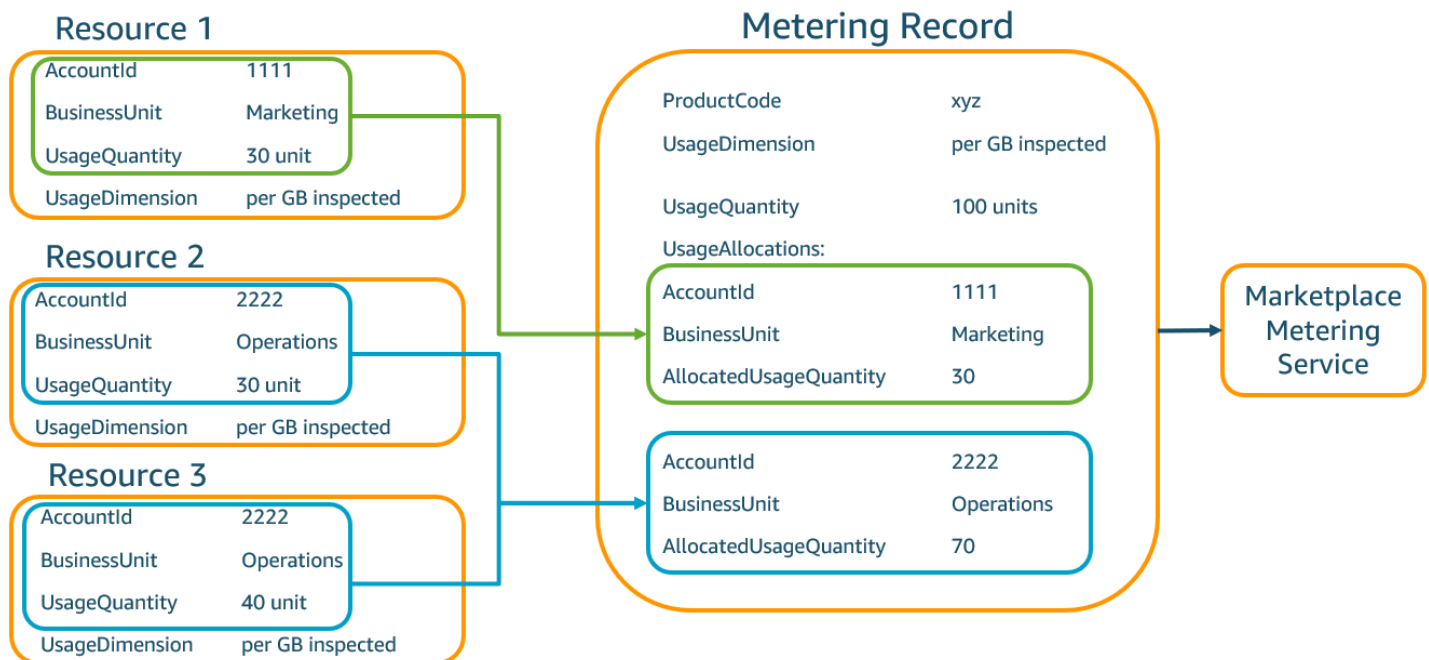
一般的なユースケースでは、購入者が 1 つの AWS アカウントを使って商品を購入します。また、購入者には同じ製品サブスクリプションに関連するユーザーが多数います。Account ID のキーを持つタグを使用して使用量の割り当てを作成し、各ユーザーに使用量を割り当てることができます。この場合、購入者は Billing and Cost Management コンソールで Account ID タグを有効にして、個々のユーザーの使用状況を分析できます。

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。

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	111122223 333	ネットワーク: 検査対象 (GB) あたり	70	2222	オペレーション
xyz	111122223 333	ネットワーク: 検査対象 (GB) あたり	30	3333	財務
xyz	111122223 333	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。

カスタム計測の設定

AWS Marketplace Metering Service の使用の詳細については、「」を参照してください [を使用したAMI製品のカスタム計測の設定 AWS Marketplace Metering Service](#)。

を使用したAMI製品のカスタム計測の設定 AWS Marketplace Metering Service

AWS Marketplace Metering Service は、使用カテゴリ別にソフトウェアを直接請求するために使用できる料金と計測機能です。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。Amazon Machine Image (AMI) ベースの製品、コンテナベースの製品、Software as a Service (SaaS) ベースの製品で Metering Service を使用できます。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しく説明します。

AWS Marketplace Metering Service は、いくつかの新しいシナリオを有効にします。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 「計測サービスAPIリファレンス」](#)を参照してください。

トピック

- [要件](#)
- [AWS Marketplace Metering Service を呼び出す](#)
- [障害処理](#)
- [制約事項](#)
- [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、以下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 () AWS Marketplace を介して から起動する必要がありますAMI。
- に既存の製品がある場合は AWS Marketplace、この機能を有効にするために新しい を送信AMIし、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。エンドユーザーは、ユーザーがソフトウェアでプロビジョニングする 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。を通じてソフトウェアをデプロイする場合、IAMロールの使用はオプションです AWS Marketplace。AWS Marketplace Metering Service ソフトウェアをデプロイするときに必要です。
- 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 を呼び出す

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すると、起動した minute-of-the-hour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

ソフトウェアは、お客様のインスタンスで設定されたIAMロールを使用して AWS Marketplace Metering Service を呼び出し、消費ディメンションと量を指定する必要があります。

ソフトウェアは、次の実装例のように、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。これには、EC2 インスタンス用に設定されたロールが必要です。ロール認証情報は、 によってSDK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、 の収集が含まれる場合があります value-per-dimension。
3. 次のパラメータを使用してSDKクライアントで meterUsageメソッドを呼び出します (使用状況があるディメンションごとにさらに を呼び出します)。
 - timestamp - 記録される時間のタイムスタンプ (内UTC)。
 - 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) の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、ソフトウェアはリージョン内の 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

インスタが提供され、無関係なリージョンの Availability 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。

計測レコードをサービスに送信するときは、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決定し、このリージョン情報を MeteringServiceClient コンストラクターに渡します。SDK コンストラクター AWS リージョンで指定しない場合、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、[「アプリケーションの現在のリージョンの決定」](#) 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

計測の失敗の中には、への接続時に一時的な問題が発生する場合があります AWS Marketplace Metering Service。AWS Marketplace では、短期間の停止やネットワークの問題を避けるため、最大 30 分間の再試行を、指数関数的にバックオフして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せんが、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。が AMI 送信されるときにソフトウェアの障害モードを含める必要があり、後で変更することはできません。

制約事項

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。

- IAM お客様のロールとインターネットゲートウェイの要件 - お客様の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 のアクセス許可](#)」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、AMI製品を製品の公開と保守 AWS Marketplace APIsに必要な と統合するのに役立ちます。

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、適切な使用量割り当てタグを含む計測レコードを AWS Marketplace に送信して、顧客に料金を pay-as-you-go 請求します。

```
# 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
)
```

の詳細についてはMeterUsage、「計測サービスリファレンス[MeterUsage](#)」の「」を参照してください。AWS Marketplace API

レスポンスの例

```
{ "MeteringRecordId": "string" }
```

でのAMI製品の契約料金 AWS Marketplace

Amazon Machine Image (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 AWS CloudFormation スタックAMI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しく説明します。

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。

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、またはユーザー から選択できます。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。

Product Load Form (PLF) を使用してAMI製品の契約を作成する場合は、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、CheckoutlicenseAPI呼び出しEntitlementの 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 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	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 ドル
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 ドル	16,000 ドル
1 時間あたりのモニタリング対象追加ホスト	10 ドル	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 ドル	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

例えば、ReadOnlyUsersとAdminUsersディメンションがある場合、に年額料金を提供する場合はReadOnlyUsers、AdminUsersにも年額料金を提供する必要があります。

自動更新

契約 AWS Marketplace を使用して製品を購入するとAMI、お客様は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AMIベースの製品へのライセンスの関連付け 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、お客様が購入したライセンス (エンタイトルメントとも呼ばれます) 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。以下のセクションでは、を使用してAWS License Manager ライセンスをAMIベースの製品に関連付ける方法について詳しく説明します。

の詳細については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

- お客様は、契約の有効期限AMIが切れた後に の新しいインスタンスを起動することはできません。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、CheckoutLicenseAPI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チェックアウトできます。

Example S3 へのバックアップの年間単位数

Amazon Simple Storage Service (Amazon S3) に、1 年間最大 1,024 単位のデータをバックアップでき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 CheckoutLicenseAPI 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呼び出しは失敗します。

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、CheckInLicenseAPIオペレーションを使用してライセンスユニットをエンタイトルメントプールに返すことができます。

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。お客様は、AWS Identity and Access Management (IAM) アクセス許可を提供することでソフトウェアを起動します。ユーザーがログインすると、アプリケーションはCheckoutLicenseAPIオペレーションを呼び出して数量を1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CheckInLicenseAPI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返します。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。ソフトウェアは、CheckoutLicenseAPI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することで、お客様がサインアップした階層を識別できます。

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

以下のステップは、AMI製品を と統合するためのワークフローを示しています AWS License Manager。

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 に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 て購入します。
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。お客様は 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、購入したエンタイトルメントを検出し、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. Product Load Form (PLF) に必要な料金情報を記入し、処理のために返送してください。
3. CheckoutLicense、および を呼び出すために必要なIAMアクセス許可を持つExtendLicenseConsumption、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CheckInLicense。

必要なIAMアクセス許可の詳細については、次のIAMポリシーを参照してください。

```
{
  "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、RegisterUsageAPIオペレーションをテスト呼び出します。

AMIベースの製品を と統合する AWS License Manager

を使用して、AMIベースの製品を License Manager [AWS License Manager](#) と統合できますAPI。AWS Marketplace AMIベースの製品を使用して Amazon EC2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 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。後で使用できるように、GetLicense レスポンスの KeyFingerprint 属性の値を書き留めておきます。
3. アプリケーションに最新のパブリック AWS SDKをダウンロードして含めます。
4. 購入者がアプリケーションのライセンスを使用する権限があ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。

ライセンスのエンタイトルメントが見つからない場合、またはエンタイトルメントの最大数を越えた場合、CheckoutLicenseAPIオペレーションは を返します NoEntitlementsAllowedException。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. (フローティングエンタイトルメントでのみ必要) CheckoutLicenseレスポンスでLicenseConsumptionToken受信した 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。

リリースプロセスに従ってアプリケーションを作成しAMI、製品発行プロセス AWS Marketplace に従って に製品を送信します。

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、[AWSLicense Manager の「Seller issued licenses」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。各リクエストには、ランダムな汎用一意識別子 (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、CheckoutLicenseAPIオペレーションを使用して という名前の設定可能なエンタイトルメントをチェックアウトするリクエストの例ですPowerUsers。

```
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、CheckoutLicenseAPIオペレーションを使用して という名前の機能エンタイトルメントをチェックアウトするリクエストの例ですEnterpriseEdition。

```
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ベースの製品のテストライセンスを作成するには

1. AWS CLI v2 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およびテスト 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ものでなければなりません。
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。使用権限でドロウダウン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の呼び出し

お客様の 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呼び出しを使用できます。

- 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、新しいエンタイトルメント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呼び出しを使用して新しいエンタイトルメントを読み取ります。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、キャンセルなどにより、製品が使用中は、定期的に `CheckoutLicense` API コールを実行することをお勧めします。定期的に `CheckoutLicenseAPI` オペレーションを使用することで、製品はアップグレードや有効期限などのエンタイトルメントの変更を検出できます。

15分ごとに `CheckoutLicenseAPI` 呼び出しを実行することをお勧めします。

でAMI製品の Amazon SNS通知を受信する 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 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受け入れるタイミングを確認できます。[Amazon SNSトピック: aws-mp-subscription-notification](#) は、AMI製品で使用できる Amazon SNSトピックです。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タイミングを通知します。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) (など)

が必要です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。Marketplace [AWSオペレーションチーム](#)に連絡して、 をリクエストしますARN。

トピック

- [Amazon SNSトピック: aws-mp-subscription-notification](#)
- [Amazon SQSキューを Amazon SNSトピックに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": "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

offer-identifier は、オファーがプライベートオファーの場合にのみ通知に表示されます。

Amazon SQSキューを Amazon SNSトピックにサブスクライブする

提供されたSNSトピックに Amazon SQSキューをサブスクライブすることをお勧めします。SQS キューの作成とトピックへのキューのサブスクライブに関する詳細な手順については、「[Amazon Simple Notification Service デベロッパーガイド](#)」の「[Amazon SNSトピックへの Amazon SQS キューのサブスクライブ](#)」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 のトピックのみをサブスクライブ AWS Marketplace SNS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SNSへの Amazon メッセージの送信」](#)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 に保存されますSQS。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。

AMI の製品チェックリスト AWS Marketplace

Amazon Machine Image (AMI) 製品リクエストを に送信する前に AWS Marketplace、このチェックリストを確認してください。この情報を検証すると、送信から公開プロセスにスムーズに進めることができます。

製品の使用:

- は本番稼働準備が整っているAMI必要があります。
- AMI では、時間やその他の測定値によって製品の使用を制限することはできません。
- は、1-Clickフルフィルメントエクスペリエンスと互換性AMIがある必要があります。
- 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または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、適切な開示を含む [the section called “製品使用ポリシー”](#) に従う必要があります。
- 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ます。あるいは、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。

- 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 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。
- Hardware Virtual Machine (HVM) 仮想化と 64 ビットアーキテクチャを使用します。
- 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。
- 購入者は、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を持っていますAMI。
- AMI セルフサービススキャンAMIを使用して を実行します。

[Windows AMIs:

- Windows Server 2012 以降のオペレーティングシステムを使用する場合は、最新バージョンの [EC2Launchv2](#) を使用します。
- AMI が Windows Server 2012 および 2012 R2 上に EC2Configで構築されている場合は、の最新バージョンを使用します[EC2Config](#)。
- EC2Launch v2 を使用している場合は、必ず以下を完了してください。
 - [Amazon EC2Launch設定](#) で、管理者アカウント の設定で Random を選択します。
 - [Amazon EC2Launch設定](#) で、SSMサービスの開始 のチェックボックスをオンにします。
 - を [EC2 v2 タスク設定](#) UserDataに追加します。
- を使用している場合はEC2Config、AMI内で Ec2SetPassword、Ec2WindowsActivate、および Ec2HandleUserData [の設定ファイル](#)を有効にします。
- 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が存在しないことを確認します。

Linux の場合AMIs :

- ルートログインはロック/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。

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。

- エラーのないすべてのURLsロード。
- 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)。

AMIのベースの製品要件 AWS Marketplace

AWS Marketplace では、すべての Amazon Machine Image (AMI) 製品およびサービスに対して以下のポリシーを維持しています AWS Marketplace。このポリシーは、お客様に安全で安心して信頼できるプラットフォームを提供するためのものです。

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を満たすようにレビューおよび調整されます。AWS Marketplace は、継続的に製品をスキャンして、セキュリティガイドラインの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違反になった場合は、AWS Marketplace から連絡があり、新しい基準を満たすようにAMI製品を更新します。同様に、新しく検出された脆弱性が に影響することが判明した場合はAMI、関連する更新AMIプログラムを含む更新プログラムを提供するよう求められます。を送信する前に、[セルフサービスのAMIスキャンツール](#)を使用する必要がありますAMI。このツールは、 がポリシーAMIを満たし AWS Marketplace ていることを確認するのに役立ちます。

トピック

- [セキュリティポリシー](#)
- [アクセスポリシー](#)
- [カスタマー情報ポリシー](#)
- [製品使用ポリシー](#)
- [アーキテクチャポリシー](#)
- [AMI 製品の使用方法](#)

セキュリティポリシー

すべての AMIsは、次のセキュリティポリシーに従う必要があります。

- AMIs は、[セルフサービスのAMIスキャンツール](#)または AWS セキュリティによって検出された既知の脆弱性、マルウェア、またはウイルスを含んではなりません。
- AMIs 現在サポートされているオペレーティングシステムやその他のソフトウェアパッケージを使用する必要があります。(EoL) オペレーティングシステムまたはその他のソフトウェアパッケージ AMIを持つ End-of-Life のバージョンは、 からリスト解除されます AWS Marketplace。更新されたパッケージAMIで新しい を構築し、新しいバージョンとして に発行できます AWS Marketplace。
- すべてのインスタンス認証では、起動時にパスワードが生成、リセット、または定義されていても、パスワードベースの認証は使用せず、キーペアのアクセスを使用する必要があります。AMIs には、何らかの理由でパスワード、認証キー、キーペア、セキュリティキー、またはその他の認証情報を含めることはできません。
- AMIs は、 リソースにアクセス AWS するために、ユーザーからアクセスキーまたはシークレットキーをリクエストまたは使用してはなりません。AMI アプリケーションがユーザーへのアクセスを必要とする場合は、 を通じてインスタンス化された AWS Identity and Access Management (IAM) ロールを使用してアプリケーションを達成する必要があります。これにより AWS CloudFormation、インスタンスが作成され、適切なロールが関連付けられます。AWS CloudFormation 配信方法を使用して製品のシングルAMI起動を有効にする場合、対応する使用手順には、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が含まれている必要があります。詳細については、「[the section called “の使用 CloudFormation”](#)」を参照してください。
- Linux ベースAMIsでは、SSHパスワード認証を許可しないでください。PasswordAuthentication を NO に設定して、sshd_config ファイルによるパスワード認証を無効にします。

アクセスポリシー

アクセスポリシーには、一般的なポリシー、Linux 固有ポリシー、および Windows 固有ポリシーの 3 つのカテゴリがあります。

一般的なアクセスポリシー

すべての AMIsは、次の一般的なアクセスポリシーに従う必要があります。

- AMIs は、オペレーティングシステム (OS) レベルの管理機能を許可して、コンプライアンス要件、脆弱性の更新、ログファイルへのアクセスを許可する必要があります。Linux ベースの AMIs と SSHWindows ベースの AMIsを使用しますRDP。
- AMIs には、承認されたパスワードや承認されたキーを含めることはできません。

- AMIs は、管理アクセスに固定パスワードを使用してはなりません。AMIs は代わりにランダム化されたパスワードを使用する必要があります。代替の実装としてインスタンスメタデータを取得し、パスワードとして `instance_id` を使用します。管理者は、自身の資格情報を設定または変更することを許可される前に、このランダム化されたパスワードの入力を求められる必要があります。インスタンスメタデータの取得については、Amazon EC2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。
- お客様の実行中のインスタンスにアクセスすることはできません。お客様は外部アクセスを明示的に有効にする必要があります、に組み込まれているアクセシビリティはデフォルトでオフAMIにする必要があります。

Linux 固有の (または Unix のような) アクセスポリシー

Linux ベースまたは Unix のような AMIs は、次のアクセスポリシーと一般的なアクセスポリシーに従う必要があります。

- AMIs は [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。
- AMIs はルートのリモートログインを無効にする必要があります。
- AMIs は、ユーザーにルート関数を実行する管理者コントロールの取得を許可する必要があります。例えば、Linux ベースの OS `sudo`へのアクセスを許可します。他のシステムでは、完全な特権レベルのアクセスを許可します。
- AMIs は、監査証跡のルートアクティビティを記録する必要があります。
- AMIs には、OS ユーザー用に承認されたパスワードを含めることはできません。
- AMIs には、承認されたキーを含めることはできません。
- AMIs に空白または null のルートパスワードを含めることはできません。

Windows 固有のアクセスポリシー

Windows ベースの AMIs は、次のアクセスポリシーと一般的なアクセスポリシーに従う必要があります。

- Windows Server 2016 以降をお使いの場合は、「EC2Launch」を参照してください。
- Windows Server 2012 R2 以前では、最新バージョンの `Ec2ConfigService` を使用し、`Ec2SetPassword`、`Ec2WindowsActivate`、および `Ec2HandleUserData` を有効にします。
- ゲストアカウントおよびリモートデスクトップユーザーを削除 (許可されていない) します。

カスタマー情報ポリシー

すべての AMIs は、次の顧客情報ポリシーに従う必要があります。

- ソフトウェアは、BYOL (お客様自身のライセンスを付与する) 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。これには、収集する内容、顧客データの保存場所、および使用方法が含まれます。例えば、この製品はお客様の名前と E メールアドレスを収集します。この情報は、<会社名> に送信され、保存されます。この情報は、<製品名> に関して購入者に連絡するためにのみ使用されます。
- 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 AMIs は、次の製品使用ポリシーに従う必要があります。

- 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。商用ソフトウェアの開発者、コミュニティ、BYOL エディションは、同等の有料バージョンがでも利用できることを条件としてサポートされています AWS Marketplace。
- すべては、ウェブサイトからの起動エクスペリエンスまたは を介した AMI ベースの配信のいずれかと互換性がある AMIs 必要があります AWS CloudFormation。ウェブサイトから起動する場合、AMI はインスタンスの作成時に顧客またはユーザーデータを正しく機能させることはできません。
- AMIs とそのソフトウェアは、セルフサービス方式でデプロイでき、追加の支払い方法やコストを必要としない必要があります。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。
- 外部依存関係が利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。
- AMIs 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 - 例えば、外部APIs または販売者や第三者によって AWS のサービス 管理される - 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ISV別の の製品へのアドオンである場合、製品の説明には、他の製品の機能を拡張し、それなしでは製品のユーティリティが非常に限られていることを示す必要があります。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。

アーキテクチャポリシー

すべての AMIs は、次のアーキテクチャポリシーに従う必要があります。

- AMIs のソースは、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。
- AMIs はHVM仮想化を使用する必要があります。
- AMIs は 64 ビットまたは 64 ビットARMアーキテクチャを使用する必要があります。

- AMIs は Amazon Elastic Block Store (Amazon EBS) によって AMIs バックアップされている必要があります。Amazon Simple Storage Service (Amazon S3) AMIs ではサポートしていません。
- AMIs は、暗号化された EBS スナップショットを使用してはなりません。
- AMIs は、暗号化されたファイルシステムを使用してはなりません。
- AMIs は、すべてで実行でき 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。AMIs 異なるリージョンに対して異なる を構築することはできません。

AMI 製品の使用方法

AMI 製品の使用方法を作成するときは、「」にあるステップとガイダンスに従ってください [the section called “AMI およびコンテナ製品の使用方法”](#)。

で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、のコンテナベースの製品について説明します AWS Marketplace。

AWS Marketplace 購入者は、利用可能な製品の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。コンテナイメージ配信オプションの場合、購入者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へのリンクを表示しますURLs。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 チャート配信オプションの場合、購入者には Helm を使用して起動する手順が表示されます step-by-step。

購入エクスペリエンスのチュートリアルについては、このビデオ「[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 Anywhere (EKS Anywhere) を使用してEKS、Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS) 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) のセルフマネージド Kubernetes クラスターにデプロイすることもできますEC2。

Docker 互換のランタイムでは、無料および独自のライセンスモデル (BYOL) 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 販売者として、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](#) asdf
- [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れを表示できます。準備ができたなら、AWS Marketplace カタログに発行して、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マーケットプレイス販売者オペレーションチーム](#)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なら、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 販売者オペレーションチームが 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、「」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、で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 に少なくとも 1 つのコンテナを作成します AWS Fargate。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. [ステップ 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 AWS Marketplace 管理ポータル、製品 ID を生成するには、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、オファーの可用性など EULA、追加のオファー情報を提供します。
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、「」を参照してください [コンテナ製品の新しいバージョンを追加する AWS Marketplace](#)。

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

使用料金のコンテナベースの製品の場合、 を使用して製品を使用するためのエンタイトルメントと、請求のための計測使用量 [AWS Marketplace Metering Service](#) の両方をチェックします。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れを表示できます。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、製品許可リストを更新して を追加することもできます AWS アカウント。このトピックでは、 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 AWS Marketplace。

製品の可視性とライフサイクルの詳細については、「 」を参照してください [製品のライフサイクル](#)。

トピック

- [製品の可視性を更新する](#)
- [の許可リストの更新 AWS アカウント IDs](#)

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動するには、AWS Marketplace 販売者オペレーション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。

の許可リストの更新 AWS アカウント IDs

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。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せん。

1. <https://aws.amazon.com/marketplace/管理/ツアー/> 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. Allowlisted AWS アカウント フィールドに を入力し、カンマで AWS アカウント IDs 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、以下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。

- [でのコンテナ製品の製品情報の更新 AWS Marketplace](#)
- [でコンテナ製品の新しいバージョンを追加する AWS Marketplace](#)

- [でのコンテナ製品の料金の管理 AWS Marketplace](#)
- [での国ごとのコンテナ製品の可用性の更新 AWS Marketplace](#)
- [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](#)
- [でのコンテナ製品のテストとリリース AWS Marketplace](#)

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。これを行うには、イメージに対して静的スキャンを実行します layer-by-layer。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。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。Clair, Twistlock, Aqua Security、または Trend Micro 取り込みと公開プロセスの遅延を回避します。

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、に送信する前に、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。


でのコンテナ製品の製品情報の更新 AWS Marketplace

製品 ID を作成して料金を設定したら、でお客様がコンテナ製品について表示することなど、製品情報を編集できます 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
- 製品ビデオ URL
- リソース
- [サポート情報]

 Note

イメージは、パブリックにアクセスできる Amazon S3 バケットにあるURLs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。

でコンテナ製品の新しいバージョンを追加する 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、バージョンを管理し、バージョン情報を更新できます。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を示します。

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。これらのステップの詳細については、「」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](#)。

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、に必要なリポジトリを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は、追加するリポジトリごとに [新しいリポジトリの追加] を選択し、一意の名前を付けます。

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して、プッシュに使用できるコマンドを含む指示のリストを開きます。Docker コンテナイメージと Helm グラフをそのリポジトリに表示します。


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 pull` を呼び出すときに、次の Amazon Elastic Container Registry (Amazon ECR) API オペレーションを使用できません `docker push`。


- `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。
- の場合 Helm チャート配信オプション、アップロード Helm AWS Marketplace コンソールで作成された Amazon ECRリポジトリにイメージをグラフ化してコンテナ化します。
- Amazon EKSコンソールのアドオン配信オプションの場合は、Helm AWS Marketplace コンソールで作成された Amazon ECRリポジトリにイメージをグラフ化してコンテナ化します。

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とき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の追加](#)」表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、異なるコンテナイメージ (例 : Kubernetes また、Ubuntu デプロイ)。お客様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し、製品のバージョンごとに最大 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では、コンソールで作成された Amazon Elastic Container Registry (Amazon ECR) リポジトリ内の AWS Marketplace 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。
 - の場合 Helm チャート配信オプション、へのパスを提供する Helm AWS Marketplace コンソールで作成された Amazon ECR リポジトリ内のグラフ。購入者は をインストールします。Helm ソフトウェアを起動するためのデプロイ環境のグラフ。
 - Amazon EKS コンソールのアドオン配信オプションでは、へのパスを指定します。Helm コンソールで作成された Amazon ECR リポジトリ内の AWS Marketplace グラフ。購入者は、Amazon EKS コンソールまたはネイティブ Amazon EKS アドオンを使用してコンテナをインストールAPIsし、ソフトウェアを起動します。詳細については、[「Amazon から利用可能な Amazon EKS アドオン EKS」](#) 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、製品バージョンソフトウェアを含むコンテナイメージECRURLに Amazon 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。各テンプレートのリソースURLにタイトルと を入力します。
- b. を追加するには Helm チャート配信オプションで、次の手順を実行します。
- i. 内Helm チャート、Amazon を ECR URL に追加します。Helm ソフトウェアを起動するために、購入者がデプロイ環境にインストールするチャート。
 - ii. コンテナイメージで、製品バージョンソフトウェアを含むコンテナイメージECRURLに Amazon 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 - In Helm リリース名、 の名前を入力します。Kubernetes の名前空間。Helm チャートがインストールされます。
 - vii. オプション - In Helm インストール名前空間、 の名前を入力します。Helm `helm install` コマンドで使用されるリリース。
 - viii. オプション - In Kubernetes サービスアカウント名、 の名前を入力します。Kubernetes AWS Identity and Access Management () への接続に使用されるサービスアカウントIAM。 - Kubernetes サービスアカウントは、ライセンスや計測などのサービスを呼び出し AWS ます。
 - ix. この製品バージョンQuickLaunchで を有効にすることを選択します。QuickLaunch はの機能です AWS Marketplace。購入者は QuickLaunch 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、[QuickLaunch 「」 の AWS Marketplace 「」](#) を参照してください。

- x. オーバーライドパラメータで、で使用されるパラメータを入力します。Helm CLI ソフトウェアを起動するコマンド。これらのパラメータにより、購入者は指定されたデフォルト値を上書きできます。を有効にしている場合は QuickLaunch、フォームの CloudFormation 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のパラメータの制限は 15 個ですが、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 製品には、DefaultValue のを持つライセンスシークレットのオーバーライドパラメータが必要です "\${AWSMP_LICENSE_SECRET}"。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、で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 APIs。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準拠していることを確認してから [Amazon EKS アドオン製品の要件](#)、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。使用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. 内Helm チャート、Amazon を ECR URL に追加します。Helm ソフトウェアを起動するために、購入者がデプロイ環境にインストールするチャート。
- ii. コンテナイメージで、製品バージョンソフトウェアを含むコンテナイメージ ECR URL に Amazon を追加します。内のすべてのイメージが Helm チャートが一覧表示されます。

- 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 のデフォルト値を維持します。
 - 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、このアドオンをインストールまたはアップグレードするときに表示されるアドオンのバージョンを入力します。の形式に従いますmajor.minor.patch。
 - 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. 内Kubernetes バージョン で、すべての を選択します。Kubernetes アドオンがサポートするバージョン。
 - 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 ですARM64。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をお勧めします。アドオンがARM64デバイスをサポートしていない場合は、すべての商用 で製品を公開する前に、サポートを追加する予定日を指定する必要があります AWS リージョン。
 - x. 名前空間 に一意の を入力します。Kubernetes アドオンがインストールされる名前空間。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 であるこれらの環境変数にマッピングできます\${AWS_EKS_CLUSTER_NAME}。
7. 追加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、前のステップの手順を繰り返して設定します。
 8. [送信] を選択します。

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 グラフ (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]
 - AddOn 名前
 - AddOn バージョン
 - AddOn タイプ
 - Helm グラフ URI
 - CompatibleKubernetesVersions
 - SupportedArchitectures
 - 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、[を更新できません Supported Architectures](#)。
7. [送信] を選択します。
 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 の下部にあるお問い合わせフォームを使用してオペレーションチームにお問い合わせください [AWS Marketplace 管理ポータル](#)。

でのコンテナ製品の料金の管理 AWS Marketplace

AWS Marketplace 販売者は、コンテナ製品の料金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 BYOL)、Amazon Elastic ECS Kubernetes Service (Amazon)、および [の無料製品](#)、Bring Your Own License Model (EKS) 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。料金ディメンションを追加または更新できます。料金モデルに応じて、契約と使用状況のディメンションを追加できます。料金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。

のコンテナ製品料金モデルの詳細については AWS Marketplace、「[」](#)を参照してください [のコンテナ製品の料金 AWS Marketplace](#)。

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

消費料金 (例: 追加の使用量の料金) を含む契約の pay-as-you-go料金ディメンションの追加は、では使用できません AWS Marketplace 管理ポータル。
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. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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 [\[料金設定ディメンションを更新\]](#) 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. 料金モデルに応じて、API識別子、表示名、および説明に関する情報を提供することで、契約ディメンションまたは使用状況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/管理/ツアー/> 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 [\[料金設定ディメンションを更新\]](#) 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/管理/ツアー/> 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. リクエスト変更リストから、パブリックオファ어의更新 を選択し、料金条件の更新 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。値上げは、変更を開始した日から 90 日間、価格をロックします。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。

製品の返金ポリシーの更新

製品の返金ポリシーを更新できます。返金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、「[での製品の返金 AWS Marketplace](#)」を参照してください。

1. <https://aws.amazon.com/marketplace/管理/ツアー/> 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、[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。

での国ごとのコンテナ製品の可用性の更新 AWS Marketplace

AWS Marketplace 販売者は、コンテナ製品を販売およびサブスクライブできる国を変更できます。詳細については、「[Countries \(国\)](#)」を参照してください。次の手順では、コンテナ製品の可用性を国ごとに更新する方法を示します。

1. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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、[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 販売者として、[の標準契約 AWS Marketplace](#)またはカスタム 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。に対する更新は、製品と製品更新をサブスクライブする新規ユーザーEULAに適用されます。を更新するにはEULA、次の手順を使用します。

1. <https://aws.amazon.com/marketplace/管理/ツアー/> 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. リクエストの変更ドロップダウンリストから、公開オファ어의更新 を選択し、更新 EULAを選択します。
4. マーケットAWSプレイスの標準契約を選択するか、カスタム を送信しますEULA。カスタム の場合EULA、Amazon S3 バケットから契約を指定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。

でのコンテナ製品のテストとリリース AWS Marketplace

コンテナ製品の新しいバージョンを に追加したら AWS Marketplace、製品をテストして一般公開できます。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ステップ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、コンテナイメージのリリースと Helm グラフ。

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「[新しいバージョンを追加する](#)」の「Asynchronous Errors」表を参照してください。

Note

製品に現在制限付き可用性が設定されている場合、製品が利用可能である購入者のみが製品バージョンにアクセスできます。製品が現在パブリック可用性に設定されている場合、すべての AWS Marketplace 購入者は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、はリクエスト AWS Marketplace を処理し、Amazon アドオンカタログで検証するための制限付き状態でEKSアドオンを発行します。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 AWS Marketplace 管理ポータル。取り込み時間は、処理するリクエストの量に応じて 5~10 営業日です。

リクエストがレビュー中ステータスの場合、アドオンは から Amazon EKSアドオンカタログ AWS Marketplace に AWS チームによって発行されています。アドオンが制限付き状態に公開されると、リクエストステータスが Success に変更されます。その後、アドオンのテストを開始できます。

- アドオンが利用可能になったら、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 は、ソフトウェアの機能を検証するための専門知識に依存しています。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、[以下の詳細な手順](#)に従います。必ず各 でテストしてください。Kubernetes ソフトウェアがサポートするバージョン。
- 有料製品を提供する場合は、次の内部 にプライベートオファーを作成します AWS アカウント。これらのアカウントは、すべての商用 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 AWS リージョン。

288092140294, 288092140294, 408202761791

- が AWS Marketplace を承認し、アドオンバージョンをパブリックに移行するまで、テストクラスターをアドオンでアクティブに保ちます。

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。

アドオンをパブリックにリリースする

Amazon EKS クラスター経由で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 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 AWS Marketplace Catalog API。

詳細については、AWS Marketplace Catalog API リファレンスの「[Amazon EKS アドオンの可視性を更新する](#)」を参照してください。

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 AWS Marketplace 管理ポータル。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 では、のすべてのコンテナベースの製品とサービスに対して以下の要件が維持されます 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 要件を満たしているか、超えていることを確認します。これらのポリシーは、進化するセキュリティやその他の使用要件を満たすようにレビューおよび調整されます。は、既存の製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 AWS Marketplace を継続的に検証します。製品がコンプライアンス違反になった場合は、AWS Marketplace から製品を更新するための連絡があります。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利用できないことがあります。

トピック

- [セキュリティ要件](#)
- [アクセスの要件](#)
- [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](#)

セキュリティ要件

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。

- Docker コンテナイメージには、既知のマルウェア、ウイルス、または脆弱性が含まれていない必要があります。コンテナ製品に[新しいバージョンを追加する](#)と、その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ます。
- コンテナベースの製品で AWS リソースを管理するためのアクセスが必要な場合は、ユーザーにアクセスキーをリクエストする代わりに、[IAMサービスアカウントのロール](#) (Amazon Elastic Kubernetes Service (Amazon EKS) 経由で実行されている場合) または[IAMタスクのロール](#) (Amazon Elastic Container Service (Amazon ECS) 経由で実行されている場合) を使用してアクセスする必要があります。
- 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、[ECS「セキュリティとEKS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。

アクセスの要件

すべてのコンテナベースの製品は、以下のアクセス要件を満たしている必要があります。

- コンテナベースの製品では、ランダム化された初期パスワードを使用する必要があります。コンテナベースの製品では、外部管理アクセス (ウェブインターフェイス経由でアプリケーションにログインする場合など) に初期固定パスワードや空白パスワードを使用してはなりません。購入者は、自身の認証情報を設定または変更することを許可される前に、このランダム化されたパスワードの入力を求められる必要があります。
- アプリケーションへの外部からのアクセスは、お客様による明示的な同意と有効化が必要です。

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、BYOL (お客様自身のライセンスを付与する) 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。これには、収集する内容、顧客データの保存場所、および使用方法が含まれます。例えば、この製品はお客様の名前と E メールアドレスを収集します。この情報は、<会社名> に送信され、保存されます。この情報は、<製品名> に関して購入者に連絡するためにのみ使用されます。
- 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デベロッパー、コミュニティ、BYOLエディションは、販売者が無料エディションの提供 AWS Marketplace から 90 日以内に同等の有料バージョンを に提供した場合にサポートされます。
- 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。
- 外部依存関係が利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。
- 外部や販売者APIsや第三者によって AWS のサービス 管理されるなど、購入者の直接管理下にならない外部リソース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ISV別の の製品へのアドオンである場合、製品の説明には、他の製品の機能を拡張し、それなしでは製品のユーティリティが非常に限られていることを示す必要があります。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 にデプロイできる必要があります [AWS Fargate](#)。
- 契約料金と の統合を伴う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 およびコンテナ製品の使用方法”](#) 」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、 に運用機能を提供するソフトウェアです。Kubernetes アプリケーションに固有ではありません。例えば、Amazon EKS アドオンにはオブザーバビリティエージェントまたは Kubernetes 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ドライバー。

コンテナ製品の販売者として、Amazon 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 EKS。製品のバージョンを AWS Marketplace アドオンとして Amazon EKSアドオンカタログに発行できます。アドオンは、AWS および他のベンダーによって管理されるアドオンの横に 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 ユーザーガイド](#) 」の「 [Amazon EKS アドオン](#) 」を参照してください。
EKS

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、公開する必要があります AWS Marketplace。
- コンテナ製品は、AMD64 および ARM64アーキテクチャの両方と互換性があるように構築する必要があります。
- コンテナ製品は、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](#)イメージのプッシュや Helm は AWS Marketplace、マネージド Amazon ECR リポジトリにグラフ化します。この要件には、などのオープンソースイメージが含まれます nginx。イメージとグラフは、[Amazon ECR Public Gallery](#)、Docker Hub および Quay。
 - Helm グラフ - を通じてデプロイされるソフトウェアを準備する Helm グラフ。Amazon EKS アドオンフレームワークは、Helm マニフェストにグラフ化します。ある程度 Helm 機能は Amazon EKS システム内ではサポートされていません。次のリストは、オンボーディング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 コマンドは Helm バージョン 3.8.1:
 - は例外ですが、すべての Capabilities オブジェクトがサポートされています。APIVersions。APIVersions は non-built-in カスタムではサポートされていません。Kubernetes APIs。
 - 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 フックと lookup 関数はサポートされていません。
 - 依存関係のあるグラフはすべて、メイン内に配置する必要があります。Helm グラフ (リポジトリパスファイル `://...` で指定)。
 - - Helm チャートは正常に合格する必要があります Helm Lint と Helm エラーのないテンプレート。コマンドは次のとおりです。
 - Helm リント - `helm lint helm-chart`
- よくある問題には、親グラフのメタデータに宣言されていないグラフが含まれます。例えば、`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 フラグで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、Helm 前述の テンプレートコマンド。マニフェストで、busybox やイメージなどの外部gcrイメージリファレンスを検索します。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、依存関係とともにすべてのコンテナイメージ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する方法の詳細については、ソフトウェア に名前を付けaws_mp_configuration_schema.json、 を使用してラッパーにパッケージ化します。Helm グラフ、 [「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](#)、 は有効なapplication/schema+jsonリソースURIを指す \$schema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受け入れてはなりません。

シークレットと証明書のインストールを処理するには、エンドユーザーにインストール後または pre-Add-onインストール手順を提供できます。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。

の制限の詳細についてはaws_mp_configuration_schema.json、 「」を参照してください [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](#)。

- 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します – 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。
- **serviceAccount** 該当する場合、 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 に接続 AWS Marketplace する必要がある場合は AWS のサービス、Helm グラフserviceAccountはデフォルトで を作成します。serviceAccount 作成がvalues.yamlファイルのパラメータによって処理される場合は、パラメータ値を に設定しますtrue。例えば、serviceAccount.create = true と指定します。これは必須です。お客様は、必要なアクセス許可を既に持っている基盤となるノードインスタンスからアクセス許可を継承することで、アドオンのインストールを選択する可能性があるためです。Helm チャートが を作成しない場合serviceAccount、アクセス許可を に関連付けることはできませんserviceAccount。
- 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、アドオンはデプロイエラーに直面します。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、アドオンが一連のカスタムリソースや追跡不可能な Kubernetes ジョブをデプロイする場合は、タミード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 のEKSお客様がARM64現在 AWS Graviton インスタンスの使用に使用されています。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する必要があります。
- APIs のライセンスまたは計測と統合 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、「」を参照してください。[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](#)。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、「」を参照してください。[使用したコンテナ製品の契約料金 AWS License Manager](#)。
- 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ず、例えば、GitHub。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 – お客様が [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をお勧めします。
- IAM ロール – ソフトウェアが他の と機能または接続するために必要なすべての AWS Identity and Access Management (IAM) ポリシーを一覧表示します AWS のサービス。
- バージョンの更新 – Amazon は、アップストリームEKS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までに 45 日間かかります。現在のバージョンのアドオンが新しい Kubernetes バージョンをサポートしている場合は、バージョン互換性マトリックスを更新できるように、そのバージョンを検証して認証します。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 を強化する運用ソフトウェアである必要がありますEKS。Gitops | モニタリング | ログ記録 | 証明書管理 | ポリシー管理 | コスト管理 | 自動スケーリング | ストレージ | 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[コンテナネットワークインターフェイス \(CNI\)](#) にすることはできません。

- ソフトウェアは、APIs有料製品のライセンス AWS Marketplace と計測を通じて販売され、統合されている必要があります。BYOL 製品は受け入れられません。

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 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EKS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、に送信された Helm Chart を含む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呼び出しを拒否します。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、 などの一般的な Kubernetes プロパティの設定nodeSelector。
- ライセンスキー、機能有効化など、アドオン固有の設定URLs。

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 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EKSことをお勧めします。

- [スキーマの要件](#)
- [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](#)

スキーマの要件

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てください。

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なく、これが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"
      "type": "object"
    },
    "podLabels": {
      "description": "Pod Labels"
      "type": "string"
    },
    "resources": {
      "type": "object"
      "description": "Resources"
    },
    "logLevel": {
      "description": "Logging Level"
      "type": "string",
      "enum": [
        "info",
        "debug"
      ]
    },
    "config": {
      "description": "Custom Configuration"
      "type": "object"
    }
  }
}
```

camelCase

設定パラメータは `camelCase` である必要があります。この形式に従わない場合 `camelCase`、拒否されます。

説明が必要です

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、各設定パラメータのラベル名を Amazon EKS コンソールで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。新しいバージョンのアドオンまたは に対処する修正のRBACアクセス許可を変更する必要がある場合CVE、アドオンプロバイダーはこの変更について 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"]
```

シークレット管理

このセクションは、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 APIs 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、お客様は設定を使用して、アドオンに必要なキーを保持する Kubernetes Secret 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、お客様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。

要約すると、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、Distro for OpenTelemetry add-on AWS のスキーマのもので、

