賣方指南

AWS Marketplace

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

AWS Marketplace: 賣方指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS Marketplace？ 1
使用 AWS Marketplace 做為賣方 1
的合約結構 AWS Marketplace 3
中的產品定價 AWS Marketplace 4
開始使用 5
賣方發佈免費軟體產品的要求 5
付費產品的其他賣方需求 6
付費產品的合格司法管轄區 6
AWS Marketplace 管理入口網站 12
註冊為 AWS Marketplace 賣方 13
步驟 1：建立您的公有設定檔 13
步驟 2：提供稅務資訊 14
步驟 3：提供銀行帳戶資訊 15
步驟 4：設定您的支出偏好設定 17
步驟 5：完成了解您的客戶 (KYC) 程序 17
步驟 6：完成銀行帳戶驗證程序 21
已經是 AWS Marketplace 賣方？ 21
支出偏好設定 22
設定您的支出偏好設定 22
AWS Marketplace 訂閱帳單 23
賣方如何取得其支出 23
可用貨幣 24
清單費用 24
了解 中標準優惠清單的費用 AWS Marketplace 25
了解 中區域優惠清單的費用 AWS Marketplace 26
Amazon Payments Europe 投訴 26
提交投訴 26
Amazon Payments Europe 投訴解決時間範圍 27
投訴上報 27
AWS Marketplace 賣方的其他工具 27
AWS Marketplace Commerce Analytics 服務 28
AWS Marketplace 欄位示範計畫 45
AWS 合作夥伴助理 46
準備您的產品 51

產品交付 51
在 上執行之產品的指定 AWS 54
產品定價 55
定價模型 56
變更定價模型 59
變更價格 60
私有優惠 60
產品退款 60
區域和國家/地區 65
AWS 區域 65
國家/地區 65
翻譯和語言 66
開始使用 66
選擇退出翻譯 67
標準化合約 68
的標準合約 AWS Marketplace 68
的經銷商合約 AWS Marketplace 70
類別和中繼資料 70
命名和描述您的產品 71
選擇類別和關鍵字 72
AMI 和容器產品使用說明 73
需求 73
撰寫版本備註 74
撰寫使用說明 74
撰寫升級指示 75
撰寫 CloudFormation 交付指示 75
搜尋引擎最佳化 75
搜尋引擎最佳化 76
AWS Marketplace 搜尋 76
促銷媒體 78
促銷媒體的最佳實務 79
新增和管理促銷媒體的提示 80
準備您的私有優惠 81
私有優惠的運作方式 81
私有優惠考量事項 82
買方的私有優惠體驗 83

報告私有優惠 83
支援的產品類型 84
AMI 產品的私有優惠 84
容器產品的私有優惠 85
專業服務產品的私有優惠 85
SaaS 產品的私有優惠 86
ML 產品的私有優惠 86
建立和管理私有優惠 87
啟動新的私有優惠 87
了解優惠狀態 88
起草和發佈私有優惠 88
新增私有優惠和示範請求按鈕 90
傳送私有優惠給買方 92
複製您的私有優惠 93
下載優惠詳細資訊 93
儲存您的私有優惠進度 93
更新私有優惠的過期時間 94
取消私有優惠 94
管道合作夥伴私有優惠 95
其他資訊 96
建立銷售授權做為 ISV 96
建立分期付款計劃 100
為私有優惠建立分期付款計劃 100
分期付款計劃報告 101
未來日期的協議 101
未來資料協議的考量事項 102
建立未來日期的協議 103
將分期付款計劃與未來日期的協議搭配使用 103
接收未來日期協議的通知 103
使用未來日期的協議與通路合作夥伴私有優惠的轉售 104
私有優惠常見問答集 105
什麼是 Private Offer Success Team (POST)，賣方如何聯絡他們？ 105
賣方如何在存取優惠索引標籤時解決錯誤？ 105
什麼是分期付款計劃或彈性付款排程？ 106
如何協助在存取私有優惠時收到 404 錯誤的買方？ 107
為什麼客戶在嘗試接受私有優惠時看到「您已經有作用中的合約」錯誤？ 109

賣方或買方是否可以取消私有優惠？ 110
如何申請退款或合約取消？ 110
何時開立買方發票？ 111
接受優惠後，賣方應採取哪些步驟？ 111
如何 AWS 支付賣方和合作夥伴的費用？ 111
如何 AWS 評估稅金？ 112
資源和支援 112
使用協議 113
協議類型 113
使用協議 113
尋找協議 114
排序協議 114
檢視協議詳細資訊 114
下載協議詳細資訊 114
修訂協議 114
公有和私有優惠修訂支援的產品類型 115
建立公有和私有優惠升級、續約和修訂 116
升級、續約和修訂的報告 117
AMI 型產品 118
AMI 型產品交付方法 118
其他資源 118
了解 AMI 型產品 119
產品生命週期 119
AMI 產品代碼 121
變更請求 122
產品載入表單 123
年度協議修訂 123
建立 AMI 型產品 124
先決條件 124
了解自助式體驗 125
建立清單 125
其他資源 127
將 CloudFormation 範本新增至您的產品 128
準備您的 CloudFormation 範本 128
架構圖 135
轉換現有產品的 CloudFormation 範本 136

新增無伺服器應用程式元件 141
管理 AMI 型產品 149
建立變更請求 150
更新產品可見性 153
新增和限制執行個體 154
管理版本 156
更新產品資訊 162
管理可用性 163
更新您的 EULA 166
更新您的退款政策 167
提供 AMI 的 AWS Marketplace 存取權 167
移除產品 168
故障診斷 169
建置 AMIs最佳實務 171
保護轉售權利 171
建置 AMI 171
為 準備和保護您的 AMI AWS Marketplace 172
掃描您的 AMI 以取得發佈需求 173
驗證您的軟體是否在 AMI 上執行 AWS Marketplace 174
AMI 產品定價 176
AMI 定價模型 176
AWS 費用和軟體費用 179
使用 AWS Marketplace Metering Service 自訂計量 180
AMI 產品的合約定價 194
使用 AWS License Manager 196
接收 Amazon SNS 通知 209
Amazon SNS 主題： aws-mp-subscription-notification 209
訂閱 Amazon SQS 佇列至 Amazon SNS 主題 210
AMI 產品檢查清單 211
產品用量 211
AMI 準備 211
Windows AMI 212
Linux AMI 212
產品載入表單或產品索引標籤 212
以 AMI 為基礎的產品需求 213
AMI 產品賣方政策 213

安全政策 213
架構政策 215
AMI 產品使用說明 216
AMI 產品版本政策 216
客戶資訊政策 216
產品使用政策 216
Image Builder 元件型產品 218
建立變更請求 218
使用自助式建立變更請求 219
建立變更請求 219
取得變更請求的狀態 220
其他資源 220
更新產品可見性 221
更新產品可見性 221
更新允許清單 （預覽帳戶） 221
新增和限制執行個體 222
新增執行個體 222
限制執行個體 223
管理版本 224
更新產品資訊 224
管理可用性 225
新增 AWS 區域 225
限制 AWS 區域 226
更新對未來的支援 AWS 區域 227
依國家/地區更新可用性 227
更新您的 EULA 228
更新您的退款政策 229
移除產品 229
容器型產品 231
取得說明 231
容器產品入門 232
產品生命週期 232
先決條件 233
概觀：建立容器產品 233
步驟 1：為您的容器產品建立產品 ID 和產品代碼 233
步驟 2：建立初始清單 234

步驟 3：新增您產品的初始版本 235
步驟 4：（僅適用於付費產品） 整合計量或合約定價 235
步驟 5：更新產品可見性 235
後續步驟 236
容器產品掃描安全問題 237
更新產品資訊 237
新增產品版本 238
管理產品定價 246
依國家/地區更新可用性 249
更新您的 EULA 249
測試並發行您的 產品 250
容器型產品需求 251
安全政策 252
客戶資訊要求 252
產品使用需求 253
架構需求 254
容器產品使用說明 254
Amazon EKS 附加元件產品的需求 254
容器產品定價 269
容器定價模型 270
容器產品的合約定價 272
容器產品計費、計量和授權整合 275
使用 的每小時和自訂計量 AWS Marketplace Metering Service 276
使用 的合約定價 AWS License Manager 278
使用 AWS Marketplace Metering Service 設定計量 279
使用 AWS Marketplace Metering Service 設定自訂計量 290
使用 的合約定價 AWS License Manager 301
容器產品的 Amazon SNS 通知 332
Amazon SNS 主題： aws-mp-subscription-notification 332
訂閱 Amazon SQS 佇列至 Amazon SNS 主題 333
機器學習產品 335
了解機器學習產品 335
SageMaker AI 模型套件 335
SageMaker AI 演算法 335
部署推論模型 336
產品生命週期 336

機器學習產品定價 338
服務限制和配額 340
安全性和智慧財產權 343
中的機器學習報告 AWS Marketplace 344
在 SageMaker AI 中準備您的產品 345
將您的程式碼封裝成映像 345
上傳您的映像 367
建立 Amazon SageMaker AI 資源 369
在 中列出您的產品 AWS Marketplace 375
先決條件 375
步驟 1：建立新的清單 378
步驟 2：提供產品資訊 379
步驟 3：新增初始產品版本 379
步驟 4：設定定價模型 380
步驟 5：設定退款政策 380
步驟 6：設定 EULA 381
步驟 7：設定允許清單 381
管理您的產品 381
..... 382
..... 382
..... 383
..... 383
..... 385
..... 385
..... 385
移除產品 386
建立私有優惠 387
需求和最佳實務 388
ML 產品的一般最佳實務 389
用量資訊的需求 389
輸入和輸出的需求 389
Jupyter 筆記本的需求 390
ML 產品清單的需求和建議摘要 391
故障診斷 395
新增產品 ARN 時出現 400 錯誤 395
新增產品 ARN 時出現 404 錯誤 395

建立產品時 SageMaker AI 存取遭拒錯誤 396
建立產品時發生 SageMaker AI 失敗 396
SaaS 產品 397
SaaS 產品入門 397
先決條件 397
後續步驟 398
SaaS 產品生命週期 398
建立 SaaS 產品 399
建立初始 SaaS 產品頁面 401
設定 SaaS 產品設定 403
整合您的 SaaS 訂閱產品 412
整合您的 SaaS 合約產品 415
整合您的 SaaS 合約與pay-as-you-go產品 418
部署無伺服器 SaaS 整合 421
規劃您的 SaaS 產品 421
規劃您的定價 422
規劃您的帳單整合 422
規劃您的 Amazon SNS 整合 422
規劃客戶將如何存取您的產品 422
SaaS 產品準則 423
產品設定準則 424
客戶資訊要求 424
產品使用準則 424
架構準則 425
SaaS 產品定價 429
SaaS 訂閱的定價 430
軟體 SaaS 合約定價 432
建立 SaaS 免費試用 435
建立 SaaS 免費試用優惠 436
取消 SaaS 免費試用優惠 437
客戶加入 437
設定您的 SaaS 產品以接受新買方 438
安全性和訂購 439
SaaS 產品的 Amazon SNS 通知 439
Amazon SNS 主題： aws-mp-entitlement-notification 440
Amazon SNS 主題： aws-mp-subscription-notification 441

訂閱 SQS 佇列至 SNS 主題 443
存取 AWS Marketplace 計量和權利服務 APIs 443
計量用量 444
檢查權利 449
SaaS 產品整合檢查清單 450
報告 453
SaaS 程式碼範例 453
ResolveCustomer 程式碼範例 454
GetEntitlement 程式碼範例 455
BatchMeterUsage 程式碼範例 456
BatchMeterUsage 使用用量配置標記程式碼範例 （選用） 458
使用 AWS PrivateLink 461
簡介 461
設定您的產品 462
將您的產品提交至 AWS Marketplace 463
買方存取 VPC 端點 463
附錄：檢查清單 464
中的專業服務產品 AWS Marketplace 466
取得說明 466
專業服務產品入門 466
必要條件 467
建立專業服務產品 467
建立私有優惠 468
編輯產品資訊 470
編輯產品定價 470
編輯產品可見性 471
移除專業服務產品 472
提供產品詳細資訊 472
產品描述 473
其他資源 474
支援資訊 474
定價維度 475
產品可見性 475
產品需求 475
產品設定準則 475
客戶資訊要求 476

產品使用準則 476
架構準則 476
專業服務產品的定價 477
私有優惠的運作方式 477
數據產品 479
提交您的產品 480
使用產品索引標籤 481
公司和產品標誌要求 482
提交付費重新封裝軟體的要求 483
具有硬體元件的產品需求 484
AWS CloudFormation推出的產品（免費或付費）或以用量為基礎的付費 AMI 產品 484
更新您的產品 488
產品變更和更新 488
時間和期望 489
將 AMIs提交至 AWS Marketplace 489
AMI 自助式掃描 490
AMI 複製和產品程式碼指派 490
最終檢查清單 490
行銷您的產品 493
180 天 GTM 學院 493
宣布您產品的可用性 493
AWS Marketplace 訊息 494
上的檢閱 AWS Marketplace 495
連結至 AWS Marketplace 495
使用 AWS Marketplace 品牌 496
了解以賣方 AWS 身分購買 496
使用 AWS 先決條件購買 497
..... 498
檢閱品牌變更 498
使用 AWS 報告購買 498
在 上直接連結至您的產品 AWS Marketplace 499
新聞發佈 499
AWS Marketplace 商標使用準則 500
通知 502
電子郵件通知 502
事件類型 503

欄位描述 511
管理 通知 514
Amazon EventBridge 通知 515
新優惠的事件 516
變更集的事件 519
安全性摘要報告的事件 522
支出的事件 523
賣方報告、資料摘要和儀表板 524
賣方交付資料摘要 524
資料饋送的儲存和結構 525
存取資料摘要 527
使用資料饋送 532
資料饋送資料表概觀 533
資料饋送查詢範例 540
資料摘要 663
賣方報告 700
存取報告 701
可用的 AWS Marketplace 賣方報告 701
每日業務報告 701
每日客戶訂閱者報告 710
支出報告 712
每月計費營收報告 720
銷售薪酬報告 727
補充報告 730
協議詳細資訊報告 730
賣方儀表板 731
存取儀表板 701
財務操作的儀表板 734
銷售操作的儀表板 760
行銷儀表板 773
AWS Marketplace 廠商洞見 789
了解 AWS Marketplace 廠商洞見 790
將 設定為賣方 790
建立安全性描述檔 791
上傳憑證 791
上傳自我評定 793

啟用 AWS Audit Manager 自動化評估 794
檢視您的設定檔 798
以賣方身分檢視您的安全設定檔 799
管理快照 800
建立快照 801
檢視快照 801
匯出快照 802
檢視最新發行的快照 802
延後快照釋放 802
變更快照清單的偏好設定 803
控制存取 803
AWS Marketplace Vendor Insights 賣家的許可 804
CreateDataSource 804
DeleteDataSource 804
GetDataSource 805
UpdateDataSource 805
ListDataSources 805
CreateSecurityProfile 805
ListSecurityProfiles 805
GetSecurityProfile 806
AssociateDataSource 806
DisassociateDataSource 806
UpdateSecurityProfile 806
ActivateSecurityProfile 806
DeactivateSecurityProfile 807
UpdateSecurityProfileSnapshotCreationConfiguration 807
UpdateSecurityProfileSnapshotReleaseConfiguration 807
ListSecurityProfileSnapshots 807
GetSecurityProfileSnapshot 808
TagResource 808
UntagResource 808
ListTagsForResource 808
其他資源 808
安全 213
適用於 的 IAM AWS Marketplace 811
建立 使用者 812

建立或使用群組 813
以使用者身分登入 814
賣方的 AWS Marketplace 政策和許可 815
AWS Marketplace 賣方的政策 816
AWS Marketplace 賣方的許可 816
範例 1：檢視 KYC 狀態的許可 820
範例 2：為私有優惠建立升級和續約的許可 820
範例 3：存取優惠頁面並建立新的私有優惠的許可 821
範例 4：存取設定頁面的許可 821
範例 5：存取檔案上傳頁面的許可 822
使用 IAM 群組 822
AWS 受管政策 823
AWSMarketplaceAmiIngestion 823
AWSMarketplaceFullAccess 824
AWSMarketplaceGetEntitlements 824
AWSMarketplaceMeteringFullAccess 824
AWSMarketplaceMeteringRegisterUsage 824
AWSMarketplaceSellerFullAccess 825
AWSMarketplaceSellerProductsFullAccess 828
AWSMarketplaceSellerProductsReadOnly 830
AWSMarketplaceSellerOfferManagement 832
AWSVendorInsightsVendorFullAccess 832
AWSVendorInsightsVendorReadOnly 832
政策更新 832
AWS Marketplace Commerce Analytics Service 帳戶許可 837
Amazon SQS 許可 838
AWS Marketplace 計量和權利 API 許可 839
SaaS 產品的 IAM 政策 839
AMI 產品的 IAM 政策 840
容器產品的 IAM 政策 840
用於轉售授權的服務連結角色 841
的服務連結角色許可 AWS Marketplace 842
為 建立服務連結角色 AWS Marketplace 844
編輯 的服務連結角色 AWS Marketplace 845
刪除 的服務連結角色 AWS Marketplace 845
AWS Marketplace 服務連結角色支援的 區域 846