```
"examples": [
  {
    "admissionWebhooks": {
      "namespaceSelector": {},
      "objectSelector": {}
    },
    "affinity": {},
    "collector": {
      "amp": {
```



```
    "enabled": true,
    "remoteWriteEndpoint": "https://aps-workspaces.us-west-2.amazonaws.com/
workspaces/ws-xxxxxxxx-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	にはデフォルトが必要ですか？
additionalLabels	アドオンによって管理されるすべての Kubernetes オブジェクトに Kubernetes ラベルを追加します。	不可
additionalAnnotations	アドオンによって管理されるすべての Kubernetes オブジェクトに Kubernetes 注釈を追加します。	不可
podLabels	アドオンによって管理されるポッドに Kubernetes ラベルを追加します。	不可
podAnnotations	アドオンによって管理されるポッドに Kubernetes 注釈を追加します。	不可
logLevel	アドオンによって管理されるコンポーネントのログレベル。	可能

パラメータ	説明	にはデフォルトが必要ですか？
nodeSelector	ノード選択制約の最も簡単な推奨形式。Pod 仕様に nodeSelector フィールドを追加し、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	許容範囲はポッドに適用されます。許容値により、スケジューラはテイントが一致するポッドをスケジューリングできます。許容範囲では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、デーモンセットでより一般的
アフィニティ	アフィニティ機能は、nodeSelector フィールドのようなノードアフィニティ関数の 2 つのタイプのアフィニティで構成されます。これはより表現性が高く、ソフトルールを指定できます。ポッド間アフィニティ/アンチアフィニティでは、他のポッドのラベルに対してポッドを制約できます。	おそらく
topologySpreadConstraints	トポロジースプレッド制約を使用して、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の障害ドメイン間で Pod をクラスター全体に分散す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	にはデフォルト が必要ですか？
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を強くお勧めします。制限はオプションです。	可能
レプリカ	アドオンによって管理されるポッドのレプリカの数。デーモンセットには適用されません。	可能

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 EBSCSI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。お客様は、コンポーネントごとに異なるノードセレクタ/許容範囲を必要とします。

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性を置き換えるものではありません。デフォルトプロパティを使用する場合は、Helm Chart に変更を加えるたびに、のデフォルトがスキーマ内のデフォルトと一致し `values.yaml`、2つのアーティファクト (`values.schema.json` と `values.yaml`) が同期していることを確認してください。

```
"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、などのクラスターメタデータパラメータはvpcIdaccountId、さまざまなアドオン (Elastic Load Balancing Controller など) で必要となる場合があります。Amazon EKS サービスで知られているこのようなパラメータは、Amazon EKS アドオンによって自動的に挿入され、ユーザーが設定オプションとしてを指定する責任は負いません。これらの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
- VPC クラスターの 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され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
- IPv4 クラスターで有効
- IPv6 クラスターで有効
- クラスターでIPv6有効になっている 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 で定義された事前定義されたparameterType属性がありますEKS。リリースメタデータは、parameterTypeと 間のマッピングを指定しますname/path of the parameter in the template. This way, the values can be dynamically passed-in by Amazon EKS without requiring customers to specify these through configurations and also gives flexibility to add-on providers to define their own template name/path。Amazon が動的に注入EKSする必要がある上記のパラメータは、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設定することをお勧めしないパラメータです。パラメータには変更不可能なデフォルトを含めるか、アドオンテンプレートにまったく含めないでください。

パラメータ	説明	にはデフォルトが必要ですか？
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ポッドを設定する。	該当なし
livenessProbe	Kubelet プロセスでは、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キャッチする可能性があります。このような状態の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可能にすることができます。	可能
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認識します。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ポッドを制御することです。ポッドの準備が整っていない場合、ポッドはサービスロードバランサーから削除されます。	可能

パラメータ	説明	にはデフォルト が必要ですか？
startupProbe	<p>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がいつ開始されたかを確認します。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、起動が遅いコンテナのライブネスチェックを採用するために使用され、起動および実行前に kubelet によって殺されるのを回避できます。</p>	<p>オプションです。</p>
podDisruptionBudget	<p>Pod Disruption Budget (PDB) を定義して、任意の中断時に PODS が引き続き実行されるようにします。は、レプリケートされたアプリケーションの Pod の数 PDB を制限します。これは、自発的な中断によって同時にダウンします。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クォーラムに必要な数を下回らないようにしたいと考えています。ウェブフロントエンドは、負荷を供給するレプリカ数が合計の特定の割合を下回らないようにしたい場合があります。</p>	<p>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</p>
serviceAccount (名前)	<p>で実行されるサービスアカウントポッドの名前。</p>	<p>可能</p>

パラメータ	説明	にはデフォルト が必要ですか？
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のIAMロール機能に使用されます。	いいえ、IAMサービスアカウントロールARNは最上位のAmazon EKSアドオンで設定されますAPI。このルールの例外は、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ロールが必要な場合ですARNs。
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のスケジュールを有効にします。	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、ポッドまたはコンテナの権限とアクセスコントロールの設定を定義します。通常、を設定するために使用します。fsGroupこれはv1.19以下のクラスターIRSAで必要でした。	AmazonがKubernetes v1.19をサポートEKSしなくなった場合、可能性は低い

パラメータ	説明	にはデフォルト が必要ですか？
securityContext	セキュリティコンテキストは、ポッドまたはコンテナの権限とアクセスコントロールの設定を定義します。	可能
updateStrategy	古いポッドを新しいポッドに置き換えるために使用される戦略を指定します。	可能
nameOverride	ポッドの名前を上書きします。	不可
podSecurityPolicy	パラメータに制限を適用します。	いいえ - PSPs 非推奨
extraVolumeMounts/extraVolumes	Amazon 以外のEKSクラスターIRSAでに使用されます。	不可

のコンテナ製品の料金 AWS Marketplace

では AWS Marketplace、Amazon Elastic Container Service (Amazon BYOL)、Amazon Elastic ECSKubernetes Service (Amazon)、および の無料製品、Bring Your Own License Model (EKS) 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、コンテナ製品の利用可能な料金モデルの概要を説明します。

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	<p>[月額固定料金]</p> <p>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</p> <p>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る 3 つの異なるコンテナイメージが含まれています。</p> <p>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。Amazon ECSまたは Amazon の</p>

料金モデル	説明
	<p>イメージから任意の数のコンテナを起動して、任意の設定EKSで実行できます。</p> <p>購入者が 1 か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ません AWS Marketplace。購入者には、最終月の未使用分が返金されます。支払いは、購入者の使用量から合意された AWS Marketplace 料金を差し引いた額に基づいて行われます。</p>
カスタムメトリクス料金ディメンション	<p>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)。</p> <p>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。</p>

料金モデル	説明
<p>タスクあたりまたはポッドあたりの時間単位料金</p>	<p>Amazon ECSタスクまたは Amazon EKS ポッド</p> <p>Amazon ECSタスクごと、または Amazon EKS ポッドごとの料金の、1 時間あたりの料金で 2 秒単位で測定します。</p> <p>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。</p> <p>コントローラーノードのコンテナイメージ内の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。</p> <p>購入者には、実行されている Amazon EKS コントローラーポッドごとに、使用量 1 時間あたり 6 ドルが課金されます。購入者が EKS コントローラーノードコンテナを含む 5 つの Amazon コントローラー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。</p> <p>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。支払いは、購入者の使用量から合意された AWS Marketplace 料金を差し引いた額に基づいて行われます。</p>

料金モデル	説明
時間単位/長期契約による使用量	<p>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。</p> <p>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) を支払うこととなります。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。</p> <p>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となります。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。</p> <p>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。</p>

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、お客様とお客様との契約に基づいて、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。

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、またはユーザー から選択できます。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。

Product Load Form (PLF) を使用してコンテナ製品の契約を作成する場合は、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、CheckoutlicenseAPI呼び出しEntitlementの 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 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	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 ドル
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 ドル	16,000 ドル
1 時間あたりのモニタリング対象追加ホスト	10 ドル	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 ドル	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

例えば、ReadOnlyUsersと AdminUsersディメンションがある場合、に年額料金を提供する場合は ReadOnlyUsers、 AdminUsersにも年額料金を提供する必要があります。

自動更新

お客様がコンテナ契約 AWS Marketplace を使用し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、購入者のエンタイトルメント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。使用料金のコンテナベースの製品の場合、を使用して、製品を使用するためのエンタイトルメントと、請求のための計測使用量 [AWS Marketplace Metering Service](#) の両方

を確認できます。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、での AWS Marketplace Metering Service 時間単位およびカスタム計測と での契約料金について詳しく説明します AWS License Manager。

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の設定](#)
- [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。請求に使用する独自の料金単位とメーターを定義する場合は、[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 AWS 測定する場合は、[RegisterUsage](#)API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

と統合 AWS Marketplace Metering Service することで、有料ソフトウェアを実行している顧客が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。エンタイトルメントを確認するには、料金モデルに応じて [MeterUsage](#) 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額料金モデルの場合は、[RegisterUsage](#)APIオペレーションを使用します。カスタム計測料金モデルの場合は、[MeterUsage](#)APIオペレーションを使用します。

購入者に製品に対する権限がない場合、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および公開し、MeterUsageまたは RegisterUsageAPIオペレーションをエンタイトルメントと計測に使用するときには、次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 から MeterUsageまたは RegisterUsageAPIオペレーションを呼び出すにはEKS、[サポートされているを使用する必要があります AWS SDK](#)。Amazon をテストMeterUsageまたはRegisterUsage統合するにはEKS、Kubernetes 1.13.x 以降を実行する 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。IAM 実行中のポッドが Amazon 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、ロールが必要ですEKS。
- 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 RegisterUsageAPIオペレーションと統合する場合は、特定の を使用するように 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自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。したがって、RegisterUsageAPIオペレーションはInvalidRegionExceptionエラーをスローします。

AWS SDK 言語はAWS_REGION、一貫した方法で を決定しません。が自動的に をピックアップSDKしない場合はAWS_REGION、ソフトウェアを手動で記述して を決定する必要がありますAWS_Region。例えば、環境変数やその他の設定が存在しない場合、AWS SDK for Java は自動的に [Amazon EC2インスタンスメタデータ](#) (具体的には ec2InstanceMetadata) を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[AWSSDK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、製品が 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、お客様が購入したライセンス (エンタイトルメントとも呼ばれます) 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。

の詳細については 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ターの作成] - 販売者は、AWS Marketplace 管理ポータルプライベートオファター作成ツールを使用して、製品のプライベートオファターを作成できます。
- [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 に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 EC2インスタンス、Amazon ECSタスク、または 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の設定

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 を使用して、のコンテナ製品で時間単位の計測を行うことができます AWS Marketplace。以下のセクションでは、AWS Marketplace Metering Service で時間単位の計測を設定する方法を示します。

RegisterUsage API オペレーションメーターのソフトウェアは、Amazon Elastic Container Service (Amazon ECS) タスクごと、または Amazon Elastic Kubernetes Service (Amazon EKS) ポッドごとに、1 時間あたりに使用し、使用量は 2 秒に比例配分されます。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。ソフトウェアは、ソフトウェアの使用を開始するための計測のために 1 RegisterUsage 呼び出すことを除いて、計測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すぐに呼び出す必要があります。コンテナの起動から 6 時間以内にコンテナを登録しない場合、AWS Marketplace Metering Service は過去 か月間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。

の AWS Marketplace Metering Control Plane は、お客様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お客様に請求を続けます。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。

トピック

- [時間単位の計測の前提条件](#)
- [時間単位の計測用の製品ロードフォーム](#)
- [RegisterUsage の統合とプレビューモードのテスト](#)
- [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す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。
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の 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、 RegisterUsageAPIオペレーションをテストコールします。

時間単位の計測用の製品ロードフォーム

時間単位の計測のために製品積載フォームに入力する際、製品を定義するその他の必須フィールドとオプションのフィールドに加えて、製品に関する次のフィールドに入力します。

- [時間料金] - 製品の 1 時間あたりの料金。
- [ディメンション長期レート] - 購入者が前払いした場合の、長期契約におけるソフトウェア料金の合計です。
- [長期期間 (日数)] - 長期契約の期間 (日単位)。

RegisterUsage の統合とプレビューモードのテスト

RegisterUsage API オペレーションを使用して、イメージを公開 AWS Marketplace のために送信する前に統合をテストします。

プレビューモードは稼働モードと同じように動作しますが、プレビューモードでは製品を使用する権限を確認しません。RegisterUsage プレビューモードで を呼び出すには、Amazon ECSまたは Amazon で製品を実行して、コンテナイメージ RegisterUsage から を呼び出します EKS。AWS アカウント 使用している を使用して、 に製品を一覧表示します 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、動的に を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 つ起動します。これにより、AWS Marketplace オペレーションチームは、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 の両方をサポートしている場合は EKS、統合を検証するために Amazon EKS で を起動するだけで済みます。

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、プレビューモードで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。

以外の例外 ThrottlingException 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 ポッドからの後続の呼び出しは、タスクまたはポッドの実行中にお客様がサブスクライブを解除 CustomerNotSubscribedException してもスローされません。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のあるエラーを示しています。各 AWS SDK プログラミング言語には、追加情報として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
InternalServiceErrorException	RegisterUsage は使用できません。
CustomerNotEntitledException	製品の有効なサブスクリプションが顧客にありません。
InvalidProductCodeException	リクエストの一部として渡された ProductCode 値が存在しません。
InvalidPublicKeyException	リクエストの一部として渡された PublicKeyVersion 値が存在しません。
PlatformNotSupportedException	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。
ThrottlingException	RegisterUsage の呼び出しが抑制されます。
InvalidRegionException	RegisterUsage 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、RegisterUsage を呼び出す際にコンテナがリージョン (withRegion("us-east-1") 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 Marketplace Metering Service と AWS SDK for Java 統合できます。ソフトウェア使用の継続的な計測は、によって自動的に処理されます。AWS Marketplace Metering Control Plane。ソフトウェアの使用を開始するために 1 RegisterUsage 回呼び出すことを除き、ソフトウェアは計測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 6 時間以内にコンテナを登録しない場合、AWS Marketplace Metering Service は過去 か月間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 統合のステップ例

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。アクセス IAM 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 のアクセス許可”](#)。
3. パブリック [AWS Java SDK](#) をダウンロードします。

Important

Amazon APIs から計測を呼び出すには EKS、[サポートされている](#) を使用し [AWS SDK](#)、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. (オプション) RegisterUsage アクションと統合していて、デジタル署名検証を実行する場合は、アプリケーションクラスパスで [BouncyCastle](#) 署名検証ライブラリを設定する必要があります。

Web JSON Token (JWT) を使用する場合は、アプリケーションクラスパスに [JWT Java](#) ライブラリも含める必要があります。JWT を使用すると、署名検証へのアプローチが簡素化されますが、必須ではなく、BouncyCastle 代わりにスタンドアロンを使用できます。JWT または を使用するかどうかにかかわらず BouncyCastle、Maven などのビルドシステムを使用して、アプリケーションクラスパス JWT に BouncyCastle または の一時的な依存関係を含める必要があります。

```
// 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 Metering Service への接続中に一時的な問題が発生する可能性があります。AWS Marketplace では、短期間の停止やネットワークの問題を避けるため、最大 30 分間の再試行を、指数関数的にバックオフして実装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を検証するために使用できる 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の新しいバージョンを再構築し、コンテナにタグを付け、Amazon や Amazon ECR ECR Public などEKS、Amazon ECSや Amazon と互換性のあるコンテナレジストリにプッシュします。Amazon を使用している場合は ECR、Amazon ECSタスクまたは Amazon ポッドを起動するアカウントが Amazon EKS ECRリポジトリに対するアクセス許可を持っていることを確認してください。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RegisterUsage、コンテナが 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。このIAMロールは、Amazon ECSタスクまたは Amazon EKS ポッド定義の [タスクロール](#) パラメータで指定する必要があります。

```
{
  "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、タスク定義の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 AWS Marketplace 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 {
```

```

        XmlNode payloadJson = Jackson.getObjectMapper().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"
        + "UWAY9iR3fy4arWNA1KoS8kVw33cJibXr8bvWUAUpaCwlvdbH6dvE0fou0/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

AWS Marketplace コンテナ製品には、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設定できます。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 と統合します。[MeterUsage](#) API オペレーションを使用して、請求 AWS のために、そ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、製品積載フォームおよび MeterUsage オペレーションを介した 2 つの場所で定義されます。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 で測定された帯域幅 MBps、ホストの数、または MB、GB、TB で測定されたデータなどです。
- 料金ディメンション - 料金ディメンションは、ユニットあたりの料金を設定している機能またはサービス (ユーザー、スキャン、vCPUs、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、パブリック製品のプライベートおよび Bring Your Own License (BYOL) オファーを定義することはできません。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。

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](#)
- [カスタム計測用の製品ロードフォーム](#)
- [MeterUsage の統合とプレビューモードのテスト](#)
- [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 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. 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. 製品積載フォームに必要なディメンション情報を入力し、処理のために当社に返送してください。

3. (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す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。
4. (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のログ記録を有効にすることをお勧めします。
5. 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、 MeterUsageAPIオペレーションをテスト呼び出します。

カスタム計測用の製品ロードフォーム

カスタム測定用の製品積載フォームに記入する場合、各製品には最大 24 のディメンションを設定できます。ディメンションは次のフィールドで定義されています。

- [ディメンション名] - コンテナアプリケーションが AWS Marketplace Metering Serviceに計測レコードを送信するときに使用される名前。この名前は、購入者が使用するディメンションを示します。この名前は請求レポートに記載されます。この名前を設定した後に変更することはできません。
- [ディメンションの説明] 購入者向けのディメンションの説明。説明は最大 70 文字です。製品が購入者に公開された後は、このフィールドを変更することはできません。
- [ディメンション料金] - 購入者が使用ごとに支払う場合のこの製品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。
- [ディメンション長期レート] - 購入者が前払いした場合の、長期契約におけるソフトウェア料金の合計です。
- [長期期間 (日数)] - 長期契約の期間 (日単位)。

MeterUsage の統合とプレビューモードのテスト

MeterUsage オペレーションを使用して、イメージを公開 AWS Marketplace のために に送信する前に統合をテストします。

プレビューモードは稼働モードと同じように動作しますが、プレビューモードでは製品を使用する権限を確認しません。MeterUsage プレビューモードで を呼び出すには、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。 を使用して に製品を一覧表示 AWS アカウント 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、

動的に を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 の両方をサポートしている場合はEKS、統合を検証するために Amazon EKS で を起動するだけで済みます。
- 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、レビューモードで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 ポッドからの後続の呼び出しは、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CustomerNotSubscribedExceptionしてもスローされません。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「」を参照してくださいMeterUsage。各 AWS SDKプログラミング言語には、追加情報として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 (ISVs) が購入者にソフトウェア使用状況をより詳細に把握し、コスト配分を実行するのに役立ちます。

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 () のタグ値によってコストを用量に分割して表示できます <https://console.aws.amazon.com/billing/>。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。

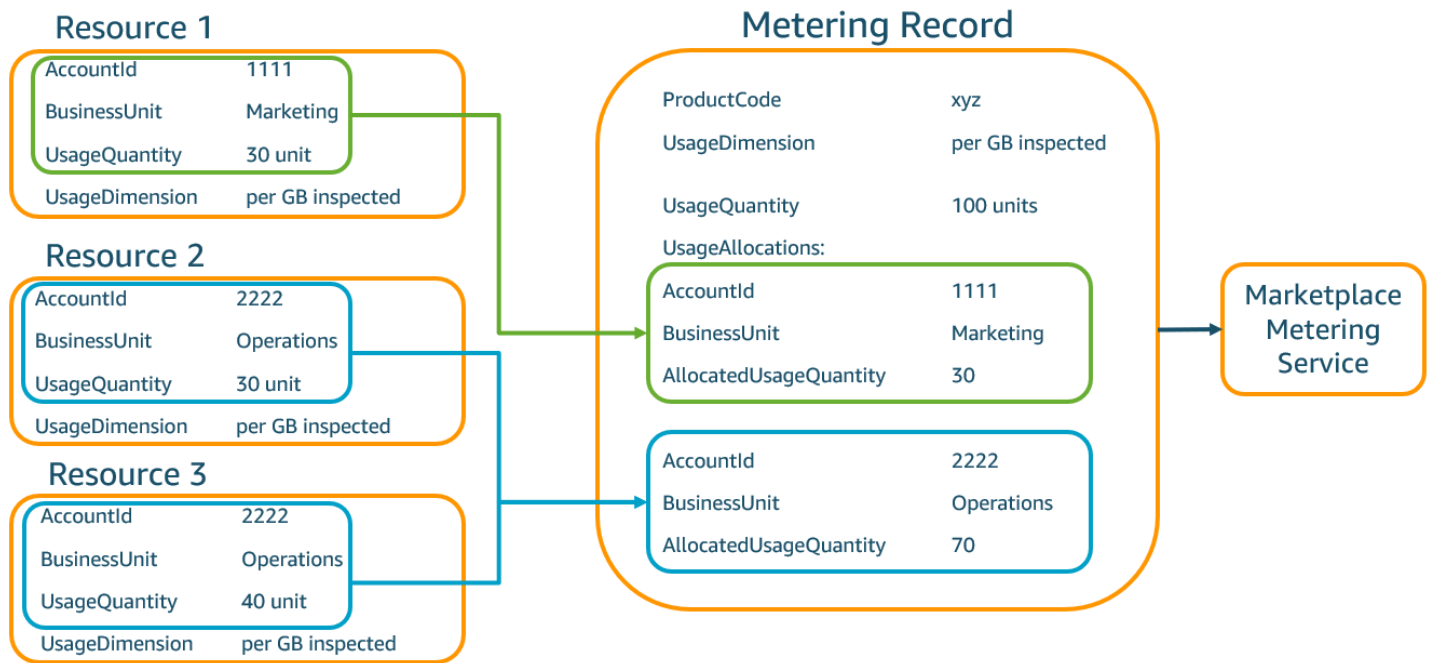
一般的なユースケースでは、購入者が1つの AWS アカウントを使って商品を購入します。また、購入者には同じ製品サブスクリプションに関連するユーザーが多数います。AccountId のキーを持つタグを使用して用量の割り当てを作成し、各ユーザーに用量を割り当てることができます。この場合、購入者は Billing and Cost Management コンソールで AccountId タグを有効にして、個々のユーザーの使用状況を分析できます。

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。

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	111122223 333	ネットワーク: 検査対象 (GB) あたり	70	2222	オペレーショ ン
xyz	111122223 333	ネットワーク: 検査対象 (GB) あたり	30	3333	財務
xyz	111122223 333	ネットワーク: 検査対象 (GB) あたり	20	4444	IT
xyz	111122223 333	ネットワーク: 検査対象 (GB) あたり	20	5555	マーケティング
xyz	111122223 333	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メンテナンスに必要な 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、適切な使用量割り当てタグを含む計測レコードを送信 AWS Marketplace して、顧客に pay-as-you-go 料金を請求します。

```
# 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
)
```

の詳細についてはMeterUsage、リファレンス[MeterUsage](#)の「」を参照してください。AWS Marketplace Metering Service API

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 コンテナ製品には、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設定できます。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 と統合します。[MeterUsage](#) API オペレーションを使用して、請求のために、その使用量 AWS の独自の料金単位とカスタム計測を定義できます。次の例は、AWS SDK for Java を使用して [AWS Marketplace Metering Service](#) MeterUsage オペレーションと統合する実装の概要を示しています。

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 のアクセス許可”](#)。
3. パブリック [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](#)、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。このAPIオペレーションでは、Dimension、Resource 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 Hour。リソースは、Amazon Elastic Container Service (Amazon ECS) タスクまたは Amazon EKS 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

への接続時に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避けるため、では、最大 30 分間の再試行を指数関数的にバックオフして実装することをお勧めします。

5. MeterUsage 呼び出しを含むコンテナイメージの新しいバージョンを再構築し、コンテナにタグを付け、Amazon Elastic Container Registry (Amazon ECR) などEKS、Amazon ECSまたは Amazon と互換性のある任意の Docker レジストリにプッシュします ECR。Amazon を使用している場合は ECR、Amazon ECS タスクまたは Amazon ポッドを起動するアカウントが Amazon EKS ECR レジストリに対するアクセス許可を持っていることを確認してください。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 MeterUsage、コンテナが 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 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) EKS ロールを指定する必要があります。

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

- 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。
- タスク ECS または ポッドを実行する Amazon または Amazon EKS クラスターを作成します。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。
- Amazon ECS または Amazon EKS クラスターを設定し、us-east-1 AWS Region EKS のステップ 8 で作成した Amazon ECS タスク定義または Amazon 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。
- 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 と 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、お客様が購入したライセンス (エンタイトルメントとも呼ばれます) 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。

License Manager を Amazon EKS Anywhere、Amazon Anywhere、Amazon Elastic Compute Cloud (Amazon EC2)、またはオンプレミスインフラストラクチャ用の AWS Marketplace for Containers ECS Anywhere 製品と統合する場合は、「」の手順に従います [AWS Marketplace for Containers Anywhere と License Manager の統合](#)。

の詳細については 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。

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、CheckoutLicenseAPI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 (権限) をチェックアウトできます。

Example S3 へのバックアップの年間単位数

Amazon Simple Storage Service (Amazon S3) に、1 年間最大 1,024 単位のデータをバックアップでき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。

この例では、ソフトウェアが CheckoutLicenseAPIオペレーションを呼び出して 10 単位のデータを調べます。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呼び出しは失敗します。

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 CheckoutLicenseAPI オペレーションを使用して、エンタイトルメントプールからエンタイトルメント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。

権限をプールに再確認するには、リソースが使用されなくなったときに CheckInLicenseAPI 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 分を超えて使用されている場合は、リソースが使用されている限り、ExtendLicenseConsumptionAPI オペレーションを使用して権限をプールからチェックアウトしておくことをお勧めします。

```
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。

フローティングエンタイトルメントの場合、CheckInLicenseAPI オペレーションを使用してライセンスユニットをエンタイトルメント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。お客様は、AWS Identity and Access Management (IAM) アクセス許可を提供することでソフトウェアを起動します。ユーザーがログインすると、アプリケーションは CheckoutLicenseAPI オペレーションを呼び出して数量を 1 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 CheckInLicenseAPI 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返します。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。ソフトウェアは、CheckoutLicenseAPI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することで、お客様がサインアップした階層を識別できます。

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、および CheckInLicenseAPI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ExtendLicenseConsumption、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については、次のIAMポリシーを参照してください。

```
{
  "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、RegisterUsageAPIオペレーションをテスト呼び出します。

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. をダウンロードします AWS SDK。

Note

Software. AWS credentials 内で AWS 認証情報を設定しないでください。購入者の認証情報は、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。

製品は、ライセンスチェックを実行する場所ならどこでも CheckoutLicenseAPI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 呼び出しは、設定した料金ライセンスの種類によって異なります。

4. で製品リストを公開します AWS Marketplace。

License Manager APIオペレーション

お客様の 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には、ソフトウェアで次のAPI操作を使用できます。

- GetLicense - ソフトウェアAPIがクエリできる。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。また、CheckoutLicenseAPIオペレーションを使用して、ユーザーがある程度のライセンスを消費したときに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時間を延長する場合は、ExtendLicenseConsumptionAPI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 - ディメンションがフローティングの場合、ライセンスをエンタイトルメントプールに戻すときは、CheckInLicenseAPIオペレーションを使用します。
- 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。追加購入後、は新しいエンタイトルメントを反映した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エンタイトルメントを読み取ります。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、キャンセルなどにより、製品が使用中は定期的に CheckoutLicenseAPIオペレーションを呼び出すことをお勧めします。定期的に CheckoutLicenseAPIオペレーションを使用することで、製品はアップグレードや有効期限などのエンタイトルメントの変更を検出できます。

15分ごとにCheckoutLicenseAPI通話を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 Anywhere、Amazon EKS、EC2またはオンプレミスインフラストラクチャ用の for AWS Marketplace Containers ECS Anywhere 製品 AWS License Manager と統合できます。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を含む AWS Marketplace、と License Manager の統合に関する一般的な情報については、「[AWS License Manager ユーザーガイド](#)」を参照してください。[AWS License Manager ユーザーガイド](#)。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 での License Manager 統合のテスト EKS](#)
- [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

for AWS Marketplace Containers Anywhere 製品を と統合するには、以下の手順に従います 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 を選択すると、 で確認できます。Containers ページには、関連する製品を含む製品のリストが表示されますIDs。

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。希望 AWS SDKのインストール手順については、 [で構築するツールAWS](#) を参照してください。

Note

Amazon EKS Anywhere または [EKS Anywhere](#) 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 SDKs](#) を使用する必要があります AWS SDK。サポートされている [AWS SDKs](#) のリストを表示するには AWS SDKs、[「サポートされている の使用 AWS SDK」](#) 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、デプロイされたコンテナアプリケーション AWS とオンプレミス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 (LicenseCredentialProvider) 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](#)

LicenseCredentialProvider は、[LicenseManagerTokenCredentialsProvider](#) を追加することで、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 が OIDC 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。LicenseCredentialProvider 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

[LicenseManagerTokenCredentialsProvider](#) を拡張 DefaultCredentialsProvider すると、オンプレミス環境で [LicenseManagerTokenCredentialsProvider](#) を実行するとき AWS と [LicenseManagerTokenCredentialsProvider](#) を実行するときに、同じコンテナアプリケーションが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. 製品オファリングの各有料コンテナイメージの `aws license-manager checkout-license` コマンドを使用して、CheckoutLicenseAPI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ときは、次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。一意のリクエストUUIDごとにランダム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、 を使用する CheckoutLicenseAPIオペレーションを使用した呼び出しの例です AWS CLI。

```
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。

6. で行われた更新、アップグレード、またはキャンセルによるお客様のライセンスの変更を特定するために、定期的に CheckoutLicenseAPIオペレーションを呼び出します AWS Marketplace。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。

7. 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。
8. License Manager を使用してライセンスにアクセスする設定を含むオプションの入力として Kubernetes シークレットを受け入れるように、アプリケーションの Helm チャートを更新します。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、コンテナアプリケーションがオンプレミスにデプロイされた APIs ときに License Manager 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。

コンテナアプリケーションをオンプレミスに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 での Helm デプロイにはサービスアカウントが必要です。EKS。コンテナイメージでサービスアカウントの IAM ロールを設定することで、License Manager API オペレーションを呼び出すアクセス許可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、[IAM 「サービスアカウントのロール」](#)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 – 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、ライセンス設定シークレットが必要です。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.awssmp.licenseConfigSecretName }}
        //Mount the license volume to the container image
        volumeMounts:
          - name: awssmp-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: awsmvp-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を含める必要があります。

10. 以下のセクションの手順EKSを実行して、License Manager の統合をローカルおよび Amazon でテストします。
 - a. [License Manager 統合のローカルテスト](#)
 - b. [Amazon での License Manager 統合のテスト EKS](#)
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、リファレンス [CreateLicense](#) の「」を参照してください。AWS License Manager API 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"
```

- 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 CreateTokenAPI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、IAMCreateRoleAPI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持つ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. の外部でホストされている Kubernetes クラスターを設定します 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 に接続 AWS License Manager API 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. 以前に生成され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 Helm 経由でアプリケーションをデプロイし、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エンタイトルメント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 での License Manager 統合のテスト EKS

Amazon で License Manager の統合をテストすることもできます EKS。アプリケーションがライセンス設定シークレットなしで License Manager API オペレーションを呼び出せることをテストします。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の IAM ロール (IRSA) を設定し、アプリケーションに関連する認証情報を提供できることを確認してください。

Amazon で License Manager 統合をテストするには EKS

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セットアップするか、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許可を作成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. 目的の設定のテスト 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 awsmc-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、IAMロールに関連付けます。次のコマンドは、でIAMロールを作成しますAWSLicenseManagerConsumptionPolicy。次に、コマンドは License Manager 統合イメージをデプロイする Amazon EKSクラスターtest_saの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す。

```
eksctl create iamserviceaccount \  
--name test_sa \  
--namespace test_namespace \  
--cluster awsmc-eks-test-example \  
--attach-policy-arn "arn:aws:iam::aws:policy/service-role/  
AWSLicenseManagerConsumptionPolicy" \  
--approve \  
--override-existing-serviceaccounts
```

4. IAM ロールが前のコマンドから関連付けられているサービスアカウントに Helm 経由で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エンタイトルメント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 CheckoutLicenseAPI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エンタイトルメントプールからエンタイトルメント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った場合、アプリケーションは CheckInLicenseAPI オペレーションを使用して、権限をプールに戻します。

```
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。これを行うには、リソースが使用されている限り、ExtendLicenseConsumptionAPI オペレーションを使用します。

```
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、を追加することで、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 は、オンプレミス環境で License Manager が OIDC発行した ID トークンを使用して認証情報を提供します。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.getStringValueOrThrow();
        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();

        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、を追加することで、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 は、オンプレミス環境で License Manager が OIDC 発行した ID トークンを使用して認証情報を提供します。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して、顧客がプライベートオファーを受け入れるタイミング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) (など) が必要です `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。Marketplace [AWSオペレーションチーム](#)に連絡して、 をリクエストしますARN。

コンテナ製品では、次の Amazon SNS トピックを使用できます。

- [Amazon SNS トピック: `aws-mp-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",
"isFreeTrialTermPresent": "true"
}
```

- `<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

offer-identifier は、オファーがプライベートオファーの場合にのみ通知に表示されます。

Amazon SQSキューを Amazon SNSトピックにサブスクライブする

提供されたSNSトピックに Amazon SQSキューをサブスクライブすることをお勧めします。SQS キューの作成とトピックへのキューのサブスクライブに関する詳細な手順については、[「Amazon Simple Notification Service デベロッパーガイド」の「Amazon SNSトピックへの Amazon SQS キューのサブスクライブ」](#)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 のトピックのみをサブスクライブ AWS Marketplace SNS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SNSへの Amazon メッセージの送信」](#)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 に保存されますSQS。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 販売者として、購入者が に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 製品タイプについて説明します AWS Marketplace。

には 2 種類の SageMaker 製品があります 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 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 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 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 で機械学習製品を作成するサポートについては SageMaker、[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。

機械学習製品の使用を開始する

AWS Marketplace は、Amazon を使用して 2 つの機械学習製品タイプをサポートしています SageMaker。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 モデルパッケージ

[Amazon SageMaker 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。トレーニング済みのモデルは、 にデプロイ SageMaker 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。販売者として、を使用してモデルをトレーニング SageMaker したり、独自のモデルを持ち込むことができます。

SageMaker アルゴリズム

購入者は [SageMaker 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。では SageMaker、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を使用してトレーニングジョブを作成します。トレーニングコンポーネントのアルゴリズムが完了すると、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。は、購入者の Amazon Simple Storage Service (Amazon S3) バケットにモデルアーティファクト SageMaker を保存します。では SageMaker、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、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。エンドポイントにデータが送信されると、SageMaker はそれをモデルコンテナに渡し、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

では 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 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 でモデルをデプロイする SageMaker」](#) 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、はモデルを SageMaker 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てから、結果を S3 バケットに返します。ジョブが完了すると、はジョブを SageMaker 停止します。詳細については、[「バッチ変換の使用」](#) を参照してください。

Note

はデータをモデルに渡し、結果を購入者に返すため SageMaker、どちらの方法もモデルに対して透過的です。

Amazon のセキュリティと知的財産 SageMaker

Amazon 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 SageMaker を保護します AWS Marketplace。以下のセクションでは、 が知的財産 SageMaker 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「[で製品を準備する SageMaker](#)」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、 は購入者のみが AWS のサービス エンドポイントを介して製品にアクセス SageMaker できるようにします。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 モデルやアルゴリズムとは異なり、購入者が から製品を起動すると AWS Marketplace、モデルや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 は、ネットワークや 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 のサービスに対してアウトバウンドAPIコールを実行できません AWS のサービス。

顧客データのセキュリティ

製品は購入者の SageMaker 内で実行されます 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ため、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 のセキュリティ SageMaker」](#) を参照してください。

の機械学習製品の料金 AWS Marketplace

では、Amazon SageMaker 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 内で実行します 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 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 SageMaker 中の 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 の SageMaker 料金](#) ページで利用できます。

ソフトウェアの料金

製品の使用に対して購入者に AWS Marketplace 請求するソフトウェア料金を決定します。機械学習製品を に追加するときに、料金と条件を設定します AWS Marketplace。

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購入者に表示されます。

トピック

- [料金無料](#)
- [時間単位の料金](#)
- [時間単位料金での年間契約](#)
- [推論の料金](#)
- [無料トライアル](#)
- [料金の変更](#)

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

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 SageMaker。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。

時間単位料金での年間契約

契約オプションでは、固定前払い料金を指定でき、サブスクリプション時に契約の全額が請求されます。年間契約の終了時に、引き続き実行するインスタンスには設定した時間単位の料金が発生します。

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、は推論料金で、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レスポンスヘッダーにカスタム計測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

推論料金の場合、HTTPレスポンスコードが のリクエストに対して AWS Marketplace のみ購入者に課金されます2XX。

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。無料トライアル期間は 5～120 日間です。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せん。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。

Note

無料トライアルを作成できるのは、時間単位で請求されるオファーのみです。推論の料金が適用される製品の無料トライアルは作成できません。

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。

料金の変更

販売者は、[AWSマーケットプレイス販売者オペレーション](#)チームに連絡して、機械学習製品の料金を変更することができます。製品 ID と新しい料金の詳細を入力します。新しい料金は 90 日後に有効になります。また、2 回目の料金変更を行うには 90 日待つ必要があります。この制限は、サポートされているインスタンスの既存のリストに新しいインスタンスタイプを追加する場合にも適用されます。例として、2023 年 11 月 1 日に機械学習製品の料金を引き上げた場合、2024 年 1 月 30 日以降に新しいインスタンスタイプの追加や、2 回目の料金変更が可能です。

で製品を準備する SageMaker

で製品を公開する前に AWS Marketplace、Amazon で準備する必要があります SageMaker。にリストされている SageMaker 製品には、AWS Marketplaceモデルパッケージとアルゴリズムの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 リソースの作成](#) - イメージが正常にスキャンされたら、イメージを使用して、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 SageMaker。

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 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および実行します。は、ロジックを含む Docker コンテナイメージ SageMaker を実行します。SageMakerは、これらのコンテナを安全でスケラブルなインフラストラクチャで実行します。詳細については、「[Amazon のセキュリティと知的財産 SageMaker](#)」を参照してください。以下のセクションでは、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 SageMaker。

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、コンテナがファイル構造に準拠 SageMaker 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 のセキュリティと知的財産 SageMaker](#)」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 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。でモデルパッケージを作成し SageMaker、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、のモデルパッケージを作成する方法を説明します 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](#)
- [モデルパッケージの推論イメージを作成する](#)

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

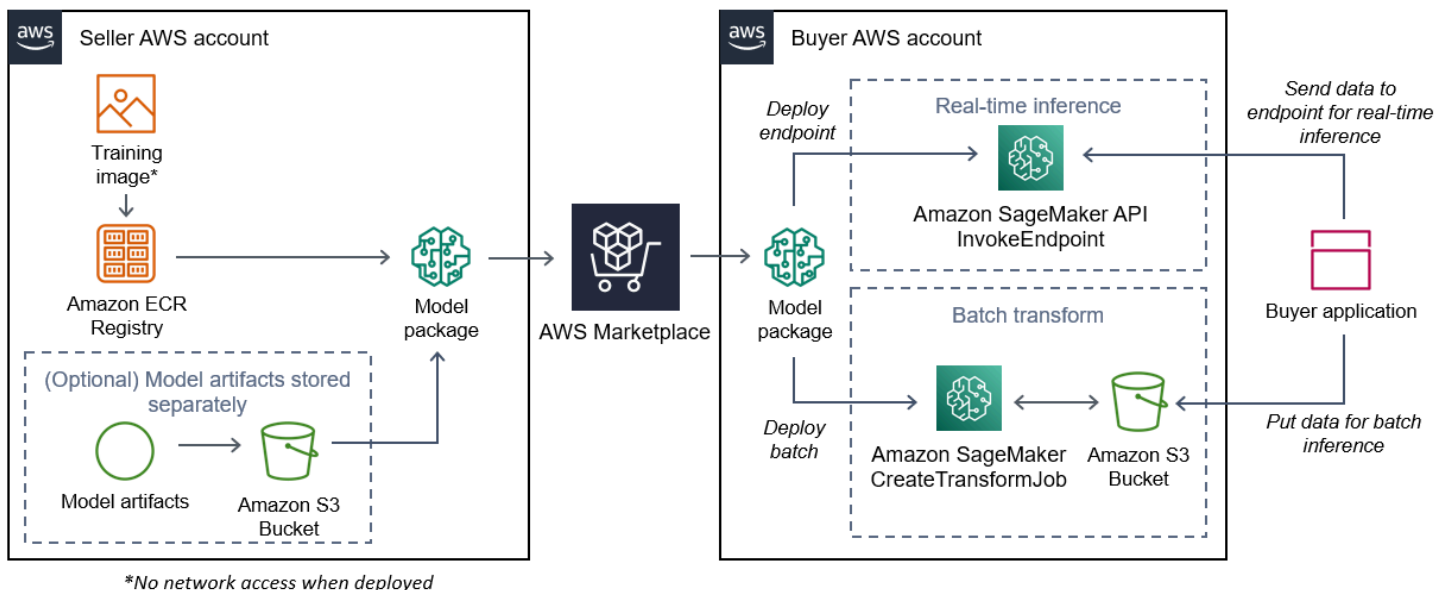
Note

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 に個別に保存することもできます SageMaker。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発行すると、これらのアーティファクトは発行され、購入者が直接アクセスできない AWS Marketplace 所有の Amazon S3 バケットに保存されます。

Tip

推論モデルが Gluon、Keras、MXNet、PyTorch TensorFlow、TensorFlow-Lite、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 Neo の使用 ONNX を検討してください。Neo は、m1.c4、m1.p2 などの、クラウドインスタンスタイプの特定のファミリー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 デベロッパーガイド](#)」の「[Neo を使用したモデルパフォーマンスの最適化](#)」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 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を作成し (デプロイ時にネットワークアクセスなし)、Amazon ECR Registry 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 でモデルパッケージリソースを作成し SageMaker、ML 製品を に公開します 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 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 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。

モデルパッケージの推論イメージを作成する

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構築してテストする](#)

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 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。

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、「[「での Docker コンテナの使用 SageMaker」](#) および [AWS Marketplace SageMaker 「」の例](#) を参照してください [GitHub](#)。

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 と互換性を持たせるには SageMaker、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。コンテナの実行中に、 はコンテナの HTTP エンドポイントへの推論のために 購入者入力を SageMaker 渡します。推論結果はレスポンスの本文に返されます HTTP。

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 CLI 開発環境で Docker 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 が使用する TCP ポート 8080 の 2 つの HTTP エンドポイントを提供する Flask ウェブサーバ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` – このエンドポイントにHTTPGETリクエスト SageMaker 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。コンテナの準備ができたら、200 HTTP レスポンスコードでこのエンドポイントのHTTPGETリクエストに応答します。
- `/invocations` – 推論のためにこのエンドポイントにHTTPPOSTリクエスト SageMaker 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 ヘッダーに渡されます。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、入力を処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 `serve SageMaker` の実行時に実行される 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 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 に追加されPATH、コンテナの実行時に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、コンテナの実行/opt/ml/model/時にコンテンツを SageMaker 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構築してテストする

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、 が HTTP リクエスト SageMaker 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 実行され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。

ping HTTPエンドポイントをテストする

がコンテナ SageMaker を実行すると、定期的にエンドポイントに ping が実行されます。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 SageMaker ることが にシグナルされます。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する

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できたと示すと、SageMaker 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"}
```

これらの2つのHTTPエンドポイントが機能すると、推論イメージはと互換性を持つようになりました SageMaker。

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。両方のデプロイで、は Docker コンテナの実行中に同じHTTPエンドポイント SageMaker 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 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ってトレーニングする必要があります。AWS Marketplace 販売者として、SageMaker を使用して、購入者がに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、のアルゴリズムイメージを作成する方法を説明します 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

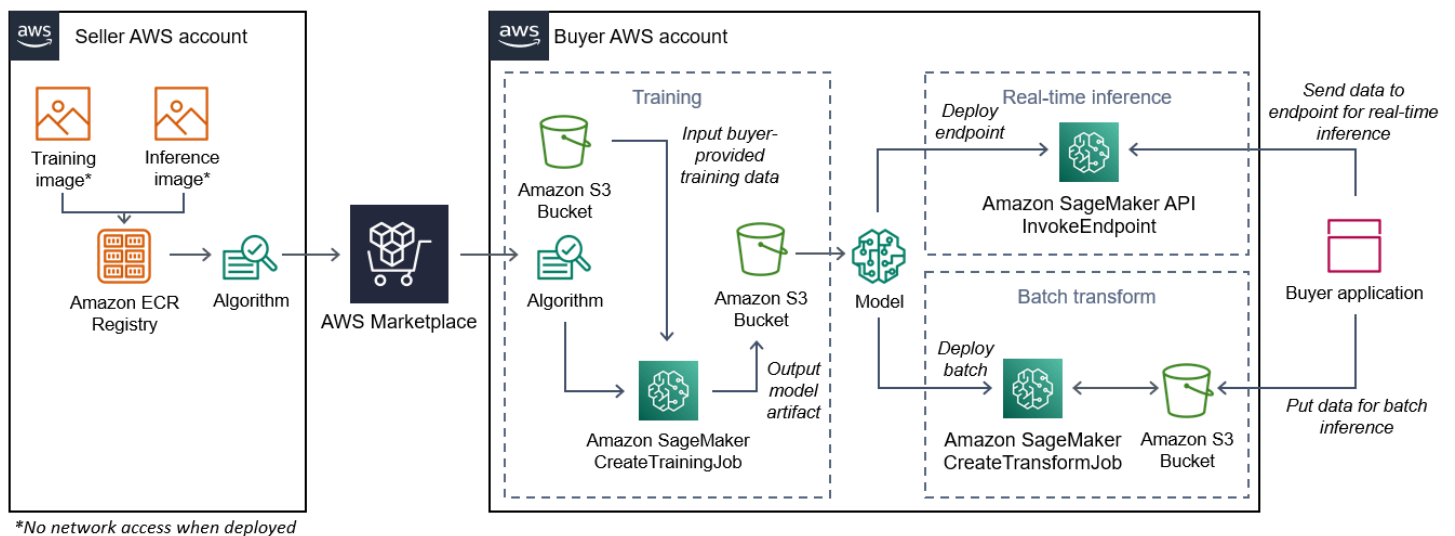
アルゴリズムには以下のコンポーネントが含まれています。

- [Amazon ECR](#)に保存されているトレーニングイメージ
- Amazon Elastic Container Registry (Amazon ECR) に保存されている推論イメージ

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 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 (デプロイ時にネットワークアクセスなし) を作成し、Amazon ECR Registry にアップロードします。
2. 次に、販売者は Amazon にアルゴリズムリソースを作成し SageMaker、ML 製品を に公開します 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。 はトレーニングイメージ SageMaker 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 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、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/このパスは、推論を提供するコードによって消費されます。
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 は、購入者に代わって推論用の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従って、SageMaker がコンテナとの間でデータをコピー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。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、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、「[で独自のアルゴリズムとモデルを使用する](#)」および [AWS Marketplace SageMaker](#) 「」の例を参照してください [GitHub](#)。

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 と互換性を持たせるには SageMaker、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 SageMaker 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 コピー元のコンテナ内の特定のディレクトリパスに保存されます。

以下は、Linux の Ubuntu ディストリビューションの開発環境に CLI インストールされた Docker を使用します。

- [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の名前と値の JSON フォーマットされたディクショナリです。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、[分散トレーニング](#) に使用されるネットワークレイアウトを説明する JSON フォーマットファイルです。トレーニングイメージが分散トレーニングをサポートしていない場合は、このファイルを無視できます。

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>/` にそのチャンネルの入力データが含まれます。チャンネルは `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ンネルが一致することが重要です。各チャンネルのファイルは [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、ディレクトリ全体の `tree`、`SageMaker packages` でも、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`) にすることができます。このファイルは、`DescribeTrainingJobAPI` 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 train コンテナイメージ SageMaker の実行時に実行される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 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は に追加されPATH、コンテナの実行時に実行できます。

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、「[で独自のアルゴリズムとモデルを使用する](#)」と [AWS Marketplace SageMaker](#) 「」の例を参照してください GitHub。

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使用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 SageMaker の実行時に同じスクリプト実行を実行します。

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 がコンテナとの間でデータを渡すようにします。

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、「[で独自のアルゴリズムとモデルを使用する](#)」および [AWS Marketplace SageMaker 「」の例](#)を参照してください [GitHub](#)。

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

推論イメージをと互換性を持たせるには SageMaker、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。コンテナの実行中に、は購入者から提供された推論SageMaker 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レスポンスの本文に返されます。

以下は、Linux の Ubuntu ディストリビューションの開発環境にCLIインストールされた Docker を使用します。

- [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 が使用する TCP ポート 8080 の 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` – このエンドポイントに HTTP GET リクエスト SageMaker 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。コンテナの準備ができたら、200 HTTP レスポンスコードでこのエンドポイントの HTTP GET リクエストに応答します。
- `/invocations` – 推論のためにこのエンドポイントに HTTP POST リクエスト SageMaker 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 ヘッダーに渡されます。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、入力を処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 `serve SageMaker` の実行時に実行される 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、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 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 に追加されPATH、コンテナの実行時に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/. SageMaker compressed the contents in that directory into a .tar.gz file and stored it in the buyer's AWS アカウント in Amazon S3.

モデルがデプロイされ、推論イメージ SageMaker が実行され、Amazon S3 の購入者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が抽出され、/opt/ml/model/ディレクトリの推論コンテナにロードされます。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 のセキュリティと知的財産 SageMaker](#)」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 \  
  my-inference-image
```

```
--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、ポートがHTTPリクエスト SageMaker を送信すること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用できるように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 SageMaker の実行時に同じスクリプト実行を実行します。

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。

ping HTTPエンドポイントをテストする

がコンテナ SageMaker を実行すると、定期的にエンドポイントに ping が実行されます。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 SageMaker ることが にシグナルされます。

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する

コンテナが 200 のステータスコードを返すことで準備できたことを示すと、 SageMaker はPOSTリクエストを介して推論データを/invocationsHTTP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"}
```

これらの2つのHTTPエンドポイントが機能すると、推論イメージは と互換性を持つようになりました SageMaker。

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。両方のデプロイで、 は Docker コンテナの実行中に同じHTTPエンドポイント SageMaker 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](#) はフルマネージドの Docker レジストリです。Amazon SageMaker は Amazon からイメージをプルECR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 は、モデルパッケージとアルゴリズム製品を公開

ECRするために Amazon からこれらのイメージも取得します。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 にアップロードするためのチュートリアルを提供します。 ECR

トピック

- [アップロードする必要があるイメージ](#)
- [どのようなIAMアクセス許可が必要ですか？](#)
- [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。

どのようなIAMアクセス許可が必要ですか？

次の手順では、ローカルマシンが販売者の AWS Identity and Access Management (IAM) ロールまたはユーザーに対して正しい AWS 認証情報を持ってい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 の両方に対して正しいポリシーを設定する必要があります ECR。例えば、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 へのアクセス用 ECR

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、現在の AWS Command Line Interface (AWS CLI) 認証情報が販売者の に属す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 Docker レジストリを使用して Docker ECR 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が構築されたこのガイドの前のセクションでは、`my-training-image`それぞれ `my-inference-image`と とタグ付けされました。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 にイメージリポジトリを作成します ECR。

```
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、次の部分で構成されます。 `<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、推論イメージを Amazon のリポジトリにプッシュします ECR。

```
sudo docker push ${fullname}
```

アップロードが完了すると、イメージは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、発行 AWS リージョン 元のを 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 リソースで使用されるコンテナイメージの Amazon ECR スキャン結果を公開する前に AWS Marketplace 確認します。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 リソースの作成](#)」に進む

Amazon SageMaker 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 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 SageMaker。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

トップでは、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 SageMaker リソースまたはアルゴリズムリソースのいずれかでリソースを作成する方法を示します。これには、検証を実行する SageMaker 方法を示す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、その方法については、[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 (Amazon S3) 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。のモデルパッケージの詳細については SageMaker、[「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 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 ECR で Amazon にアップロードした推論イメージは、同じリージョンにある必要があります。
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を指定します (例 : *my-model-package*) .
2. 推論イメージの場所には、Amazon にアップロードされた推論イメージ URI のを入力します ECR。は、[Amazon ECR コンソール](#) でイメージを検索 URI することで取得できます。
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. [Next (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。検証仕様では、検証を実行する SageMaker 方法について説明します。

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、[こ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、ロールの新しいロールを作成する を選択します IAM。Amazon SageMaker 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 からのイメージのプル ECR や Amazon S3 からのアーティファクトなどのアクションが含まれます。設定を確認し、ロールの作成 を選択します。ここでロールを作成すると、[Amazon SageMaker Full Access](#) IAM ポリシーで説明されているアクセス許可が、作成したロールに付与されます。

3. 検証プロファイルJSONで を編集します。許可される値の詳細については、「」を参照してください [TransformJobDefinition](#)。
 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: テストデータコンテンツタイプ (、 image/png 、またはその他の値など) application/json text/plainを指定します。SageMaker は実際の入力データを検証しません。この値は、Content-typeヘッダー値のコンテナ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 は Amazon から推論イメージをプルし ECR、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。「[AWS Marketplaceでの製品の公開](#)」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 に保存されている推論イメージ ECR
- Amazon に保存されているトレーニングイメージ ECR

- 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 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 ECR で Amazon 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は、同じリージョンにある必要があります。
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を入力します (例 : *my-algorithm*)。
2. トレーニングイメージ の場合、Amazon 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 場所を貼り付けます ECR。は、[Amazon ECR コンソール](#) でイメージを検索 URI することで取得できます。
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. [チャンネルの仕様] 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 つのチャンネルを追加します (入力ソースの最大数は 20 チャンネル)。詳細については、「[入力データ設定](#)」を参照してください。

5. [Next (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. [Next (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 の場合、Amazon にアップロードされた推論イメージURIの を貼り付けますECR。は、[Amazon ECRコンソール](#) でイメージを検索URIすることで取得できます。
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. [Next (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。検証仕様では、検証を実行する SageMaker 方法について説明します。

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証](#) 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、ロール の新しいロールを作成する を選択しますIAM。Amazon 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このロール SageMaker を使用します。これには、Amazon からイメー

ジをプルするECR、Amazon S3 にアーティファクトを保存する、Amazon S3 からトレーニングデータをコピーするなどのアクションが含まれます。設定を確認し、ロールの作成 を選択します。ここでロールを作成すると、[AmazonSageMakerFullAccessIAM](#)ポリシーで説明されているアクセス許可が、作成したロールに付与されます。

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配列で、トレーニング仕様ステップで指定したチャンネルごとに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. Transform 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、[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 は実際の入力データを検証しません。この値は、Content-typeヘッダー値のコンテナ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 API リファレンス [SplitType](#)」の「」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 は、Amazon からトレーニングイメージをプルしECR、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 から

推論イメージをプルし ECR、Amazon S3 から推論コンテナにアーティファクトをコピーし、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。「[AWS Marketplaceでの製品の公開](#)」に進みます。

AWS Marketplaceでの製品の公開

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、Amazon SageMaker リソースを作成したら、[AWS Marketplace](#) で機械学習製品を公開できます。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、製品の更新など、公開プロセスについて説明します。

トピック

- [前提条件](#)
- [公開プロセスの概要](#)
- [必要なアクセス許可](#)
- [製品リストの作成](#)
- [製品のテスト](#)
- [公開のサインオフ](#)
- [製品の更新](#)

前提条件

でモデルパッケージまたはアルゴリズムを発行する前に AWS Marketplace、以下が必要です。

- 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ァイル。
- 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 にアクセスするためのアクセス許可が必要です SageMaker。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。

公開プロセスの概要

公開プロセスには次の 4 つのステップがあります。

1. 製品の送信 - モデルパッケージまたはアルゴリズム製品の説明、使用情報、その他の詳細を記載したリストを作成します。製品を公開用に提出してから、ステータスが次のステップに変わるまでに約 1 時間かかります。
2. テスト製品 – AWS Marketplace 販売者として登録 AWS アカウント されている を使用して、のリストをプレビューし AWS Marketplace、サブスクライブして製品をテストします。さらに、許可されている他の AWS アカウント は、製品をプレビューおよびテストできます。変更が必要な場合は、戻ってリストの詳細を編集できます。
3. 発行のサインオフ – 製品が本番稼働する準備ができたなら、 に戻り AWS Marketplace 管理ポータル、サインオフを選択して を発行します。
4. 製品の公開開始 - 製品が AWS Marketplaceで公開されました。更新や製品修正を含む新しいバージョンを公開して、製品を維持できます。

必要なアクセス許可

Amazon SageMaker 製品を公開するには、としてサインインしている AWS Identity and Access Management ユーザーまたはロールに、次のいずれかまたは両方の IAM アクションが必要です。

- sagemaker:DescribeModelPackage – モデルパッケージを一覧表示する場合
- sagemaker:DescribeAlgorithm – アルゴリズムを一覧表示する場合

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。

製品リストの作成

以下は、モデルパッケージとアルゴリズム製品の両方 AWS Marketplace について、で製品リストを作成するためのチュートリアルです。

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。

このプロセスには以下のステップがあります。

ステップ

- [ステップ 1: 新しいリストを作成する](#)
- [ステップ 2: 一般的な製品情報を提供する](#)
- [ステップ 3: 起動オプションを追加する](#)
- [ステップ 4: 価格と条件を設定する](#)
- [ステップ 5: 製品を公開用に送信する](#)

ステップ 1: 新しいリストを作成する

新しい機械学習製品リストを作成するには

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. トップメニューで [製品]、[機械学習] の順に移動します。
3. [新しいリストを作成] を選択します。

Note

[新しい製品] ページの [製品概要] セクションでは、現在のステータス、プライバシー設定、製品タイプ、作成者、製品 ID を確認できます。

ステップ 2: 一般的な製品情報を提供する

一般的な製品情報を提供するには

1. [一般的な製品情報] セクションの [製品の説明] で、[追加] を選択します。
 - a. [製品の可視性] セクションでは、以下のいずれかのオプションを選択します。
 - パブリック – 製品は、最初はテスト AWS アカウント 用に限定された セットで利用できます。サインオフして公開すると、製品は一般に公開され、すべてのお客様がサブスクライブできるようになります。
 - プライベート – 製品は、AWS アカウント 指定した のみに表示されます。今後、この製品を公開することはできません。
 - b. [製品のタイトル]、[簡単な製品の説明]、[製品の概要]、[製品カテゴリ 1]、およびその他の詳細を入力します。これらの値は、後で変更できます。製品の説明については、「[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。
 - c. 完了したら [続行] を選択します。
2. [プロモーションリソース] には、製品ロゴ、検索キーワード、関連するリソースリンクを入力します。これらの値は、後で変更できます。
 - 完了したら [続行] を選択します。
3. [サポート情報] で、製品のサポートを提供するかどうかを選択します。
 - a. [はい] を選択した場合は、サポートと連絡先の詳細を入力してください。これらの値は後で変更できます。
 - b. 完了したら [続行] を選択します。
4. リージョンの可用性 では、製品を一覧表示 AWS リージョン する特定の を選択します。

デフォルト値は、[現在および将来サポートされるすべてのリージョンで利用可能にする] です。

 - 完了したら [続行] を選択します。

Note

公開用に下書きを送信した後は、この選択を変更することはできません。

製品を公開する次のステップでは、販売するモデルパッケージまたはアルゴリズムである起動オプションを提供します。

ステップ 3: 起動オプションを追加する

起動オプションを追加するには

1. 起動オプションセクションの「を入力」にARN、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) を入力します。

は、Amazon SageMaker コンソール [のモデルパッケージ](#) または [アルゴリズム](#) ページARNにあります。

Example ARN モデルパッケージの

```
arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>
```

Example ARN アルゴリズムの

```
arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>
```

2. [追加] を選択します。
3. 以下の手順は、モデルパッケージを公開するか、アルゴリズム製品を公開するかによって異なります。購入者向けのバージョン番号を除き、バージョンの詳細は後で変更できます。
 1. ステップ 1: バージョンの詳細と Git リポジトリリンクを入力し、バージョン番号、リリースノート、URLsをサンプル Jupyter ノートブックと GitHub リポジトリに提供します。
 2. アルゴリズム製品の場合のみ、[ステップ 2: トレーニングデータ入力の詳細の入力]で、トレーニングデータの説明、トレーニングアルゴリズムの概要の他、トレーニングデータリソースの例も記載してください。

アルゴリズムメトリクス、チャンネル仕様、ハイパーパラメータは、でアルゴリズムリソースを作成したときに指定した値に基づいて、製品の詳細ページに自動的に表示されます SageMaker。

以下の例は、セラーにはトレーニングデータ入力の詳細がどのように表示されるか、購入者にはトレーニングデータ入力の詳細がどのように表示されるかを示しています。

Example トレーニングデータ入力の例 - セラービュー

Enter details describing the training data inputs

Information to train a model

Describing the training data including an example training data resource along with an overview of training algorithm. Algorithm Metrics, Channel specification and Hyperparameters will be automatically displayed on the product detail page based on details in your algorithm container image. See examples [here](#). Note, if you have any details related to trained model input and outputs, please remove from here and enter in Steps 3 and 4.

Example input(s) for training job:
[\[Bank Marketing Dataset from UCI\]](#) (<https://archive.ics.uci.edu/ml/datasets/bank+marketing>)
 AutoGluon-Tabular requires no manual data preprocessing as long as your data is a valid CSV table
 Rows in your table represent different examples (data points), columns represent different variables (features).
 The first line of your CSV file should contain names for each column.
 Columns in your CSV file can be strings/text-fields/Numeric.
 Your data must contain the column that you identify as 'label' in your hyperparameter configuration.

572 of 600 characters used

Example トレーニングデータ入力の例 - 購入者ビュー

Usage Information

Training

Example input(s) for training job:
[Bank Marketing Dataset from UCI](#)
AutoGluon-Tabular requires no manual data preprocessing as long as your data is a valid CSV table
 Rows in your table represent different examples (data points), columns represent different variables (features).
 The first line of your CSV file should contain names for each column.
 Columns in your CSV file can be strings/text-fields/Numeric.
 Your data must contain the column that you identify as 'label' in your hyperparameter configuration.

[Show less](#)

次の例は、カスタム属性 (呼び出しパラメータ) がセラーにどのように表示されるか、カスタム属性 (呼び出しパラメータ) が購入者にどのように表示されるかを示しています。

Example カスタム属性 (呼び出しパラメータ) の例 - セラービュー

Custom attributes (invocation parameters) - optional Remove

Names
Short label for parameter data, can be comma delimited list

Maximum 100 characters

Description of parameter data
Brief summary of the parameter field

55 of 500 characters used

Parameter data type
Select one

Minimum Value - optional **Maximum Value - optional**

Maximum 9 characters Maximum 9 characters

Is this parameter variable always required?
 Yes
 No, it's optional (must enter a default value)

Default Value

Maximum 200 characters

Example カスタム属性 (呼び出しパラメータ) の例 - 購入者ビュー

▼ Custom attributes (invocation parameters)

Field name
threshold

Description
Threshold of the confidence score of detected objects

Data type Continuous (Float)	Range Min: 0.0 Max: 1.0	Required No
---------------------------------	-------------------------------	----------------

Default value
0.3

3. ステップ 3: 入力の詳細 を入力し、サンプル入力ファイルのモデルまたはアルゴリズムの入力の詳細 と URLs を指定します。

以下の例は、セラーにはモデルデータ入力の詳細がどのように表示されるか、購入者にはモデルデータ入力の詳細がどのように表示されるかを示しています。

Example モデルデータ入力の例 - セラービュー

Model input details
Help customers understand your model capabilities by providing details related to model input (summary, limitations, mime types, and sample data for realtime and batch invocation) and input data descriptions (required for text/csv and application/json mime types). See examples [here](#).

Model input summary
Describe the model input format specification in text.

B I H | </> | ≡ | ≡ | % | 🔍 | 🗑️

This model can analyze images that are supplied as image bytes or stored in an Amazon S3 bucket.

96 of 500 characters used

Limitation for input type - optional
Define any limitations on the input data, such as file size and/or image aspect ratio

The minimum size is 80 pixels for both height and width. The image can be no larger than 1024X1024 pixels, otherwise the detection performance may degrade dramatically. Note, that images with aspect ratio close to 1.0 are best.

227 of 300 characters used

Input mime types
Select all the input data formats permitted

- Select multiple -

image/bmp X image/png X image/jpeg X application/x-image X

Choose to provide raw text or URL for realtime sample input data

URL
 Text

Realtime: Sample input data

https://github.com/zreshold/gluoncv-sagemaker-examples/blob/master/example/input/playground.jpg

Maximum 150 characters

Batch job: Sample input data
Provide URL to a folder that shows an example of a batch input that supports multiple records

https://github.com/zreshold/gluoncv-sagemaker-examples/blob/master/example/input/playground.jpg

Maximum 150 characters

Example モデルデータ入力の例 - 購入者ビュー

Input

Summary

This model can analyze images that are supplied as image bytes or stored in an Amazon S3 bucket.

Limitations for input type

The minimum size is 80 pixels for both height and width. The image can be no larger than 1024X1024 pixels, otherwise the detection performance may degrade dramatically. Note, that images with aspect ratio close to 1.0 are best.

Input MIME type

image/bmp, image/png, image/jpeg

Sample input data

[view data](#)

- ステップ 4: 出力の詳細を入力するには、モデルまたはアルゴリズムの出力の詳細とサンプル出力をテキストまたはとして指定しますURLs。

使用に関する情報については、「[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。

次の例は、モデルデータ出力の詳細がセラーにどのように表示されるか、モデルデータ出力の詳細が購入者にどのように表示されるかを示しています。

Example モデルデータ出力の例 - セラービュー

Model output details

Help customers understand your model capabilities by providing details related to model output (summary, limitations, mime types, and sample output data for realtime and batch invocation) and output data descriptions (required for text/csv and application/json mime types). See examples [here](#).

Model output summary

Describe the model output format specification in text.

B I H | </> | ≡ | ≡ | % | 🗑️ | ⓘ

The model detects instances of common objects such as computer, in an image. The response includes an array of detected object labels (id field) with bounding box pixel coordinates, and an associated level of confidence.

220 of 500 characters used

Limitation for output type - optional

Define any limitations on the output data

Maximum file size is...

0 of 300 characters used

Output mime types

Select one or more output formats

- Select multiple - ▾

application/json ×

Choose to provide raw text or URL for realtime sample output data

URL

Text

Realtime: Sample output data

Ensure the sample output data corresponds to the input sample data

```
[{"right":603,"bottom":528,"top":177,"score":0.9921523332595825,"id":"person","left":439},
{"right":687,"bottom":539,"top":184,"score":0.9885265231132507,"id":"person","left":577},
{"right":611,"bottom":246,"top":228,"score":0.21156705915927887,"id":"cell phone","left":598}]
```

273 of 1000 characters used

Batch job: Sample output data

Provide a link to a file or folder and ensure example data corresponds to the input sample data

<https://github.com/zhrshold/gluoncv-sagemaker-examples/blob/master/example/input/playground.jpg>

Maximum 150 characters

Example モデルデータ出力の例 - 購入者ビュー

Output

Summary

The model detects instances of common objects such as computer, in an image. The response includes an array of detected object labels (id field) with bounding box pixel coordinates, and an associated level of confidence.

Output MIME type

application/json

Sample output data