使用 記錄 AWS Marketplace API 呼叫 AWS CloudTrail 846
AWS Marketplace 計量 API 日誌檔案項目範例 846
故障診斷 852
常見的賣方註冊錯誤訊息 852
文件歷史紀錄 855
AWS 詞彙表 872
..... dccclxxiii

什麼是 AWS Marketplace？

AWS Marketplace 是精選的數位目錄，客戶可用來尋找、購買、部署和管理第三方軟體、資料和服務，以建置解決方案並執行其業務。 AWS Marketplace 包含來自安全、商業應用程式、機器學習等熱門類別的數千種軟體清單，以及特定產業的資料產品，例如醫療保健、金融服務和電信。客戶可以快速啟動預先設定的軟體，並選擇 Amazon Machine Image (AMIs)、軟體即服務 (SaaS) 和其他格式的軟體解決方案。也提供專業服務，協助客戶設定、部署和管理第三方軟體。如需交付方法的完整清單，請參閱產品交付。

您可以使用 AWS Marketplace 做為買方 （訂閱者）、賣方 （供應商） 或兩者。具有 的任何人 AWS 帳戶 都可以使用 AWS Marketplace 做為買方，並且可以註冊成為賣方。賣方可以是獨立軟體廠商 (ISV)、管道合作夥伴、受管服務供應商 (MSP) 或個人，其可提供可與 AWS 產品和服務搭配使用的服務。

> **Note**
>
> 資料產品提供者必須符合 AWS Data Exchange 資格要求。如需詳細資訊，請參閱《AWS Data Exchange 使用者指南》中的在 AWS Data Exchange 上提供資料產品。

符合資格的合作夥伴可以透過程式設計方式列出 以外的 AWS Marketplace 產品 AWS Marketplace。如需成為合格合作夥伴的詳細資訊，請聯絡您的 AWS Marketplace 業務開發合作夥伴。

下列影片會詳細說明如何在 中銷售 AWS Marketplace。

簡介 AWS Marketplace

使用 AWS Marketplace 做為賣方

在 中銷售軟體產品的程序 AWS Marketplace 包含下列七個步驟。

賣方程序

步驟	動作	描述
1	註冊	身為賣方，您可以從註冊 開始 AWS Marketplace 管理入口網站。我們建議您實作新的專用 AWS 帳戶 ，以便與現有的 AWS 組織輕鬆連結。確認 AWS 合作夥伴的稅務資訊符合司法管轄區的資格條件。若要接收付款，您必須提供銀行帳戶。

步驟	動作	描述
2	決定產品類型	決定您要銷售的產品類型。如需在 中建立產品類型的詳細資訊 AWS Marketplace，請參閱下列內容： • 中的 AMI 型產品 AWS Marketplace • 上的容器型產品 AWS Marketplace • 中的機器學習產品 AWS Marketplace • 中的 SaaS 型產品 AWS Marketplace • 中的專業服務產品 AWS Marketplace • 資料產品 （如需資料產品的詳細資訊，請參閱《AWS Data Exchange 使用者指南》中的什麼是 AWS Data Exchange？。）
3	準備產品	設定您的套件、設定定價方案、決定要列出產品的相關類別，以及新增關鍵字，讓您的產品出現在相關搜尋中。為了簡化採購程序，您可以針對公開產品清單和私有優惠使用標準化授權條款。
4	提交產品	使用產品提交程序讓您的產品可在 中使用 AWS Marketplace。產品可能很簡單，例如，具有一個價格結構的單一 Amazon Machine Image (AMI)。或者，產品可能很複雜，具有多個 AMIs、 AWS CloudFormation 範本，以及複雜的定價選項和付款排程。
5	市場產品	透過直接將流量導向您的產品頁面，提高 AWS Marketplace 和 的意識，為產品的成功做出貢獻 AWS Marketplace。
6	檢視報告和資料摘要	將 註冊為賣方之後，請使用 AWS Marketplace 管理入口網站 來存取您產品的用量報告。 AWS Marketplace 提供工具來收集和分析產品銷售的相關資訊。
7	管理 產品	使用 AWS Marketplace 管理入口網站來管理您的帳戶和產品頁面。

身為賣方，請前往 AWS Marketplace 管理入口網站 進行註冊。如果您要收取使用產品的費用，您也必須在註冊時提供稅務和銀行資訊。註冊時，您可以為公司或您自己建立可探索的設定檔 AWS Marketplace。您也可以使用 AWS Marketplace 管理入口網站來建立和管理產品的產品頁面。

的合約結構 AWS Marketplace

在 中銷售的軟體、服務和資料產品的使用 AWS Marketplace 受買方和賣方之間的協議約束。 AWS 不是這些協議的一方。

身為賣方，您的協議包含下列項目：

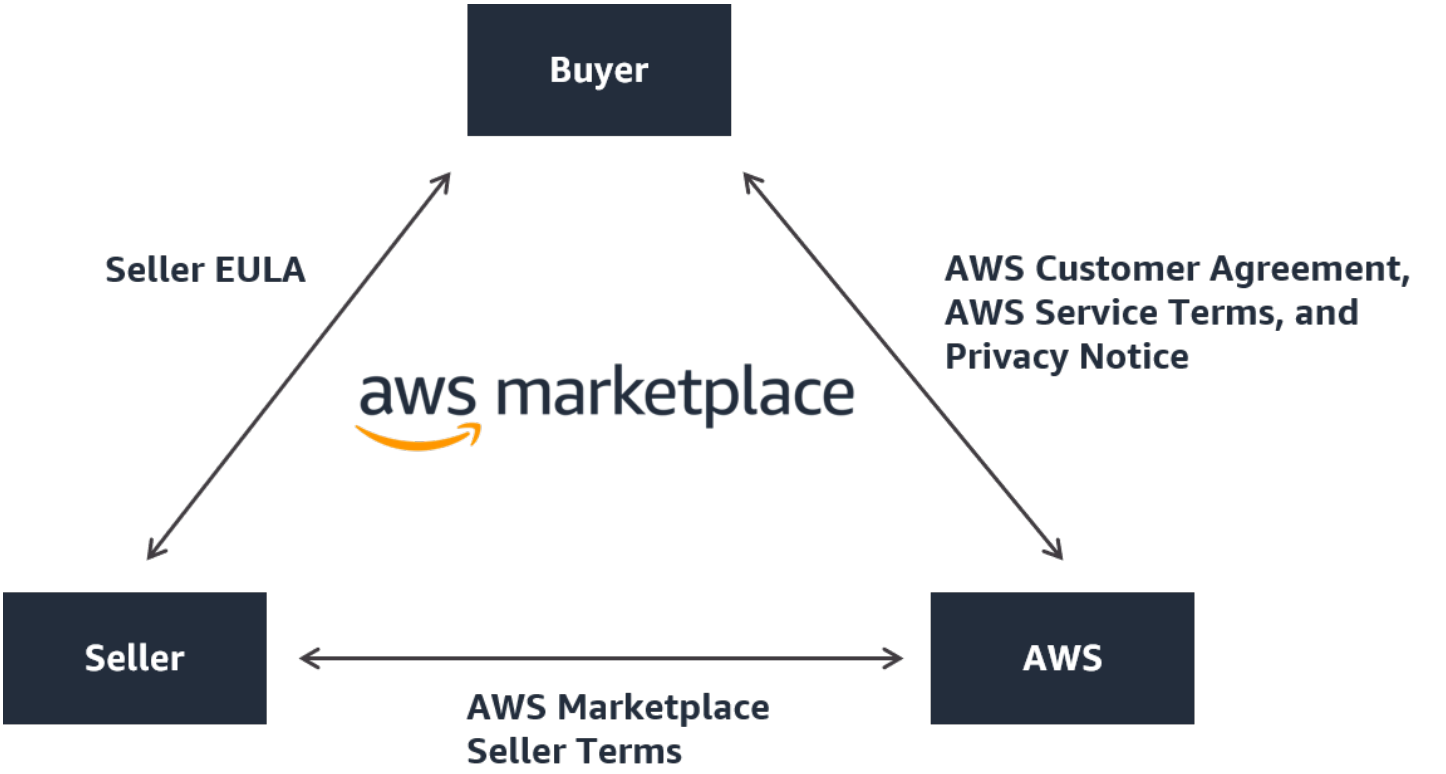
- 您與買方簽訂的最終使用者授權合約 (EULA)，位於 中公有軟體清單的產品清單頁面上 AWS Marketplace。許多賣方使用 AWS Marketplace (SCMP) 標準合約做為其預設 EULA。您也可以使用 SCMP 做為私有優惠中交涉的基礎，並使用修訂範本來修改 SCMP。私有優惠也可以包含雙方協商的自訂合約條款。