```
[
  {
    "right": 603,
    "bottom": 528,
    "top": 177,
    "score": 0.9921523332595825,
    "id": "person",
    "left": 439
  },
  {
    "right": 687,
    "bottom": 539,
    "top": 184,
    "score": 0.9885265231132507,
    "id": "person",
    "left": 577
  },
  {
    "right": 611,
    "bottom": 246,
    "top": 228,
    "score": 0.21156705915927887,
    "id": "cell phone",
    "left": 598
  }
]
```

- [ステップ 5: サポートされているインスタンスを確認して作成する] では、推奨インスタンスを設定します。

- これがモデルパッケージ製品の場合は、バッチ変換とリアルタイムデプロイの両方で、サポートされているインスタンスから推奨インスタンスタイプを選択してください。
- アルゴリズム製品の場合は、推奨インスタンスタイプトレーニングジョブも選択してください。

モデルパッケージやアルゴリズムリソースがサポートしていないインスタンスタイプは選択できません。サポートされているインスタンスタイプは、Amazon でこれらのリソースを作成したときに選択されました SageMaker。

4. 完了したら [続行] を選択します。

Note

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。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。

製品を公開する次のステップでは、価格と条件を設定します。

ステップ 4: 価格と条件を設定する

価格と条件を設定するには

1. [価格と条件] セクションで、[オファーを追加] を選択します。
2. [料金表] を設定します。

ソフトウェアを無料で提供したり、有料価格を設定したり、無料試用期間を有効にしたりできます。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。

3. エンドユーザーライセンス契約 () として使用するプレーンテキストファイルをアップロードしますEULA。
4. [保存して閉じる] を選択します。

製品に関するすべての情報を入力しました。次のステップでは、製品をテストできるように、限定販売で公開します。

ステップ 5: 製品を公開用に送信する

製品を公開用に送信するには

1. 新製品ページで、公開用に送信セクションの「追加のテストアカウント – オプション」の「」で、追加のテスターに 1 つ以上の AWS アカウント IDs を入力します。
2. [公開用に送信] を選択します。

これにより、ユーザー (およびオプションのテスター) AWS Marketplace がサブスクライブしてテストに使用できるプレビューリストを作成して、公開プロセスが開始されます。

これで、製品をテストする準備が整いました。機械学習製品のテストの詳細については、「[製品のテスト](#)」を参照してください。

製品をテストした後、変更が必要な場合は上記の手順をやり直しできます。購入者に製品を提供する準備が整ったら、[公開を承認](#)できます。

製品のテスト

製品を最初に送信してから、プレビューリストの準備が整うまでに約 1 時間かかります。ステータスがテスト製品に変更されると、販売者アカウントやその他の許可リストに登録されている AWS アカウント アカウントは、のリストをプレビューし AWS Marketplace、製品をサブスクライブしてテストできます。

出品情報のプレビューを見るには

1. で AWS Marketplace 管理ポータル、製品概要ページに移動します。
2. [ステージングされた製品に移動] を選択します。
3. 変更を加える場合は、[製品の編集] を選択し、「[製品リストの作成](#)」と同じ手順に従ってください。
4. すべての購入者が閲覧できるように製品を公開する準備が整ったら、「[公開のサインオフ](#)」の手順に従ってください。

パブリッシュ前に製品をテスト AWS アカウント するために他の を追加するには、[AWS Marketplace 販売者オペレーションチーム](#)に連絡し、を指定します AWS アカウント IDs。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。

公開のサインオフ

このステップは、説明、価格、使用情報を記入し、製品をテストした後に行います。

公開をサインオフするには

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. トップメニューで [製品]、[機械学習] の順に移動します。
3. 製品の [製品の概要] に移動します。
4. [サインオフして公開] を選択します。

製品の更新

の [Machine Learning リスト](#) ページを使用して AWS Marketplace 管理ポータル、次の方法でモデルパッケージまたはアルゴリズム製品を更新できます。

- [新しいバージョンの追加](#) - 新しいモデルパッケージまたはアルゴリズムリソースを既存の製品の新しいバージョンとして追加できます。
- [バージョンの制限](#) - 既存の製品の以前のバージョンを制限できます。
- [製品の削除](#) - 製品全体を削除できます。

新しいバージョンを追加する

モデルパッケージまたはアルゴリズムリソースの新しいバージョンを追加するには

1. AWS Marketplace 管理ポータルの「[Machine Learning Listings](#)」ページに移動します。
2. 既存の製品の [製品の概要] に移動します。
3. [製品の編集] を選択します。
4. [起動オプション] で [編集] を選択します。
5. リソースARNの を追加するには、バージョンページに移動し、新しいバージョンの追加 を選択します。

起動オプションの追加方法の詳細については、「[製品リストの作成](#)」を参照してください。

Note

使用情報は製品バージョンごとに異なります。新しいバージョンに使用情報を追加する場合は、引き続き「[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](#)」に従ってください。

購入者が AWS Marketplace リストから製品を起動すると、異なるバージョンを選択できます。購入者が Amazon SageMaker コンソールから製品を起動すると、最新バージョンのみが表示されます。

バージョンを制限する

モデルパッケージまたはアルゴリズムリソースのバージョンを制限するには

1. 既存の製品の [製品の概要] に移動します。
2. [製品の編集] を選択します。
3. [起動オプション] で [編集] を選択します。
4. [バージョン] ページで [バージョンを制限] を選択します。
5. [製品の概要] に戻り、[公開用に送信] を選択します。

Note

既に製品をサブスクライブしている購入者は、制限付きバージョンのモデルパッケージまたはアルゴリズムを引き続き使用できます。ただし、新規購入者にはそれらの制限付きバージョンをオプションとして表示することはできません。

製品を削除する

製品を削除するには

1. AWS Marketplace 管理ポータル「[Machine Learning Listings](#)」ページにある公開製品のリストに移動します。
2. 削除する製品を選択し、[アクション] ドロップダウンリストで [リストを非公開にする] を選択します。
3. リクエストに関して AWS Marketplace 担当者から連絡があった場合に備えて、E メールアドレスと出品を削除する理由を記入してください。

Note

から製品を削除すると AWS Marketplace、新しい購入者は製品をサブスクライブできなくなります。ただし、既存の購入者は製品を使用し続けられ、これは最低 90 日サポートされる必要があります。非公開の出品情報を別の製品に置き換える予定がある場合は、削除リクエストの詳細に新しい出品情報を明記してください。

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、機械学習 (ML) 製品リストを作成するための要件と 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

公開された製品がこれらの要件を満たしていない場合は、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

- [必須アセット](#)
- [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](#)

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) – AWS リージョン 発行元の にモデルパッケージまたはアルゴリズムリソースARNの を指定します (「」を参照[公開 AWS リージョン でサポート](#))。

- モデルパッケージARNの には、次の形式があります。
arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>
- アルゴリズムARNの には、次の形式があります。
arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>
- [the section called “使用状況情報の要件”](#) - 入力、出力、および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のいずれかを提供します。
- [the section called “Jupyter Notebook の要件”](#) - 製品の使用方法をすべてデモンストレーションします。

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、非公開で製品を作成することもできます。詳細については、「[製品リストの作成](#)」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。
- (オプション) モデルパッケージ製品について、商品一覧ページでリアルタイムの製品デモを有効にする場合は、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。製品デモでは、見込み客が自分でモデルをサブスクライブしたりデプロイしたりしなくても、出品ページでモデルを直接試すことができます。

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。

初めて公開する新製品の使用情報を追加するには、AWS Marketplace 管理ポータル コンソールにサインインします。[製品] ドロップダウンから [機械学習] を選択します。製品を選択します。起動オプションの製品概要で、モデルパッケージまたはアルゴリズムリソースARNのを指定し、の追加を選択します。

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、[起動オプション] の [編集] を選択して、[バージョンを編集] を選択します。

入力と出力の要件

バイヤーが製品を理解して使用できるよう、入力と出力の例を含め、形式を明確に説明することが重要です。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 リソースを製品リストに追加すると、次のメッセージが表示されます。

推論の入力と出力

推論入力では、リアルタイムエンドポイントとバッチ変換ジョブの両方の入力形式を指定します。データの前処理に必要なコードスニペットを含めてください。サポートされているMIMEコンテンツタイプ (例: image/jpeg、image/png、image/bmp)、該当する場合は値の説明、および制限を含めます。でホストされている入力サンプルを含めます [GitHub](#)。

推論出力では、リアルタイムエンドポイントとバッチ変換ジョブの両方の出力形式を指定します。必要に応じて、出力MIMEコンテンツタイプ (application/json、image/jpeg など) と値の説明を含めます。でホストされている出力サンプルを含めます [GitHub](#)。

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。サポートされているMIMEコンテンツタイプ (例: image/jpeg

、image/png、image/bmp)、該当する場合は値の説明、および制限を含めます。でホストされている入力サンプルを必ず含めてください[GitHub](#)。

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。[分散トレーニング](#) (1 CPUインスタンスあたりGPU 1つ以上のトレーニング) がサポートされている場合は、これを指定します。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 リソースを製品リストに追加するときは、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す でホスト[GitHub](#)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。

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、「」の「[モデルパッケージのリストとサンプルノートブック](#)」を参照してください [GitHub](#)。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。ノートブックはすべてので機能します。パラメータを入力する必要はなく AWS リージョン、購入者がサンプルデータを見つける必要もありません。

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は、完全なトレーニング、チューニング、モデル作成、リアルタイム推論用の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を示します。詳細については、「」の「[アルゴリズムのリストとサンプルノートブック](#)」を参照してください [GitHub](#)。サンプルノートブックについては、「」の「[amazon_demo_product](#)」と「[automl](#)」を参照してください [GitHub](#)。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		
サポートされているコンテンツタイプ (例: 画像で X を検出する) に対して製品が何をするかを詳しく説明します。	必須	必須
製品に関する説得力のある差別化情報を提供します (「best」や根拠のないクレームなどの形容詞は避けてください)。	推奨	推奨
この製品の最も重要なユースケースを一覧表示します (複数可)。	必須	必須
トレーニングしたデータ (ソースとサイズ) を説明し、既知の制限があればリストします。	必須	該当しない
モデルが構築されたコアフレームワークについて説明します。	推奨	推奨

詳細	モデルパッケージリスト用	アルゴリズムリスト用
<p>検証データに関するモデルパフォーマンスメトリクスを要約する (Z データセットを使用してベンチマークされた「XX.YY パーセントの精度」など)。</p>	必須	該当しない
<p>推奨されるインスタンスタイプのモデルレイテンシーおよび/またはスループットメトリクスを要約します。</p>	必須	該当しない
<p>アルゴリズムカテゴリについて説明します。例えば、この決定フォレスト回帰アルゴリズムは、ブートストラップ集約の一般的な手法とランダムな特徴の選択を使用して構築された木構造化分類子のアンサンブルに基づいています。</p>	該当しない	必須
Usage information		
<p>推論には、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の入力形式を指定します。サポートされているMIMEコンテンツタイプ (例: image/jpeg、image/png、image/bmp)、該当する場合は値の説明、および制限を含めます。「入力と出力の要件」を参照してください。</p>	必須	必須

詳細	モデルパッケージリスト用	アルゴリズムリスト用
推論には、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の入力サンプルを指定します。サンプルはでホストする必要があります GitHub。「 入力と出力の要件 」を参照してください。	必須	必須
推論には、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の出力形式を指定します。必要に応じて、出力MIMEコンテンツタイプ (application/json、image/jpeg など) と値の説明を含めます。「 入力と出力の要件 」を参照してください。	必須	必須
推論には、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の出力サンプルを指定します。サンプルはでホストする必要があります GitHub。「 入力と出力の要件 」を参照してください。	必須	必須
推論には、エンドポイントまたはバッチ変換ジョブを使用する例を指定します。 AWS Command Line Interface (AWS CLI) コマンドまたはを使用してコード例を含め ず AWS SDK。	必須	必須

詳細	モデルパッケージリスト用	アルゴリズムリスト用
<p>トレーニングには、入力形式を指定します。サポートされているMIMEコンテンツタイプ (image/jpeg、image/png、image/bmp など)、該当する場合は値の説明、制限 (必要なデータの最小行など) を含めます。「入力と出力の要件」を参照してください。</p>	該当しない	必須
<p>トレーニングでは、でホストされている入力サンプルを提供します GitHub。「入力と出力の要件」を参照してください。</p>	該当しない	必須
<p>トレーニングでは、トレーニングジョブの実行例を示します。サポートされているハイパーパラメータ、その範囲、および全体的な影響について説明します。アルゴリズムがハイパーパラメータチューニング、分散トレーニング、またはGPUインスタンスをサポートしているかどうかを指定します。AWS CLI コマンドやの使用などのコード例を含めます AWS SDK。</p>	該当しない	必須
<p>製品の完全な使用 GitHub を示すためにホストされている Jupyter ノートブックを提供します。「Jupyter Notebook の要件」を参照してください。</p>	必須	必須

詳細	モデルパッケージリスト用	アルゴリズムリスト用
ユーザーマニュアルやサンプルデータなど、製品の使用に関連する技術情報を提供します。	推奨	推奨

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 イメージサイズは、Amazon Elastic Container Registry (Amazon ECR) [サービスクォータ](#) 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 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 SageMaker 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 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 デベロッパーガイド」の [「ホストインスタンスストレージボリューム」](#) を参照してください。

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 では、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を選択できます。インスタンスタイプは、CPU、GPUメモリ、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 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 Service Quotas」](#)を参照してください。

非同期推論

で発行されたモデルパッケージとアルゴリズムは、[Amazon SageMaker 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 SageMaker Serverless Inference](#)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。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

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、MaxWaitTimeInSecondsの値は3,600秒(60分)に設定されます。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、イメージを販売 AWS アカウント 者が所有する 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発行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 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 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 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 SageMaker
- Amazon ECRリポジトリにアップロードされた推論イメージとトレーニングイメージ
- Amazon Simple Storage Service (Amazon S3) 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

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 SageMaker。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。

リストプロセス中に、AWS リージョン 発行元の に関係なく、発行先のリージョンを選択し、製品を使用可能にすることができます。

機械学習製品発行のエラー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 エラーになります。AWS Marketplace 管理ポータル

Amazon SageMaker 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公開するプロセスの最終ページでは、このアルゴリズムを公開 AWS Marketplace するには `はい` を選択する必要があります。AWS Marketplace[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。

を使用して AWS SDK [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 `true` に設定されていることを確認します。

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、ARN に新しい `はい` を追加します AWS Marketplace 管理ポータル。

全般: モデルパッケージまたはアルゴリズムARNの `はい` を追加すると、404 エラーが発生します。
AWS Marketplace 管理ポータル

このエラーは、さまざまな理由で発生します。

- `はい` が無効であるARN可能性があります。正しい `はい` を使用していることを確認しますARN。
 - モデルパッケージの場合、`はい` に似ているARNsは `arn:aws:sagemaker:us-east-2:000123456789:model-package/my-model-package-name` です。
 - アルゴリズムの場合、`はい` に似ているARNsは `arn:aws:sagemaker:us-east-2:000123456789:algorithm/my-algorithm` です。
- 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。
- 公開に使用するユーザーまたはロールには、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正しいIAMアクセス許可がありません。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。

全般: `はい` でアルゴリズム製品の料金を指定すると 500 エラーになります。AWS Marketplace 管理ポータル

このエラーは、トレーニングイメージのみで、それに付随する推論イメージがないアルゴリズムリソースを公開しようとした場合に発生することがあります。で公開されるアルゴリズムリソースには、両方のコンポーネント AWS Marketplace が必要です。詳細については、「[で製品を準備する SageMaker](#)」を参照してください。

Amazon SageMaker: 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」の失敗メッセージが表示される

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 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可能性があります AWS アカウント。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。イメージを、公開 AWS アカウント に使用する が所有する Amazon ECR リポジトリにコピーします。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: 「Not Started」と「Client error: No scan scheduled...」を取得します。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

このエラーは、SageMaker が Amazon 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 ECR。この場合、[Amazon ECR コンソール](#) を開き、イメージがアップロードされたリポジトリを検索し、イメージを選択してから、スキャン を選択します。

で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 製品のレポートを生成します。すべてのレポートは、AWS Marketplace 管理ポータル [レポートページの](#) 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、機械学習製品の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。購入者は匿名の一意のカスタマーリファレンス ID によって識別されます。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 [での販売者の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。

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。

APIs 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「[Sell your Gateway APIs through AWS Marketplace](#)」を参照してください。API

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 製品の使用を開始する 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ターの作成 AWS Marketplace](#)
- [による SaaS 製品への顧客のオンボーディング AWS Marketplace](#)
- [SaaS 製品の Amazon SNS通知](#)
- [Metering and Entitlement Service へのアクセス AWS Marketplace APIs](#)
- [での SaaS 製品のレポート AWS Marketplace](#)
- [SaaS 製品統合のコード例](#)
- [VPC を使用して Amazon 経由で製品を配信する AWS PrivateLink](#)

での SaaS 製品の使用を開始する AWS Marketplace

AWS Marketplace 販売者として、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から始めて、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、で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 製品設定の設定 AWS Marketplace](#)
- [SaaS サブスクリプション製品を と統合する AWS Marketplace](#)
- [SaaS 契約製品を と統合する AWS Marketplace](#)
- [との SaaS 契約と pay-as-you-go製品の統合 AWS Marketplace](#)
- [でのサーバーレス SaaS 統合ソリューションのデプロイ AWS Marketplace](#)

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、最初に可視性が制限されて公開され、アカウントのみがそれにアクセスできます。準備ができたら、AWS Marketplace カタログに発行して、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 製品の作成の詳細については、[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マーケットプレイス販売者オペレーションチーム](#)にお問い合わせください。
- パブリック - 購入者が商品を見つけ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なら、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 販売者オペレーションチームが 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 販売者として、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、以下の幅広い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[契約を pay-as-you-go](#) 製品と統合します AWS Marketplace。
- [サブスクリプション](#)、[契約](#)、または 製品と [pay-as-you-go](#) の統合契約をテストします AWS Marketplace。
- 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

で販売する 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、との契約の 3 つのオファータイプがあります pay-as-you-go。オファータイプの選択は、SaaS 製品を と統合する方法に影響します 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

製品の送信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URLしている製品のロゴの明確なイメージを含む、パブリックにアクセス可能な Amazon S3。
- エンドユーザーライセンス契約 (EULA) URL – 製品には、PDF ファイルとして利用可能な EULA が必要です。顧客が EULA 製品の AWS Marketplace ページで を確認できる Amazon S3 バケットへのリンクを指定する必要があります。
- 製品登録 URL – これは、 で製品を正常にサブスクライブした後に URL 購入者がリダイレクトされる です AWS Marketplace。
- 製品に関するメタデータ - AWS Marketplace 管理ポータルでの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ンネルURLsの E メールアドレスとが含まれます。

4. 製品を統合に送信する

販売者アカウントと AWS Marketplace Management Portal を使用して、AWS Marketplace は製品を限定製品として公開します [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) のトピックは、製品概要ページで確認できます。

Note

お客様と AWS Marketplace 販売者オペレーションチームが大きなコストをかけずに製品をテストできるように、製品は割引価格で維持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、製品に使用するオファertype AWS Marketplace によって異なります。オファertype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](#)
- [統合との契約 pay-as-you-go](#)

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テストすることです。

6. 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。AWS Marketplace Seller Operations チームは、可視性を Public に更新する前に、製品を確認し、料金を更新します。

Note

AWS Marketplace 販売者オペレーションは、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。このプロセスは、公開された可視性を更新するのに 7~10 営業日かかり、チームがエラーを見つけた場合はより長くなります。タイミングの詳細については、このガイドの [「タイミングと期待値」](#) 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。その後、製品情報、製品デプロイの詳細、公開オファ어의詳細を追加することもできます。必要に応じて、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。詳細については、次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

テスト中は、デフォルトで1ディメンションあたり 0.01 USD に設定されます。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. [送信] を選択します。次に、 が情報 AWS Marketplace を検証します。検証が成功すると、 は製品を制限付きステータスで 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 に使用されると AWS Marketplace、販売者オペレーションチームのテストアカウントのみがアクセスできます。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 製品設定の設定 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、料金の詳細の更新、製品の可視性、その他の設定など、製品設定を変更する方法を示します。

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](#)
- [の許可リストを更新する AWS アカウント IDs](#)
- [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](#)
- [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](#)
- [エンドユーザーライセンス契約を更新する \(EULA \)](#)

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、AWS Marketplace 管理ポータル [リクエストタブ](#)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. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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. [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. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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. [\[SaaS 製品\]](#) 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 URL](#)
- [\[ハイライト\]](#)
- [\[製品カテゴリ\]](#)
- [キーワード](#)
- [製品ビデオ 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。

の許可リストを更新する AWS アカウント IDs

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. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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. [\[SaaS 製品\]](#) 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. Allowlisted AWS アカウント フィールドに を入力し、カンマで AWS アカウント IDs区切ります。
5. の許可リストを更新するには AWS アカウント IDs、「送信」を選択します。

製品の可視性を更新する

で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、可視性の更新 を使用できます。

1. [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. [\[SaaS 製品\]](#) 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 に移動するには、AWS Marketplace 販売者オペレーション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 製品の料金をディメンションごとに変更する場合は、[\[料金条件を更新\]](#) を使用できます。

Note

いずれかのディメンションの料金を引き上げると、少なくとも今後 90 日間は料金更新オプションを利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. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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. **[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. [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. **[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識別子、表示名、説明を指定して、新しいディメンションを製品に追加し、**次へ** を選択します。

Note

API 識別子と名前は、すべてのディメンションで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識別子と単位を変更することはできません。

- 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。

- [送信] を選択して、リクエストを送信します。
- [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。

- [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。
- [\[SaaS 製品\]](#) タブから、変更する製品を選択します。
- [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。
- 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。
- 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。

Note

ディメンション名は一意である必要があります。

- [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. [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. [\[SaaS 製品\]](#) 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。

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”](#) – お客様は、 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URLに を使用します AWS Marketplace。
- [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URLに使用される を更新するには、フルフィルメントオプションの更新タブを使用します。

1. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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. [\[SaaS 製品\]](#) ページの [SaaS 製品] タブで、変更する製品を選択します。
3. リクエストの変更ドロップダウンリストから、パブリックオファターの更新 を選択し、デフォルトのフルフィルメントの編集 URL を選択します。

- フルフィルメントURLフィールドに、SaaS URL 製品フルフィルメントオプションの新しい を入力します。
- 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。
- [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、step-by-step AWS CloudFormation テンプレートを使用して、購入者にガイド付き、手順、リソースデプロイを提供できます。購入者は CloudFormation テンプレートを使用して製品を設定および起動します。

Tip

クイック起動設定プロセスの詳細については、[SaaS クイック起動を有効にする](#) 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。


- [\[SaaS 製品\]](#) 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。


- 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。
- [クイック起動] で、[有効化と設定] ボタンを選択します。
- アカウントログインの詳細 URL については、購入者がログインまたはアカウントを作成できるサイトの を指定します。これにより、購入者エクスペリエンスに新しいタブURL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] 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。
 1. タイトル – CloudFormation デプロイの名前を指定します。
 2. [説明] - テンプレートの説明を入力します。
 3. [スタック名] -スタックの名前を指定します。この名前は、 の購入者のスタック名です CloudFormation。
 4. CloudFormation テンプレート URL – URL テンプレートに Amazon Simple Storage Service (Amazon S3) を提供します。AWS はこのテンプレートを確認し、最終的なテンプレート AWS を提供しますURL。

 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ご質問は、AWS Marketplace 事業開発パートナーまたは [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。

5. 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 IDs](#)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、「[」を参照してください PutDeploymentParameter](#) AWS Marketplace Deployment Service APIリファレンスの「」。

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. 起動の詳細については、CloudFormation スタックのデプロイ後に購入者URLが製品にアクセスする を指定します。
8. (オプション) Quick Launch の Allowlisted アカウントには、可視性が制限され AWS アカウント また Quick Launch 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、[設定と起動] ページを使用して設定をテストできます。
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 販売者オペレーションチームが設定を確認し、購入者テストを実施し、エクスペリエンスを発行します。

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. [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. [\[SaaS 製品\]](#) 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。
 - 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. [managementhttps://aws.amazon.com/marketplace//tour/](https://aws.amazon.com/marketplace//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. [\[SaaS 製品\]](#) 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。

エンドユーザーライセンス契約を更新する (EULA)

製品をサブスクライブしているEULA新規ユーザーの を更新できます。

1. [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. [\[SaaS 製品\]](#) タブから、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から、公開オファーの更新 を選択し、 を更新しますEULA。
4. [Standard Contract for AWS Marketplace \(SCMP \)](#) を選択するか、カスタム を送信できます EULA。カスタム の場合EULA、URL契約の Amazon Simple Storage Service (Amazon S3) を指定する必要があります。

Note

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 AWS Marketplace 、 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) 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、複数の顧客シナリオに正常に対応できることを示す必要があります。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、 [で software-as-a-service \(SaaS \) 製品に適した料金モデルが選択されていることを確認してください AWS Marketplace](#)。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、一時的な `x-amzn-marketplace-token` トークンを持つ HTTPPOST リクエスト URL である登録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. [ResolveCustomer](#) API オペレーションを呼び出し `ProductCode` で `CustomerIdentifier`、トークンを `CustomerAWSAccountId`、および と交換します AWS Marketplace Metering Service。

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使用するための計測の設定](#) 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 販売者オペレーション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の提供や取り消しなど、顧客サブスクリプションの変更に関する通知を受け取ります。

Note

Amazon SNS トピックの Amazon リソースネーム (ARN) は のようになり
ます `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-
notification-<product code>`。

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 経由で受信した通知から顧客の最新のステータスを保存SQSして、顧客がアクセスできるかどうかを確認します。

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、統合が成功するように詳細なテストを行う必要があります。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. 許可されたアカウントをサブスクライブしたら、アカウントが登録にリダイレクトされURL、リダイレクトが一時的なトークンを含む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。

- 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。
- テストカスタマーがオンボーディングされたら、 の BatchMeterUsageAPI オペレーションを使用して、AWS 課金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。
- 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。
- サブスクリプションの成功を確認します。サブスクリプションに成功したメッセージを含むテストアカウントの Amazon SNS 通知を受け取ったら、計測を開始できます。Amazon SNS 通知を受信する AWS Marketplace Metering Service 前に に送信される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完了し、ソリューションをテストしたら、AWS Marketplace 販売者オペレーション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することで、ソリューションに対して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、製品をパブリックに一覧表示できます 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 契約製品を と統合する 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、複数の顧客シナリオに正常に対応できることを示す必要があります。以下のセクションでは、これらのシナリオについて説明し、対応方法を説明し、統合のテストの概要を示します。

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、登録にリダイレクトされます。登録はURL、一時的なx-amzn-marketplace-tokenトークンを持つHTTPPOST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. で [ResolveCustomer](#) APIオペレーションを呼び出すProductCodeこと
でCustomerIdentifier、トークンをCustomerAWSAccountId、およびと交換します
AWS Marketplace Metering Service。
2. 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 AWS Marketplace Entitlement Service。
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。
- カスタマー ID があり、GetEntitlementsAPIオペレーションは適切なエンタイトルメントを返します。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 はありますが、GetEntitlementsAPIオペレーションはエンタイトルメントを返さないか、リクエストを満たすのに十分な数量を返しません。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、製品の作成時にオペレーションチームから AWS Marketplace 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、特定の顧客に対するアクセスの提供や取り消しなど、顧客権限の変更に関する通知を受け取ります。

Note

SNS トピック Amazon リソースネーム (ARN) は のようになります `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>`。

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ました。新しいステータスを確認するには、GetEntitlements API オペレーションを呼び出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、「[を使用したエンタイトルメントの確認 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、統合が成功するように詳細なテストを行う必要があります。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. アカウントが契約を取得したら、アカウントが登録にリダイレクトされURL、リダイレクトが一時的なトークンを含む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. オンボーディング中または継続的な検証パスのいずれかで、GetEntitlementsAPI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、アプリケーションは、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正しく管理する必要があります。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完了し、ソリューションをテストしたら、AWS Marketplace オペレーションチームに通知します。次に、GetEntitlementsAPIオペレーションを正常に呼び出し、新規顧客を十分にオンボーディングしたことを検証して、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、製品をパブリックに一覧表示できます 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 “AWS Marketplace 製品の返金プロセス”](#)」を参照してください。

との SaaS 契約と pay-as-you-go製品の統合 AWS Marketplace

製品をと統合することは AWS Marketplace、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、複数の顧客シナリオに正常に対応できることを示す必要があります。以下のセクションでは、これらのシナリオについて説明し、それらにどのように対応するかを説明し、統合のテストの概要を示します。

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](#)
- [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

お客様が製品をサブスクライブすると、登録にリダイレクトされます。登録は URL 一時的な x-amzn-marketplace-token トークンを持つ HTTPPOST 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. で [ResolveCustomer](#) API オペレーションを呼び出し ProductCode で CustomerIdentifier、トークンを CustomerAWSAccountId、およびと交換します AWS Marketplace Metering Service。

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。
- カスタマー ID があり、GetEntitlementsAPIオペレーションは適切なエンタイトルメントを返します。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 はありますが、GetEntitlementsAPIオペレーションはエンタイトルメントを返さないか、リクエストを満たすのに十分な数量を返しません。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使用するための計測の設定](#) 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) キューをセットアップし、製品の Amazon SNS トピックをサブスクライブします。2つのSNSトピックがあり、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 販売者オペレーション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の提供や取り消しなど、顧客サブスクリプションの変更に関する通知を受け取ります。

Note

サブスクリプション変更の SNS トピック Amazon リソースネーム (ARN) は、のようになります `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>`。権限変更 ARN の SNS トピックは のようになります `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>`。

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 トピックで) – 顧客のエンタイトルメントが変更されました。新しいステータスを確認するには、`GetEntitlements` API オペレーションを呼び出す必要があります。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) – 顧客はサブスクライブされており、顧客 ID に対して正常に計測できます。
- `unsubscribe-pending` (サブスクリプション SNS トピックで) – お客様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) – お客様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) – カスタマー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、「[を使用したエンタイトルメントの確認 AWS Marketplace Entitlement Service](#)」を参照してください。

SaaS 契約製品統合のテスト

契約を pay-as-you-go 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行う必要があります。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詳しくは、[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. アカウントが契約を取得したら、アカウントが登録にリダイレクトされ URL、リダイレクトが一時的なトークンを含む 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. オンボーディング中または継続的な検証パスのいずれかで、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、アプリケーションは、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正しく管理する必要があります。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 BatchMeterUsageAPI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 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。
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 販売者オペレーションチームに通知します。次に、GetEntitlementsAPIオペレーションを正常に呼び出し、新規顧客を十分にオンボーディングしたことを検証して、ソリューションをテストします。また、BatchMeterUsageAPI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も検証されます。

統合とテストが完了したら、最終レビューを実行し、製品をパブリックに一覧表示できます 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。

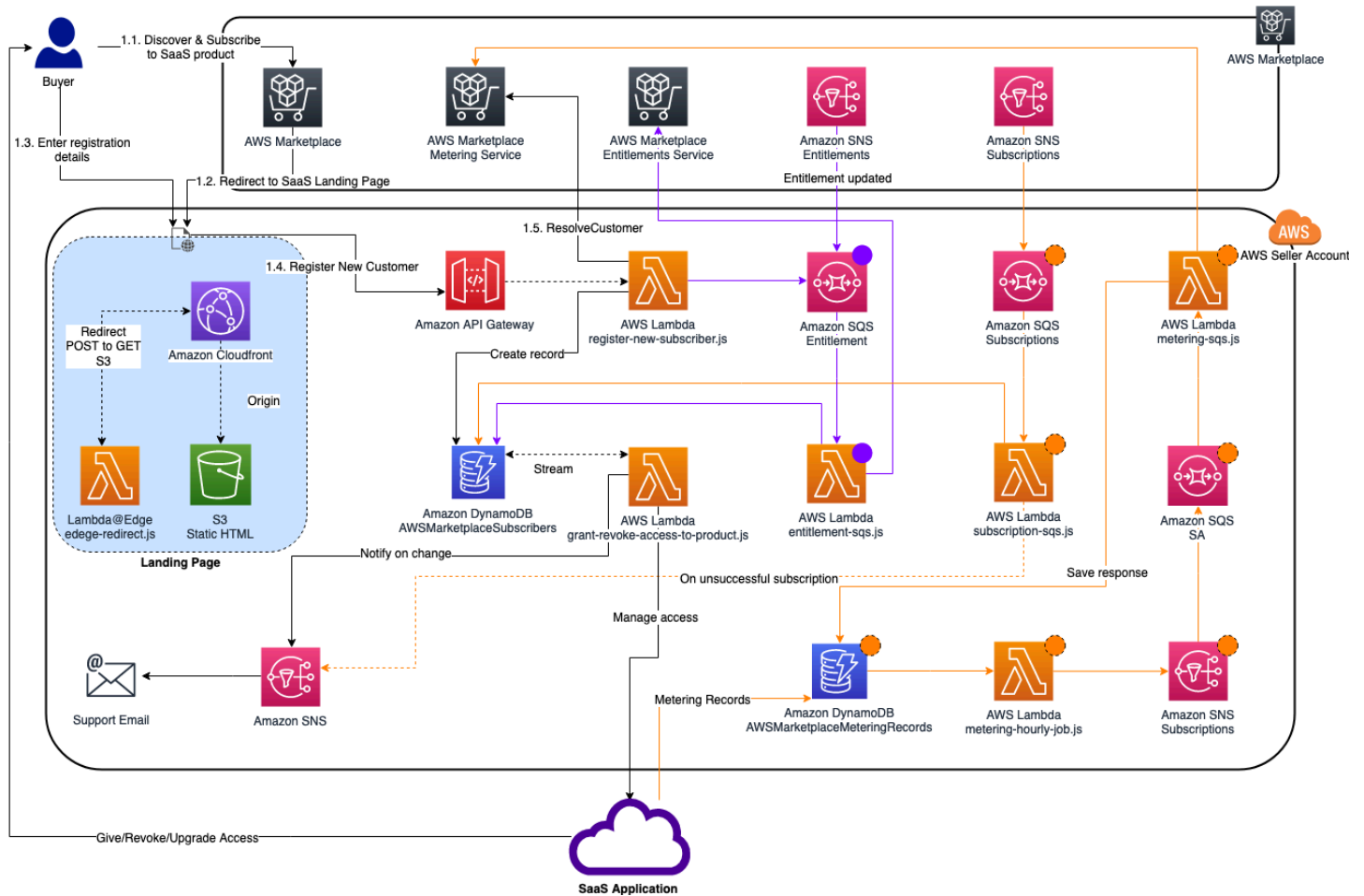
でのサーバーレス SaaS 統合ソリューションのデプロイ AWS Marketplace

AWS Marketplace サーバーレス SaaS 統合デプロイは、ベンダーの SaaS ソリューションを の対応するリストと正常に統合するために必要なコア機能を果たします AWS Marketplace。これらの機能には、新規顧客登録の受け入れ、顧客のアクセスの許可と取り消し、顧客の資格の更新、計測された使用量の報告などが含まれます。

ここで紹介するビデオでは、AWS Marketplace サーバーレス SaaS 統合の AWS クイックスタートをデプロイする方法について説明します。この自動リファレンスデプロイを使用すると、AWS Marketplace 販売者アカウントに新しい Software as a Service (SaaS) ソリューションを簡単に統合し、リスティングプロセスを高速化し、時間を大幅に短縮 go-to-marketできます。

[AWS AWS Marketplace Serverless SaaS 統合のクイックスタート](#)

次の図は、AWS 環境上の AWS Marketplace サーバーレス SaaS 統合が、次のイベントのワークフローをどのようにセットアップするかを示しています。



に serverless SaaS 統合をデプロイする方法の詳細については AWS クラウド、[AWS Marketplace Serverless SaaS 統合クイックスタートリファレンスデプロイガイド](#)を参照してください。このクイックスタートリファレンスガイドは、新しい SaaS 出品で必要な統合を完了するための軽量のサーバーレスソリューションを求めている、登録済み AWS Marketplace 販売者を対象としています。

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統合を計画する](#)
- [顧客が製品にアクセスする方法の計画](#)

料金の計画

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

SaaS 製品を使用する利点の 1 つは、請求の統合 AWS Marketplace です。この利点を利用するには、選択した料金モデルに応じて AWS Marketplace Entitlement Service、AWS Marketplace Metering Service または 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、「[Metering and Entitlement Service へのアクセス AWS Marketplace APIs](#)」を参照してください。

Amazon SNS統合を計画する

SaaS 製品用に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、お客様が開始 AWS したサブスクリプションや契約に対する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、お客様が製品登録ウェブサイトで新しいアカウントに登録できるようにするプログラムによるトリガーです。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 のお客様を検証するには、「[SaaS のお客様オンボーディング](#)」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。

AWS PrivateLink を使用して SaaS 製品にアクセスする

[VPC を使用して Amazon 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。お客様は VPC エンドポイントを作成し、仮想ネットワーク経由で AWS クラウド 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、お客様のために安全で安全で信頼できるプラットフォームを推進 AWS Marketplace するために、上のすべての 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、上の SaaS 製品のガイドラインを示します AWS Marketplace。

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 ガイドラインを満たしているか、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、AWS Marketplace は製品を継続的にレビューし、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。

- 少なくとも 1 つの価格ディメンションの料金が \$0.00 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。

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した後 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ISV別の の製品へのアドオンである場合、製品の説明には、他の製品の機能を拡張し、それなしでは製品に非常に限られたユーティリティがあることを示す必要があります。

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。

アーキテクチャガイドライン

すべての SaaS 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。

- アプリケーションの一部は、所有 AWS アカウント している でホストする必要があります。
- すべてのアプリケーションコンポーネントは、管理するインフラストラクチャでホストされている必要があります。顧客のインフラストラクチャに追加の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。
 - (AWS STS) や AWS Security Token Service () 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 AWS Identity and Access Management IAM。
 - プロビジョニングされたすべての AWS のサービスの説明、IAMポリシーステートメント、IAM ロールまたはユーザーのデプロイ方法と使用状況を含む追加のドキュメントを提供します。
 - 顧客がトランザクションとは別に追加の AWS インフラストラクチャ料金が発生した場合、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 AWS Marketplace 。
 - 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 アカウントにデプロイする方法についての説明について顧客に提供する必要があります。
 - お客様のインフラストラクチャで実行されているリソースを必要とするアプリケーションには AWS Marketplace、 による追加のレビューが行われます。これには 2~4 週間かかる場合があります。
- プロバイダーとして登録され、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 料金モデルでは、呼び出すAPIs必要があるものを決定します。
 - SaaS 契約 – [GetEntitlements](#)の AWS Marketplace Entitlement Service。
 - 消費を伴う SaaS 契約 – [GetEntitlements](#) AWS Marketplace Entitlement Service および AWS Marketplace Metering Service [BatchMeterUsage](#)で。
 - SaaS サブスクリプション – AWS Marketplace Metering Service [BatchMeterUsage](#)で。
- 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上の境界、製品のユースケース、製品に推奨されないワークロードの概要を示す必要があります。

での SaaS 製品の料金 AWS Marketplace

購入者がで 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。次に、お客様は AWS 環境内または作成した仮想プライベートクラウド (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 pay-as-you-go製品の時間単位の使用に対して購入者に請求するモデル。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。
との SaaS 契約 pay-as-you-go	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ます。詳細については、「 SaaS 契約の料金 」を参照してください。

SaaS 製品を で使用できるようにするには 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。その後、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 (時間)、またはユーザー (時間) から選択できます。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。

- ディメンションAPI名 — 計測レコードを に送信するときに使用するAPI名前[AWS Marketplace Metering Service](#)。この名前は顧客がどのディ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。
- ディメンションの説明 – 製品のディ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 — における、この製品のFCPユニット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 3 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. SaaS 製品には、その顧客の Amazon SNS トピックを通じて 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 を通じて製品に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. お客様は、AWS Marketplace ウェブサイトの Your Marketplace ソフトウェアページで、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。

SaaS 製品は、その顧客の Amazon SNS トピックを通じて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、 で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。契約に基づき、お客様は SaaS 製品の指定された使用量の使用権限を付

与されます。はこれらの使用権限を SaaS アプリケーションに AWS Marketplace 伝達します。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。代わりに、 を呼び出して使用権限を検証します AWS Marketplace Entitlement Service。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、お客様とお客様の間の契約に基づいて、前払いで、またはお客様が定義した支払いスケジュールでお客様に請求します。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については、ソフトウェアが使用状況を報告し、 を通じて当社が受け取った計測レコードに基づいて顧客に AWS Marketplace 請求する必要があります AWS Marketplace Metering Service。

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. で新しい SaaS 製品を作成し AWS Marketplace 管理ポータル、新しい 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。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 (GBps、MBps)、データ (GB、MB、TB)、ホスト (時間)、リクエスト (時間)、またはユーザー (時間) から選択できます。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。定義するディメンション

ごとに、名前、説明、料金、API名前を追加します。名前、価格、および説明は顧客に表示されません。このAPI名前は、AWS Marketplace 次のようにでの追跡とレポートに使用します。

- [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 - エンタイトルメント を呼び出すときに使用される名前API。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 名前の最大長は 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	追加使用の Pay-as-you-go 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	追加使用の Pay-as-you-go 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	4000 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

お客様が SaaS 契約 AWS Marketplace を使用して製品を購入すると、契約条件の自動更新に同意できます。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。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 通知を受け取りません。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 Support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。

お客様は、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 AWS Support。

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. SaaS 製品は、その顧客の Amazon SNSトピックを通じて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。今後料金が発生しないようにするには、でキャンセルを確認する必要があります AWS Marketplace。

での SaaS 無料トライアルオファーの作成 AWS Marketplace

AWS Marketplace 販売者として、AWS Marketplace 管理ポータル () で Software as a Service (SaaS) の無料トライアルオファーを作成できますAMMP。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 (Amazon S3) 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。販売者として、新しいサブスクリプションが作成されると、[Amazon Simple Notification Service \(Amazon SNS\) の通知](#)も届きます。Amazon SNS通知には、無料トライアル契約を識別するための `isFreeTrialTermPresent` フラグが含まれています。

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 () で SaaS 無料トライアルオファーを作成できます AMMP。

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 バージョンでは、またはカスタム EULAの標準契約 AWS Marketplaceを選択し、オファーの確認 を選択できます。
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。

による SaaS 製品への顧客のオンボーディング AWS Marketplace

Software as a Service (SaaS) サブスクリプションと SaaS 契約では、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にリダイレクトされ、自分のアカウントを登録して製品を設定します。

SaaS 製品リストを作成するときは、登録ランディングページURLに を指定します。サブスクライブ後に登録ランディングページにリダイレクトURLするために、これを使用します。販売者のソフトウェアの登録ランディングページで、販売者は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、 のフォームデータでx-amzn-marketplace-tokenトークンを識別し、請求先の顧客の識別子 AWS Marketplace で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、一意の顧客識別子、顧客 AWS アカウント ID、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。発信者が同じトークンAPIで を呼び出す限り、トークンの有効期限が切れるまで同じレスポンス値を返し続けることとなります。

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、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 アカウントは、お客様の製品にサブスクライブされています。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、お客様の AWS 請求書の一部になります。
3. その顧客に対して、その顧客の顧客 ID と販売者の製品コードが含まれている登録トークンが生成されます。
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. お客様のブラウザは、ソフトウェアの登録ランディングページに POST リクエストを送信します URL。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識別子、顧客 AWS アカウント ID、製品コードに引き換えるには、ウェブサイトが AWS Marketplace Metering Service [ResolveCustomer](#) を呼び出す必要があります。顧客識別子は顧客の AWS アカウント ID ではありませんが、製品間で共通なので、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 1 つの一意の製品コードがあり、登録時に割り当てられます。

Note

ResolveCustomer 呼び出しの例については、「[ResolveCustomer コード例](#)」を参照してください。

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 Marketplace 固有の顧客 ID (ステップ 6 で取得) を解決したら、顧客に通知メッセージを表示します。通知には、アカウントの設定中であることを記載し、販売者からの連絡を待つように伝えます。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、ステップ 6 で取得した顧客識別子を入力する AWS 顧客識別子の列が必要です。販売者のシステム内の他のアカウントでこの顧客 ID が共有されていないことを確認します。を通じて複数の製品をサブスクライブするお客様の場合 AWS Marketplace、お客様識別子は同じままで、各サブスクリプションには一意の製品コードがあります。
9. 販売者登録プロセスでは、顧客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する Amazon SNS トピックをサブスクライブします。これらは、カスタマー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。

これらのメッセージをキャプチャするには、Amazon Simple Queue Service (Amazon SQS) を使用することをお勧めします。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。これを行う方法については、「[Amazon Simple Notification Service デベロッパーガイド](#)」の「[ステップ 2: SQS Amazon SNS トピックにメッセージを送信するアクセス許可を付与する](#)」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顧客識別子を使用して、AWS Marketplace Metering Service で使用状況を計測するか、でエンタイトルメントを確認します AWS Marketplace Entitlement Service。

セキュリティと注文

販売者として、すぐに返される顧客識別子、AWS またはシステムが署名した顧客識別子のみを信頼するのはお客様の責任です。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、顧客のブラウザセッションで署名済みの属性として顧客 ID を保存します。

SaaS 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

製品作成プロセス中に、SNS トピックに実際の Amazon リソースネーム (ARN) を受け取ります。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せん。

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 契約製品を と統合する AWS Marketplace](#)
- [との SaaS 契約と pay-as-you-go製品の統合 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。
- 消費 (超過) による SaaS 契約の場合、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 (FDAs) の場合、このトピックは契約開始日 (契約署名日ではなく) に開始されます。また、契約のキャンセル、置き換え、更新、失効など、エンタイトルメントに後続の変更が発生したときにも開始されます。

契約価格 (との契約を含む pay-as-you-go) の製品は、これらのメッセージに応答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": "n0123EXAMPLEXXXXXXXXXXXX",
  "offer-identifier": "offer-abcexample123",
  "isFreeTrialTermPresent": "true"
}
```

offer-identifier は、オファーがプライベートオファーの場合にのみ通知に表示されます。

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値はブール値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 subscribe-fail が生成された場合、購入者が既に 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。
- 将来の日付のアグリーメント (FDAs) の場合、`subscribe-success` アクションはアグリーメントの開始日 (アグリーメント署名日ではなく) に開始されます。

サブスクリプション料金の商品 (との契約を含む `pay-as-you-go`) は、これらのメッセージに応答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](#)
- [との SaaS 契約と `pay-as-you-go` 製品の統合 AWS Marketplace](#)

SNS トピックへのSQSキューのサブスクライブ

提供されたSNSトピックに Amazon SQSキューをサブスクライブすることをお勧めします。SQSキューの作成とトピックへのキューのサブスクライブに関する詳細な手順については、[「Amazon Simple Notification Service デベロッパーガイド」](#)の[「Amazon SNSトピックへの Amazon SQS キューのサブスクライブ」](#)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 のトピックのみをサブスクライブ AWS Marketplace SNS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-SNSへの Amazon メッセージの送信」](#)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 に保存されますSQS。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。

Metering and Entitlement Service へのアクセス AWS Marketplace APIs

このセクションでは、Software as a Service (SaaS) 製品の顧客使用に関する請求とレポートが正確であることを確認する AWS Marketplace Entitlement Serviceのために使用される AWS Marketplace Metering Service または と統合するプロセスの概要を説明します。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 (Amazon SageMaker Ground Truth、 など AWS WAF) と統合されている場合、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使用するための計測の設定](#)
- [を使用したエンタイトルメントの確認 AWS Marketplace Entitlement Service](#)
- [SaaS 製品統合チェックリスト](#)

の設定と認証情報については AWS CLI、AWS Command Line Interface ユーザーガイドの「[の設定 AWS CLI](#)」を参照してください。AWS Python を初めて使用する場合は SDK、「[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使用するための計測の設定

Software as a Service (SaaS) サブスクリプションの場合、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 によって請求されます。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 製品で使用するために計測を設定する方法について説明します。

お客様の使用状況を計測すると、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

- [時間単位で計測する](#)
- [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 \(オプション\)](#)

時間単位で計測する

すべてのお客様の使用状況を AWS、一度に最大 25 のバッチで 1 時間単位で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できるだけ詳細に把握できます。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、次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、送信したレコードが正確であることを確認することができます。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 計測API呼び出しのログ記録 AWS CloudTrail](#)」を参照してください。

- これが料金モデル「サブスクリプション」(料金モデル「Contract」または「Contract with Consumption」ではない)の SaaS である場合、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除アクションを開始すると、販売者は通知を受け取り、最終unsubscribe-pending通知の前に報告されていないすべての使用状況を送信するために 1 時間かかりますunsubscribe-success。2 回目の通知以降は、すべて受け入れられません。他の 2 つの料金モデルにはサブスクリプション時間に基づいて設定された期間があり、購入者はサブスクリプション中にサブスクリプションを解除することはできません。自動更新はオフにしかできません。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。
- レポート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、その時間についてレポートする使用量がない場合は 0 の数量を記録できます。購入者のディメンションの使用状況を 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 時間の使用状況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、その請求サイクルまたはこれまでの請求サイクルで消費されたその明細項目の合計ユニット数と合計コストが表示されます。ただし、ディメンションの価格が 1 ユニットあたり \$0 の場合も同様です。有効にすると、[AWS コストと使用状況レポート \(CUR\)](#) に、単位あたりの料金を含むこの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 オペレーションチームは、SaaS SaaS アプリケーションが製品の公開を許可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 (Amazon SageMaker Ground Truth、など AWS WAF) と統合されている場合、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。AWS Marketplace は、AWS WAF 現時点では新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。

Marketplace AWS Metering Service の BatchMeterUsage オペレーションを使用して、計測レコードを に配信します AWS。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 ごとに重複排除されます product/customer/hour/dimension。
 - 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 に対して複数のリクエストを送信する場合 customer/dimension/hour、レコードは集計されません。
- 顧客が商品を購読している場合、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、「」を参照してください [BatchMeterUsage](#)。

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。お客様は、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 (ISVs) が購入者にソフトウェア使用状況をより詳細に把握し、コスト配分を実行するのに役立ちます。

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/billing/>) 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。

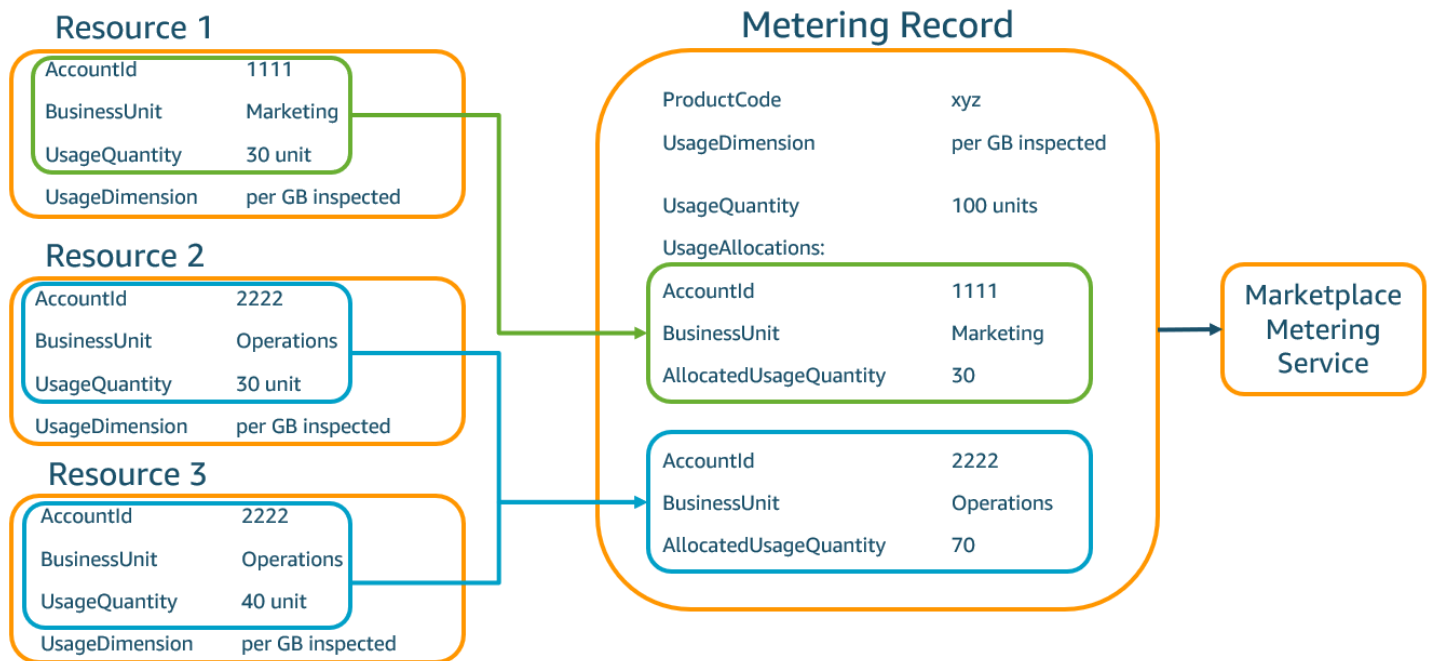
一般的なユースケースでは、購入者が 1 つの AWS アカウントを使って商品を購入します。また、購入者には同じ製品サブスクリプションに関連するユーザーが多数います。Account ID のキーを持つタグを使用して使用量の割り当てを作成し、各ユーザーに使用量を割り当てることができます。この場合、購入者は Billing and Cost Management コンソールで Account ID タグを有効にして、個々のユーザーの使用状況を分析できます。

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

ペイロード制限に違反していないことを確認するには、ビジネス要件に基づいて最大サイズのサンプルリクエストオブジェクトを作成し、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呼び出しで1MB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	11112222333	ネットワーク: 検査対象 (GB) あたり	70	2222	オペレーション

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	30	3333	財務
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。

を使用したエンタイトルメントの確認 AWS Marketplace Entitlement Service

製品が SaaS 契約製品である場合、製品は AWS Marketplace Entitlement Service を呼び出して、[GetEntitlements](#) を使用して顧客のエンタイトルメントを取得しますAPI。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 その容量に対するエンタイトルメントサービスをチェックする必要があります。

お客様の製品に対する権限を確認するには、AWS Marketplace 権限サービスで `GetEntitlements` オペレーションを使用します。AWS Marketplace エンタイトルメントサービスは、からアクセスできる米国東部 (バージニア北部) リージョンでのみ利用できません `entitlement.marketplace.us-east-1.amazonaws.com`。

GetEntitlements はフィルターとして顧客識別子およびディメンションを受け付けます。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。お客様が自動更新を設定している場合、ExpirationDate フィールドの日付は更新日です。

コードの例については、「[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) 製品が稼働する前に、必要な設定が完了していることを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して、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	使用する目的の AWS アカウント AWS Marketplace 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。

カテゴリ	要件
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。
製品	製品の作成ウィザードの Notes タブでテストIDs用の AWS アカウントを提供しました AMMP。
製品	製品タブEULAで URLの を .txt 形式で指定しました。
製品	から製品コードと Amazon SNSトピック情報を受信しました AWS Marketplace。
製品	Amazon SNSトピックをサブスクライブし、Amazon SNSトピックをサブスクライブする Amazon SQSキューを作成しました。
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	検証済み SaaS 契約製品の AWS Marketplace エンタイトルメントサービスから顧客のエンタイトルメントを検証できます。
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IDsやキャンセルされたサブスクリプションなどの状況に対してテストされません。

カテゴリ	要件
製品	発行 AWS Marketplace のために製品リクエストを に送信しました。
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ました。
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)	AWS トークンCustomerIdentifier から ProductCode と を取得ResolveCustomer するために を使用していることが検証されました。
Registration (登録)	から受信し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。

カテゴリ	要件
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が検証されました。すべての管理アクセスは、AWS Identity and Access Management (IAM) で作成された ID を介して行われます。共有アカウントがありません。
セキュリティ	すべてのプログラムによる Amazon Elastic Compute Cloud (Amazon EC2) アクセスに IAM ロールが使用されていることを検証しました。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ました。
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。

カテゴリ	要件
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應できること、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。
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、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。次の表は、使用可能なレポートに関する情報を示しています。レポートを使用して、顧客基盤をより深く理解し、売上や税金などの財務をより深く理解できます。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せん。

レポートを行う	SaaS の内容
	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。
顧客サブスクリプションレポート	<p>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。</p> <p>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。</p>

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、AWS Marketplace APIs 製品の発行と保守に必要なと統合できます。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。

の詳細についてはResolveCustomer、「計測サービスリファレンス[ResolveCustomer](#)」の「」を参照してください。AWS Marketplace API

```
# 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。

の詳細についてはGetEntitlement、「エンタイトルメントサービスリファレンス[GetEntitlement](#)」の「」を参照してください。AWS Marketplace API

```
# 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' : {
        'CUSTOMER_IDENTIFIER': [
            'customerID',
        ]
    },
    '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",
      "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、計測レコードを に送信 AWS Marketplace して、顧客に pay-as-you-go料金を請求します。

```
# 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  
#  
# customerID is obtained from the ResolveCustomer response  
  
# Import AWS Python SDK  
import boto3  
  
usageRecord = [  
    {  
        'Timestamp': datetime(2015, 1, 1),  
        'CustomerIdentifier': 'customerID',  
        'Dimension': 'string',  
        'Quantity': 123  
    }  
]  
  
marketplaceClient = boto3.client('meteringmarketplace')  
  
response = marketplaceClient.batch_meter_usage(usageRecord, productCode)
```

の詳細についてはBatchMeterUsage、「計測サービスリファレンス[BatchMeterUsage](#)」の「」を参照してください。AWS Marketplace API

レスポンスの例

```
{  
  'Results': [  
    {  
      'Timestamp': '2015-01-01T00:00:00',  
      'CustomerIdentifier': 'customerID',  
      'Dimension': 'string',  
      'Quantity': 123  
    }  
  ]  
}
```

```
{
  'UsageRecord': {
    'Timestamp': datetime(2015, 1, 1),
    'CustomerIdentifier': 'string',
    'Dimension': 'string',
    'Quantity': 123
  },
  'MeteringRecordId': 'string',
  'Status': 'Success' | 'CustomerNotSubscribed' | 'DuplicateRecord'
},
],
'UnprocessedRecords': [
  {
    'Timestamp': datetime(2015, 1, 1),
    'CustomerIdentifier': '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および消費料金モデルを使用した契約に関連しますが、追加使用量のない SaaS 契約製品には該当しません。Python の例では、適切な使用量配分タグを含む計測レコードを AWS Marketplace に送信して、顧客に料金を pay-as-you-go 請求します。

```
# 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
#
# customerID is obtained from the ResolveCustomer response

# Import AWS Python SDK
import boto3
import time

usageRecords = [
  {
    "Timestamp": int(time.time()),
```

```
"CustomerIdentifier": "customerID",
"Dimension": "Dimension1",
"Quantity":3,
"UsageAllocations": [
  {
    "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.batch_meter_usage(UsageRecords=usageRecords,
ProductCode="testProduct")
```

の詳細についてはBatchMeterUsage、リファレンス[BatchMeterUsage](#)の「」を参照してください。AWS Marketplace Metering Service API

レスポンスの例

```
{
  "Results": [
    {
      "Timestamp": "1634691015",
      "CustomerIdentifier": "customerID",
      "Dimension": "Dimension1",
      "Quantity":3,
    }
  ]
}
```

```
    "UsageAllocations": [
      {
        "AllocatedUsageQuantity": 2,
        "Tags":
          [
            { "Key": "BusinessUnit", "Value": "IT" },
            { "Key": "AccountId", "Value": "123456789" },
          ]
      },
      {
        "AllocatedUsageQuantity": 1,
        "Tags":
          [
            { "Key": "BusinessUnit", "Value": "Finance" },
            { "Key": "AccountId", "Value": "987654321" },
          ]
      },
    ],
    "MeteringRecordId": "8fjef98ejf",
    "Status": "Success"
  },
],
"UnprocessedRecords": [
  {
    "Timestamp": "1634691015",
    "CustomerIdentifier": "customerID",
    "Dimension": "Dimension1",
    "Quantity": 3,
    "UsageAllocations": []
  }
]
```

VPC を使用して Amazon 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、を通じて販売 AWS のサービスする製品へのアクセスを購入者に提供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。

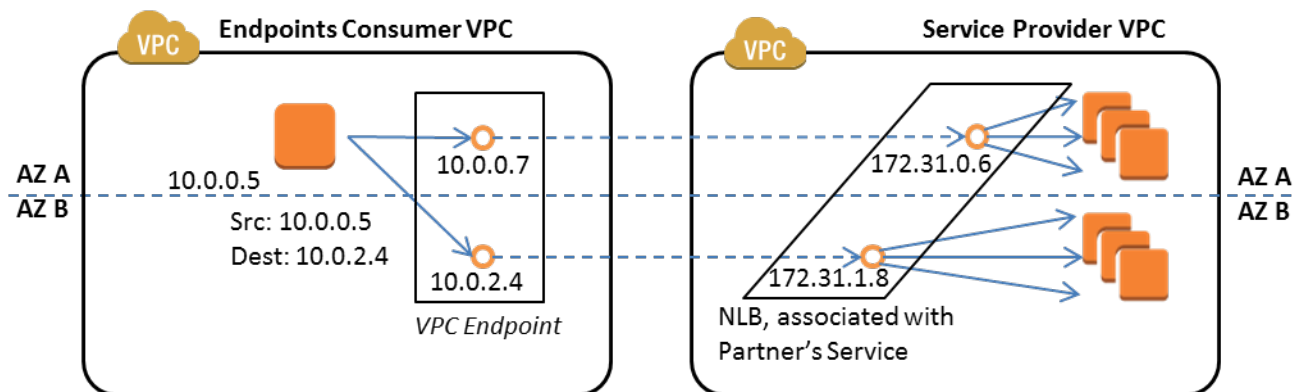
このトピックでは、複数の AWS サービスと AWS Marketplace 環境に関する実用的な知識があることを前提としています。

トピック

- [序章](#)
- [製品の設定](#)
- [製品の送信先 AWS Marketplace](#)
- [VPC エンドポイントへの購入者アクセス](#)
- [付録: チェックリスト](#)

序章

AWS Marketplace 販売者として、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。このサービスを作成および配信 AWS Marketplace するために使用すると、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に利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、AWS 顧客がインターネット、デバイス、接続、または 経由でアクセスすることなく、VPCと別の AWS サービスの間にプライベートVPN接続を作成できるようにする仮想 NAT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、購入者 AWS Marketplace 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、既存のリソースを作成または使用して、の設定VPC、サービスインスタンスの設定、ネットワークロードバランサーの設定、サービスエンドポイントの作成によるネットワークロードバランサーへのサービスの登録を行います。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。

購入者がVPCエンドポイントを作成すると、プライベートDNS名を有効にするオプションがあります。このオプションを選択すると、購入者のVPCサービスは [プライベートホストゾーン](#) を設定します。プライベートDNS名を指定すると、購入者はサービスに接続するようにVPCエンドポイントを設定するときにそれを使用できます。購入者のプライベートホストゾーンでは、プライベートDNSネーム (api.example.com) は、エンドポイントサービス (s) 用に作成されたランダムに生成されたDNS名前 (s) (vpce-1111111111111111-yyy-yyyyy.api.vpce.example.com) を指します。購入者のEC2インスタンスは、異なる間で同じ統合DNS名 (api.example.com) を呼び出しますVPCs。また、パブリックDNS名とプライベート名が同じ場合、購入者は内外からサービスにアクセスするときに同じパブリック名を使用できますVPC。

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 販売者オペレーション](#) 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、そのサービスはマーケットAWSプレイスサービスの下に表示されます。AWS Marketplace 販売者オペレーションチームは、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、わかりやすいDNS名前を使用します。

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エンドポイントを介して製品を使用可能にするには：

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 (または既存のインスタンス) を作成 (または使用) します。
3. 製品を提供する各リージョンに [ネットワークロードバランサー](#) を作成します。AWS では、リージョンのすべての [アベイラビリティーゾーン](#) (AZs) 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 SDKs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名前の[証明書](#)を [AWS Certificate Manager \(ACM\) にリクエスト](#)します。証明書 ACMを発行する前に、証明書リクエストのドメイン名を所有または制御していることを検証します。
7. `api.vpce.example.com` などのわかりやすいDNS名前のサブドメインを、AWS Marketplace 販売者オペレーション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、名前 (`vpce-0ac6c347a78c90f8.api.vpce.example.com` など) がパブリックに解決されるように、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DNS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: サポートされている SDKまたはこの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.`

製品の送信先 AWS Marketplace

サービスを発行するプロセス中に AWS Marketplace、AWS Marketplace 販売者オペレーション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. サービスのわかりやすいDNS名前。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名前です。
 - 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ネーム。

AWS Marketplace 販売者オペレーションチームは、登録するサービスに使用する会社の ID とDNS名前 (`api.vpce.example.com` など) を検証します。検証後、DNS名前はデフォルトのベースエンドポイントDNS名を上書きします。

VPC エンドポイントへの購入者アクセス

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と、プライベートホストゾーンをに関連付けるオプションがありますVPC。ホストゾーンには、 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るサービスのデフォルトのプライベートDNS名のレコードセットが含まれていますVPC。

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 コールに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、サブネット内のサービスに接続するVPCエンドポイントを確立AZs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 アドレスとリージョン名、ゾーンDNS名はエンドポイントに割り当てられます。

クライアント側DNSの名前	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
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、VPCエンドポイントの作成時に購入者がプライベートDNS名を有効にする (プライベートホストゾーンに関連付けられている) を選択した場合、購入者はサービスに接続するリージョンのデフォルトのプライベートDNS名を表示します。

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 販売者オペレーションチームに送信する前に、製品を設定してテストします。

製品の作成チェックリスト

- 作成 (または既存の を使用) VPC してから設定します。
- 内で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。
- VPC エンドポイントサービスを作成して、サービスをネットワークロードバランサーに登録します。
- 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 チームに提供します。
- 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 販売者オペレーションチームに提供します。
- ユーザーフレンドリーなサービスDNS名 (必須) を指定して、ランダムに生成されたサービスDNS名を上書きします。ユーザーフレンドリーなサービスDNS名に使用されるサブドメインACMの からSSL証明書をリクエストします。これらの証明書と、それらをリクエストするために使用した AWS アカウント ID を AWS Marketplace 販売者オペレーションチームに提供します。
- 推奨: プライベートDNS名を指定します。
- 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IDsの AWS アカウントを許可されたアカウントのリストに追加します。

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

販売者として、AWS Marketplace 購入者にプロフェッショナルな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

独立した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として、別のパートナーに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プロフェッショナルサービス製品を再販することを許可することもできます。詳細については、[「としてのチャネルパートナーの再販機会の作成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、および料金単位で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サービスには請求されます。などのツールを使用して、支払いAWS Cost Explorer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使用を開始する](#)
- [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](#)
- [としてチャネルパートナーの再販機会を作成する ISV](#)

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。

[AWS Marketplace でプロフェッショナルサービス製品を管理します。](#)

ヘルプの利用

プロフェッショナルサービス製品に関する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 販売者オペレーション](#) 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使用開始 AWS Marketplace

販売者として、AWS Marketplace 購入者にプロフェッショナルな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。製品定義は、購入者に、提供するサービスについて伝え、そのサービスのために会社を選択すべき理由を説明します。AWS Marketplace は購入者が連絡することを許可します。契約に同意し、購入者が固定料金でサービスを購入できるようにす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、お客様に提供するために必要なステップを示します。

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を作成する](#)
- [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](#)
- [製品の可視性を編集する](#)
- [プロフェッショナルサービス製品を削除する](#)

前提条件

でプロフェッショナルサービスを販売するには AWS Marketplace、次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、で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](#)」を参照してください。
- サービスまたは 内の少なくとも 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を作成する

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が。

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 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、「[製品の可視性を編集する](#)」を参照してください。

プライベートオファーを作成する

潜在的な購入者がで製品を表示する場合 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、お客様がプライベートオファーをリクエストしたことを通知する E メールメッセージを 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 AWS Marketplace に送信します。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 ドルを超えるプライベートオファーを作成する場合 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、を通じてプライベートオファー成功チーム (POST)、ベンダー財務成功チーム (VFS)、または販売者報告チームにお問い合わせください [AWS Marketplace 管理ポータル](#)。

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 (テキストまたはPDF形式) は 1 つのPDFドキュメントにまとめて追加されるため、内容を理解するためにファイル名が必要でない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で利用可能になったら、そのオファーのアクションメニューからオファーのコピー を選択しURL、それを承諾する E メールメッセージで購入者に送信します。

製品情報の編集

次の手順では、AWS Marketplace 管理ポータル内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ない場合は、「」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

新しい料金ディメンションは、を通じてのみ追加できます 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。

製品の可視性を編集する

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、既に作成した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のため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した問題を表示します。

Note

製品をパブリック AWS Marketplace カタログに表示するには、AWS Marketplace 販売者オペレーション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を削除する

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、既に作成した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として「制限あり」を選択し、「の送信」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がある製品は、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、から削除されます 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

でプロフェッショナルサービス製品を公開するときは 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を提供します。

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) で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ト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URLを指すパブリック S3 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。の内容を変更URLしても、送信 AWS Marketplace 後に でロゴは変更されません。

Note

URL 指定する S3 は公開されている必要があります。これは、ファイルが保存されている 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「[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

End-to-end お客様に代わって環境管理を行います。

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 Supports と連携する 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、URLリソースの名前と 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。

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) などの問題に関するサポートを期待します。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。購入前の質問と購入後の問題の両方に関するサポートの詳細を含めることを検討してください。

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、「[製品の可視性を編集する](#)」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 では、のすべての製品とサービスの要件が維持されます 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その関連メタデータを確認し、現在の AWS Marketplace 要件を満たしているか、超えていることを確認します。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 は製品を継続的にレビューし、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。これらは、で利用可能な AWS 利用規定に準拠している必要があります <https://aws.amazon.com/aup/>。

- 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の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、顧客が の採用または管理に関連する特定の成果を達成できるように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、 に記載されているディメンションで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 AWS Marketplace 、 のプロフェッショナルサービス製品リストの顧客支払い情報を収集することは許可されていません。
- プロフェッショナルサービス製品に関連して から AWS 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 販売者向け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) に関連していなければなりません。これらのパブリック製品では、製品がそれらの関連製品のサービスを直接提供するか、それらの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IAM。
 - 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 およびロールまたはユーザーがカスタマーアカウントでデプロイおよび使用される方法を含む追加のドキュメントを提供します。
 - 製品説明に、顧客が AWS Marketplace トランザクションとは別に追加の AWS インフラストラクチャコストを負担した場合、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 販売者として、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、 から製品を購入するために使用されるネゴシエートされた用語です AWS Marketplace。販売者として、購入者と交渉してプライベートオファーに到着します。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の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の作成](#)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、 のオファーページから作成および管理できます[AWS Marketplace 管理ポータル](#)。オファーの製品を指定して、一意の ID と を生成しますURL。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に拡張します。AWS アカウント。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、潜在的な購入者に通知した後、購入者はオファーを表示して承諾できます。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

購入者は、リンクされたアカウントまたは管理アカウントに拡張しない限り、オファーを表示できません。オファーでサービス制限を指定することはできません。そのため、製品に制限がない限り、購入者は交渉済みの価格で製品を必要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、[「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。

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、[「プライベートオファーのレポート」](#)と[「販売者レポートガイド」](#)を参照してください。

データ製品

AWS Data Exchange は、AWS お客様が 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 AWS クラウド。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 のプロバイダーには、AWS 顧客にリーチし、既存の顧客にサブスクリプションをより効率的に付与するための、安全で透過的で信頼性の高いチャネルがあります。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。

資格要件を含む詳細については、[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。

で公開するための製品の送信 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。1 つの価格構造を持つ 1 つの Amazon マシンイメージ (AMI)、複数の AMIs、AWS CloudFormation テンプレート、複雑な料金オプションと支払いスケジュールなど、製品は非常にシンプルです。製品提供を定義し、次の 2 つの方法のいずれか AWS Marketplace 管理ポータルで経由で送信します。

- [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 - より複雑で、より詳細な定義が必要な製品については、製品ロードフォーム (PLF) をダウンロードし、製品の詳細を追加し、ファイル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、他の 3 つの列は製品を顧客にデプロイする方法を示しています。

料金モデル	単一ノードを使用して起動される製品 AMI	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	単一ノードを使用して起動される製品 AMI	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	
有料使用量 (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。Products 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、まず製品タブを使用します。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備えた製品の要件](#)
- [AWS CloudFormation- 起動済み製品 \(無料または有料\) または使用ベースの有料AMI製品](#)
- [製品の変更と更新](#)
- [タイミングと心構え](#)

- [AMIs への送信 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。Products タブから、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。

その製品タイプのダッシュボードが表示され、現在の製品が表示されます。リクエストタブを選択すると、ダッシュボードに未処理のリクエストと完了した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、必要に応じて複数の異なるセッションでリクエストを作成できます。

製品リクエストを送信すると、AWS Marketplace チームはそれを確認します。リクエストした製品のタイプについては、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。新製品の場合、リクエストの公開が承認されると、送信のプレビューと承認URLに使用する限定リストが届きます。製品オファーは、送信を承認するまで公開されません。

製品更新をリクエストすると、変更を確認および承認することなく公開されます。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、詳細情報を必要としています。
- 承認が必要 - AWS Marketplace チームはURL製品の限定リストを作成しました。AWS Marketplace が公開URLする前に、を確認して承認または拒否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変ります。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](#)、輸出管理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、再パッケージソフトウェアには、Windows AMIを搭載したなど、別のベンダーによって作成されたオープンソースAMIまたは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。製品タイトルの例には、「強化」「製品>」、「製品>」、「製品2>」の「製品1>」などがあります。
- 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。

- 製品ロゴは、販売者登録プロセスで使用された会社ロゴと同じである必要があります。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。明示的なアクセス許可を取得した場合、そのドキュメントへのリンクを変更リクエストのメモセクションに含めるか、製品ロードフォームを使用する際にファイルアップロードページの簡単な説明を入力フィールドに含める必要があります。
- AMI 製品の場合、AMI名前を元の製品から再利用しないでください。AMI 名前は販売者名で始まり、次の形式に従う必要があります: [販売者名] []name-given-to-ami。

AMI 名前が命名規則に準拠していない場合は、AWS コンソールAMIから をコピーして名前を変更できます。詳細については、[「Amazon ユーザーガイドEC2」の「Amazon のコピーAMI」](#)を参照してください。 EC2

有料リストが、会社によって作成されていないスタンドアロンソフトウェア製品に関するもので、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、製品に知的財産が追加されていない場合は、以前の要件とともに、次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、追加サポートのオープンソース製品の課金に関する長い説明は、次のように始まる場合があります。これは、{SLA Details} のサポートには追加料金が適用される、再パッケージ化されたオープンソースソフトウェア製品です。

ハードウェアコンポーネントを備えた製品の要件

ハードウェア製品の販売は ではありません 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、次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を で販売することはできません 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。

- リスト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。このリスト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、製品の詳細を確認してください。

AWS CloudFormation- 起動済み製品 (無料または有料) または使用ベースの有料AMI製品

製品ロードフォーム (PLF) を使用して、AWS Marketplace お客様が 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、 から入手できます AWS Marketplace 管理ポータル。

製品を送信するには、以下の幅広いステップに従います。

- 料金モデルを選択します。
- Microsoft Excel スプレッドシートである製品ロードフォーム (PLF) 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。

各ステップの完了の詳細については、リスト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、に入力した料金情報を制御しますPLF。サポートされている料金モデルのリストについては、このガイドの[AMI の製品料金 AWS Marketplace](#)「」を参照してください。

新製品PLFの のダウンロード

1. [AWS Marketplace Management Portal](#) を起動します。
2. アセットタブの右側のペインで、[CloudFormation 製品リンクAMIを含む単一](#)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 を使用している場合は OneDrive、「コピーの編集」を選択します。これにより、が PLF に保存され OneDrive、そこで編集できます。

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。

PLF 既存の製品の 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、製品の以前のバージョンのデータが含まれています。そのデータをそのままにして、次の空白の行に新しい製品を追加します。

への入力 PLF

次の手順では、製品ロードフォーム () を完了する方法について説明しますPLF。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. から返金およびキャンセルポリシーSKUの列に、製品に関連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、詳細で明確な使用指示 を指定する必要があります。[の製品の
の使用手順の作成AMIとコンテナ AWS Marketplace](#) このガイドに記載されている要件に従ってください。

2. Type to Endpoint URL Relative URL 列に、 の必要な情報を入力しますAMI。

⚠ Important

AMI を AWS Marketplace と共有する必要があります。これを行うには、このガイドの [AMIs への送信 AWS Marketplace](#) 「」のステップに従います。

3. エンドユーザーライセンス契約URLの列には、標準の AWS Marketplace 契約へのリンクが表示されます。その契約を承諾するか、EULA使用する へのリンクを入力できます。リンクを指定する場合は、Amazon S3 バケットからのリンクなどEULA、 のダウンロードをお客様に許可する必要があります。

標準契約の詳細については、このガイド [での標準化された契約の使用 AWS Marketplace](#) の「」を参照してください。

4. us-east-1 Availability to Make available for all future instance types 列で、使用する各 AWS リージョン **FALSE**に **TRUE**または を入力します。

i 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、「[ユーザーガイド](#)」の「[のセットアップ方法](#)」を参照してください。AWS GovCloud

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。

i Note

- ほとんどのお客様は、推奨される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 を含める国の列で、インスタンスタイプ **TRUE**の下に を入力します。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 **FALSE**には を入力します。インスタンスタイプの詳細については、<https://aws.amazon.com/ec2/> 「[インスタンスタイプ](#)」を参照してください。

7. 含める国と列を除外する国には、含める国または除外する国の米国などの 2 文字の国コードを入力します。
8. Pricing Model 列に、製品の料金モデルを入力します。

次のリストでは、料金モデルと、完了する必要がある追加の列について説明します。

- BYOL URL – ライセンス を入力しますURL。料金情報を入力する必要はありません。
 - 時間単位 — に設定したインスタンスタイプの料金を入力しますTRUE。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 時間単位料金から z1d.metal 時間単位料金までの列です。
 - 時間単位の年間料金 – 前のステップでリストされた列と、a1.medium 年間料金から z1d.metal 年間料金 までで始まる列を入力します。に設定したインスタンスタイプの料金を入力します TRUE。他のすべての列は空のままにできます。
 - 使用状況 - Category FCP から FCPDimension24 Rate の列に使用状況ディメンションに関連する情報を入力します。
 - Contract – Contracts Category to Contracts Dimension24 の 36-Month料金列に、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。形式に従いますtcp, #, #, 0.0.0.0/0。例えば、tcp, 22, 22, 0.0.0.0/0 には SSHを、 tcp, 3389, 3389, 0.0.0.0/0には を使用します RDP。
 10. クラスターとAWSリソーストポロジ 1: クラスターとAWSリソーストポロジ 3: アーキテクチャ図URLの列に、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:
The image shows a box titled "Usage Information". Inside, there is a section for "Fulfillment Options" which includes a small icon of a server rack and the text "Topology 1" in a yellow box, followed by "CloudFormation Template" below it.
 - Pricing Estimate 列に、 値を含む [AWSCalculator](#) 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 テンプレートへの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への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、 に記載されている要件に従う必要があります [アーキテクチャ図](#)。

の送信 PLF

次の手順では、完了した を送信する方法について説明しますPLF。

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. ファイルアップロードページで、 PLFと任意の 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 に自動的に通知し 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。チームがリクエストについて質問や問題がある場合は、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PLF、 を使用して、それらの製品に変更を加えます。PLF 完了した元の を変更することも、使用できない場合は新しい から開始することもできますPLF。[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 を使用するかPLF、 を起動し[AWS Marketplace 管理ポータル](#)、アセットタブでファイルアップロード を選択します。製品ロードフォームと販売者ガイド で、製品の PLFをダウンロードできます。
2. で製品を更新しますPLF。
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. ファイルアップロードページで、更新された AWS CloudFormation PLFと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 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、料金の変更や AWS リージョン 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。

- 新しいバージョン – ソフトウェアの新しいバージョンとパッチまたは更新のロールアウト。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、URLs、および使用手順)。
- 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。
- 製品削除 – 新しい顧客がサブスクライブできない AWS Marketplace ように、から製品ページを削除します。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

リクエストをできるだけ早く処理するよう努めますが、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として、以下を使用します。

- 通常、合計リクエスト時間は 2~4 週間かかります。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。
- それに応じてリクエストを優先できるように、完了した製品リクエストと、予定されているイベントやリリースAMIの少なくとも 45 日前に行う必要があります。

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。

AMIs への送信 AWS Marketplace

にAMIs構築および送信されるすべての 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AMI前に、 の最終チェックをいくつか行うことをお勧めします。

- 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。
- リージョンに AWS GovCloud (US) デプロイAMIするために を送信する場合は、[アクティブな AWS GovCloud アカウントを持ち](#)、適用される輸出管理[AWS GovCloud 要件](#)を含む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

セルフサービスAMIスキャンは、内で利用できます AWS Marketplace 管理ポータル。この機能を使用すると、 のスキャンを開始しAMIs、通常 1 時間未満でスキャン結果を 1 つの場所で明確なフィードバックですばやく受信できます。

セルフサービススキャンAMIで の共有とスキャンを開始するには

1. management <https://aws.amazon.com/marketplace//manage-products/> に移動します。
2. AMI 共有する を選択します。

3. 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 () [をアップロード](#)して AWS Marketplace、販売者オペレーションチームに送信できますPLF。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、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AMI必要があります。への他のアカウント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の割り当て

AMI が送信されると、は、ソフトウェアが利用可能である必要があると指定したリージョンAMIsごとにクローン AWS Marketplace を作成します。このクローン作成と公開プロセス中、はクローンされたに製品コードを AWS Marketplace アタッチしますAMIs。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る必要があります。

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。
- ソフトウェアには、クライアントアプリケーションなど、製品を使用するために必要なすべてが含まれています。
- 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 準備

- 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
- 購入者は、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を持っていますAMI。
- AMI セルフサービススキャンAMIを使用して を実行します。

の場合 Windows AMIs

- [の設定で説明されているようにEc2ConfigService、最新バージョンの を使用します。Windows を使用するインスタンス EC2ConfigAmazon EC2 ユーザーガイド の サービス](#)。
- Ec2SetPassword、Ec2WindowsActivate、および Ec2HandleUserDataプラグインは、「[の設定](#)」で説明されているように有効になっています。[Windows を使用するインスタンス EC2ConfigAmazon EC2 ユーザーガイド の サービス](#)。
- 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 の場合 AMIs

- 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。
- エラーのないすべてのURLsロード。
- 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構築に関するご質問やご意見がございましたら、[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。

での製品のマーケティング AWS Marketplace

製品の成功に貢献するには、の商品ページへのトラフィックを直接促進 AWS Marketplace し、の認識を高める必要があります AWS Marketplace。以下のセクションでは、にリストした製品のマーケティングに役立つ情報とサポートを提供します AWS Marketplace。

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](#)
- [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) リソースを提供して、で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、の Marketplace リソースセクションからアクセスできます [AWS Marketplace 管理ポータル](#)。

製品が入手可能になったことの発表

で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、その他のプレス以外のリリース発表を確認し、AWS メッセージングやブランドガイドライン、音声との整合性を確保します。レビューのためにリクエストを 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。のビジター 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、時間または月単位で使用した分のみを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 で購入する理由 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。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、開発者ツール、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など、商用および無料の IT およびビジネスソフトウェア AWS Marketplace を幅広く用意しています。AWS Marketplace お客様は、オプションを比較したり、レビューを読んだり、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料金に表示されます。

ソフトウェアまたは SaaS 販売者が で販売される理由 AWS Marketplace

を使用すると AWS Marketplace、 で実行される サービスを持つソフトウェアおよびソフトウェア・アズ・ア・サービス (SaaS) 販売者は、顧客認識の向上、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とサービスのソフトウェアと SaaS 販売者が、個々のソフトウェア開発者から大企業まで、数十万人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。

で販売 AWS Marketplace することで、独立した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 して、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免除されるため、ソフトウェアデベロッパーは優れたソフトウェアの構築により多くの時間を割くことができます。

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マーケットプレイスである business-to-business G2 などのシンジケートされたレビューワーが、シンジケートされたレビューを に統合する機能も提供します AWS Marketplace。

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、レビューがレビュー基準を満たしていることを確認するためにレビュー 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

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。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ドとブランドアセットを提供します。でリスティング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 パートナークリエイティブとメッセージングガイド 2023
- AWS Marketplace ブランディングアセットで利用可能

リソースをダウンロードするには

1. [AWS Partner Central](#) にサインインします。
2. ブラウザのアドレスバーURLに以下を貼り付け、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. [Available in AWS Marketplace](#) のブランドアセットについて、ステップ 2 と 3 を繰り返します。

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、 の特定の製品ページに顧客を誘導します AWS Marketplace。次のリンク形式を使用します。

```
https://aws.amazon.com/marketplace/pp/prodview-prodview ID
```

置換 *prodview ID* AWS Marketplace で製品の一意的英数字コードを使用します。例:

Example

```
https://aws.amazon.com/marketplace/pp/prodview-1234567890abcdef
```

prodview ID は、 aws.amazon.com/marketplace でアプリケーションを検索するURLと に表示されます。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、顧客に正しいページを案内してい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署名 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。からの引用を使用して、プレスリリースを case-by-case ベースでサポートします 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 したり、を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を作成する必要があります。次の見出しをお勧めします: [製品名を挿入] で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ず。
- 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 ロゴと商標 (それぞれ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. お客様は、Amazon が明示的に許可する目的に限り、本商標を使用することができます。また、お客様は、(i) 本商標の使用に関する Amazon とのすべての契約 (総称して「契約」) の最新バージョン up-to-date に準拠し、(ii) 本ガイドラインの最新バージョン up-to-date に準拠し、(iii) 本商標の使用に適用される Amazon が随時発行するその他の条件またはポリシーに準拠する必要があります。
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

E メール、Amazon EventBridge イベント、Amazon Simple Notification Service (Amazon SNS) topics. AWS Marketplace provides を通じて、AWS Marketplace 製品に関連する通知をタイムリーに受信できます。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、販売者 AWS Marketplace、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。通知にカスタム E メールエイリアスを追加したり、E メール通知から受信者のサブスクライブ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 – オファーの作成時 AWS Marketplaceなど、で発生するアクションのイベントを受信 EventBridge するために を使用できます。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、 の通知とイベントの詳細について説明します AWS Marketplace。

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、および顧客は 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、「」を参照してください [イベントタイプ](#)。

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ている E メールアドレスを更新するには、[AWS アカウントのプライマリ連絡先の更新](#) を参照してください。通知に [カスタム E メールエイリアスを追加](#) したり、E メール通知から [受信者のサブスクライブを解除](#) 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メールの設定を調整します。Google や Yahoo などのプロバイダーは、これらをフィルタリングできます。手順については、「[Yahoo Mail の](#)」で、[有効な E メールがスパム \(Google\) またはブロックおよびブロック解除されるのを防ぐ](#)」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](#)
- [通知を管理する](#)

イベントタイプ

次のイベントタイプは、機械学習製品を除くすべての製品と料金タイプの E メール通知でサポートされています。

Offers

次の表は、オファーのイベントを示しています。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。

Email(メール)	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 Channel Partner プライベート オファ ーの公 開。	チャネ ルパー トナー	AWS Marketplace 公 開され たチャ ネル パート ナープ ライベ ートオ ファー	顧客AWSアカウント ID、ISV名前、ISVAWSアカウ ント ID、製品名、製品 ID、オファー ID、機会 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 Channel Partner プライベート オファ ーの公 開。	ISV	Channel Partner Private Offer が 公開さ れまし た	顧客 AWS アカウント IDs、チャンネルパートナー、 チャンネルパートナー AWS アカウント ID、製品名、 製品 ID、オファー名、オファー ID、機会 ID、オ ファー公開日、オファー有効期限、卸売価格
OfferPubl SVNotific ationSCP -1.0	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	ISV ま たは AWS Marketplace チャ ネル パート ナー	プライ ベート オファ ーが公 開され ました	カスタマー 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 ファー有効期限、合計契約額

Email(メール)	イベント	受取人	タイトル	フィールド
	発行します。			
リセラーの機会の有効期限が切れました	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を表示する	ISV、製品名、製品 ID
リセラー機会の取り消し	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を表示する	ISV、製品名、製品 ID

Email(メール)	イベント	受取人	タイトル	フィールド
作成されたりセラーの機会	ISV は、AWS Marketplace チャネルパートナーの販売承認または販売機会を作成します。	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、販売承認期間

契約

ISVs およびチャネルパートナーは、購入者が公開オファー、プライベートオファー、または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、置き換え、障害が含まれます。次の表は、契約イベントのために送信される E メールを示しています。

Email(メール)	イベント	受取人	タイトル	フィールド
内定承諾 – プライベートまたはパブ	AWS Marketplace お客様はブリックオファー	ISV	顧客が AWS Marketplace オファーを受け入れた	お客様の会社名、お客様の AWS アカウント 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

Email(メール)	イベント	受取人	タイトル	フィールド
リック内定	または プラ イベー トオ ファー を受け 入れま す。			
オ ファー 承諾 – チャネ ルパー トナー プライ ベート オファ ー	AWS Marketpla ce お 客様は AWS Marketpla ce チャ ネル パート ナー のプラ イベー トオフ アーを 受け入 れます 。	ISV	顧客が AWS Marketpla ce チャ ネル パート ナー のプラ イベー トオフ アーを 受け入 れた	お客様の会社名、お客様の 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、契約 承諾日、卸売価格

Email(メール)	イベント	受取人	タイトル	フィールド
オ ファー 承諾 - チャネ ルパー トナー プライ ベート オフ ー	AWS Marketpla ce お 客様は AWS Marketpla ce チャ ネル パート ナー のプラ イベー トオフ アーを 受け入 れます 。	AWS Marketpla ce チャ ネル パート ナー	顧客が AWS Marketpla ce チャ ネル パート ナー のプラ イベー トオフ アーを 受け入 れた	お客様の会社名、お客様の AWS アカウント ID、ISV名前、ISV AWS アカウント ID、製品名、製 品 ID、オフアー名、オフアー 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

Email(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	契約またはサブスクリプション製品の AWS Marketplace 契約が、将来の開始日を持つ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、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

Email(メール)	イベント	受取人	タイトル	フィールド
契約開始 - チャンネルパートナーのプライベート	契約またはサブスクリプション製品の AWS Marketplace 契約が、チャンネルパートナーのプライベート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、チャンネルパートナー 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

Email(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 Channel Partner オファーの承諾により、契約またはサブシヨ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 名前、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
キャンセルされた契約 – 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 Marketplace 契約がキャンセルされました。	カスタマー 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

Email(メール)	イベント	受取人	タイトル	フィールド
契約がキャンセルされた - チャンネルパートナーのプライベートオファー	AWS Marketplace チャンネルパートナーのプライベートオファー契約のキャンセル。	ISV	AWS Marketplace 契約がキャンセルされました。	顧客 AWS アカウント ID、チャンネルパートナー、チャンネルパートナー 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
契約キャンセル - チャンネルパートナーのプライベートオファー	AWS Marketplace チャンネルパートナーのプライベートオファー契約のキャンセル。	ISV	AWS Marketplace 契約がキャンセルされました。	カスタマー 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

Email(メール)	イベント	受取人	タイトル	フィールド
コンプライアンスの失敗 – 記録販売者	顧客の支払いが失敗したため、AWS Marketplace 契約は失敗します。	ISV またはオファアを作成した 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、オファア名、オファア ID、契約 ID
支払いの失敗 – 記録販売者	顧客の支払いが失敗したため、AWS Marketplace 契約は失敗します。	ISV またはオファアを作成した AWS Marketplace チャネルパートナー	AWS Marketplace アグリメントの支払い失敗	顧客 AWS アカウント ID、製品名、製品 ID、オファア名、オファア ID、契約 ID
契約が置き換えられた – プライベートオファア	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、オファア名、オファア ID、新しい契約 ID、契約開始日、契約終了日、新しい購入額

Email(メール)	イベント	受取人	タイトル	フィールド
契約が置き換えられた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。	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、チャンネルパートナー、チャンネルパートナー 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
契約が置き換えられた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 名前、ISV AWS アカウント ID、製品名、製品 ID、オファー名、オファー ID、新規契約 ID、契約開始日、契約終了日、卸売価格、マージン、新規購入額

フィールドの説明

次の表は、[Offers](#)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ています。

フィールド	説明			
顧客の会社名	サブスクライバーの会社の名前。			

フィールド	説明			
カスタマー 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 名前	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フレンドリーで一意的識別子。			
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	顧客の製品サブスクリプションが開始される日付。の形式MM-DD-YYYY。この日付は、将来の日付のアップグレードの承諾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。形式はMM-DD-YYYY。			

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。形式は <code>MM-DD-YYYY</code> 。計測または <code>pay-as-you-go</code> サブスクリプションの場合、この日付は <code>JAN-1-9999</code> に設定されます。			
契約終了日	契約の有効期限が切れる日付。形式は <code>MM-DD-YYYY</code> 。計測または <code>pay-as-you-go</code> サブスクリプションの場合、この日付は <code>JAN-1-9999</code> 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せん。			

フィールド	説明			
チャンネルパートナーの会社名	から卸売コストで製品またはサービスを購入し、顧客に再販したアカウントの名前。			
チャンネルパートナーAWSアカウント ID	から製品またはサービスを購入し、顧客に再販したチャンネルパートナーのAWSアカウントの ID。			
卸売価格	からチャンネルパートナーISVに製品を再販するための卸売コスト。			
通貨コード	契約の推定コストに関連付けられたオファー料金通貨。			
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。			

フィールド	説明			
オファターの有効期限	オファターの有効期限が切れた日付。			
機会 ID	登録されたオポチュニティの一意的識別子。			
販売承認期間	販売承認で指定されている割引を使用してオファターを作成することを再販業者に許可されている期間。			

通知を管理する

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. [送信] を選択します。

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. [送信] を選択します。

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

販売者として、Amazon を使用して のイベントの通知 EventBridge を受信できます AWS Marketplace。例えば、オファーの作成 AWS Marketplace 時 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、コンソールの Marketplace Catalog サービス EventBridgeで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](#)

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、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の製品は、チャンネルパートナーがオファーを作成するために使用します。	Offer Released	the section called “新しいオファーのイベント”
チャン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”
の製品でセキュリティISVの脆弱性が検出されました	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 ユーザーガイド」の「Amazon EventBridge ルール」](#)を参照してください。 EventBridge

以下は、によって作成された新しいオファーのイベント本文の例ですISV。

```
{
  "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の製品がチャネルパートナーによってオファ어의作成に使用される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/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"],
  }
}
```

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、チャンネルパートナー、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。Catalog 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したときにイベント AWS Marketplace API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 カタログAPIリファレンスの [「変更セットの操作」](#) 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 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 ユーザーガイド」](#) の [「Amazon EventBridge ルール」](#) を参照してください。 EventBridge

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`、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 ユーザーガイド」の「Amazon EventBridge ルール」](#)を参照してください。 EventBridge セキュリティイベントの管理の詳細については、[ブログのブログ記事の「製品カタログのセキュリティを向上させる方法 AWS MarketplaceAWS」](#)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"
}
}
```

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 販売者として、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、登録済みのすべての AWS Marketplace 販売者が利用できる[レポート](#)。
- これらのレポートのセクションにアクセスするために[API](#)使用できる。
- [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。
- [Amazon QuickSight を搭載したダッシュボード](#)には、財務データへのアクセスと分析に役立つグラフ、グラフ、インサイトが含まれています。

AWS Marketplace は、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、以下に従います。

- 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 販売者オペレーション](#) チームに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

- [での販売者の配信データフィード AWS Marketplace](#)
- [販売者がでレポートする AWS Marketplace](#)

- [補足レポート](#)
- [販売者ダッシュボード](#)

での販売者の配信データフィード AWS Marketplace

AWS Marketplace は、up-to-date販売 AWS Marketplace 者が所有するビジネスインテリジェンス ツール間で、構造化された製品および顧客情報を販売者 Amazon S3 バケットに送信 ETL (抽出、変換、ロード) するメカニズムとして、データフィードを提供します。データフィードは、カンマ区切りの値 (CSV) ファイルを収集して、指定した暗号化された 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。以下のセクションでは、データフィードの概要と、それら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、前日の 24 時間分のデータを含む前日の更新 UTC 後、毎日午前 0 時に配信されます。更新は、顧客サブスクライブ、請求対象の顧客、または AWS 支払い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の収集と分析](#)
- [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、up-to-date販売 AWS Marketplace 者が所有するビジネスインテリジェンス ツール間で、構造化された製品および顧客情報を販売者 Amazon S3 バケットに送信 ETL (抽出、

変換、ロード)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関する詳細について説明します。

データフィードは、指定した暗号化された Amazon S3 バケットにカンマ区切りの値 (CSV) ファイルを収集して配信します。CSV ファイルには次の特徴があります。

- [4180 標準](#)に準拠しています。
- 文字エンコーディングは なしで UTF-8 ですBOM。
- 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、8601 ISO の日付と時刻の形式であり、1 秒以内に正確です。
- 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たびに、その月の既存の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 2010/01/01 から 2020/04/30 (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](#)で利用できます。

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 S3 バケットにデータが配信されたときに通知SNSするように Amazon を設定できます。通知を設定する方法については、「[Amazon Simple Notification Service デベロッパーガイド](#)」の「[Amazon の使用開始SNS](#)」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、データフィードをメカニズムとして使用して、up-to-date 販売者所有のビジネスインテリジェンスツール間で Amazon S3 バケットを AWS Marketplace システムから構造化された製品および顧客情報を送信 ETL (抽出、変換、ロード) できます。暗号化された Amazon S3 バケットへの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ライブ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のサブスクライブ解除](#)

データフィードにアクセスする

1. SQL および ETL (抽出、変換、ロード) の経験を持つ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。このユーザーには、 の設定経験も必要です APIs。
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 製品リストに関連付けられた 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、以下のステップに従います。