  > **Note**
  >
  > 如需有關何時進行 EULA 更新的資訊，根據優惠類型和定價模型，請參閱《 AWS Marketplace 買方指南》中的 EULA 更新。

- AWS Marketplace Sellers 的服務條款，其控管您的活動 AWS Marketplace。

買方對 的使用 AWS Marketplace 受 AWS 服務條款、AWS 客戶協議和隱私權聲明的約束。

下圖顯示 的合約結構 AWS Marketplace。



中的產品定價 AWS Marketplace

在 中 AWS Marketplace，產品可以免費使用或產生相關費用。費用會成為買方 AWS 帳單的一部分，在買方付款之後，支付賣方 AWS 的費用。產品可以採用多種形式。例如，產品可以做為使用買方的 執行個體化的 Amazon Machine Image (AMI) 提供 AWS 帳戶。產品也可以設定為使用 CloudFormation 範本交付給買方。產品也可以是來自 ISV、Web 存取控制清單 (Web ACL)、規則集或條件的 SaaS 產品 AWS WAF。產品也可以是來自 ISV、頻道合作夥伴或 MSP 的專業服務。

彈性的定價選項包括免費試用、每小時、每月、每年、多年和自攜授權模型 (BYOL)，以及從單一來源計費。 AWS 處理帳單和付款，費用會顯示在客戶的 AWS 帳單上。

您可以使用 ISV 的標準最終使用者授權合約 (EULA)，以列出的價格購買軟體產品。此外，軟體產品可以透過私有優惠提供自訂定價和 EULA。您也可以在具有指定時間或使用界限的合約下購買產品。訂閱產品後，買方可以使用 AWS Service Catalog 複製產品，並管理產品在買方組織中的存取和使用方式。如需買方體驗的詳細資訊，請參閱 https://docs.aws.amazon.com/marketplace/latest/buyerguide/service-catalog.html。如需定價的詳細資訊，請參閱 the section called “產品定價”。

以 AWS Marketplace 賣方身分開始使用

如果您是獨立軟體廠商 (ISV)、管道合作夥伴、受管服務供應商 (MSP) 或個人，且其提供適用於 AWS 產品和服務的內容，則可以註冊為 上的賣方 AWS Marketplace。註冊為賣方是發佈產品以進行銷售的先決條件 AWS Marketplace。根據您的居住地和銷售的產品類型， AWS Marketplace 有不同的註冊要求。下列各節提供賣方註冊程序、需求和相關工具的概觀。

若要在 中販售您的軟體 AWS Marketplace，請依照下列步驟進行：

- 檢閱免費產品的賣方要求和付費產品的要求。
- 完成賣方註冊程序。
- 進一步了解後續步驟和賣方工具。

> **備註**
>
> - 註冊為 AWS Marketplace 賣方是列出 AWS Data Exchange 上資料產品並使其可供使用的先決條件 AWS Marketplace。如需這些要求的詳細資訊，請參閱《AWS Data Exchange 使用者指南》中的在 AWS 資料交換上提供資料產品。
> - 如需 AWS Marketplace 賣方所需許可的資訊，請參閱 賣方的 AWS Marketplace 政策和許可。
> - 如需產品清單費用的詳細資訊，已註冊的賣方可以在 中檢視AWS Marketplace 賣方條款 AWS Marketplace 管理入口網站。
> - 如需常見問答集的解答，請參閱 AWS Marketplace Sellers 常見問答集。

賣方發佈免費軟體產品的要求

無論您在提供產品時是否收取產品費用 AWS Marketplace，您都會銷售該產品。客戶的成本是 0.00 USD，但您和客戶同意使用產品的交互合約。如果您只提供免費產品，則不需要提供銀行資訊給 AWS Marketplace。

若要在 中建立並提供免費產品 AWS Marketplace，您必須：

- 銷售公開可用的全功能生產就緒軟體。
- 擁有已定義的客戶支援程序和支援組織。

- 提供讓軟體定期更新且無漏洞的方法。
- 在 中行銷您的產品時，請遵循最佳實務和準則 AWS Marketplace。
- 成為信譽良好的 AWS 客戶，並符合 AWS Marketplace 賣方條款與條件中的要求。

付費產品的其他賣方需求

如果您收取產品或優惠使用自有授權模型 (BYOL) 產品的費用，您還必須符合下列要求，並提供此額外資訊：

- 您必須是合格司法管轄區的永久居民或公民，或在其中一個區域中組織或註冊的商業實體。
- 您必須提供稅務和銀行帳戶資訊。對於美國實體，需要來自美國銀行的 W-9 表單和銀行帳戶。您必須將銀行帳戶資訊與支出偏好設定建立關聯，才能建立任何公有或私有優惠。
- 非美國賣方必須提供 (i) W-8 表單、增值稅 (VAT) 或商品和服務稅 (GST) 註冊號碼，以及 (ii) 在合格司法管轄區中具有 SWIFT 代碼的銀行帳戶。如有必要，您可以從 Hyperwallet 註冊虛擬美國銀行帳戶。
- 若要提供資料產品，您還必須透過建立案例精靈來請求加入 支援。
- 若要透過 Amazon Web Services EMEA SARL 將產品銷售給 AWS 帳戶 位於歐洲、中東和非洲 (EMEA) 國家和領土的客戶 （土耳其和南非除外），您必須完成了解您的客戶程序。除此之外：
  - 您最多會收到兩個支出 （適用於透過 AWS Inc. 和 Amazon Web Services EMEA SARL 的交易）。
  - 視位置而定，某些交易的清單費用可能會向您收取稅金。如需稅金的詳細資訊，請參閱AWS Marketplace 賣方稅務說明頁面。如果評估列出費用的增值稅 (VAT)， AWS Marketplace 將提供符合稅務規範的發票。
  - 如需 Amazon Web Services EMEA SARL 的詳細資訊，請參閱 Amazon Web Services Europe FAQs網站上的 AWS EMEA Marketplace - Sellers。

若要銷售到 AWS GovCloud (US) Region，賣方必須擁有 AWS GovCloud (US) 帳戶。如需 ITAR 需求的詳細資訊，請參閱 AWS GovCloud (US) 使用者指南。

有關 AWS Marketplace 賣方要求或註冊程序的問題，請聯絡AWS Marketplace 賣方營運團隊。

付費產品的合格司法管轄區

若要在 銷售付費軟體 AWS Marketplace，您必須是下列其中一個國家/地區或 SARs 的永久居民或公民，或是組織或併入其中的商業實體：

- 澳洲1
- 巴林 1 2
- 哥倫比亞1 2
- 歐盟 (EU) 成員國1
- 香港特別行政區
- 以色列1 2
- 日本3
- 紐西蘭1
- 挪威 1 2
- 卡達
- 南韓
- 瑞士1 2
- 阿拉伯聯合大公國 （阿拉伯聯合大公國）1 2
- 英國 (UK)1
- 美國 (US)

1 這些國家/地區的付費產品賣方必須在建立國家/地區提供增值稅註冊資訊。

2 如果您是賣方與買方位於相同的國家/地區，您可能需要負責開立稅務發票、收款和匯款。請洽詢您的稅務顧問。

3 從 2025 年 4 月 1 日開始，下列程序適用於 10% 日本消費稅 (JCT) 的收集和匯款，以及 AWS 日本針對銷售產品發行符合稅務資格的發票 (TQI) AWS Marketplace：

賣方和客戶案例	稅務收集和發票程序
Seller – 地址在日本以外的獨立軟體廠商 (ISV)。 客戶 – AWS 地址在日本的帳戶。	根據指定的平台稅務規則， AWS 日本會執行下列動作： • 收取 10% 日本消費稅 (JCT)。 • 向 AWS 日本 JCT 編號 T6011001106696 的客戶開立稅務合格發票 (TQI)。 • 將已收集的日本消費稅 (JCT) 匯至日本稅務局 (JTA)。