ビデオとステップでは、設定を簡素化する[AWS CloudFormation テンプレート](#)の使用方法を説明します。

- 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 を選択して、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 - が新しいデータを Amazon S3 バケットに AWS 配信するときに通知を受け取るトピック。
- 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 ステータスと詳細を含む新しいページが開きます。
- e. リソースタブから、次のリソースの Amazon リソースネーム (ARNs) を CloudFormation ページから AWS Marketplace [カスタマー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) が新しいデータを Amazon S3 バケットに AWS 配信するときに通知を受け取るための 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 (抽出、変換、ロード) オペレーションを使用して、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、データフィードを設定して、データツールでプライマリキーと外部キーを適用します。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[データフィードテーブルの概要](#)の「」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。各一意のフィードからのデータが S3 バケットに配信されたときにアラートを送信するように Amazon SNS通知を設定できます。これらの通知は、販売者データツールがこの機能をサポートしている場合、データフィードを介して新しいデータを受信すると、販売者データウェアハウスを自動的に更新するために活用できます。Amazon Simple Notification Service デベロッパーガイドの「Amazon [の使用開始SNS](#)」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状態があります。

- SKIPPED – 販売者には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](#)、またはのクエリは GitHub <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、AWS Athena 用に記述されています。ツールで使用するクエリを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。
 - SalesforceCRM、InforERP、Netsuite などの、または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使用してバケットと 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、はreports.marketplace.amazonaws.comサービスプリンシパル AWS Marketplace を使用して Amazon S3 バケットにデータをプッシュします。を指定しました *amzn-s3-demo-bucket* CloudFormation テンプレートで。

AWS Marketplace レポートサービスが Amazon S3 AWS KMSまたは Amazon を呼び出すと SNS、バケットに書き込む予定のARNデータの 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なるようにするには、ポリシーの条件に aws:SourceArn を指定します。次の例では、*account-id* の ID を使用します AWS アカウント。

```

{
  "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のサブスクリプション解除

ウェブブラウザを開き、[AWSマーケットプレイス管理ポータル](#) 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、配信停止リクエストを AWS Marketplace 販売者オペレーション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の収集と分析

AWS Marketplace は、up-to-date販売 AWS Marketplace 者が所有するビジネスインテリジェンス ツール間で、構造化された製品および顧客情報を販売者 Amazon S3 バケットに送信 ETL (抽出、変換、ロード) するメカニズムとして、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。

- スプレッドシートでデータを表示 [データフィードへのアクセス](#) できるように、 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、データを収集および分析します。

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。

の詳細については 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、[内のデータに対してSQLクエリを実行します AWS Glue Data Catalog](#)。

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. [Amazon 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](#)。

Amazon の詳細については QuickSight、「[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「[Seller Data Feed Delivery Service、Amazon Athena および Amazon を使用して販売者レポートを作成する QuickSight](#)」を参照してください。

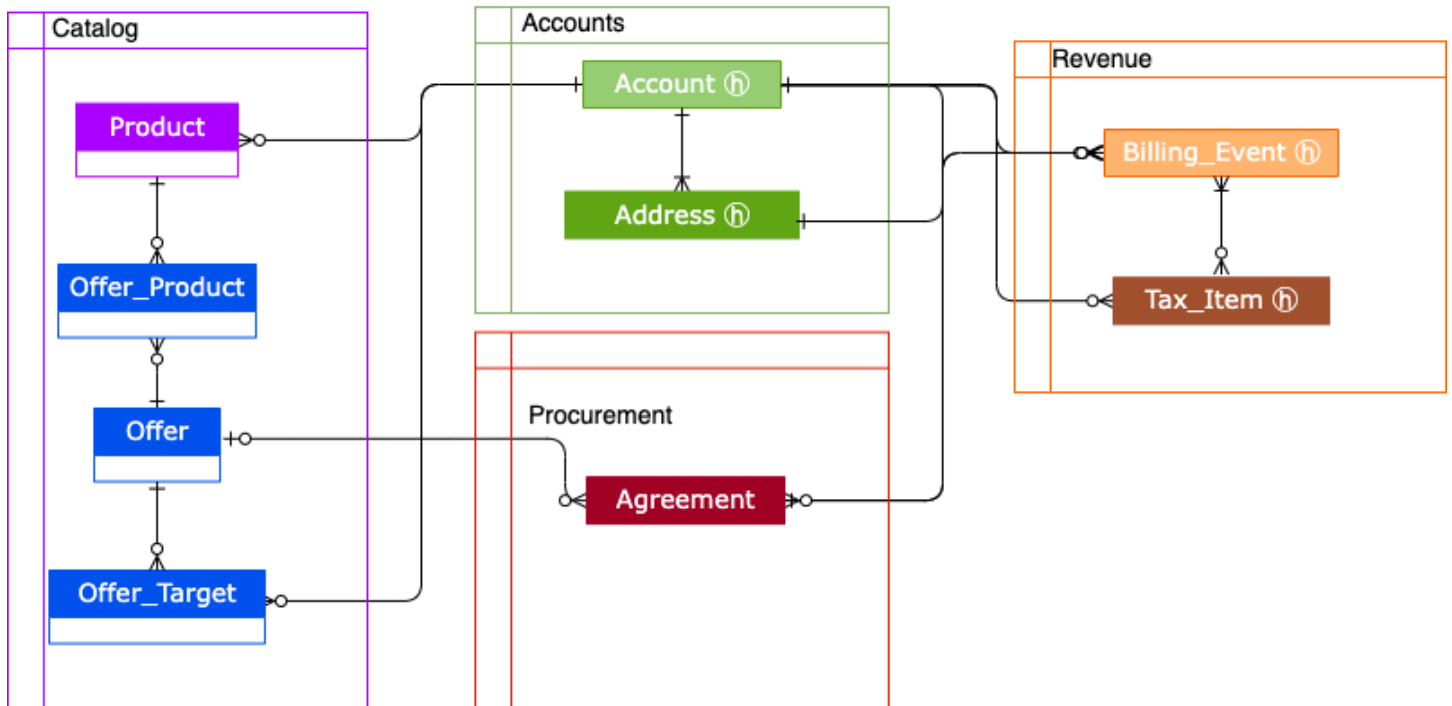
データフィードテーブルの概要

AWS Marketplace 提供されるデータフィードは、結合してクエリのコンテキストを増やすことができる一連のテーブルです。

データフィードには、次の 3 つの一般的なドメイン、つまり関心のあるカテゴリがあります。

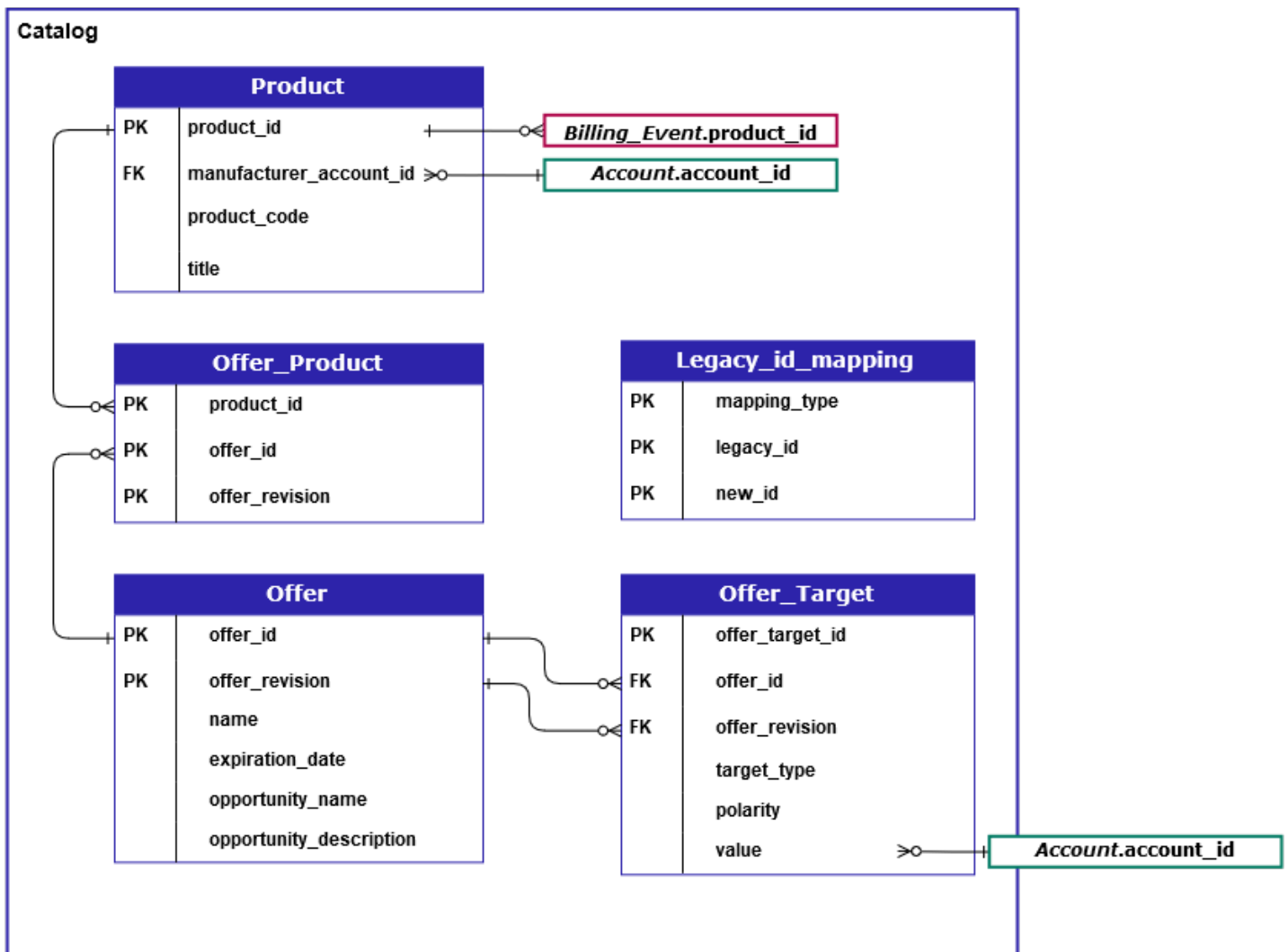
- カタログ - アカウント内の製品やオファーに関する情報が含まれます。
- アカウント - で製品を提供または購入するアカウントに関する情報 AWS Marketplace が含まれます (チャンネルパートナーや購入者など、お客様自身のアカウントまたはお客様が協力する相手のアカウント)。
- 収益 - 請求、支払い、税金に関する情報が含まれます。
- 調達 — 登録販売者として作成した製品オファーの契約に関する情報が含まれます。

次の図は、各ドメインのテーブルと、それぞれのテーブル間の関係を示しています。この図は、カタログ、アカウント、収益の各ドメインとその中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

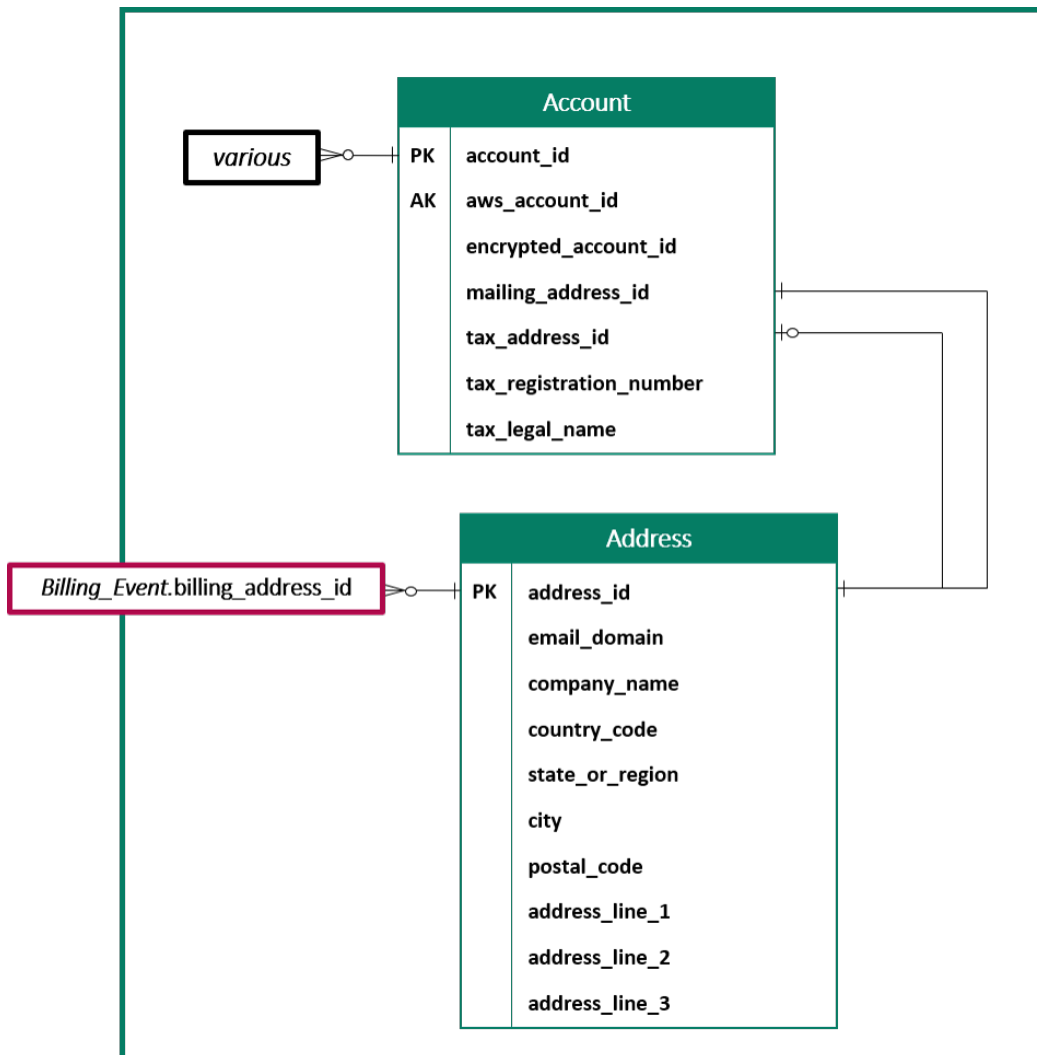
テーブル内の各フィールドの説明や作成できる結合など、これらのテーブルの詳細については、以下のトピックを参照してください。

- [製品データフィード](#)
- [オファー製品データフィード](#)

- [オファーデータフィード](#)
- [オファ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。



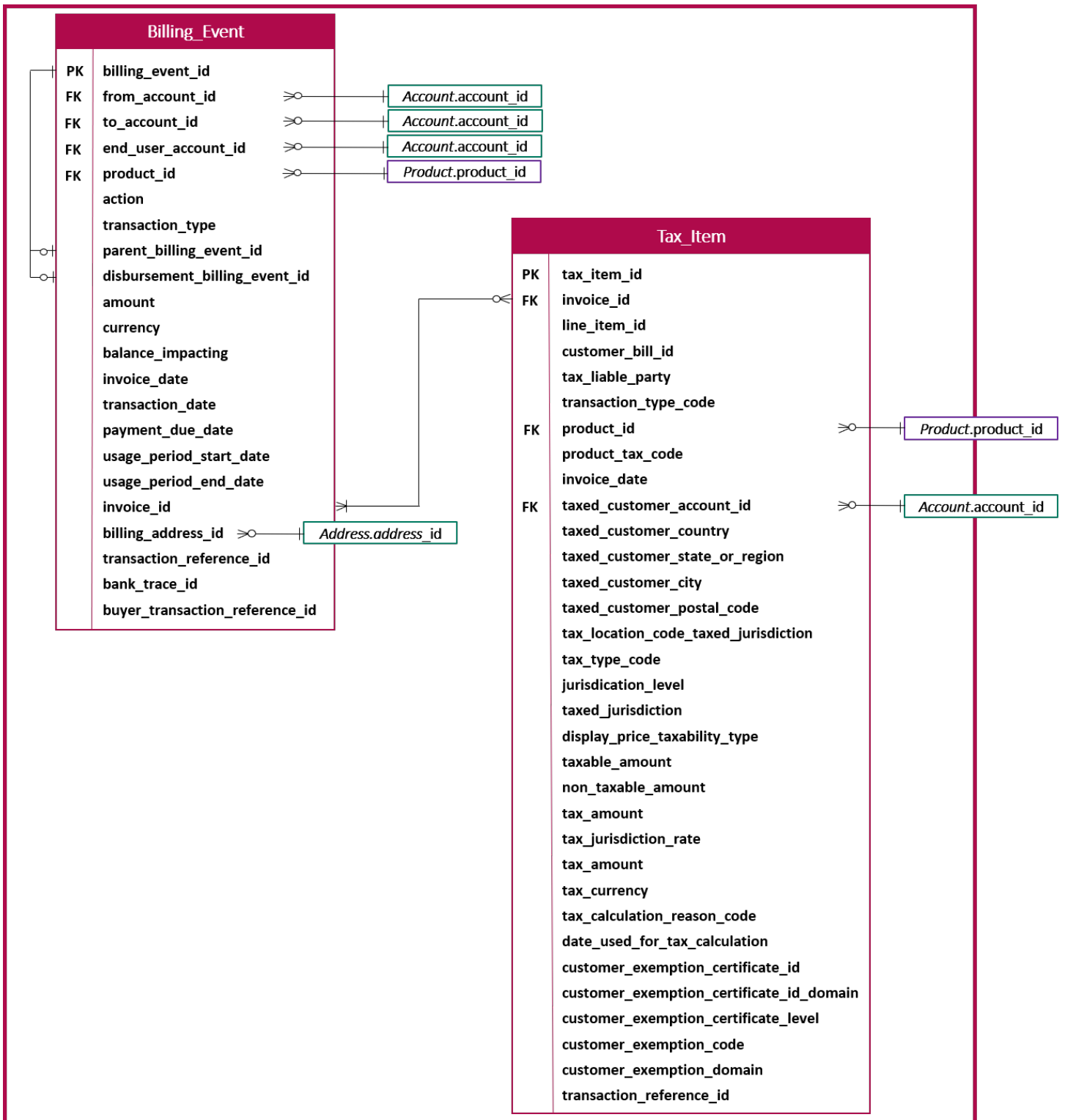
Note

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、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ルド](#)
- [住所データフィールド](#)

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、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可能な結合など、これらのテーブルの詳細については[アグリーメントデータフィード](#)、このガイドの「」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。

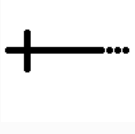
次の表では、ER 図に示されている記号について説明します。




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 」を参照してください。複数のフィールドがプライマリ

記号	説明
	<p>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。</p> <p>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。</p> <div data-bbox="829 562 1507 877" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p>Note</p> <p>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。</p> </div>
	<p>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。</p>
	<p>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。この例では、one-to-many 接続を表します。</p>

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
	<p>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。</p>

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](#)
- [未回収または支払い済みの請求書](#)
- [税金請求書](#)
- [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、詳細な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ことも、データやユースケースに合わせてカスタマイズすることもできます。

クエリのコメントには、クエリの動作と変更方法が説明されています。

```
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、請求済み収益レポートが作成されます。図のように例を使用することも、データやユースケースに合わせてカスタマイズすることもできます。

クエリのコメントには、クエリの動作と変更方法が説明されています。

```
-- 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 '4Other 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,
aggrement.proposer_account_id,
aggrement.offer_id,
aggrement.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 cpo 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
end

```



```

        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)
```

未回収または支払い済みの請求書

未回収または支払い済みの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、コレクション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ことも、データやユースケースに合わせてカスタマイズすることもできます。

クエリのコメントには、クエリの動作と変更方法が説明されています。

```
-- 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 '4Other 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,
        aggrement.proposer_account_id,
        aggrement.offer_id,
        aggrement.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_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
),
cppo_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'
  end

```



```

        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)
```

税金請求書

税金がかかる請求書を見つけるには、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、税金レポートを作成します。図のように例を使用することも、データやユースケースに合わせてカスタマイズすることもできます。

クエリのコメントには、クエリの動作と変更方法が説明されています。

```
-- 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

製品によって支払われた金額を確認するには、次のような一連のクエリを実行します。この例は販売[支払いレポート](#)者レポートと同等です。

これらのクエリ例は相互に構築され、支払いを含む製品の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、図のように使用することも、データやユースケースに合わせてカスタマイズすることもできます。

クエリのコメントには、クエリの動作と変更方法が説明されています。

Note

このクエリを実行する場合、データは列と update列の2つの時間軸を使用して取り込まれることを前提valid_fromとしています。軸の詳細については、「」を参照してください[AWS Marketplace データフィードのストレージと構造](#)。

```
-- 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、次のような一連のクエリを実行します。この例は、[販売補償レポート](#)販売者レポートと同等です。これらのクエリ例は相互に構築され、ソフトウェアの使用に対して各顧客に請求される合計金額を含む顧客詳細の最終リストを作成します。クエリは、図のように使用することも、データやユースケースに合わせてカスタマイズすることもできます。

クエリのコメントには、クエリの動作と変更方法が説明されています。

Note

このクエリを実行する場合、取り込まれたデータが2つの時間軸(列valid_fromとupdate列)を使用していると仮定します。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 は、販売者が製品の販売に関する情報を収集および分析できるように、多数のデータフィードを提供します。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](#)
- [アグリーメントデータフィード](#)
- [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に一意的識別子 (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せん。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 アカウントのアカウント AWS 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。の値は Metering Service AWS

列名	説明
	Marketplace によって encrypted_account_id 使用され、例えば、 ResolveCustomer アクションによって返 CustomerIdentifier される の値として使用されます。
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	25o3k46eN6eViOfFiiqtxwX8e3kaOiPaUiofjyFa3			
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	555567679999	373vuQUqmQ8v	5oJ6vTjSzMrrF2gvh2Vj9HfqiM800MuLEHmyFY5Lr42s8	5oJ6vTjSzMrrF2gvh2Vj9HfqiM800MuLEHmyFY5Lr42s8	SE823935083345	

account_id	aws_account_id	encrypted_account_id	mailing_address_id	tax_address_id	tax_registration_number	tax_legal_name
VleGa2t9j	737399988	8SPxAYmi8	NLUc5UeiM	V5NhBYBiY	DE4695580	
3MuxioH9w	88	MwX	IGFTTrDWCo	ogwy0WMhn	25	
c8lsndXXC			ftDPhDUF1	dGU4AfMgg		
gGCGUreeX			oaSd8xgl5	muoTC2j7P		
riocM5			QM8Db7	m8ZKKNNyT		

住所データフィード

⚠ Important

AWS Marketplace は、2023 年 12 月に販売者配信データフィードサービスを使用して配信される PIIFeed を廃止します。 <https://docs.aws.amazon.com/marketplace/latest/userguide/data-feed-service.html> アドレスデータのニーズには AddressFeed_V1 データフィードを使用します。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	住所の一意的キー。
	billing_address_id フィールドの Billing_Event データフィー

列名	説明
	ド、mailing_address_id または tax_address_id フィールドの Account データフィールドからの結合に使用できます。
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
V5NhBYBiYogwy0WMhndGU4AfMgg	a.com	Mateo Jackson's Company	DE		Hamburg	67568

address_id	email_domain	company_name	country_code	state_or_region	city	postal_code
muoTC2j7P m8ZKKNNyT						
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	

アグリーメントデータフィード

このデータフィードは、契約に関する情報を提供します。契約とは、提案者 (レコード販売者) とアクセプター (AWS購入者) の間で署名された契約で、製品の使用を開始します。このデータフィードは、登録販売者として作成した製品オファターの契約に関する情報を提供します。

アグリーメントデータフィードは 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	<p>アグリーメントの現在のステータス。サポートされているステータスは次のとおりです。</p> <ul style="list-style-type: none"> • アクティブ – 契約の条項の一部またはすべてが有効である。 • 終了 – AWS開始された終了イベントにより、契約は事前に合意された終了日より前に終了しました。最も一般的なのは支払いの失敗です。 • キャンセル済み – アクセプターは終了日より前に契約を終了することを選択しています。 • 期限切れ – 契約は、事前に合意された終了日に終了しました。 • 置き換え済み – 契約は代替オファーを使用して置き換えられました。 • 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ました。 • アーカイブ済み - 契約が終了しました。契約の終了、終了、キャンセル、期限切れの正確な理由が指定されていません。 • ロールバック – エラーのため、契約リビジョンがロールバックされました。以前のリビジョンがアクティブになりました。非アクティブな契約リビジョンにのみ適用されません。

列	[Description] (説明)
	<ul style="list-style-type: none"> 置き換え済み – 契約リビジョンはアクティブでなくなり、別の契約リビジョンがアクティブになりました。非アクティブな契約リビジョン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が支払う可能性のある既知の金額の合計。
currency_code	契約のライフサイクル全体で顧客が支払う可能性のあ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 を提案した販売者は PurchaseAgreement、販売者のアカウントのグローバルに一意の識別子 (GUID) で表されます。アカウントデータフィールドに結合するために使用できます。
acceptor_account_id	この を受け入れた購入者。購入者のアカウントのグローバルに一意の識別子 (GUID) で PurchaseAgreement表されます。アカウントデータフィールドに結合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。Offer および Offer ターゲットデータフィールドに結合するために使用できます。
start_time	契約が開始される日時。
end_time	契約が終了した日時。終了日のない pay-as-you-go アグリーメントでは、フィールドは null です。

列	[Description] (説明)
acceptance_time	<p>オファーが承諾された日時、または契約が作成された日時。</p> <ul style="list-style-type: none"> bring-your-own-license アグリーメントのバックデートが可能 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
intent	契約が最後に変更されたときの購入者の意図。
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。

アグリーメントデータフィールドの例

validity_start_time	agreement_id	product_id	acceptance_time	offer_id	offer_status	start_time	end_time	acceptance_time	intent	preceding_agreement_id	status	status_code	estimated_value	currency
2021-07-20T05:44:20Z	agn88a44c78a5t	88a44c78a5t	88a44c78a5t	offer_krx_x	1	2021-07-20T05:44:20Z	2021-07-20T05:44:20Z	2021-07-20T05:44:20Z	NE		AC		1,000,000	USD

請求イベントデータフィールド

このデータフィールドは、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、このデータフィールドを使用して、購入者への請求日や請求内容を確認できます。[サンプル SQLクエリ](#)を使用して、このデータフィールドからのデータを分析することもできます。

このデータフィールドは、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、このデータフィール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。

列名	説明
アクション	<p>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。</p> <ul style="list-style-type: none">• 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	<p>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。</p> <ul style="list-style-type: none"> • 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 クレジットを付与する金額。 • 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
	<p>ng_event_id が表示され、balance_impacting の値は 0 です。</p> <ul style="list-style-type: none">DISBURSEMENT_FAILURE - 取引を無効にします。

列名	説明
parent_billing_event_id	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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。 <p>broker_id の値が AWS_EUROPE の場合、parent_billing_event_id は次の</p>

列名	説明
	<p>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。</p> <ul style="list-style-type: none"> <code>action</code> の値は <code>DISBURSED</code> で、<code>transaction_type</code> の値は <code>DISBURSEMENT_FAILURE</code> です。 <p><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。</p>
<code>disbursement_billing_event_id</code>	<p><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。</p> <ul style="list-style-type: none"> <code>transaction_type like ('SELLER%')</code> <code>transaction_type like ('AWS%')</code> <p>他のすべてのシナリオでは、この値は <code>null</code> です。</p>
<code>amount</code>	請求イベントの金額。
<code>currency</code>	639 ISO 通貨コード。
<code>balance_impacting</code>	<p>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。</p>
<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。 <ul style="list-style-type: none">• 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	<p>取引を促進した事業者の識別子。可能な値は以下のとおりです。</p> <ul style="list-style-type: none"> • AWS_INC <ul style="list-style-type: none"> – AWS, Inc. (米国に拠点を置く) の識別子。 • AWS_EUROPE <ul style="list-style-type: none"> – Amazon Web Services の識別子 EMEA SARL (ルクセンブルクをベース)。 • NULL <ul style="list-style-type: none"> - 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	<p>GROUP BY または のウィンドウ関数コンストラクトを使用して、請求フィールド内のすべての関連レコードをグループ化する識別子 SQL。これらの関連レコードには、出品料金に対する購入者請求書、販売者請求書、および付加価値税 (VATs) を含めることができます。</p>

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 によって収集され AWS、販売者に支払い、販売者が税務機関に送金されます。この場合、transaction_type は SELLER_TAX_SHARE です。
- 税金は によって収集されません 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、税金計算を実行 AWS Marketplace したり、税金情報を受け取ったりしません。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ルドの例

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。

- データフィールドの `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。

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ルド

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ル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-o4grxfafcx	INVOICED	SELLER_RE V_SHARE

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
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

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
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	777788889999	111122223333		prod-o4grxfafcxxxx	INVOICED	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
16	77778888999	11112222333		prod-o4grxfafcxxxx	INVOICED	SELLER_TAX_SHARE
17	11112222333	AWS		prod-o4grxfafcxxxx	INVOICED	AWS_REV_SHARE

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		73.5		2018-12-3 1T00:04:07Z	788576665
		6.5		2018-12-3 1T00:04:07Z	788576665
		-7.35		2018-12-3 1T00:04:07Z	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
130	77778888999	11112222333		prod-o4grxfafcxxxx	DISBURSED	SELLER_REV_SHARE
131	77778888999	11112222333		prod-o4grxfafcxxxx	DISBURSED	SELLER_TAX_SHARE
132	11112222333	AWS		prod-o4grxfafcxxxx	DISBURSED	AWS_REV_SHARE

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	アクション	transaction_type
I33	AWS	111122223333			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の収集と分析](#) 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: 販売者に代わって金額が収集 AWS できる

販売者に代わって が収集 AWS できる金額から返金、クレジット、および許可アカウントを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 収集する出品手数料と税金を削除し、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 GUIDs は古いレポートで使用され、新しい IDs はデータフィードと AWS Marketplace APIs で使用されます。

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-dacpxznflfwin
PRODUCT	1368541d-890b-4b6c-9bb9-4a55306ab642	prod-o4grxfafcxxxxy
OFFER	558d8382-6b3a-4c75-8345-a627b552f5f1	offer-gszhmlle5npzip

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。
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。

列名	説明
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_rev ision	name	expiratio n_date	opportuni ty_name	opportuni ty_descri ption	seller_ac count_id	opportuni ty_id
offer-dac pxznflfwi n	1	エンター プライズ 契約プロ グラムオ ファー	9999-01-0 1T00:00:0 0Z				
offer-gsz hmle5npzi p	1	販売者 が作成し たプライ	2020-10-3 1T00:00:0 0Z				

offer_id	offer_revision	name	expiration_date	opportunity_name	opportunity_description	seller_account_id	opportunity_id
		ベートオファー					
offer-hmzhyle8nphilp	1	エンタープライズ契約プログラムオファー	9999-01-01T00:00:00Z				

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。 Offer 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す。

列名	説明		
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

オファーターゲットデータフィールド

このデータフィールドは、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ルドに含まれます。

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ルドに作成されます。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。

オファーターゲットデータフィールドは 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、データフィールドの列の名前と説明を示しています。

列名	説明
offer_target_id	フィールドの主キー。
offer_id+offer_revision	オファタ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ターを参照します。 Targetデータフィールドの 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	オファターの受取人が BuyerAccounts (プライベートオファターを示す) であるか、ParticipatingPrograms であるかを示します。
polarity	オファターが 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。 <ul style="list-style-type: none"> PositiveTargeting オファターは、この target_type に適用されます。 NegativeTargeting オファターは、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 またはオファターで使用できるプログラム。例えば、 AWS Marketplace (SCMP) の標準契約 、

列名	説明
	または AWS Marketplace フィールドデモンストラーションプログラム (FDP) などです。

オファーターゲット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。

offer_target_id	offer_id	offer_revision	target_type	polarity	value
925ddc73f6a373b7d5544ea3210610803b600	offer-dacpxznflfwin	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract
471ff22ae3165278f1fb960d3e14517bcd601	offer-gszhml5npzip	1	ParticipatingPrograms	PositiveTargeting	FieldDemonstration
511ff22adfj65278f1fb960d3e14517bcd6e602	offer-gszhml5npzip	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 および Offer_Product データフィードの product_id フィールドに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、データをレポートと結合したり、AWS Marketplace Metering Service で提供される内容を参照したりするためにも使用されます。
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

product_id	manufacturer_account_id	product_code	title
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。が売上税を AWS 徴収 GST する場所 VAT、または、または、および、AWSInc. という名前で現地の税務機関に税金を送金する場所の詳細については、[「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 の値を組み合わせて、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。
customer_bill_id	顧客請求書の一意的識別子。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。

列名	説明
tax_liable_party	<p>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 が納税義務当事者である場合、売上税は 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。</p> <p>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。</p>
transaction_type_code	<p>取引のタイプ。可能な値は以下のとおりです。</p> <ul style="list-style-type: none">• 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 <p>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</p>
product_id	<p>製品の外部キー。</p> <p>product_id フィールドの Product データフィールドへの結合に使用できます。</p>
product_tax_code	<p>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。</p>
invoice_date	<p>請求書が作成された日付。</p>

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 フィールドへの結合に使用できます。
taxed_customer_country	税計算に使用される住所の ISO3166 アルファ 2 国コード。
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	上記の金額の ISO 4217 アルファ 3 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。 <ul style="list-style-type: none"> 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 Marketplace ファシリテータ税は、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ドのすべての AWS_TAX_SHARE および SELLER_TAX_SHARE レコードには、税金項目データフィードに対応するレコードが含まれることが予想されます。

税品目データフィードの例

税品目データフィードの例を次に示します。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。

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
f1r4jobxjzww8czdsr q4noue2uxd56j39wxw 0k7_0001	250816266	764000000 00000000000	572000000 0000000000
gfkjjobxjzw56jgkrs rqgjdk52uxd56j39wg 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
fnohdid8kwwgqq9lvii 2k30spn3ftgwihbe8h 75x_0001	229987654	921000000 00000000000	639000000 0000000000

tax_liable_party	transaction_type_code	product_id	product_tax_code	invoice_date
販売者	AWS	prod-o4gr xfafcxxx	AWSMPSOFT WARE_RA	2018-12-3 1T00:00:00Z
販売者	AWS	prod-o4gr xfafcxxx	AWSMPSOFT WARE_RA	2018-12-3 1T00:00:00Z
販売者	AWS	prod-t3gr xfafcxyy	AWS_REMOT E_ACCESS_ SOFTWARE	2018-08-3 1T00:00:00Z
販売者	REFUND	prod-t3gr xfafcxyy	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_customer_account_id	taxed_customer_country	taxed_customer_state_or_region	taxed_customer_city	taxed_customer_postal_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	NonTaxable	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	NonTaxable	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、AWS Marketplace はいくつかのレポートとデータセットを中止します。

- Marketplace レガシーカンマ区切り (csv) 販売者レポート
- 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は、レポートページを確認するか、5/30/2024 に通信を発行した AWS 管理者にお問い合わせください。

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 に生成UTCされ、前日UTCの 24:00 までカバーされます。
- 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 S3 バケットにデータが配信されたときに通知SNSするように Amazon を設定できます。通知を設定すると、は で登録した 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については、Amazon Simple Notification Service デベロッパーガイドの「[Amazon の使用開始SNS](#)」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 による製品および顧客データへのアクセス](#)を使用すると、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 のように直接ではなく、Amazon Simple Storage Service (Amazon S3) のファイルに非同期的にデータを返しますAPI。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

では、次のレポートを使用できます 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、に関するデータが記載されています。このレポートを受け取るのは、関連情報が入手できる場合のみです。

- AWS 顧客が製品を日常的に使用する方法
- その使用量による推定収益。

このレポートが届かず、必要と思われる場合は、[AWS Marketplace Seller Operations](#) チームにお問い合わせください。また、レポートの E メール通知をキャンセルする場合は、そのチームに連絡します。

このレポートは、[AWS Marketplace 管理ポータル](#) でアクセスできます。に登録している場合は [the section called “AWS Marketplace コマース分析サービス”](#)、を使用してレポートにアクセスすることもできます AWS SDK。

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 に公開 UTC され、前日 UTC の 00:00 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。
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 標準を使用しています。
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。
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)。
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。

列名	説明
提供タイプの説明	製品提供タイプの説明。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、トランザクションが米ドルの場合、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 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。
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 財務レポートの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的識別子。

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。
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 標準を使用しています。
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、トランザクションが米ドルの場合、エントリは USD です。
料金の説明	料金の理由 (月額料金、年間料金、返金など)。
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 標準を使用しています。
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。
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 財務レポートの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的識別子。

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の識別子。

セクション 4: 新しいインスタンス

このセクションでは、新しい EC2 インスタンスとインスタンスタイプの一覧表示し、過去 24 時間について説明します。

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。

列名	説明
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 標準を使用しています。
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 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 標準を使用しています。
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 財務レポートの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 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。
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 標準を使用しています。
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。
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。

列名	説明
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 標準を使用しています。
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 財務レポートの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的識別子。

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せん。

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。
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 標準を使用しています。
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。
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 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 標準を使用しています。
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。

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。
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 財務レポートの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。

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がわかります。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。このレポートが届かず、必要と思われる場合は、[AWS Marketplace Seller Operations](#) チームに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します。

このレポートは、[AWS Marketplace 管理ポータル](#) でアクセスできます。に登録している場合は [the section called “AWS Marketplace コマース分析サービス”](#)、を使用してレポートにアクセスすることもできます AWS SDK。

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 IDs すべての顧客の AWS アカウント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 に公開UTCされ、前日UTCの 00:00 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 時点におけるすべての使用量ベースのサブスクリプションのデータを示しますUTC。

列名	説明
顧客 AWS 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。
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。
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

列名	説明
	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を追跡するのに役立ちます。
リセラーアカウント ID	チャン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	チャン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

このセクションでは、前日の 23:59:59 時点におけるすべての料金ベースのサブスクリプションのデータを一覧表示しますUTC。

列名	説明
顧客 AWS 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。
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。

列名	説明
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を追跡するのに役立ちます。
リセラーアカウント ID	チャン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	チャン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 日に利用できなくなります。これら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](#)
- の [collections_and_disbursements_report](#) SQLクエリ [GitHub](#)。

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。では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 [での販売者の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](#) チームに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金額。
AWSRefFeeRefund	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る AWS Marketplace 料金の金額。
SellerRevCredit	がお客様のアカウントに AWS Marketplace 付与した AWS クレジット。
AWSRefFeeCredit	がアカウントに 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。この列は、列から SellerRev AWSRefFee 列を引いた値に等しくなります。顧客に返金が行われると、この列は列から SellerRevRefund 列を引いた数と等しい負の数になります AWSRefFeeRefund。
取引リファレンス ID	トランザクションを AWS Marketplace レポート間で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。
SellerUSSales税金	この取引で請求された米国の消費税と使用税の合計額。

列名	説明
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 標準を使用しています。
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 財務レポートの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的識別子。

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	支払いの一意的識別子。
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 標準を使用しています。
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。
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、米国売上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
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、米国売上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の支払いのデータを示します。

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。
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的識別子。

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。

列名	説明
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。
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。

列名	説明
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
支払人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	製品の一意的識別子。
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 収益分配	決済時に請求金額から差し引かれたAWS料金額。
総返金	取引の返金の合計金額。
AWS 返金分配	取引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、米国の売上税と使用税を引いた金額。
通貨	取引の通貨。例えば、トランザクションが米ドルの場合、エントリは USD です。
AR 期間	取引の月と年 (YYYY-MM 形式)。

列名	説明
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 財務レポートの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 日に利用できなくなります。セクション 1~3 では、これらの代替方法を使用できます。

- [請求済み収益ダッシュボード](#)
- の [billed_revenue_report](#) SQLクエリ GitHub。

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。

月別請求済み収益レポートは、会計およびその他の財務レポートの目的で毎月の請求済み収益に関する正式な情報を提供します。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。AWS Marketplace コマース分析サービスに登録している場合は、API呼び出しを使用してこのレポートのプルダウンセクションを使用できます。詳細については、「[the section called “AWS Marketplace コマース分析サービス”](#)」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](#) チームに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 に毎月公開されます UTC。

請求および収益データは翌月の 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 から前月の最後の日の 23:59 までを対象とし UTC ます。

例えば、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 を対象としています UTC。

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。
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 標準を使用しています。
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。
製品コード	製品の一意的識別子。
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	請求金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サブスクリプション費用の合計金額。

列名	説明
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	取引に対して請求される合計金額、AWS Marketplace 手数料、返金、および米国の売上税と使用税を差し引いた金額。
通貨	取引の通貨。たとえば、取引が US ドルの場合、エントリーは USD です。
取引リファレンス ID	トランザクションを AWS Marketplace レポート間で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、米国売上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、米国売上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 AWS アカウント番号	料金の請求先となるアカウントの ID。

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。例えば、E メールアドレスが <code>liu-jie@example.com</code> 、エンタリは <code>example.com</code> 。
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 財務レポートの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。
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 標準を使用しています。

列名	説明
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。
AWS 出品料金	請求金額から差し引かれた 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)。

列名	説明
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 手数料、返金、および米国の売上税と使用税を差し引いた金額。
通貨	取引の通貨。例えば、トランザクションが米ドルの場合、エントリーは <code>USD</code> です。
取引リファレンス ID	トランザクションを AWS Marketplace レポート間で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、米国売上税レポートの関連レコードには「納税義務者」 <code>==SELLER「」</code> 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、米国売上税レポートの関連レコードには「納税義務者」 <code>==SELLER「」</code> と表示されます。)
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 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 <code>liu-jie@example.com</code> の場合、エントリーは <code>example.com</code> 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。

列名	説明
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 財務レポートの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	フィールドデモンストレーション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	料金の請求先となる支払いアカウントの 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 標準を使用しています。
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 ZIP コード	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。例えば、E メールアドレスが <code>liu-jie@example.com</code> 、エントリは <code>example.com</code> 。
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。

列名	説明
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。
契約総額 (USD)	の契約の合計値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)	の請求書に請求される金額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 財務レポートの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。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](#)
- の [billed_revenue_report](#) SQLクエリ GitHub。

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。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。