賣方和客戶案例	稅務收集和發票程序
	為了避免重複，ISV 應停止收集或匯款日本消費稅 (JCT) 和開立稅務合格發票 (TQIs)。
Seller – 地址在日本的獨立軟體廠商 (ISV)。 客戶 – AWS 地址在日本的帳戶。	根據特殊機構規則， AWS 日本會執行下列動作： • 收取 10% 日本消費稅 (JCT)。 • 向 AWS 日本 JCT 編號 T6011001106696 的客戶開立稅務合格發票 (TQI)。 • 將收集的 JCT 分發至 ISV。 ISV 必須執行下列動作： • 在 Amazon Marketplace 管理入口網站中輸入其日文公司號碼和日文使用稅 (JCT) 號碼。 • 使用 Amazon Marketplace 管理入口網站中提供的稅務合格發票 (TQI) 資訊，將收集到的 JCT 傳送至日本稅務局 (JTA)。

賣方和客戶案例	稅務收集和發票程序
管道合作夥伴私人優惠 (CPPO)，日本以外的 ISV 1. 第一個交易 • Seller – 地址在日本以外的獨立軟體廠商 (ISV)。 • 客戶 – AWS Marketplace 與日本地址的通路合作夥伴。 2. 第二個交易 • 賣方 – 相同的 AWS Marketplace 通路合作夥伴。 • 客戶 – AWS 地址在日本的帳戶。	對於第一個交易，頻道合作夥伴必須根據反向收費稅務機制報告日本消費稅 (JCT)。 對於第二個交易： • 根據特殊機構規則， AWS 日本會執行下列動作： • 收取 10% 日本消費稅 (JCT)。 • 向 AWS 日本 JCT 編號 T6011001106696 的客戶開立稅務合格發票 (TQI)。 • 向頻道合作夥伴支付收取的日文使用稅 (JCT)。 • 頻道合作夥伴必須執行下列動作： • 在 Amazon Marketplace 管理入口網站中輸入其日文公司號碼和日文使用稅 (JCT) 號碼。 • 使用 Amazon Marketplace 管理入口網站中提供的稅務合格發票 (TQI) 資訊，將收集到的 JCT 傳送至日本稅務局 (JTA)。

賣方和客戶案例	稅務收集和發票程序
通路合作夥伴私人優惠 (CPPO)，日本的所有賣方和客戶 1. 第一個交易 • Seller – 地址在日本的獨立軟體廠商 (ISV)。 • 客戶 – AWS Marketplace 與日本地址的通路合作夥伴。 2. 第二個交易 • 賣方 – 相同的 AWS Marketplace 通路合作夥伴。 • 客戶 – AWS 地址在日本的帳戶。	對於第一個交易： • 管道合作夥伴從 ISV 收到稅務合格發票 (TQI)。 • AWS 日本不會開立發票或收取日本消費稅 (JCT)。 對於第二個交易： • 根據特殊機構規則，AWS 日本會執行下列動作： • 收取 10% 日本消費稅 (JCT)。 • 使用 AWS 日本 JCT 編號 T6011001106696 向客戶開立稅務合格發票 (TQI)。 • 向頻道合作夥伴支付收取的日文使用稅 (JCT)。 • 頻道合作夥伴必須執行下列動作： • 在 Amazon Marketplace 管理入口網站中輸入其日文公司號碼和日文使用稅 (JCT) 號碼。 • 使用 Amazon Marketplace 管理入口網站中提供的稅務合格發票 (TQI) 資訊，將收集到的 JCT 傳送至日本稅務局 (JTA)。

賣方和客戶案例	稅務收集和發票程序
日本境外的通路合作夥伴私人優惠 (CPPO)、ISV 和通路合作夥伴 1. 第一個交易 • Seller – 地址在日本以外的獨立軟體廠商 (ISV)。 • 客戶 – AWS Marketplace 與日本以外地址的通路合作夥伴。 2. 第二個交易 • 賣方 – 相同的 AWS Marketplace 通路合作夥伴。 • 客戶 – AWS 地址在日本的帳戶。	對於第一個交易， AWS 日本不會開立發票或收取日本消費稅 (JCT)，因為這超出 JCT 的範圍。 對於第二個交易： • 根據指定的平台商業規則， AWS 日本會執行下列動作： • 收取 10% 日本消費稅 (JCT)。 • 向 AWS 日本 JCT 編號 T6011001106696 的客戶開立稅務合格發票 (TQI)。 • 將收集到的日本消費稅 (JCT) 匯至日本稅務局 (JTA)。 為了避免重複，頻道合作夥伴應停止收集或匯款日本消費稅 (JCT) 和開立稅務合格發票 (TQIs)。
通路合作夥伴私有優惠 (CPPO)，日本以外的通路合作夥伴 1. 第一個交易 • Seller – 地址在日本的獨立軟體廠商 (ISV)。 • 客戶 – AWS Marketplace 與日本以外地址的通路合作夥伴。 2. 第二個交易 • 賣方 – 相同的 AWS Marketplace 通路合作夥伴。 • 客戶 – AWS 地址在日本的帳戶。	對於第一個交易， AWS 日本不會開立發票或收取日本消費稅 (JCT)，因為這超出 JCT 的範圍。 對於第二個交易： • 根據指定的平台商業規則， AWS 日本會執行下列動作： • 收取 10% 日本消費稅 (JCT)。 • 向 AWS 日本 JCT 編號 T6011001106696 的客戶開立稅務合格發票 (TQI)。 • 將收集到的日本消費稅 (JCT) 匯至日本稅務局 (JTA)。 為了避免重複，頻道合作夥伴應停止收集或匯款日本消費稅 (JCT) 和開立稅務合格發票 (TQIs)。

如需增值稅、發票和您身為賣方的稅務義務的詳細資訊，請參閱 Amazon Web Service Tax Help 上的AWS Marketplace 賣方。

如果商業實體未納入上述其中一個國家/地區或 SARs，請參閱AWS Marketplace 司法管轄區以外的公司資源。

> **Note**
>
> AWS India Private Limited (AWS India) 帳戶先前稱為 Amazon Internet Services Private Limited，無法用來註冊帳戶 AWS Marketplace。

AWS Marketplace 管理入口網站

您可以使用 AWS Marketplace 管理入口網站來管理您銷售的產品 AWS Marketplace。所有已註冊的賣方都可以 AWS Marketplace 管理入口網站 使用其用來建立其產品的帳戶的 AWS 登入資料來存取 。您使用的帳戶定義為客戶訂閱您的產品時的記錄賣方。如果您需要協助判斷您產品記錄賣方的特定帳戶，請聯絡AWS Marketplace 賣方營運團隊。

您可以在 入口網站上完成下列任務：

- 註冊為 AWS Marketplace 賣方。
- 使用產品頁面提交新的軟體產品並更新現有的軟體產品。
- 監控請求的狀態。
- 上傳建立和管理新軟體產品所需的檔案。
- 利用go-to-market活動，將軟體產品管理為增量管道收入。
- 在啟動後數小時內測量行銷活動的結果，包括行銷活動驅動的用量和收入。
- 可讓客戶服務代表即時擷取客戶資料。
- 啟動自動 Amazon Machine Image (AMI) 掃描以偵測漏洞。

> **Note**
>
> 資料產品會從 AWS Data Exchange 主控台發佈和管理。AWS Data Exchange 供應商可以使用 AWS Marketplace 管理入口網站 註冊為賣方、請求 AWS Data Exchange 加入、存取賣方報告，以及提交退款請求。

註冊為 AWS Marketplace 賣方

如果您是獨立軟體廠商 (ISV)、管道合作夥伴、受管服務供應商 (MSP) 或個人，且其提供適用於 AWS 產品和服務的內容，則可以註冊為 的賣方 AWS Marketplace。註冊為賣方是發佈產品以供銷售的先決條件 AWS Marketplace。下列各節會引導您完成成功註冊為賣方的步驟 AWS Marketplace。

若要在 中註冊為賣方 AWS Marketplace，您可以使用現有的 AWS 帳戶 或建立新帳戶。所有 AWS Marketplace 互動都會繫結至您選擇的帳戶。 AWS Marketplace 強烈建議使用 AWS Identity and Access Management (IAM) 角色登入 ， AWS Marketplace 管理入口網站 而不是使用您的根帳戶登入資料。如需詳細資訊，請參閱 AWS Marketplace 安全性。您也可以使用 IAM 來設定主要 AWS 帳戶 ，以允許具有各種許可的多個使用者存取 AWS Marketplace 管理入口網站。如需詳細資訊，請參閱the section called “適用於 的 IAM AWS Marketplace”。