販売補償レポートおよびこのプログラムの一部として共有される情報は、お客様との非開示契約に基づく Amazon の機密情報、またはそのような契約が存在しない場合は AWS Marketplace 販売者の利用規約を構成します。この情報を使用できるのは、会社名、地域、AWS アカウント ID で AWS Marketplace 収益を担当者とマッピングすることで、販売担当者を補償するためだけです。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。このような情報の使用と共有は、お客様との機密保持契約における義務と、AWS Marketplace 販売者向けの利用規約の第 3.8 条を含むがこれらに限定されない 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 [販売補償レポート](#) として使用し AWS Marketplace [での販売者の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。

掲載スケジュール

このレポートは毎月、毎月 15 日の 00:00 に公開されます UTC。レポートは、その月の最終暦日の 00:00 から 23:59 UTC UTC までの前暦月を対象としています。例えば、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 を対象としています UTC。

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 AWS アカウント番号	料金の請求先となるアカウント。

列名	説明
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 標準を使用しています。
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	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	製品の一意的識別子。
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。
AWS 収益配分	決済時に請求金額から差し引かれた AWS 料金額。 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 返金共有	取引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、米国の売上税と使用税を引いた金額。

列名	説明
通貨	取引の通貨。例えば、トランザクションが米ドルの場合、エントリーは 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 財務レポートの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。

補足レポート

AWS Marketplace は [での販売者の配信データフィード AWS Marketplace](#)、を通じて、販売者の AWS Marketplace 出品に関連する 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、「[送信先の Amazon S3 バケットを作成する](#)」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。
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的識別子。

名前	説明
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](#) を搭載したダッシュボードに、財務データと売上データへのアクセスと分析に役立つグラフ、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銀行口座に AWS 収集および支払いされた資金に関する情報を提供します。
- [the section called “税金ダッシュボード”](#) - 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。

[the section called “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。

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、レガシーアクセスIAM許可で有効化された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す。以下のコード例で、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

請求収益ダッシュボードおよび回収と支払いのダッシュボード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。

次のコード例を使用して、現在および将来のデータフィードに基づいて、現在および将来の AWS Marketplace リソース (ダッシュボードとレポートを含む) へのアクセスを提供できます。

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



```

    "aws-marketplace:GetSellerDashboard"
  ],
  "Resource": [
    "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/*",
  ]
}]
}

```

または、次のコード例に示すように、特定のARNセクションを含めることで、1つ以上のダッシュボードへのアクセスを提供できます。例えば、請求収益ダッシュボード、契約と更新のダッシュボード、および使用状況ダッシュボードのみにアクセスできるようにするには、arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/BillingEvent_V1/Dashboard/CollectionsAndDisbursements_V1 のコード例からこの行を削除します。

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/BillingEvent_V1/Dashboard/BilledRevenue_V1",
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/BillingEvent_V1/Dashboard/CollectionsAndDisbursements_V1",
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/Agreement_V1/Dashboard/AgreementsAndRenewals_V1",
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/Usage_V1/Dashboard/Usage_V1",
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/TaxItem_V1/Dashboard/Tax_V1"
    ]
  }]
}

```

Note

AWS Identity and Access Management (IAM) ポリシーの作成の詳細については、「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。AWS Identity and Access Management

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](#)
- [税金ダッシュボード](#)

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。

データをエクスポートおよびダウンロードできます (.csv または Microsoft Excel ファイル) は、Amazon QuickSight ダッシュボード上の任意のチャート、グラフ、テーブルから取得できます。詳細については、「Amazon 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。

運用プロセスと財務プロセスについては、以下のトピックをご確認ください。

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

ダッシュボードは毎日 午前 0 時に更新されます UTC。請求書が 5 月 10 日 午前 0 時前に作成された場合 UTC、5 月 11 日の更新により請求書が表示されます。請求書が 5 月 10 日の午前 0 時以降に作

成された場合UTC、5月12日の更新によって請求書が表示されます。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、次の手順を実行します。

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 アグリーメントタブ](#)
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 オファ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、パートナー機会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。例えば、[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、特定のカスタマーアカウント 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、「[Amazon ユーザーガイド](#)」の「[Amazon でのデータのフィルタリング QuickSight](#)」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 レコード販売者	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： <ul style="list-style-type: none"> • AWS_INC: AWS, Inc. (米国ベース) の識別子。 • AWS_EUROPE: Amazon Web Services の識別子 EMEA SARL (ルクセンブルクにベース)。 • AWS_AUSTRALIA: Amazon Web Services Australia Pty Ltd の識別子 • AWS_JAPAN: Amazon Web Services Japan G.K. 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。

コントロール名	説明
リセラー AWS アカウント ID	顧客ISVへの再販のために から卸売で製品またはサービスを購入したアカウントの 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。

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、オファーの通貨と請求日付範囲に基づいてレコードを絞り込むフィルターが用意されています。請求日付範囲を使用して、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、メトリクス、傾向、内訳、詳細なデータセクションに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には、選択したフィルターを使用して、キーパフォーマンスインジケータ (KPI) が表示され、キー請求収益の数値の比較が視覚化されます。KPI は、指定した請求書の日付範囲と通貨のグロス収益、グロス返金、リスティング料金、卸売コスト (該当する場合)、販売者の純収益に表示されます。フィルター条件を更新して、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。日付、month-over-month、または などの指定された日付集計で傾向を表示 year-over-yearして

quarter-over-quarter、請求された収益を把握できます。請求済み収益のトレンド情報は、集計レベル、またはオファーの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。
- オファーの可視性別の請求トレンド - [プライベートオファー](#)、パブリックオファー (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ー数と総収益の推移のスナップショットを、オファーの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、レコードの販売者、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、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 請求される合計金額を含む、請求されたすべての売上が表示されます。は、次の 3 つの頻度を使用して顧客に 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

ては、「Amazon 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	形式の支払い期日 YYYY-MM-DD。
支払い条件	顧客の AWS 請求書支払い条件。
請求書 ID	顧客に請求される料金に割り当てられた AWS ID。
出品料金の請求書 ID	AWS Marketplace サブスクリプションが SARL、日本、またはオーストラリアの法人 (登録販売者) AWS EMEA を通じて取引される場合、販売事業者は販売者に対し、販売者リスティング料金に付加価値税 (VAT) を請求し、税準拠の請求書を渡す必要があります。該当するトランザクションの場合、リスト料金で VAT 評価された の請求書 ID は、ソフトウェアまたは製品サブスクリプションの請求書 ID とは異なります。AWS, Inc. からのトランザクションは、一覧表示料金請求書 ID が適用されないため、「Missing_listing_fee_invoice_id」の値を持ちます。
サブスクリバイバーの会社名	製品をサブスクライブしているアカウント名。
サブスクリバイバー 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。
サブスクリバイバーの 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

列	[Description] (説明)
	メールアドレスが liu-jie@example.com の場合、エントリーは 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が開始される日付。形式は <code>MM-DD-YYYY</code> 。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。形式は <code>MM-DD-YYYY</code> 。

列	[Description] (説明)
契約終了日	契約の有効期限が切れる日付。形式は <code>MM-DD-YYYY</code> 。計測/pay-as-you-go サブスクリプションの場合、この日付は <code>JAN-1-9999</code> 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。
支払いステータス	AWS が前回の支払い以降に銀行口座に資金を収集して支払ったことを確認するための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。支払い銀行トレース ID を使用して、販売者銀行が提供する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。
出品料金	請求額から差し引かれる AWS Marketplace 料金額。

列	[Description] (説明)
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。
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 タックスシェアの返金	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。
販売者の純収益	AWS Marketplace 手数料、返金、米国売上税と使用税を差し引いた、取引に対して請求される合計金額。
通貨	取引の通貨。例えば、トランザクションが米ドルの場合、エントリーは 必ず USD。

列	[Description] (説明)
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。レポート間でトランザクションを関連付けるために使用できません AWS Marketplace。
AWS レコード販売者	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。 <ul style="list-style-type: none">• AWS_INC: AWS, Inc. の識別子 (米国ベース)• AWS_EUROPE: Amazon Web Services の識別子 EMEA SARL (Luxembourg にベース)• AWS_AUSTRALIA: Amazon Web Services Australia Pty Ltd の識別子• AWS_JAPAN: Amazon Web Services Japan G.K. 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	登録されたオポチュニティの ISV で定義された説明。
再販会社名	顧客に再販 ISV するために、 から卸売コストで製品またはサービスを購入したアカウントの名前。
支払者 AWS アカウント ID	製品またはサービスを から卸売価格で購入 ISV し、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、エントリーは example.com です。

列	[Description] (説明)
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 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。

回収と支払いダッシュボード

コレクションと支払いダッシュボードは、前回の支払い以降に銀行口座に AWS 収集および支払いされた資金に関するデータを提供します。また、未払いの請求書のすべてのリストも表示されます。

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。請求書に記載されている全額が顧客から回収されるまで、資金の支払いは受け取れません。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、顧客の支払いにタイムリーにアクセスできるようになります。支払いが送信されてから 5 日後に作成されるレガシー [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

お客様は異なる支払い条件を使用しているため AWS、未収集の年齢カテゴリの一部の資金は、お客様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。

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。銀行に預金を受け取ってから 1 日以内に支払い済みの請求書が表示されることが予想されます。

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。例えば、[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、特定のカスタマーアカウント ID、サブスクライバーの会社名、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、「[Amazon ユーザーガイド](#)」の「[Amazon でのデータのフィルタリング QuickSight](#)」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。

コントロール名	説明
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 レコード販売者	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: <ul style="list-style-type: none"> • AWS_INC: AWS, Inc. の識別子 (米国ベース)。 • AWS_EUROPE: Amazon Web Services の識別子 EMEA SARL (ルクセンブルクにベース)。 • AWS_AUSTRALIA: Amazon Web Services Australia Pty Ltd (AWS オーストラリア) の識別子 • AWS_JAPAN: Amazon Web Services Japan G.K. の識別子。
支払いステータス	AWS が前回の支払い以降に銀行口座に資金を収集して支払ったことを確認するための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。

コントロール名	説明
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	から卸売で製品またはサービスを購入ISV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。
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

ダッシュボードのこのセクションには、オファー通貨と 2 つの異なる日付ディメンションに基づいてレコードを絞り込むフィルターが用意されています。日付フィールドの値が指定された日付の前か後か、日付範囲内かです。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払い出される請求書に結果を制限します。支払い期日には、支払い日に関係なく、指定された範囲内の期日の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

ダッシュボードのこのセクションには、支払い済み収益と未支払い収益の比較を視覚化するための主要パフォーマンスインジケータ (KPI) が表示されます。KPI グロス収益、純収益、卸売コスト (該当する場合)、支出額、および指定されたフィルター基準に対して未支出額に対してが表示されません。

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。
- 支払いの経過期間 - 純収益のスナップショットと、標準の経過期間売掛金バケット (期日なし、1 ~ 30 日、31 ~ 60 日など) によって分類された支払い済み請求書の数を提供します。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。
- 未払い支払いの経過期間 - 未払いおよび未払いの請求書の純収益と数を、期限を過ぎたバケット (期日なし、1 ~ 30 日、31 ~ 60 日など) 別にまとめたスナップショットを提供します。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。この傾向は、今日の日付から支払い期日までの日数を測定して、受信債権を表示します。

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、「Amazon 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	形式の支払い期日 YYYY-MM-DD。
支払い条件	顧客の AWS 請求書支払い条件。
請求書 ID	顧客に請求される料金に割り当てられた AWS ID。
出品料金の請求書 ID	AWS Marketplace サブスクリプションが 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 EMEA取引される場合、AWS EMEA販売のマーケットプレイスオペレーター (例: SARL) は販売者リスティング料金VATに を請求する必要があります。該当するトランザクションの場合、リスト料金で VAT評価された の請求書 ID は、ソフトウェアまたは製品サブスクリプションの請求書 ID とは異なります。
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。

列	[Description] (説明)
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が開始される日付。の形式になります MM-DD-YYYY。この日付は、将来の日付の契約であれば、承諾日とは異なる場合があります。

列	[Description] (説明)
契約終了日	契約の有効期限が切れる日付。形式は <code>MM-DD-YYYY</code> 。計測/pay-as-you-go サブスクリプションの場合、この日付は <code>JAN-1-9999</code> 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。形式は <code>MM-DD-YYYY</code> 。
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	AWS が前回の支払い以降に銀行口座に資金を収集して支払ったことを確認するための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、販売者銀行が提供する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。
卸売コスト	チャン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	チャン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。
販売者の純収益	AWS Marketplace 手数料、返金、米国売上税および使用税を差し引いた、取引に対して請求される合計金額。
通貨	取引の通貨。例えば、トランザクションが米ドルの場合、エントリーは 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。レポート間でトランザクションを関連付けるために使用できません AWS Marketplace 。

列	[Description] (説明)
AWS レコード販売者	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。 <ul style="list-style-type: none">• AWS_INC: AWS, Inc. の識別子 (米国ベース)• AWS_EUROPE: Amazon Web Services の識別子 EMEA SARL (ルクセンブルクにベース)• Amazon Web Services Australia Pty Ltd (AWS オーストラリア)• Amazon Web Services 日本合同会社
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	登録されたオポチュニティの ISV で定義された説明。
リセラーの会社名	顧客に再販 ISV するために、 から卸売コストで製品またはサービスを購入したアカウントの名前。
リセラー AWS アカウント ID	顧客 ISV への再販のために から卸売コストで製品またはサービスを購入したアカウントの 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。

列	[Description] (説明)
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 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トランザクションの合計金額。
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
支払い期間	資金が収集された売掛金の範囲を説明するカテゴリ (未払い、1~30 日、31~60 日など)。

税金ダッシュボード

税ダッシュボードには、での取引に対する米国および国際税に関する視覚化データと詳細なデータが表示されます AWS Marketplace。このダッシュボードには、財務オペレーションの AWS Marketplace 管理ポータル Insights タブからアクセスできます。

Amazon QuickSight ダッシュボードの任意のチャート、グラフ、およびテーブルから (.csv または MicrosoftExcelファイルとして) データをエクスポートおよびダウンロードできます。詳細について

は、「[Amazon 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。

運用プロセスと財務プロセスについては、以下のトピックをご確認ください。

トピック

- [税金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: 日付をフィルタリングする](#)
- [セクション 3: メトリクス](#)
- [セクション 4: 税金の傾向](#)
- [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 免除コード](#)

税金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受信した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 日の遅延が発生する可能性があります。

セクション 1: コントロール

ダッシュボードのこのセクションには、税金データを絞り込むためのフィルターが用意されています。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、「[Amazon ユーザーガイド](#)」の「[Amazon でのデータのフィルタリング QuickSight](#)」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。
納税対象顧客 AWS アカウント ID	製品サブスクリプションに対して税金が課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者の識別子。
顧客の国に税金がかかる	税制上の顧客に関連付けられている 2 文字の国コード
顧客の州または地域に税金がかかる	税制上の顧客に関連付けられた請求先住所の州または地域。
顧客の市区町村	税制上の顧客に関連付けられた請求先住所の市区町村
課税対象	税金、非税金、税金の返金の対象となる出典取引。
納税義務者	<p>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 が納税義務当事者である場合、売上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 販売者の税務ヘルプ」を参照してください。</p> <p>税金が徴収されない場合は、値が表示されません。販売者は税金徴収の責任を負うため、請求書ごとに一部の税金が徴収されたかどうかを判断する必要があります。</p>
税金タイプ	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

コントロール名	説明
	、Tax registration type、VAT (付加価値税)、CNPJ、IGST、ですCT。

セクション 2: 日付をフィルタリングする

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請求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。請求書日付内で選択されたフィルターは、メトリクス、傾向、内訳、および詳細なデータセクションに表示されるデータを更新します。例えば、日付フィルターを選択するとPast 30 days、すべてのセクションに過去 30 日以内に税金が課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、税金に関連するメトリクスを視覚化するための主要なパフォーマンス指標 (KPI) が表示されます。これには、税金の金額、非税金の金額、税金の金額、税金が課される請求書の数、取引の合計数が含まれます。フィルターセクションで日付フィルターを更新することで、日付範囲を更新できます。

セクション 4: 税金の傾向

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税金の傾向が表示されます。日次、、、など month-over-month、指定した日付集計で傾向を表示 year-over-yearして quarter-over-quarter、税金に関するインサイトを得ることができます。次の税金の傾向情報を利用できます。

- 税金の傾向 - 金額 - 選択した期間の税金と税金の両方のスナップショットを請求日別に表示します。
- 税金の傾向 - 請求書数 - 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、税金タイプ、製品タイトル、顧客地域別の税務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、での組織の取引に関する詳細な税金データが表示されます AWS Marketplace。

詳細なデータの説明

列	[Description] (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。
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	<p>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 が納税義務当事者である場合、売上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 販売者の税務ヘルプ」を参照してください。</p> <p>税金が徴収されない場合は、値が表示されません。販売者は税金徴収の責任を負うため、請求書ごとに一部の税金が徴収されたかどうかを判断する必要があります。</p>
トランザクションタイプコード	<p>取引のタイプコード。値には以下のものが含まれます。</p> <ul style="list-style-type: none"> • AWS – 転送税取引。 • REFUND – 全額または一部の返金。 • TAXONLYREFUND – 税金のみの返金。 <p>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</p>
製品 ID	ソフトウェア製品のわかりやすい一意的識別子。
製品のタイトル	購入された製品の名前。

列	[Description] (説明)
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に対して顧客が請求され、税金が課された日付。
納税対象顧客 AWS アカウント ID	税制上の顧客の AWS アカウント ID。
顧客の国に税金がかかる	税制上の顧客に関連付けられている 2 文字の国コード。
顧客の州または地域に税金がかかる	税制上の顧客に関連付けられた請求先住所の州または地域。
顧客の市区町村	税制上の顧客に関連付けられた請求先住所の市区町村。
税金がかかる顧客の郵便番号	税額計算に使用する郵便番号。
税金タイプ	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、Tax registration type、VAT、CNPJ、IGST です CT。
管轄レベル	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。
税制上の管轄区域	課税管轄地域の名前。
価格の納税義務タイプを表示する	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。
税務管轄率	管轄区域で適用される税率。
税額	管轄区域で課される税額。

列	[Description] (説明)
税金通貨	請求された税金の通貨。例えば、トランザクションが米ドルの場合、エントリは USD。
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。
税計算に使用される日付	取引の税金の計算に使用される日付。
顧客免除証明書 ID	免除証明書の証明書 ID。
顧客免除証明書 ID ドメイン	証明書が Amazon システムに保存されている場所。
顧客免除証明書レベル	免除を提供した管轄区域。
顧客免除コード	免除を指定するコード。例えば、RESALE と指定します。
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。
AWS 登録販売者	取引を促進した事業者の識別子。

Note

合計税額と購入者料金の合計は、請求済み収益ダッシュボードの「税AWS 配分」フィールドと「総収益」フィールドに移動しました。 <https://docs.aws.amazon.com/marketplace/latest/userguide/billed-revenue-dashboard.html>

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	チャリティ組織
DIRECT	直接支払いの許可
DISTSBT	ディストリビューター
DP	国際連合/外交省
DV	無効な退役軍人
EDI	教育機関
FG	連邦政府
HCP	医療プロバイダー
HO	勲章 (非営利または州)
INSUR	保険
IPM	IPM
LB	[Library] (ライブラリ)
MPU	複数の使用場所の免除、デジタル製品/ソフトウェア免除
該当なし	ネイティブアメリカ語
NAI	ネイティブアメリカ個人
NP	非営利組織
NPR	非営利の宗教組織

免除コード	説明
OT	その他
RESALE	リセラー
SBE	中小企業の免除
SLG	州/地域政府

EU 免除コード

免除コード	説明
SpecialOrg	の請求が免除される組織VAT。
SpecialZone	免税の対象となる地域。
SplitPayment	購入者は、請求書VATに記載されているすべての支払いを税務機関に直接行います。

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

契約および更新ダッシュボードには、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。dashbaord は、期限切れのサブスクリプションを追跡して更新を有効に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、セールスオペレーションを選択し、インサイトタブを選択します。

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

ダッシュボードの本セクション内には、契約および更新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。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、[「Amazon ユーザーガイド」の「Amazon でのデータのフィルタリング QuickSight」](#)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、契約書の一意的な契約データフィードリファレンス。

コントロール名	説明
CPPO フラグ	チャン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、登録販売者がチャン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。
ISV 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的識別子。
製品のタイトル	製品のタイトル。
再販承認 ID	がチャンネルパートナーISVに提供する再販認証の ID。
再販承認名	がチャンネルパートナーISVに提供する再販認証の名前。
リセラー AWS アカウント ID	顧客ISVへの再販のために から卸売で製品またはサービスを購入したアカウントの 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

ダッシュボードのこのセクションには、キーパフォーマンスインジケータ (KPI) が表示され、キーアグリーメントデータの比較を視覚化できます。アクティブなアグリーメントの数と終了されたアグリーメントの数には 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。トレンドは、日次、month-over-month、など、指定された日付集計別に表示 year-over-yearして quarter-over-quarter、アグリーメント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では、アクティブなサブスクライバー、および ISVs/リセラーの会社名全体でのビジネスのメトリクスを提供します。契約数 IDs、終了契約 IDs、オファー IDs、更新済み契約 IDs、サブスクライバーを表示できます。

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、サブスライバー、再販承認、再販業者、およびの詳細なデータが表示されますISVs。

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、「Amazon 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)
サブスライバーの会社名	サブスライバーの会社の名前。
サブスライバーAWSアカウント ID	製品をサブスライブしているアカウントの ID。
サブスライバーの E メールドメイン	製品をサブス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。
サブスライバーの国	製品をサブス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
サブスライバーの州または地域	製品をサブス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
サブスライバーの市町区村	製品をサブス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。
サブスライバーの郵便番号	製品をサブス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。

列	[Description] (説明)
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的識別子。
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。
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が開始される日付。MM-DD- の形式になりますYYYY。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- の形式ですYYYY。
契約終了日	MM-DD- の形式で、契約の有効期限が切れる日付YYYY。計測/pay-as-you-go 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。

列	[Description] (説明)
ステータス	<p data-bbox="829 222 1503 304">アグリーメントの現在のステータス。サポートされているステータスは次のとおりです。</p> <ul data-bbox="829 352 1503 1812" style="list-style-type: none"><li data-bbox="829 352 1503 434">• アクティブ – 契約の条項の一部またはすべてが有効になります。<li data-bbox="829 457 1503 632">• 終了 – AWS開始された終了イベントにより、契約は事前に合意された終了日より前に終了しました。最も一般的なのは支払いの失敗です。<li data-bbox="829 655 1503 737">• キャンセル済み – アクセプターは終了日より前に契約を終了することを選択しています。<li data-bbox="829 760 1503 842">• 期限切れ – 契約は、事前に合意された終了日に終了しました。<li data-bbox="829 865 1503 947">• 置き換え済み – 契約は代替オファーを使用して置き換えられました。<li data-bbox="829 970 1503 1052">• 更新済み – 契約は、自動更新などの機能を使用して新しい契約に更新されました。<li data-bbox="829 1075 1503 1205">• アーカイブ済み – 契約が終了しました。契約の終了、終了、キャンセル、期限切れの正確な理由が指定されていません。<li data-bbox="829 1228 1503 1451">• ロールバック – エラーのため、契約リビジョンがロールバックされました。以前のリビジョンがアクティブになりました。非アクティブな契約リビジョンにのみ適用されます。<li data-bbox="829 1474 1503 1648">• 置き換え済み – 契約リビジョンはアクティブでなくなり、別の契約リビジョンがアクティブになりました。非アクティブな契約リビジョンにのみ適用されます。<li data-bbox="829 1671 1503 1801">• ロールバック – エラーのため、契約リビジョンがロールバックされました。以前のリビジョンがアクティブになりました。非アク

列	[Description] (説明)
	タイプな契約リビジョンにのみ適用されます。
推定契約値	「契約総額」または「購入額」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、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	契約の推定コストに関連するオファー料金通貨。
リセラーの会社名	顧客に再販ISVするために、 から卸売コストで製品またはサービスを購入したアカウントの名前。
リセラーAWSアカウント ID	製品またはサービスを から卸売価格で購入ISVし、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	登録されたオポチュニティの ISVで定義された説明。
CPPO フラグ	Amazon の計算フィールド QuickSight。

列	[Description] (説明)
契約の意図	<p>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:</p> <ul style="list-style-type: none"> • 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	<p>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。</p>
次の契約 ID	<p>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。</p>
以前のオファー ID	<p>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。</p>

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 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	の計算フィールド QuickSight。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視覚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の消費を追跡し、製品サポート、料金、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、および製品の中止に関する意思決定を行うことができます。ダッシュボードには、過去 6 か月間のデータがローリングウィンドウとして表示されます。

ダッシュボードを開くには、 を起動し AWS Marketplace 管理ポータル、セールスオペレーションを選択し、インサイトタブを選択します。

データを .csv または Microsoft Excel Amazon QuickSight ダッシュボード上の任意のチャート、グラフ、テーブルのファイル。詳細については、「[Amazon 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。

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 Machine Image (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 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

ダッシュボードは毎日午後 4 時 PST (午前 0 時) に更新されます UTC。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。

運用プロセスと財務プロセスについては、以下のトピックをご確認ください。

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、[「Amazon ユーザーガイド」の「Amazon でのデータのフィルタリング QuickSight」](#)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。
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、データをレポートに結合したり、AWS Marketplace Metering Service で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、契約書の一意的な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。

コントロール名	説明
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。
リセラー AWS アカウント ID	顧客ISVへの再販のために から卸売で製品またはサービスを購入したアカウントの ID。
再販承認 ID	顧客ISVへの再販のために から卸売で製品またはサービスを購入したアカウントの 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

ダッシュボードのこのセクションには、主要なパフォーマンスインジケータ (KPI) が表示され、推定使用量単位、使用状況のあるお客様、使用状況のある製品などの消費に関連するメトリクスを視覚化できます。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。トレンドは、日次、または などの指定された日付集計別に表示 year-over-year して month-over-month quarter-over-quarter、使用状況を把握できます。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

ダッシュボードのこのセクションには、会社名、製品タイトル、ディメンションキー、選択したユニットタイプのオファーにわたるビジネスIDsにおける推定使用量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

ダッシュボードのこのセクションには、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 () 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 up-to-date の追跡のために、毎月ダッシュボードを毎日訪問する必要がある場合があります。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、「[Amazon 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、エントリーは 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的識別子。
製品 ID	製品のわかりやすい一意的識別子。
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UTCした の日付タイムスタンプ。
契約開始日	顧客の製品サブスクリプションが開始されUTC したときの の日付タイムスタンプ。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	契約の有効期限UTCが切れる の日付。計測/pay-as-you-go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	購入者が Amazon EC2 インスタンスを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、トランザクションが米ドルの場合、エントリーは USD です。

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 abc@example.com の場合、エントリーは 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、エントリーは 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

列	[Description] (説明)
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
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	顧客に再販ISVするために、 から卸売コストで製品またはサービスを購入したアカウントの名前。
リセラー AWS アカウント ID	製品またはサービスを から卸売価格で購入ISVし、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的識別子。
再販承認名	登録された再販オポチュニティの一意的名前。
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	チャン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、登録販売者がチャンネルパートナーです。いいえの場合、登録販売者は製品メーカー () ですISV。
ISV 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。

マーケティング用のダッシュボード

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](#)

パフォーマンスダッシュボードの一覧表示

顧客契約ダッシュボード

カスタマーアグリーメントダッシュボードには、で製品をサブスクライブする契約と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ロック解除するには、[AWS Marketplace Seller Prime](#) プログラムを登録する必要があります。
- 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、マーケティング 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: IAM アクセス許可の更新](#)
- [セクション 2: フィルター](#)
- [セクション 3: 日付フィルターの詳細](#)
- [セクション 4: 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 5: 顧客メトリクス](#)

セクション 1: IAM アクセス許可の更新

カスタマーアグリーメントダッシュボードにアクセスするには、[AWS Marketplace Seller Prime](#) に登録する必要があります。

登録後、既存のIAMポリシーを更新する必要があります。2024年9月以降、販売者ダッシュボードへのアクセスにレガシーIAMアクセス許可を使用することはサポートされなくなります。アクセスIAM許可を更新するには、次の例に示す Amazon リソースネーム (ARN) 形式を使用します。

この例では、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許可します。


```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace:::AWSMarketplace/*",
    ]
  }]
}
```

この例では、`*`を含めることで1つのダッシュボードへのアクセスを許可しますARN。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::<123456789012>:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/CustomerAgreements_V1"
    ]
  }]
}
```

Note

レガシーIAMアクセス許可の詳細については、「[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。IAM

セクション 2: フィルター

次のフィルターを使用して、アグリーメント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。

コントロール名	説明
日付フィルター	過去 30、60、90 日間のデータ、過去 12 か月 (TTM)、および年初来のデータ (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します。
顧客業界	販売者製品をサブスクライブした顧客のために AWS を定義する業界。
顧客セグメント	販売者製品をサブスクライブした顧客のために 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 Amazon ユーザーガイド」の「Amazon でのデータのフィルタリング QuickSight」](#)を参照してください。 QuickSight

セクション 3: 日付フィルターの詳細

この日付フィルターは、カスタマーアグリーメント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 year-to-date フィルター値を使用すると、following メトリクスが表示されます。

メトリクス	説明
公開オファー契約数	内で少なくとも 1 日アクティブな日がある公開オファーをオファー可視性とする契約の数 YTD。
プライベートオファー契約数	内で少なくとも 1 日アクティブな日を持つ、プライベートオファーをオファーの可視性とするアグリーメントの数 YTD。
新しい公開オファー契約数	内で承認日があるオファーの可視性として公開オファーを含むアグリーメントの数 YTD。

メトリクス	説明
新しいプライベートオファー契約数	内で承諾日を持つオファーの可視性としてプライベートオファーを持つアグリーメントの数 YTD。
アクティブなお客様	サブスクライバーAWSアカウント ID として識別され、内で少なくとも1つのアクティブな契約を持つ一意の顧客の数 YTD。アクティブアグリーメントは、期間中に少なくとも1日アクティブなアグリーメントとして定義されます。
新しい有料顧客	サブスクライバーAWSアカウント ID として識別され、内に最初の請求月がある一意の顧客の数 YTD。

セクション 4: 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。主要業績評価指標 (KPIs) には、公開オファー契約の数、プライベートオファー契約の数、新しい公開オファー契約の数、新しいプライベートオファー契約の数が含まれます。ボリュームと割合には、year-over-year または period-over-period の変更が表示されます。フィルターセクションで日付フィルターを更新することで、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) とアクセプター (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せん。

次の表は、契約ステータスを一覧表示し、説明しています。

ステータス	説明
[アクティブ]	契約の条項の一部またはすべてが有効である。

ステータス	説明
期限切れ	契約は、事前に合意された終了日に終了しました。
キャンセル済み	アクセプターは、契約終了日より前に契約を終了することを選択します。
終了済み	支払いの失敗など、AWSが開始した終了イベントにより、契約は事前に合意された終了日より前に終了しました。
更新済み	この契約は、自動更新などの機能を使用して新しい契約に更新されました。
置換済み	契約は代替オファーを使用して置き換えられました。

セクション 5: 顧客メトリクス

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顧客傾向が表示されます。KPIs には、アクティブな顧客の数と新しい有料顧客の数が含まれます。月次傾向 - グラフ内のフィルターに基づいて、アクティブな顧客数と新しい支払い顧客数を月次で表示します。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、アクティブな顧客の数と新規支払い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、各顧客セグメントまたは顧客業界が顧客メトリクスの合計にどのように貢献しているかを理解できます。

Note

アグリーメントには特定のステータスがあります。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。

次の表は、メトリクスを一覧表示し、説明しています。

メトリクス	説明
公開オファー契約	公開オファーの可視性IDsを持つ契約として識別される契約の数。契約の長さは、選択した日付範囲と少なくとも1日重複しています。長さは、契約開始日と契約終了日の間で識別されます。
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IDsを持つ契約として識別される契約の数。契約の長さは、選択した日付範囲と少なくとも1日重複しています。長さは、契約開始日と契約終了日の間で識別されます。
新しい公開オファー契約	公開オファーの可視性を持つ契約IDとして識別される契約の数。契約の承諾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IDsを持つ契約として識別される契約の数。契約の承諾日は、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IDsが1つ以上あるサブスクライバーAWSアカウント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
新しい有料顧客	選択した日付範囲内に最初の請求月IDsがあるサブスクライバーAWSアカウントとして識別された顧客の数。

契約と収益の詳細については、このセクションの [契約および更新ダッシュボード](#) と [請求済み収益ダッシュボード](#) の両方を参照してください。

パフォーマンスダッシュボードの一覧表示

一覧表示パフォーマンスダッシュボードには、AWS マーケットプレイスのリスト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と、顧客が製品をサブスクラ

イブ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のロックを解除するには、[AWS Marketplace Seller Prime](#) プログラムを登録する必要があります。
- 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、[マーケティング](#) 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: IAM アクセス許可の更新](#)
- [セクション 2: フィルター](#)
- [セクション 3: 日付フィルターの詳細](#)
- [セクション 4: 概要メトリクス](#)
- [セクション 5: 月次トレンドとファネル変換](#)
- [セクション 6: チャネル別のトラフィック傾向](#)
- [セクション 7: 月間トラフィックと契約メトリクス](#)
- [セクション 8: 一意の訪問者のウェブトラフィックソース](#)
- [セクション 9: ウェブトラフィックソースの内訳](#)

セクション 1: IAM アクセス許可の更新

登録後、IAMアクセス許可を更新する必要があります。2024 年 9 月をもって、レガシーアクセス IAM許可で有効化された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す。アクセス許可を更新するには、次の例に示す Amazon リソースネーム (ARN) 形式を使用します。

この例では、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許可します。

```
{
```

```

"Version": "2012-10-17",
"Statement": [{
  "Effect": "Allow",
  "Action": [
    "aws-marketplace:GetSellerDashboard"
  ],
  "Resource": [
    "arn:aws:aws-marketplace:::AWSMarketplace/*",
  ]
}]
}

```

この例では、 を含めることで 1 つのダッシュボードへのアクセスを許可しますARN。

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::<123456789012>:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/ListingPerformance_V1"
    ]
  }]
}

```

AWS Identity and Access Management (IAM) ポリシーの作成については、「IAMser Guide」のIAM「[ポリシーの作成](#)」を参照してください。

セクション 2: 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、60、90 日間、過去 12 か月 (TTM)、および年初来 () が含まれますYTD。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、「[Amazon ユーザーガイド](#)」の「[Amazon でのデータのフィルタリング QuickSight](#)」を参照してください。 QuickSight

セクション 3: 日付フィルターの詳細

この日付フィルターは、一覧表示された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 year-to-date 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
Unqie 訪問者	の AWS Marketplace リスティングページにアクセスした一意の個人の数YTD。
新しい公開オファー契約	内で少なくとも 1 日アクティブな日を持つ公開オファーをオファーの可視性とする契約の数 YTD。
新しい一意の訪問者	で初めて AWS Marketplace リスティングページにアクセスした一意の個人の数YTD。
一意の訪問者を返す	選択した年の初め以前に AWS Marketplace のリストページを訪問し、で再度訪問した一意の個人の数YTD。

セクション 4: 概要メトリクス

ダッシュボードのこのセクションには、トラフィックと公開オファー契約の概要が表示されます。主要なパフォーマンス指標 (KPIs) には、一意の訪問者の数、新しい公開オファー契約の数、新しい一意の訪問者の数、および戻りの一意の訪問者の数が含まれます。ボリュームとパーセンテージには、year-over-year または period-over-period の変更が表示されます。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することで、日付範囲を更新できます。

メトリクス	説明
一意の訪問者	AWS Marketplace リスティング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	提案者 (製品またはサービス所有者) とアクセプター (顧客) の間で署名された契約で、製品の使用を開始します。
新しい公開オファー契約	選択した日付範囲内に承認日があるオファーの可視性としてパブリックオファーを持つアグリーメントの数。には、ステータスがアクティブ、期限切れ、キャンセル、終了が含まれます。
[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です。期限切れ - 契約は事前に合意された終了日に終了しました。キャンセル済み - アクセプターは終了日より前に契約を終了することを選択しています。終了 - 支払いの失敗などAWS、が開始した終了イベントにより、契約は事前に合意された終了日より前に終了しました。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ました。置き換え済み - 契約は代替オファーを使用して置き換えられました。

セクション 5: 月次トレンドとファネル変換

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トラフィックとアグリーメントの傾向が表示されます。主要業績評価指標には、新しい公開オファー契約を参照する、一意の訪問者の数と新しい契約の数が含まれます。月別の一意の訪問者 - は、一意の訪問者の数を毎月表示します。1人の一意のビジターが複数の月を訪問できるため、月間の一意のビジターの合計は、サマリーメトリクスセクションの一意のビジターの合計を超えることができます。月別の新しい契約 - は、新しい公開オファー契約の数の月間ビューを提供します。ファネル変換 - は、step-by-step 変換ビューを提供します。これには、リストページ、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、一意の訪問者、ページビュー、一意の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変換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
一意の訪問者	AWS Marketplace リスティング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 リスティングページへのアクセス回数。

メトリクス	説明
一意のアカウント	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、訪問者が認証した後にのみ適用されます。

セクション 6: チャンネル別のトラフィック傾向

ダッシュボードのこのセクションでは、毎月のトラフィック傾向について詳しく説明します。一意の訪問者とページビューは、バーが表示するものとして選択できます。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 7: 月間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、一意の訪問者、新しいアグリーメント、変換率、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。変換率は、毎月の一意の訪問者で割った新しい契約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 Month 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 8: 一意の訪問者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による一意の訪問者の詳細を説明します。カテゴリには、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 有機的検出、AWS および 主導のキャンペーンが含まれます。これらのカテゴリは、訪問者または訪問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
販売者の有機的検出	ウェブサイトまたはアプリ内通知のアクションの呼び出しと検出ポイントを通じてリスティングに到着する一意の訪問者。

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ンネルを通じてリスティングに到着する一意の訪問者。
AWS 有機検出	有機 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リスティングに到着する一意の訪問者。
AWS マーケティングキャンペーン	AWS主導型キャンペーンと有料プロモーションを通じてリスティングに到着する一意の訪問者。

セクション 9: 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、トラフィックを駆動する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 2 つの主要なカテゴリとして 主導型と販売者主導型が含まれます。
- **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 有機的検出、AWSおよび 主導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。
- **パブリッシャー** - Facebook、Google、Seller-led など LinkedIn 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` を使用できます。

メトリクス	説明
属性アグリーメント	各プロモーションの訪問によって寄与された契約の合計。
変換率	各プロモーションの属性付きアグリーメントを一意的訪問者で割ったもの。

Amazon QuickSight ダッシュボードの任意のチャート、グラフ、テーブルから、.csv または Microsoft Excel ファイルとしてデータをエクスポートおよびダウンロードできます。詳細については、「Amazon 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。

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、信頼できる調達ソフトウェアを保護し、基準を満たす機能です。AWS Marketplace Vendor Insights を使用すると、購入者は 1 つ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、以下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
- 業界標準の監査レポート (国際標準化機構 27001 ISO など)
- 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、[AWS Config を有効にする](#) で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、[AWS Audit Manager を有効にする](#) で 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 up-to-date を自動的に保持する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 () が含まれます CAIQ。
- 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 rules を使用して AWS リソースの設定を評価することで生成されます。は証拠 AWS Audit Manager をキャプチャし、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 ルールの作成と AWS Audit Manager 自動評価。

- 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 レポートとSOC2タイプ II レポート – コントロールカテゴリは、国際標準化機構 (ISO) およびシステムおよび組織コントロール (SOC2) レポート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ステップについて説明します。

SaaS リストで 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、認証、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る] を選択します。
5. フォームに入力し、[送信] を選択します。

AWS Marketplace 販売者オペレーション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

認証は、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、以下の認証をサポートしています。

- FedRAMP 認証 – 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) の要件に準拠し、個人データと個人のプライバシー権を保護する
- HIPAA コンプライアンスレポート – 医療保険の相互運用性と説明責任に関する法律 (HIPAA) の規制の遵守を示し、保護された医療情報を保護します
- ISO/IEC 27001 audit report – Confirms compliance with International Organization for Standardization (ISO)/International 情報セキュリティ標準を強調した電気技術委員会 (IEC) 27001
- PCI DSS 監査レポート – Security Standards Council によって設定されたPCIセキュリティ標準への準拠を示します。
- SOC 2 タイプ 2 監査レポート – Service Organizational Control (SOC) データプライバシーとセキュリティコントロール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。

認証ステータスは、認証の詳細が検証ValidationPendingされるまでに変わります。データソースの処理中と処理後に代替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 – AWS Marketplace Vendor Insights が読み取るために、データソースの外部ソースリファレンス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リファレンスは、VendorInsights が読み取ることができなくなります。
- 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されています。この段階では実行できるアクション項目はありません。ステータスは、使用可能 ResourceNotSupported、またはに更新されず ValidationFailed。
- 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、以下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チブアンケート (CAIQ)。詳細については、Cloud Security Alliance ウェブサイトの「[とは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リファレンスは、VendorInsights が読み取ることができなくなります。
- ResourceNotFound – データソースの外部ソースリファレンスは、VendorInsights が読み取ることができなくなります。
- 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されています。この段階では実行できるアクション項目はありません。ステータスは、使用可能 ResourceNotSupported、または に更新されます ValidationFailed。
- 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、以下の AWS のサービスと 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、AWS CloudFormation スタックと を使用します。スタックとは StackSets、必要なリソースのプ

ロビジョニングと設定を行います。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。

の詳細については 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、AWS Config 必要な AWS Config ルールを設定するためのパフォーマンスパックをデプロイします。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、アカウントでのアクティビティの増加に気付くかもしれません。最初のブートストラッププロセス中に、は、記録するために AWS Config 選択した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えることがあります。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 オフの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 - このスタックセットは、以下の 2 つの Amazon Simple Storage Service (Amazon S3) バケットを作成します。
 - vendor-insights-stack-set-output-bucket-{account number} – このバケットには、スタックセット実行からの出力が含まれています。AWS Marketplace 販売者オペレーションチームは、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 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 Audit Manager リソースの評価を読み取るアクセスを AWS Marketplace Vendor Insights に提供します。 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が AWS Marketplace Vendor Insights プロファイルの設定を支援します。
 - このAWSAuditManagerAdministratorAccessロールには AWS Audit Manager、 を有効または無効にする、設定を更新する、評価、コントロール、フレームワークを管理するための管理アクセス権限があります。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、 AWS CloudFormation スタックとを使用します。スタックとは StackSets、必要なリソースのプロビジョニングと設定を行います。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は、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 StackSets できます。

の詳細については CloudFormation StackSets、 AWS CloudFormation ユーザーガイドの「[の使用 AWS CloudFormation StackSets](#)」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 CloudFormation StackSets で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することができます。このスタックは、 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で作成します StackSets。

を作成する VendorInsightsPrerequisiteCFT スタック

VendorInsightsPrerequisiteCFT CloudFormation スタックを実行することで、スタックセットのオンボーディングを開始するIAMアクセス許可を設定します。

を作成するには VendorInsightsPrerequisiteCFT スタック

1. ウェブサイトの [AWS Sample Repo for Vendor Insights テンプレートフォルダ](#) から最新の VendorInsightsPrerequisiteCFT.yml ファイルを確認してダウンロードします GitHub。
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> でコンソールを開きます AWS CloudFormation 。
3. CloudFormation コンソール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の作成と新しいリソース (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。

[Next (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。

[送信] を選択します。

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

を作成する VendorInsightsOnboardingCFT スタックセット

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。

複数のアカウントソリューションがある場合、または個別の販売アカウントと本番アカウントがある場合は、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 これにより、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実行できます。

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。

を作成するには VendorInsightsOnboardingCFT スタックセット

1. ウェブサイトの [AWS Sample Repo for Vendor Insights テンプレートフォルダ](#) から最新の VendorInsightsOnboardingCFT.yml ファイルを確認してダウンロードします GitHub。
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、コンソールを AWS CloudFormation <https://console.aws.amazon.com/cloudformation> で開きます。
3. CloudFormation コンソールナビゲーションペインで、 の作成 StackSet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、管理者ロールでIAMロール名 を選択し、ドロップダウンからAWSVendorInsightsOnboardingStackSetsAdminロール名 を選択します。
5. 実行ロール名**AWSVendorInsightsOnboardingStackSetsExecution**として を入力します。 IAM
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、 の自動 AWS Audit Manager 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ができる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、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. 必要に応じて StackSet オプションを設定します。[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。[Next (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、単一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。

[送信] を選択します。

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、AWS アカウント 認証情報を保護し、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せん。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 Vendor AWS Marketplace 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 をセットアップ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ァイルを確認する

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ァイルを確認することが重要です。プロファイ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、このプロファイルは購入者には表示されません。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ァイ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、[AWS Support](#) に連絡してサポートケースを開始します。

購入者が表示するようにセキュリティプロファイルを表示するには

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ァイ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 point-in-time 体制です。AWS Marketplace Vendor Insights では、スナップショットを使用して販売者の製品をいつでも評価できます。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 スナップショットの概要セクションで、作成およびリリーススケジュールに関する次のスナップショットの詳細を表示できます。

- [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。
- [次回の予定リリース] - 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. にサインイン AWS Management Console し、[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。[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. にサインイン AWS Management Console し、[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. にサインイン AWS Management Console し、[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. にサインイン AWS Management Console し、[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

プロフィー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. にサインイン AWS Management Console し、[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. にサインイン AWS Management Console し、[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、販売者のデータ、評価、販売者の自己認証、および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、Amazon EC2インスタンス、Amazon S3 バケットなどの特定のリソースに対してアクションを許可することを許可できます。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、複数のアクセス許可を組み合わせた が AWS Marketplace 作成および管理する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、「[サービス認証リファレンス](#)」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ポリシーに結合して、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、ユーザーは既存のと AWS Marketplace Vendor Insights プロファイルDataSource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、 の開始方法と使用に関する詳細が記載されています IAM。

- [IAM でのセキュリティのベストプラクティス](#)
- [IAMポリシーの管理](#)
- [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 ユーザーグループの作成](#)
- [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、最もセキュリティに敏感な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の恩恵を受けることができます。

セキュリティは、AWS とユーザーの間で責任を共有します。[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。は、安全に使用できるサービス AWS も提供します。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。に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、を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます AWS Marketplace。次のトピックでは、セキュリティとコンプライアンスの目標を達成するために 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と保護 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。

で提供する製品に関するセキュリティや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。

- [AMIの 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 のセキュリティの詳細については、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 販売者の 管理ポリシー AWS Marketplace](#)
- [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 のアクセス許可](#)
- [での再販認証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 計測API呼び出しのログ記録 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 リソースの使用を許可する (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。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、Amazon EC2インスタンス、Amazon S3 バケットなどの特定のリソースに対してアクションを許可することを許可できます。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SQSのように Amazon にリクエストを送信するたびに、サービスは Jane にアクセス許可があるかどうかを確認します。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、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、複数のアクセス許可を組み合わせた が AWS Marketplace 作成および管理するポリシーもあります。

以下のリソースでは、 の開始方法と使用に関する詳細情報を提供しますIAM。

- [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ポリシーの管理](#)
- [IAM ユーザーグループにポリシーをアタッチする](#)
- [IAM ID \(ユーザー、グループ、ロール\)](#)
- [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 に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、 でIAMコンソールを開きます<https://console.aws.amazon.com/iam/>。
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. 認証情報のダウンロード 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 に移動する必要がありますURL。URL これは、先ほどダウンロードした認証情報ファイルにあり、IAMコンソールでも使用できます。詳細については、[「ユーザーガイド」のIAM「ユーザーが にサインイン AWS アカウント」する方法IAM](#)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。管理権限を持つユーザーを自分で作成する方法については、[「ユーザーガイド」の「管理ユーザーの作成」](#)を参照してください。
IAM

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. でIAMコンソールを開きます<https://console.aws.amazon.com/iam/>。
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. でIAMコンソールを開きます<https://console.aws.amazon.com/iam/>。
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コンソールを開きます <https://console.aws.amazon.com/iam/>。
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

でユーザーを作成するとIAM、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。そのためには、URLに関連付けられている一意の を使用する必要があります AWS アカウント。サインインを取得URLしてユーザーに配布できます。

アカウントの一意のサインインを取得するには URL

1. でIAMコンソールを開きます <https://console.aws.amazon.com/iam/>。
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で、IAMユーザーのサインインリンクを見つけ、次のような形式のサインインリンクを書き留めます。