> **Note**
>
> 若要接收重要的通知和帳戶更新，您必須提供組織內適當的聯絡人可存取的電子郵件地址。別名無法取代電子郵件地址，因為有些 AWS 團隊只能從註冊的電子郵件地址接收電子郵件。

若要註冊為賣方，請完成下列步驟。

主題

- 步驟 1：建立您的公有設定檔
- 步驟 2：提供稅務資訊
- 步驟 3：提供銀行帳戶資訊
- 步驟 4：設定您的支出偏好設定
- 步驟 5：完成了解您的客戶 (KYC) 程序
- 步驟 6：完成銀行帳戶驗證程序
- 已經是 AWS Marketplace 賣方？

步驟 1：建立您的公有設定檔

註冊的第一步是選取要 AWS 帳戶 做為主要 AWS Marketplace 帳戶的 ，並提供在 主控台中 AWS Marketplace 向潛在買方顯示的資訊。此帳戶將是 中您產品的記錄賣方， AWS Marketplace 並用於向您報告、分配和通訊 AWS Marketplace 。

使用 AWS 帳戶 註冊成為賣方並列出產品後 AWS Marketplace，您就無法變更與產品相關聯的帳戶。我們建議您使用新帳戶註冊為 AWS Marketplace 賣方。不過，如果該帳戶是在 2017 年 9 月 27 日之後建立的，您可以使用現有的帳戶。

建立您的公有設定檔

1. 從 AWS Marketplace 管理入口網站(AMMP) 選擇立即註冊，然後登入您的賣方 AWS 帳戶。
2. 選取新增公有設定檔以提供您的賣方資訊。

完成公有設定檔後，您可以發佈和銷售免費產品。若要銷售付費產品，您必須提供您的稅務和銀行資訊。

步驟 2：提供稅務資訊

您必須提供您的稅務資訊和增值稅 (VAT) 資訊，以便 AWS Marketplace 可以準確地報告和預扣產品銷售的稅金。

提供您的稅務資訊

1. 登入 AWS Marketplace 管理入口網站，然後選擇設定。
2. 在付款資訊區段中選取前往稅務儀表板。
3. 完成美國稅務面試。
   - 若要銷售專業服務 AWS Marketplace，您必須完成 DAC7 的稅務問卷。
   - 如果您看到錯誤訊息「稅務面試位置與業務位置不符」，請確定 Billing and Cost Management 中提供的銀行和稅務資訊符合輸入 的內容 AWS Marketplace 管理入口網站。您的稅務面試位置必須符合業務位置。
4. 完成稅務資訊後，請返回設定頁面，然後在有可用時選取完成增值稅資訊。此選擇會將您重新導向至 AWS Billing 主控台的稅務設定頁面。

> **Note**
>
> 只有在您位於支援 VAT AWS 區域 的 中時，才能使用 VAT 資訊區段。

補充稅務註冊

如果您使用當地稅務登記號碼 (TRN) 進行特定地區銷售，您可以將它新增為 AWS Marketplace 帳戶的補充 TRN。例如，補充瑞士 TRN 用於管理瑞士買方的稅務收集和發票開立。與補充 TRN 相關聯的商業地址可能會與獨立軟體供應商、管道合作夥伴和買方共用。

如果沒有補充 TRN， 會使用主要 TRN AWS Marketplace 確定稅務管轄權。如需詳細資訊，請參閱 如何 AWS 判斷您帳戶的位置？

新增補充稅務註冊

1. 登入 AWS Marketplace 管理入口網站。
2. 選擇設定。
3. 在設定頁面上，選擇稅務設定索引標籤。
4. 選擇新增補充稅務資訊。
5. 輸入補充稅務註冊詳細資訊。TRN 可以是增值稅 (VAT) 號碼、VAT ID、VAT 註冊號碼或商業註冊號碼。
6. 選擇提交。

存取稅務文件

您可以從 存取您的稅務文件，例如 1099 表格 AWS Marketplace 管理入口網站。

存取您的稅務文件

1. 登入 AWS Marketplace 管理入口網站，然後選擇設定。
2. 前往付款資訊區段。
3. 選取相關的稅務表單 (1099K 或 DAC7)。
4. 如果您的稅務表單可用，您可以從稅務儀表板頁面下載。

步驟 3：提供銀行帳戶資訊

所有想要銷售付費產品的賣方都需要位於合格司法管轄區的銀行帳戶 AWS Marketplace。您的銀行帳戶必須能夠接受 USD 支出。

> **Note**
>
> 如需您可以在其中提供付費產品的國家/地區清單 AWS Marketplace，請參閱 付費產品的合格司法管轄區。

提供銀行資訊

1. 登入 AWS Marketplace 管理入口網站，然後選擇設定。
2. 在付款資訊區段中選取完成銀行資訊。
3. 提供銀行帳戶的必要資訊。您可以選擇提供一或多個銀行帳戶。這些銀行帳戶可以是美國 ACH 帳戶、來自合格司法管轄區的 SWIFT 銀行帳戶，或 Hyperwallet 帳戶。

> **Note**
>
> 如果您尚未提供您的稅務資訊（以及適用的增值稅資訊），您將無法提供銀行資訊。

如果您偏好使用美國銀行帳戶接受 USD 支出， Hyperwallet可以為您提供美國帳戶。

Hyperwallet 是獨立的服務供應商，可讓您以支援的貨幣將資金轉移到另一個銀行帳戶。在有限時間內，您不需要支付與 AWS Marketplace 支付相關的特定Hyperwallet服務費用。

- 將Hyperwallet您的帳戶詳細資訊新增至 AWS Marketplace 您的賣方帳戶，即表示您同意並確認 AWS Marketplace 會與 共用您的姓名、電子郵件地址和帳戶號碼Hyperwallet，以確認您身為 AWS Marketplace 賣方的狀態。
- 使用 Hyperwallet服務可能會收取額外費用（包括轉移費用和將資金轉移至您的當地貨幣所需的匯率費用），以及匯率。Hyperwallet 服務費用將在有限的時間內免收，而且僅限於將已付費產品的收益 AWS Marketplace 支付到Hyperwallet您的帳戶。如需詳細資訊，請參閱Hyperwallet網站的費用區段或聯絡人Hyperwallet以取得詳細資訊，並檢閱適用的費用。如需其服務的詳細資訊，請參閱 Hyperwallet支援網站。

開始向 註冊Hyperwallet並取得您的美國銀行帳戶資訊

1. 登入 AWS Marketplace 管理入口網站，然後選擇設定，然後在付款資訊區段中選取完成銀行資訊。

2. 如果您沒有Hyperwallet帳戶，且需要在 中使用 AWS Marketplace，請選擇否以回應您是否擁有美國銀行帳戶？ 和是否已向 註冊Hyperwallet？ 您會收到個人識別號碼 (PIN) 和用於註冊 的連結Hyperwallet。
3. 啟用Hyperwallet帳戶後，請依照Hyperwallet註冊入口網站中所述的步驟完成註冊並接收您的存款帳戶資訊。
4. 當您從 取得帳戶時Hyperwallet， AWS 帳戶 請登入 ，將Hyperwallet您的帳戶資訊新增至您的 AWS Marketplace 管理入口網站。然後，選擇設定，然後在付款資訊區段中選取完成銀行資訊。

> **Note**
>
> 若要更新Hyperwallet帳戶中的資訊，請聯絡Hyperwallet入口網站中的 Hyperwallet 支援團隊。如需時數和聯絡資訊，請參閱支援索引標籤。

步驟 4：設定您的支出偏好設定

選取每日或每月支出選項，選擇當月接收支出的日期，並將貨幣指派給銀行帳戶。如需詳細資訊，請參閱 AWS Marketplace 賣方的支出偏好設定。

步驟 5：完成了解您的客戶 (KYC) 程序

賣方需要完成了解您的客戶 (KYC) 程序，才能：

- 獲得韓國交易的付款
- 使用以英國為基礎的銀行帳戶進行交易
- 簡化對 EMEA 客戶的銷售

此步驟是選用的，但建議用於其他區域。CSV 涉及提供有關貴公司、聯絡要點、受益所有權和支援文件的其他資訊。

> **Note**
>
> 只有經過驗證的使用者才能處理退款、修改 KYC 資訊，以及變更銀行帳戶詳細資訊等財務資訊。已驗證的使用者是完成驗證的經 CSV 驗證的使用者和次要使用者。