```
https://AWS_account_ID.signin.aws.amazon.com/console/
```

Note

URL サインインページの 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、 をカスタマイズ 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、[「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。IAM

4. これをURL、それぞれに作成したサインイン認証情報とともに AWS Marketplace、 を使用できる社内のユーザーに配布します。にアクセスする前に、アカウントの一意のサインインを使用してURLサインインするように指示します AWS Marketplace。

AWS Marketplace 販売者のポリシーとアクセス許可

AWS Marketplace には、 で使用できる複数の管理ポリシーがあります 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により、以下を実行できます。

- 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 AWS Data Exchange のポリシーとアクセス許可の詳細については、AWS「Data Exchange ユーザーガイド」の [AWS「Data Exchange でのアイデンティティとアクセスの管理」](#) 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 販売者向けポリシー

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、マネージドポリシーを使用して、購入したサブスクリプションを管理できます。で使用する マネージドポリシーの名前は、で AWS Marketplace 管理

ポータル 始まりますAWSMarketplaceSeller。でポリシーを検索するときはIAM、で始まるポリシー名を必ず検索してくださいAWSMarketplaceSeller。

AWS Marketplace には、特定のシナリオ向けの特殊なマネージドポリシーも用意されています。AWS Marketplace 販売者向けの AWS マネージド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 販売者の 管理ポリシー AWS Marketplace](#)。

AWS Marketplace 販売者のアクセス許可

のIAMポリシーでは、次のアクセス許可を使用できます AWS Marketplace 管理ポータル。

aws-marketplace-management:PutSellerVerificationDetails

Know Your Customer (KYC) プロセスを開始するアクセスを許可します。

aws-marketplace-management:GetSellerVerificationDetails

でKYCステータスを表示するアクセスを許可します AWS Marketplace 管理ポータル。

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

に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 メールコンタクト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できるようにします。

payments:CreatePaymentInstrument

銀行口座を に追加するための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アクセスを許可します AWS Marketplace 管理ポータル。

aws-marketplace:ListTasks

販売者のアクションが保留になっているタスクのリストを表示するアクセスを許可します。これはレガシーIAMポリシーアクションであり、不要になりました。

aws-marketplace:DescribeTask

販売者のアクションが保留になっているタスクの詳細を表示するアクセスを許可します。これはレガシーIAMポリシーアクションであり、不要になりました。

aws-marketplace:UpdateTask

販売者のアクションが保留になっているタスクを編集するアクセスを許可します。これはレガシーIAMポリシーアクションであり、不要になりました。

aws-marketplace:CompleteTask

販売者のアクションが保留になっているタスクに対する編集を送信するアクセスを許可します。これはレガシーIAMポリシーアクションであり、不要になりました。

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 [\[ファイルのアップロード\]](#)、[\[オファー\]](#)、[\[パートナー\]](#) ページにアクセスするために必要です。

Note

アクセスに 設定 タブの表示を許可するには、このアクセス許可、アクセスListEntity許可、および次の Amazon リソースネーム (ARN): を使用できますarn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}。

aws-marketplace:DescribeEntity

でオブジェクトの詳細を表示する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 [\[ファイルのアップロード\]](#)、[\[オファー\]](#)、[\[パートナー\]](#)、[\[契約\]](#) ページにアクセスするために必要です。

Note

アクセスに 設定 タブの表示を許可するには、このアクセス許可、アクセスDescribeEntity許可、および次の を使用できますARN。arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/*

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 [\[ファイルのアップロード\]](#)、[\[オファー\]](#)、[\[パートナー\]](#)、[\[契約\]](#) ページに変更を加えるために必要です。

Note

アクセスを で販売者として登録できるようにするには AWS Marketplace、このアクセス許可、`catalog:ChangeType: "CreateSeller"`条件キー、および次の を使用できますARN。 `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`

で販売者プロフィールの更新へ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 を使用できますARN。 `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アクセスを許可するには、このアクセス許可、`catalog:ChangeType: "UpdateDisbursementPreferences"`条件キー、および を使用できますARN。 `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`

aws-marketplace:SearchAgreements

[契約](#) ページの契約の概要リスト、および [パートナー](#) ページの ISVs とチャネルパートナー間の機会を表示できます。

aws-marketplace:DescribeAgreement

契約ページで大まかな契約の詳細を表示し、パートナーページで ISVs とチャネルパートナー間の機会を表示できます。

aws-marketplace:GetAgreementTerms

契約ページですべての契約期間の詳細を表示し、パートナーページで ISVs とチャネルパートナー間の機会を表示できます。

aws-marketplace:GetSellerDashboard

AWS Marketplace 管理ポータルの [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。

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。

上記のアクセス許可を 1 つの IAM ポリシーに結合して、必要なアクセス許可を付与できます。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。

KYC の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。

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

例 2: プライベートオファ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

[\[契約\]](#) ページを表示および使用してプライベートオファ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。

```
{
  "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。

```

{
  "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、内の個々のページへのアクセスを許可するための個別のIAMグループを作成することもできます AWS Marketplace 管理ポータル。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 Insights ページへのアクセス許可を付与し、別のグループを作成し、そのグループにファイルアップロードページへのアクセス許可を付与します。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IAM](#)」の「[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。

AWS 販売者の 管理ポリシー AWS Marketplace

AWS マネージドポリシーは、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 マネージドポリシーは、ユーザー、グループ、ロールへのアクセス許可の割り当てを開始できるように、多くの一般的なユースケースに対するアクセス許可を提供するように設計されています。

AWS マネージドポリシーは、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権限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。ユースケース別に[カスタマーマネージドポリシー](#)を定義して、マネージドポリシーを絞り込むことをお勧めします。

AWS マネージド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 マネージドポリシーで AWS 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用でき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、「[ユーザーガイド](#)」の[AWS 「管理ポリシー」](#)を参照してください。IAM

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、[AWS Marketplace 「購入者ガイド」](#)の「[購入者のマネージドポリシーAWS Marketplace](#)」を参照してください。

トピック

- [AWS マネージドポリシー: AWSMarketplaceAmiIngestion](#)
- [AWS マネージドポリシー: AWSMarketplaceFullAccess](#)
- [AWS マネージドポリシー: AWSMarketplaceGetEntitlements](#)
- [AWS マネージドポリシー: AWSMarketplaceMeteringFullAccess](#)
- [AWS マネージドポリシー: AWSMarketplaceMeteringRegisterUsage](#)
- [AWS マネージドポリシー: 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: AWSVendorInsightsVendorFullAccess](#)
- [AWS マネージドポリシー: AWSVendorInsightsVendorReadOnly](#)
- [AWS MarketplaceAWS マネージドポリシーの更新](#)

AWS マネージドポリシー: **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、「[AWS Marketplace へのアクセスの許可 AMI](#)」を参照してください。

このポリシーは、AWS Marketplace が Amazon マシンイメージ (AMIs) をコピーして に一覧表示するアクセス許可を寄稿者に付与するために使用されます AWS Marketplace。

許可の詳細

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:ModifySnapshotAttribute"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:us-east-1::snapshot/snap-*"
    },
    {
      "Action": [
        "ec2:DescribeImageAttribute",
        "ec2:DescribeImages",
        "ec2:DescribeSnapshotAttribute",
        "ec2:ModifyImageAttribute"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

AWS マネージドポリシー: **AWSMarketplaceFullAccess**

ID IAM にAWSMarketplaceFullAccessポリシーをアタッチできます。

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。

許可の詳細

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:*",
        "cloudformation:CreateStack",
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStacks",
        "cloudformation:List*",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:CreateTags",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
        "ec2>DeleteSecurityGroup",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeTags",
        "ec2:DescribeVpcs",
```

```

        "ec2:RunInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CopyImage",
        "ec2:DeregisterImage",
        "ec2:DescribeSnapshots",
        "ec2>DeleteSnapshot",
        "ec2>CreateImage",
        "ec2:DescribeInstanceStatus",
        "ssm:GetAutomationExecution",
        "ssm:UpdateDocumentDefaultVersion",
        "ssm>CreateDocument",
        "ssm:StartAutomationExecution",
        "ssm:ListDocuments",
        "ssm:UpdateDocument",
        "ssm:DescribeDocument",
        "sns:ListTopics",
        "sns:GetTopicAttributes",
        "sns:CreateTopic",
        "iam:GetRole",
        "iam:GetInstanceProfile",
        "iam:ListRoles",
        "iam:ListInstanceProfiles"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket",
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3::*image-build*"
    ]
},
{

```



```

    "Effect": "Allow",
    "Action": [
      "sns:Publish",
      "sns:setTopicAttributes"
    ],
    "Resource": "arn:aws:sns:*:*:*image-build*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "*"
    ],
    "Condition": {
      "StringLike": {
        "iam:PassedToService": [
          "ec2.amazonaws.com",
          "ssm.amazonaws.com"
        ]
      }
    }
  }
]
}

```

AWS マネージドポリシー: **AWSMarketplaceGetEntitlements**

ID IAM にAWSMarketplaceGetEntitlementsポリシーをアタッチできます。

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たかどうかを確認するための読み取り専用アクセス許可を付与します。

許可の詳細

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid" : "AWSMarketplaceGetEntitlements",
      "Effect" : "Allow",
      "Action": [

```

```
        "aws-marketplace:GetEntitlements"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS マネージドポリシー: **AWSMarketplaceMeteringFullAccess**

ID IAM にAWSMarketplaceMeteringFullAccessポリシーをアタッチできます。

このポリシーは、AMI および コンテナ製品に対応する計測使用量を、の柔軟な消費料金で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。

アクセス許可の詳細

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

AWS マネージドポリシー: **AWSMarketplaceMeteringRegisterUsage**

ID IAM にAWSMarketplaceMeteringRegisterUsageポリシーをアタッチできます。

このポリシーは、の時間単位の料金でコンテナ製品に対応する計測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。

許可の詳細

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

```
        "aws-marketplace:RegisterUsage"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }
]
```

AWS マネージドポリシー: **AWSMarketplaceSellerFullAccess**

ID IAM にAWSMarketplaceSellerFullAccessポリシーをアタッチできます。

このポリシーは AWS Marketplace、を含む のすべての販売者オペレーションへのフルアクセスと AWS Marketplace 管理ポータル、AMIベースの製品EC2AMIで使用される Amazon の管理を許可する管理アクセス許可を付与します。

アクセス許可の詳細

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MarketplaceManagement",
      "Effect": "Allow",
      "Action": [
        "aws-marketplace-management:uploadFiles",
        "aws-marketplace-management:viewMarketing",
        "aws-marketplace-management:viewReports",
        "aws-marketplace-management:viewSupport",
        "aws-marketplace-management:viewSettings",
        "aws-marketplace:ListChangeSets",
        "aws-marketplace:DescribeChangeSet",
        "aws-marketplace:StartChangeSet",
        "aws-marketplace:CancelChangeSet",
        "aws-marketplace:ListEntities",
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:ListTasks",
        "aws-marketplace:DescribeTask",
        "aws-marketplace:UpdateTask",
        "aws-marketplace:CompleteTask",
        "aws-marketplace:GetSellerDashboard",
        "aws-marketplace:ListAssessments",
        "aws-marketplace:DescribeAssessment",
```

```

        "ec2:DescribeImages",
        "ec2:DescribeSnapshots",
        "ec2:ModifyImageAttribute",
        "ec2:ModifySnapshotAttribute"
    ],
    "Resource": "*"
},
{
    "Sid": "AgreementAccess",
    "Action": [
        "aws-marketplace:SearchAgreements",
        "aws-marketplace:DescribeAgreement",
        "aws-marketplace:GetAgreementTerms"
    ],
    "Effect": "Allow",
    "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",
        "payments:ListPaymentInstruments",
        "payments:ListTagsForResource",
        "payments:TagResource",

```

```

        "payments:UntagResource",
        "tax:GetTaxInterview",
        "tax:PutTaxInterview",
        "tax:GetTaxInfoReportingDocument"
    ],
    "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/*"
}
]
}

```

AWS マネージドポリシー:

AWSMarketplaceSellerProductsFullAccess

ID IAM にAWSMarketplaceSellerProductsFullAccessポリシーをアタッチできます。

このポリシーは、およびへの製品の管理 AWS Marketplace 管理ポータル、およびAMIベースの製品EC2AMIで使用される Amazon の管理へのフルアクセスを許可するアクセス許可を寄稿者に付与します。

アクセス許可の詳細

```

{
    "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:ListTasks",
    "aws-marketplace:DescribeTask",
    "aws-marketplace:UpdateTask",
    "aws-marketplace:CompleteTask",
    "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/*"
  }
]
}

```

AWS マネージドポリシー: **AWSMarketplaceSellerProductsReadOnly**

ID IAM にAWSMarketplaceSellerProductsReadOnlyポリシーをアタッチできます。

このポリシーは、で製品を表示し AWS Marketplace 管理ポータル、AMIベースの製品EC2AMIで使
用される Amazon を表示するアクセスを許可する読み取り専用アクセス許可を付与します。

許可の詳細

```

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

```



```

    "Effect": "Allow",
    "Action": [
      "aws-marketplace:ListChangeSets",
      "aws-marketplace:DescribeChangeSet",
      "aws-marketplace:ListEntities",
      "aws-marketplace:DescribeEntity",
      "aws-marketplace:ListTasks",
      "aws-marketplace:DescribeTask",
      "aws-marketplace:ListAssessments",
      "aws-marketplace:DescribeAssessment",
      "ec2:DescribeImages",
      "ec2:DescribeSnapshots"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:ListTagsForResource"
    ],
    "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
  },
]
}

```

AWS マネージドポリシー: **AWSVendorInsightsVendorFullAccess**

ID IAM にAWSVendorInsightsVendorFullAccessポリシーをアタッチできます。

このポリシーは、AWS Marketplace Vendor Insights で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 は、このガイドの目的上、購入者とベンダーが販売者と同等であると評価者を識別します。は、契約検索の追加、プロファイルスナップショットの更新、ベンダーのタグ付け、AWS Artifact およびサードパーティーレポートへの読み取り専用アクセスを許可AWSVendorInsightsVendorFullAccessするために AWS Marketplace 更新されました。

許可の詳細

```

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

```

```

    "Effect": "Allow",
    "Action": "aws-marketplace:DescribeEntity",
    "Resource": "arn:aws:aws-marketplace:*:*:*/*SaaSProduct/*"
  },
  {
    "Effect": "Allow",
    "Action": "aws-marketplace:ListEntities",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "vendor-insights:CreateDataSource",
      "vendor-insights:UpdateDataSource",
      "vendor-insights>DeleteDataSource",
      "vendor-insights:GetDataSource",
      "vendor-insights:ListDataSources",
      "vendor-insights:CreateSecurityProfile",
      "vendor-insights:ListSecurityProfiles",
      "vendor-insights:GetSecurityProfile",
      "vendor-insights:AssociateDataSource",
      "vendor-insights:DisassociateDataSource",
      "vendor-insights:UpdateSecurityProfile",
      "vendor-insights:ActivateSecurityProfile",
      "vendor-insights:DeactivateSecurityProfile",
      "vendor-insights:UpdateSecurityProfileSnapshotCreationConfiguration",
      "vendor-insights:UpdateSecurityProfileSnapshotReleaseConfiguration",
      "vendor-insights:GetSecurityProfileSnapshot",
      "vendor-insights:ListSecurityProfileSnapshots",
      "vendor-insights:TagResource",
      "vendor-insights:UntagResource",
      "vendor-insights:ListTagsForResource",
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:AcceptAgreementApprovalRequest",
      "aws-marketplace:RejectAgreementApprovalRequest",
      "aws-marketplace:GetAgreementApprovalRequest",
      "aws-marketplace:ListAgreementApprovalRequests",
      "aws-marketplace:CancelAgreement",
      "aws-marketplace:SearchAgreements"
    ]
  }

```

```
    ],
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "aws-marketplace:AgreementType": "VendorInsightsAgreement"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "artifact:GetReport",
      "artifact:GetReportMetadata",
      "artifact:GetTermForReport",
      "artifact:ListReports",
    ],
    "Resource": "arn:aws:artifact:*::report/*"
  }
]
```

AWS マネージドポリシー: **AWSVendorInsightsVendorReadOnly**

ID IAM にAWSVendorInsightsVendorReadOnlyポリシーをアタッチできます。

このポリシーは、AWS Marketplace Vendor Insights プロファイルおよび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付与します。AWS Marketplace Vendor Insights は、購入者とベンダーが本ガイドの目的上販売者と同様であると評価者を識別します。タグを一覧表示するアクセス許可を追加し、AWS Artifact サードパーティーレポートへの読み取り専用アクセスを許可するAWSVendorInsightsVendorReadOnlyのように AWS Marketplace 更新されました。

許可の詳細

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:DescribeEntity",
      "Resource": "arn:aws:aws-marketplace:*:*:/SaaSProduct/*"
    },
    {
```

```

    "Effect": "Allow",
    "Action": "aws-marketplace:ListEntities",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "vendor-insights:GetDataSource",
      "vendor-insights:ListDataSources",
      "vendor-insights:ListSecurityProfiles",
      "vendor-insights:GetSecurityProfile",
      "vendor-insights:GetSecurityProfileSnapshot",
      "vendor-insights:ListSecurityProfileSnapshots",
      "vendor-insights:ListTagsForResource"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "artifact:GetReport",
      "artifact:GetReportMetadata",
      "artifact:GetTermForReport",
      "artifact:ListReports"
    ],
    "Resource": "arn:aws:artifact:*::report/*"
  }
]
}

```

AWS Marketplace AWS マネージドポリシーの更新

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。このページの変更に関する自動アラートについては、ページの RSS フィードにサブスクライブしてください AWS Marketplace [ドキュメント履歴](#)。

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DescribeA	2024 年 10 月 22 日

変更	説明	日付
	<p>assessments 許可が追加されました。変更により、S SLv2ユーザーは評価データにアクセスできます。</p>	
<p>AWSMarketplaceSellerProductsFullAccess - ポリシーの更新</p>	<p>AWS Marketplace に ListAssessments および アクセスDescribeAssessments 許可が追加されました。変更により、S SLv2ユーザーは評価データにアクセスできます。</p>	<p>2024 年 10 月 22 日</p>
<p>AWSMarketplaceSellerProductsReadOnly - ポリシーの更新</p>	<p>AWS Marketplace に ListAssessments および アクセスDescribeAssessments 許可が追加されました。変更により、S SLv2ユーザーは評価データにアクセスできます。</p>	<p>2024 年 10 月 22 日</p>
<p>AWSMarketplaceSellerFullAccess - ポリシーを更新</p>	<p>、aws-marketplace-management:viewMarketing、aws-marketplace-management:viewSettings およびのアクションの削除を反映するようにAWSMarketplaceSellerFullAccess ドキュメントを更新しましたaws-marketplace-management:uploadFiles。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。</p>	<p>2024 年 6 月 4 日</p>

変更	説明	日付
AWSMarketplaceGetEntitlements - ポリシーを更新	AWS Marketplace ポリシーステートメントsidに を追加AWSMarketplaceGetEntitlements するために更新されました。	2024 年 3 月 22 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace 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を追加AWSMarketplaceSellerFullAccess するために更新されました。	2024 年 3 月 15 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace は、税金情報にアクセスするためのアクセス許可を追加AWSMarketplaceSellerFullAccess するために更新されました。	2024 年 2 月 8 日
AWSVendorInsightsVendorFullAccess - ポリシーの更新	AWS Marketplace を更新AWSVendorInsightsVendorFullAccess して、データソースを更新するアクセス許可を追加しました。	2023 年 10 月 18 日
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace エンティティを共有するためのアクセス許可を追加AWSMarketplaceSellerFullAccess するために更新されました。	2023 年 6 月 1 日

変更	説明	日付
AWSMarketplaceSellerFullAccess - ポリシーを更新	AWS Marketplace アカウントの検証、銀行口座の検証、ケース管理、販売者通知の詳細に関連するアクセス許可を追加AWSMarketplaceSellerFullAccess するために更新されました。	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へ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
AWS Marketplace 更新済み AWSVendorInsightsVendorFullAccess	AWS Marketplace を更新AWSMarketplaceSellerProductsFullAccess して、契約検索の追加、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、AWS Artifact サードパーティーレポートへの読み取り専用アクセス (プレビュー) を許可しました。	2022 年 11 月 30 日

変更	説明	日付
<p>AWS Marketplace 更新済み AWSVendorInsightsVendorReadOnly</p>	<p>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への読み取り専用アクセス AWS Artifact を許可するAWSVendorInsightsVendorReadOnly ように更新されました (プレビュー)。</p>	<p>2022 年 11 月 30 日</p>
<p>AWSVendorInsightsVendorFullAccess および AWSVendorInsightsVendorReadOnly – 新しいポリシーを追加</p>	<p>AWS Marketplace AWS Marketplace Vendor Insights AWSMarketplaceSellerProductsFullAccess の新しい機能のポリシーを追加しました。AWSVendorInsightsVendorReadOnly</p>	<p>2022 年 7 月 26 日</p>
<p>AWSMarketplaceSellerProductsFullAccess および AWSMarketplaceSellerFullAccess – ポリシーの更新</p>	<p>AWS Marketplace AWS Marketplace Vendor Insights: AWSMarketplaceSellerProductsFullAccess および の新機能のポリシーが更新されましたAWSMarketplaceSellerFullAccess 。</p>	<p>2022 年 7 月 26 日</p>
<p>AWSMarketplaceSellerFullAccess および AWSMarketplaceSellerProductsFullAccess - 既存のポリシーの更新</p>	<p>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 。</p>	<p>2021 年 11 月 22 日</p>

変更	説明	日付
AWSMarketplaceFullAccess - 既存ポリシーへの更新。	AWS Marketplace はAWSMarketplaceFull Access 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 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

AWS Marketplace Commerce Analytics Service に登録するには、次の 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。

```
{
  "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。

```
{
  "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 による製品および顧客データへのアクセス](#)」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を許可するための IAM ポリシーは必要です。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。

```
{
  "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キューのサブスクリプション”](#) および [the section called “Metering and Entitlement Service へのアクセス AWS Marketplace APIs”](#) を参照してください。

AWS Marketplace 計測と使用権限 API のアクセス許可

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、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。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のある 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。

```
{
  "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、「[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。

```
{
  "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、「[で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。

での再販認証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、に直接リンクされる一意のタイプの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、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いため、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、特に定義されていない限り、はロールのみを引き受け AWS Marketplace することができます。定義されたアクセス許可には、信頼ポリシーとアクセス許可ポリ

シーが含まれ、そのアクセス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は、まずその関連リソースを削除しなければ削除できません。これによりAWS Marketplace、リソースへのアクセス許可を誤って削除できないため、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「と連携するIAMサービス」](#)を参照してください。また、「サービスにリンクされたロール」列で「はい」のサービスを探します。サービスにリンクされた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、[Yes] (はい) リンクを選択します。

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、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AWSServiceRoleForMarketplaceResaleAuthorization、が再販認証 AWS Marketplace に使用または管理する AWS サービスやリソースへのアクセスが可能になります。

AWSServiceRoleForMarketplaceResaleAuthorization サービスにリンクされたロールは、次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

という名前のロールアクセス許可ポリシーAWSMarketplaceResaleAuthorizationServiceRolePolicyではAWS Marketplace、は指定されたリソースに対して次のアクションを実行できます。

```
{
  "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、ロールなどがサービスにリンクされたロールを作成、編集、削除できるようにするには、アクセス権を設定する必要があります。詳細については、「[ユーザーガイド](#)」の「[サービスにリンクされたロールのアクセス許可](#)」を参照してください。IAM

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作成すると AWS Marketplace 管理ポータル、 はサービスにリンクされたロール AWS Marketplace を作成します。

サービスにリンクされたロールを作成するには

1. で[AWS Marketplace 管理ポータル](#)、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、サービスにリンクされたロールタブを選択します。
3. サービスにリンクされたロールページで、再販認証または再販認証統合 のサービスにリンクされたロールを選択し、サービスにリンクされたロールの作成または統合の設定を選択します。
4. 再販売認証または再販売認証統合の作成の サービスにリンクされたロール ページで、情報を確認し、サービスにリンクされたロールの作成 または統合の作成を選択して確認します。

サービスにリンクされた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。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ方法でアカウントにロールを再作成できます。でサービスにリンクされたロールを作成すると AWS Marketplace 管理ポータル、 はサービスにリンクされたロールを再度 AWS Marketplace 作成します。

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を作成した後は、多くのエンティティによってロールが参照され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、を使用してロールの説明を編集できますIAM。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

サービスリンク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、新しい再販認証はターゲットチャネルパートナーに自動的に共有 AWS Resource Access Manager されません。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認証は自動的に承諾 AWS Resource Access Manager されません。

を使用してサービスにリンクされたロールを手動で削除するには IAM

IAM コンソール、AWS CLI、または AWS API を使用して、サービスにリンクされたロールを削除します `AWS::IAM::ServiceRoleForMarketplaceResaleAuthorization`。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 計測API呼び出しのログ記録 AWS CloudTrail

AWS Marketplace はと統合されています AWS Marketplace。これは AWS CloudTrail、. CloudTrail captures のユーザー、ロール、またはによって実行されたアクションの記録を提供するサービス AWS のサービスであり、をイベント AWS Marketplace としてAPI呼び出します。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 アカウントを作成する AWS アカウントと、でが有効になります。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 CloudTrail イベ

ント履歴の他のAWSのサービスイベントとともに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とAWS Identity and Access Management ユーザー認証情報のどちらを使用して行われたか。
- リクエストが、ロールとフェデレーテ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、BatchMeterUsageオペレーションをイベントとしてログファイルに記録するCloudTrailことをサポートします。

AWS Marketplace 計測APIログファイルエントリの例

例: BatchMeterUsage

次の例は、からのBatchMeterUsageアクションを示すCloudTrailログエントリを示しています。販売者が[計測レコードを送信して、に記載されているSoftware as a Service \(SaaS\) 製品の顧客使用状況を報告する](#)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/Alice",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2018-04-19T16:32:51Z",
  "eventSource": "metering-marketplace.amazonaws.com",
  "eventName": "BatchMeterUsage",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.0.2/24",
  "userAgent": "Coral/Netty14",
```

```

    "requestParameters": {
      "usageRecords": [
        {
          "dimension": "Dimension1",
          "timestamp": "Apr 19, 2018 4:32:50 PM",
          "customerIdentifier": "customer1",
          "quantity": 1
        }
      ],
      "productCode": "EXAMPLE_proCode"
    },
    "responseElements": {
      "results": [
        {
          "usageRecord": {
            "dimension": "Dimension1",
            "timestamp": "Apr 19, 2018 4:32:50 PM",
            "customerIdentifier": "customer1",
            "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、からの RegisterUsage アクションを示す CloudTrail ログエントリを示しています。AWS Marketplace Metering Service。からの時間単位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

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 に記録されます AWS アカウント。

```
{
  "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 **MeterUsage** のコンテナの場合 EKS

次の例は、Amazon 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EKS。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 を呼び出し AWS アカウントで 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 に記録されます AWS アカウント。

```

{
  "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
    boto3/1.17.26",
    "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: MeterUsage の AMIs

次の例は、Amazon マシンイメージ () 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。AMI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、MeterUsage購入者の内のソフトウェアAWS MarketplaceがAMI呼び出しAWSアカウントで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、購入者のに記録されますAWSアカウント。

```

{
  "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"
}
```


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フィードをサブスクライブできます。

変更	説明	日付
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。 <ul style="list-style-type: none">• AWSMarketplaceSellerFullAccess• AWSMarketplaceSellerProductsFullAccess• AWSMarketplaceSellerProductsReadOnly	2024 年 10 月 9 日
非推奨のアクセス許可を削除しました	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 Machine Image (AMI) の年間契約変更オプションを更新しました	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
Know Your Customer (KYC) プロセスを更新しました	販売者の KYC プロセスに ステップを追加しました。	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を更新	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工	2024 年 5 月 20 日

	クスペリエンスのためのコンテンツを追加しました。	
Amazon EKS アドオン製品の要件を更新	「コンテナ製品を AWS Marketplace アドオンとして準備する」セクションを更新し、「アドオンの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	料金モデルを使用した pay-as-go 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
での新しいデモとプライベートオファターのオプション AWS Marketplace	AWS Marketplace は、一部の販売者の製品詳細ページで デモ および プライベートオファ ー リクエストオプションをサポートするようになりました。	2024 年 4 月 1 日
税金の新しいダッシュボード	AWS Marketplace では、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する課税ダッシュボードがサポートされ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sidに追加されました。	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の製品手順の削除を更新しました。	2024 年 3 月 19 日
AMI アクセスポリシーを更新しました	Linux 固有の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
の検索エンジン最適化情報を追加 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
のセラーの マネージドポリシーを更新しました AWS Marketplace	サービスにリンクされたロールの作成に関連するアクセス許可を追加AWSMarketplaceSellerFullAccess するために更新されました。	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 では、によって再販売認証 AWS Marketplace のために使用または管理されるAWSサービスや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が提供されるようになりました。	2024 年 3 月 15 日

[のセラーの マネージドポリシーを更新しました AWS Marketplace](#)

税金情報へのアクセスに関連するアクセス許可を追加AWSMarketplaceSellerFullAccess するために更新されました。

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 EventBridge 、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な場合に、以前は Amazon CloudWatch Events と呼ばれていた Amazon 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](#)

コンテナベースの製品から AWS Marketplace Amazon EKS アドオンへの発行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を追加](#)

販売者は、独立したソフトウェアベンダー () としてチャネルパートナーの再販機会を作成できるようになりました ISVs。

2024 年 1 月 18 日

[での将来の日付のアグリメントの一般提供 AWS Marketplace](#)

すべての ISVsおよび AWS Marketplace Channel Partners は、前払い料金でプライベートオファーを発行する一環として、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の柔軟な支払いスケジュール (FPS) が、内のすべてのお客様が利用できるようになりました AWS Marketplace。	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、販売者向け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したときに、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。	2023 年 10 月 31 日

AWS Marketplace Vendor Insights で販売者の管理ポリシーを更新しました	AWS Marketplace を更新した新AWSVendorInsightsVendorFullAccess して、データソースを更新するアクセス許可を追加しました。	2023 年 10 月 18 日
AWS Marketplace for Desktop アプリケーション (AMDA) のサポート終了	AWS Marketplace は 2023 年 10 月 2 AMDA日に のサポートを終了しました。に関連するすべてのコンテンツと手順がガイドから削除AMDAされました。	2023 年 10 月 2 日
SaaS 製品のセルフサービスコンテンツを追加しました	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
AWS Marketplace 販売者が毎日の支払いを利用できるようになりました	の売り手 AWS Marketplace 管理ポータル は、毎日または毎月支払いを受け取ることができます。	2023 年 9 月 7 日
Amazon のサポート EventBridge	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
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	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

[のセラーの マネージドポリシーを更新しました AWS Marketplace](#)

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

[のセラーの マネージド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 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 Commerce Analytics Service を使用した Product Support Connection および共有カスタマーの連絡先情報は、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 AWS アカウントに拡張された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

のセラーの マネージド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へ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
でのAMIセルフサービスリスト (バージョン 2) の販売者エクスペリエンスの更新 AWS Marketplace	の販売者は、単一AMI製品のセルフサービスリストを作成AWS Marketplace 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 タグベースの承認機能用に 3 つのポリシー (AWSMarketplaceSellerFullAccess 、AWSMarketplaceSellerProductsFullAccess 、AWSMarketplaceSellerProductReadOnly) を更新しました。	2022 年 12 月 9 日

AWS Marketplace Vendor Insights で販売者のポリシーを更新しました	AWS Marketplace Vendor Insights 販売者AWSVendorInsightsVendorReadOnly 用の マネージドポリシー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 AWSVendorInsightsVendorFullAccess 、 AWSVendorInsightsVendorReadOnly 管理ポリシーを更新しました。	2022 年 11 月 28 日
販売者は Amazon でアドオン製品を公開できます EKS	AWS Marketplace と Amazon の統合EKS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](#)

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 2 つのポリシーを追加しました。](#)

AWS Marketplace Vendor Insights AWSVendorInsightsVendorReadOnly には 2 つのポリシーAWSVendorInsightsVendorFullAccess と、ソフトウェアリスク評価を提供する機能が追加されました。

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](#)

AWS Marketplace Commerce Analytics Service 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
で販売者が契約用の SaaS 無料トライアルを利用できるようにになりました 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 メール通知を有効にする新機能 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、チャン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セクションの最小限の更新	コンテナベースの要件のポリシーへの最小限の更新と、AMI料金契約の誤った情報の削除	2022年2月14日
コンテナバージョンニングの更新	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

[AWS License Manager と AWS Marketplace for Containers Anywhere 製品を統合するためのドキュメントを追加しました](#)

License Manager と統合することで AWS Marketplace、for Containers Anywhere 製品に契約料金を追加する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 を通じて EMEA への購入者と取引できる機能 EMEA SARL](#)

対象となる AWS Marketplace 販売者は、Amazon Web Services EMEA を通じて、欧州、中東、アフリカ (EMEA) の国と地域に拠点を置く AWS アカウント 顧客と取引できるようになりましたSARL。

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 QuickLaunch に対して有効にできます。QuickLaunch 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、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 (ISVs) は、新しい AMI ベースの製品またはコンテナベースの製品を一覧表示し、購入者に前払い契約料金を提供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

[Amazon Simple Notification Service for AMIまたは Container 製品](#)

独立系ソフトウェアベンダー (ISVs) は、顧客が Amazon Simple Notification Service を通じて AMI または Container 製品をサブスクライブまたはサブスクライブ解除すると、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、オファード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](#)

ペイロード推論が最大数を超えることが予想される機械学習ソフトウェア、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機械学習ソフトウェアの場合、購入者は Amazon SageMaker Asynchronous Inference 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ァーを選択またはアップロードEULA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の再販機会を作成する際に、エンドユーザーライセンス契約 (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 (ISVs) は、コンサルティングパートナーの再販機会を作成する際に SaaS 契約製品のディメンション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用の 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 SAR 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な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更新	販売者は、を使用して Amazon マシンイメージ (AMI) ベースの製品を更新する方法をより簡単かつ迅速にできるようになりました AWS Marketplace 管理ポータル。	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 () での販売者になる資格を得ましたUAE。	2020年6月17日

<u>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</u>	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として提供できません。	2020 年 5 月 28 日
<u>詳細はデータフィードで利用できます。</u>	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
<u>標準化されたライセンス条項が利用可能になりました</u>	カスタムの代わりに標準化されたライセンス条件を提供EU LAsして、契約プロセスを簡素化できます。	2020 年 4 月 28 日
<u>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</u>	現在、AWS Marketplace(i) オーストラリア (AU) ニュージーランド (NZ) の永住者および市民、または (ii) いずれかの地域で組織または法人化された事業体で販売者になる資格があります。	2020 年 4 月 2 日
<u>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</u>	請求に使用する独自の料金単位とメーターを定義する場合は、AWS Marketplace Metering Service の meterUsage アクションと統合します。	2019 年 11 月 14 日
<u>AWS Marketplace は Data Exchange AWS を介してデータ製品をサポートします</u>	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) 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EKS。	2019年9月25日
AMI製品機能の更新	AMIs と Lambda 関数を使用し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 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を更新しました	2019年4月11日
バージョン情報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
コンテナベースの製品セクションを追加しました	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

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	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
消費コンテンツに関する SaaS 契約を追加しました	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
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	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
アクセスIAM許可コンテンツを更新しました	AMMP 読み取り専用アクセスの新しいIAMアクセス許可をサポートするコンテンツを追加しました。	2018 年 10 月 9 日
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	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
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	AMIs 機能のプライベートイメージビルドのリリースをサポートする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](#)

すべてのPDF販売者ガイドを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。