金融機構和線上企業用來驗證其客戶身分的合規性要求。此要求是由於：

- 韓國報告和使用指定金融交易資訊法案第 5-2 條
- 付款服務指令 (PSD 2)
- 管理金融機構的歐盟反洗黑錢指令

AWS Marketplace 透過 Amazon Web Services EMEA SARL 的 交易會透過 Amazon Payments Europe， S.C.A. (APE) 處理，後者是盧森堡的授權電子金融機構，需要驗證您的身分才能使用付款服務。

> **Note**
>
> 位於英國的銀行帳戶只能收到歐元和英鎊的支出。對於其他貨幣的支付，您將需要位於不同司法管轄區的帳戶。此外，您將僅限於透過 Amazon Web Services EMEA SARL 銷售。

完成 KYC 程序

1. 在 AWS Marketplace 管理入口網站上，選擇設定。
2. 在帳戶摘要區段中，確認顯示的國家/地區正確無誤。

   > **Note**
   >
   > 選擇資訊連結，了解如何變更您的國家/地區。

3. 選擇前往 DNS 資訊，或選擇了解您的客戶 (KYC) 索引標籤，然後選擇開始執行 DNS 合規，然後系統會將您重新導向至 LDAP 註冊入口網站。

   如需如何使用和共用資訊的詳細資訊 AWS Marketplace，請參閱 Amazon Payments Europe 隱私權聲明。
4. 選擇前往 全球開發中心概觀。
5. 在了解您的客戶 (KYC) 概觀上，閱讀所需資訊和文件的清單，並收集所需文件 （如果您尚未這麼做）。然後選擇繼續執行 CSV 合規。
6. 依照指示輸入基本詳細資訊。檢閱 Amazon Payments Europe 條款與條件後，請選擇同意並繼續。

   當您繼續前往下一頁或下一步驟，該動作表示您接受 Amazon Payments Europe 條款與條件。

   如果您有任何問題，請參閱主控台右側的常見問答集 (FAQ)。

7. 依照指示輸入所需的商業資訊，然後選擇下一步。

> **Note**
>
> 每次選擇下一步以前往下一個步驟時，都會儲存您的資訊。

8. 依照指示輸入所需的聯絡人資訊，然後選擇下一步。
9. 選擇受益擁有者是否與聯絡人相同，視需要新增受益擁有者（最多四個），確認您的新增，然後選擇下一步。
10. 選擇法務代表是否與聯絡窗口或受益擁有者相同。如果法定代表是不同的實體，請提供必要的資訊、儲存您的項目，然後選擇下一步。
11. 對於其他文件，請上傳您的商業授權、身分文件和授權書（如適用）。

> **Note**
>
> 如果您需要授權書，您可以使用下列範例：

```
*Letterhead of the company*

POWER TO ACT ON BEHALF OF THE COMPANY
The undersigned **name of Company here** (herein after, the "Company"), duly
 represented by
(name, date of birth, and function) **add full name, date of birth, and
 function of the signatory here**,
confirms that **add full name of the Person of Contact here** is authorized
 to open an Amazon Web
Services account with Amazon Payments, accept the User Agreement and other
 Policies, have access
to the Amazon Web Services account, and initiate transactions in the name
 and on behalf of the Company.

    Dated this:

    Signed by:
```

12. 在檢閱和提交時，檢閱並驗證您已輸入的所有資訊。

    如有必要，您可以選取編輯以返回任何上一節。

13. 選擇提交以進行驗證。

系統將會檢閱您（通常在 24 小時內）的 DNS 合規狀態。審核完成後，您將會透過電子郵件收到通知。整個 ABAC 程序通常需要大約 2 週的時間。

您可以返回設定索引標籤，在帳戶摘要卡上檢視您的 GMT 合規狀態。如需有關您的 帳戶摘要狀態的詳細資訊，請選擇帳戶摘要卡下的了解您的客戶 (KYC) 索引標籤。它會顯示審核中，直到審核完成為止。

通過身分驗證後，您必須在付款資訊索引標籤上提供銀行陳述式，才能透過 APE 接收付款。

（選用）為了解您的客戶程序新增次要使用者

> **Note**
>
> 使用者需要啟用多重要素驗證 (MFA) 才能更新支出資訊。如需 MFA 的詳細資訊，請參閱 IAM 的Multi-Factor Authentication (MFA)。

次要使用者是可以修改 KYC 資訊、控制資金或退款流程，以及變更銀行帳戶詳細資訊等財務資訊的個人。

只有經過身分驗證的次要使用者才能進行上述更新。這些次要使用者受到與根帳戶擁有者相同的持續篩選控制。

若要通過 LDAP 驗證，次要使用者必須完成 中的程序步驟 5：完成了解您的客戶 (KYC) 程序。

為了解您的客戶程序新增次要使用者

1. 要求使用者登入 AWS Marketplace 管理入口網站。
2. 導覽至設定索引標籤。
3. 選擇了解您的客戶 (KYC) 索引標籤，並參閱次要使用者資訊的一節。
4. 選擇完成次要使用者資訊。

   系統會將您重新導向至次要使用者註冊入口網站。
5. 在次要使用者註冊入口網站中，完成必要欄位，然後選擇下一步。
6. 在檢閱和提交頁面上，上傳身分文件 (上傳護照) 和地址證明 (上傳文件) 的副本。
7. 選擇提交以進行驗證。

系統將會檢閱您（通常在 24 小時內）的 DNS 合規狀態。審核完成後，您將會透過電子郵件收到通知。整個 ABAC 程序通常需要大約 2 週的時間。

步驟 6：完成銀行帳戶驗證程序

若要從 Amazon Payments Europe (APE) 接收付款，您必須提供其他資訊，以驗證列於 付款資訊索引標籤中的付款銀行帳戶 AWS Marketplace 管理入口網站。

提供其他銀行資訊

提供其他銀行資訊

1. 登入 AWS Marketplace 管理入口網站，然後選擇設定。
2. 在付款資訊區段中選取更新銀行資訊。
3. 選取適當的支出帳戶。

   驗證狀態會顯示未驗證。
4. 選擇 Verify (驗證)。
5. 系統會將您重新導向至銀行帳戶驗證註冊入口網站，您可以在其中上傳和提交您的銀行對帳單。

   如果您使用Hyperwallet虛擬銀行帳戶解決方案，請參閱 the section called “從 下載您的銀行帳戶驗證陳述式 Hyperwallet”。
6. 在入口網站中，選擇上傳銀行文件，然後選擇提交。

從 下載您的銀行帳戶驗證陳述式 Hyperwallet

對於使用Hyperwallet虛擬銀行帳戶解決方案的賣方，您可以使用下列程序下載Hyperwallet銀行陳述式。然後，您可以按照 中的指示上傳銀行文件步驟 6：完成銀行帳戶驗證程序。

從 下載您的銀行陳述式 Hyperwallet

1. 登入Hyperwallet您的帳戶。
2. 前往存款帳戶資訊頁面。
3. 下載您的銀行帳戶驗證陳述式。

已經是 AWS Marketplace 賣方？

註冊成為賣方之後，請檢閱下列主題，以取得 AWS Marketplace 和後續步驟的詳細資訊：

- 支出 – 身為 AWS Marketplace 賣方，您可以設定支出偏好設定，以接收您所選貨幣的未結餘額。如需詳細資訊，請參閱AWS Marketplace 賣方的支出偏好設定。

- 列出費用 – 如需 AWS Marketplace 賣方列出費用的詳細資訊，請參閱 了解 AWS Marketplace 賣方的清單費用。
- AWS Marketplace 使用 Amazon Payments Europe (APE) 處理 的投訴政策 – 如果您對 Amazon Payments Europe (APE) 提供的服務有任何問題 AWS Marketplace，請告訴我們。您的意見回饋有助於我們為您和所有買方和賣方創造更好的體驗。如需詳細資訊，請參閱提交 Amazon Payments Europe （ APE ） 的 AWS Marketplace 賣方投訴。
- 其他賣方工具 – AWS Marketplace 為您提供其他賣方工具，可讓您更深入了解您的客戶群，並協助您更深入了解您的銷售。如需詳細資訊，請參閱AWS Marketplace 賣方的其他工具。
- 準備您的產品 – 完成將帳戶註冊為賣方之後，您可以建立產品以透過 銷售給買方 AWS Marketplace。如需詳細資訊，請參閱準備您的 產品 AWS Marketplace。

AWS Marketplace 賣方的支出偏好設定

AWS Marketplace 賣方，包括獨立軟體廠商 ISVs) 和通路合作夥伴，可以設定支出偏好設定，以接收其未結餘額。身為賣方，您可以選擇每日或每月支出選項、選擇每月的哪一天來接收支出，以及將貨幣指派給銀行帳戶。多個貨幣可以指派給銀行帳戶。這些區段說明如何設定您的支出偏好設定。他們也提供 AWS Marketplace 訂閱帳單以及 AWS 付款方式的相關資訊。

> **Note**
>
> 美國 ACH 帳戶和 Hyperwallet 帳戶只能以 USD 接收付款。對於非 USD 支出，您必須提供 SWIFT 銀行帳戶。

主題

- 設定您的支出偏好設定
- AWS Marketplace 訂閱帳單
- 賣方如何取得其支出
- 可用貨幣

設定您的支出偏好設定

若要設定您的支出偏好設定，請遵循下列步驟。如果您最近已註冊，則可能需要等待兩個工作天，才能建立公有設定檔和新增支出方法。

1. 登入 AWS Marketplace 管理入口網站，然後選擇設定。
2. 選取付款資訊索引標籤。
3. 在支出方法區段中，選擇新增支出方法。或者，您可以選擇支出方法，然後選擇編輯以更新您的設定。
4. 針對貨幣，選取您偏好的支出貨幣。然後，對於銀行帳戶，選擇您可以接收所選貨幣付款的帳戶。
5. 若要從每日支付變更為每月支付，請選取每月，並在您希望支付處理的月份中選擇一個介於 1-28 之間的數字。
6. 選擇新增支出方法。

> **Note**
>
> 賣方應加入到精細的 IAM 許可，以存取支出偏好設定選項。若要加入至精細的 IAM 許可，請參閱 the section called “賣方的 AWS Marketplace 政策和許可”。

AWS Marketplace 訂閱帳單

AWS 會代表您擔任帳單機制。買方可用的兩個最常見付款選項是信用卡和發票。

如需詳細資訊 AWS Marketplace，請注意下列事項：

- 使用預付付款的購買會在訂閱後立即計費。
- 私有優惠的帳單排程由買方和賣方共同協議。
- 發票付款條件 （包括帳單到期日） 由買方與 達成協議 AWS。這些條款不會向廠商公開。
- 使用彈性付款排程器的私有優惠，必須開立發票做為付款選項。
- 您可以使用帳單營收儀表板來驗證發票。此儀表板總結 AWS 代表您開立的發票，並提供發票建立日期和發票到期日的可見性。

賣方如何取得其支出

- 需要有效的付款方式、已註冊的銀行帳戶和提交的 W9 表單才能支付。
- 付費產品的賣方必須提供 W-8、增值稅 (VAT) 或商品和服務稅 (GST) 註冊號碼，以及在合格司法管轄區的銀行帳戶。
- AWS 以下列方式支付付款：

  - 每日 – 每日支出會在可用時產生。賣方必須具有正餘額才能接收支出。
  - 每月 – 賣方選擇每月一天 (1–28) 來接收支出。集合和支出儀表板會顯示您的支出日期。
- AWS 在買方支付發票之後，使用自動清算中心 (ACH) 轉移或 SWIFT 轉移來支付付款。
- 資金只會在從客戶收集之後撥付。
- 付款大約需要 1 到 2 個工作天才能在付款日期之後送達賣方的銀行。確切的時間點取決於銀行和時區。
- 支出儀表板會在支出後的 AWS Marketplace 管理入口網站 3–5 天內更新。
- 有關已撥付資金和未收帳資金的詳細資訊，請參閱撥付報告，包括任何未結帳戶應收款項。

> **Note**
>
> 如果買方重新安置到不支援其私有優惠中約定貨幣的司法管轄區，您必須以支援的貨幣協商新的私有優惠。如需詳細資訊，請參閱the section called “私有優惠”。

可用貨幣

賣方可以使用合約定價建立私有優惠，並以下列貨幣接收其支出：

- 美元 (USD)
- 歐洲 （ 歐元 ）
- 英國英鎊 (GBP)
- 澳幣 (AUD)
- 日元 （ 日元 ）

> **Note**
>
> 所有具有消費定價的公開優惠和私有優惠只能以 USD 建立。

了解 AWS Marketplace 賣方的清單費用

AWS Marketplace 為 AWS Marketplace 賣方提供標準和區域列示費。列出費用會在 發出的列出費用發票中顯示為單一明細項目 AWS。

> **Note**
>
> 這些列出費用自 UTC 2024 年 1 月 5 日午夜起生效。

了解 中標準優惠清單的費用 AWS Marketplace

了解標準公開優惠、私有優惠、管道合作夥伴私有優惠 (CPPOs) 和專業服務的清單費用。 AWS Marketplace 提供下列產品的清單費用。

公開優惠清單費用

軟體和資料公開優惠的清單費用取決於部署方法：

- Software-as-a-service(SaaS) – 3%
- 伺服器 (Amazon Machine Image (AMI)、容器和機器學習 ） – 20%
- AWS 資料交換 – 3%

私有優惠清單費用

私有優惠的清單費用取決於合約總價值，以及私有優惠是否從先前的私有優惠或 外部的先前協議續約 AWS Marketplace：

- 低於 100 萬 – 3%
- 介於 100 萬到 1000 萬 – 2% 之間
- 等於或大於 1，000 萬 – 1.5%
- 所有續約 – 1.5%

通路合作夥伴私有優惠 (CPPO) 清單費用

無論優惠類型或部署方法為何，CPPO 產品都會收取 0.5% 的列示費用。例如，如果產品是總合約價值低於 100 萬美元的 SaaS 私有優惠，則列出費用將為 3.5%。

專業服務清單費用

所有專業服務方案都會收取 2.5% 的私有優惠列示費。

了解 中區域優惠清單的費用 AWS Marketplace

與下列司法管轄區的買方交易時，標準列示費用將適用額外的區域列示費用（含）。

> **Note**
>
> 區域清單費用與標準清單費用相加。例如，如果產品是 SaaS 私有優惠，其總合約價值低於 100 萬美金，則列示費將為 4% (3% 的標準私有優惠列示費加上 1% 的區域列示費）。

區域	其他區域清單費用	生效日期
南韓	1%	04/01/2025

提交 Amazon Payments Europe（APE）的 AWS Marketplace 賣方投訴

身為 AWS Marketplace 賣方，如果您對 Amazon Payments Europe S.C.A（APE）提供的服務有任何問題，您可以提交投訴。如果您對 Amazon Payments Europe（APE）提供的服務有任何問題，請告訴我們。您的意見回饋有助於我們為您和所有買方和賣方建立更好的體驗。下列各節詳細說明根據我們的投訴政策提交與 Amazon Payments Europe（APE）相關的投訴所需的特定步驟。本主題也為您提供 Amazon Payments Europe（APE）的投訴政策相關資訊，包括解決時間範圍和投訴上報。

> **Note**
>
> 只有 的特定投訴 AWS Marketplace 才會透過下列程序解決。Amazon Payments Europe S.C.A. 提供的服務包括處理付款交易、驗證可能出現在費用中的錯誤，以及資金的支出。

提交投訴

如果您有 Amazon Payments Europe S.C.A. AWS Marketplace 的帳戶，您的投訴將由 Amazon Payments Europe S.C.A. 處理。

若要提交投訴

1. 登入您的AWS Marketplace賣方帳戶。

2. 前往聯絡我們 。
3. 選取商業市場 、賣方帳戶 、註冊 。
4. 提供有關投訴的詳細資訊，然後選擇提交 。

Amazon Payments Europe 投訴解決時間範圍

Amazon Payments Europe S.C.A.（APE） 會在收到您投訴的 15 個工作天內回應您的投訴更新。在無法控制 的特殊情況下APE，投訴的解決時間可能會延長至 35 個工作日，時間是APE第一次收到該投訴的次日。

投訴上報

如果您對我們的回應不滿意，您可以選擇聯絡下列人員，向上呈報您的投訴：

- Amazon Payments Europe 資深管理

  透過傳送電子郵件訊息給資深管理層，提交您的投訴：<ape-management@amazon.lu>。我們會仔細考慮您的意見，並在資深管理層收到您投訴後 15 個工作日內回應。在無法控制 Amazon Payment Europe 的特殊情況下，投訴的解決可能會延長至 35 個工作日，時間是資深管理層第一次收到投訴的那一天。

- Commission de Surveillance du Secteur Financier (CSSF)

  CSSF 是負責盧森堡金融業公司審慎監督的主管機關。您可以在 CSSF 110 聯絡 Route d'Arlon L-2991 Luxembourg 或使用聯絡頁面：https://www.cssf.lu/contacts/。若要取得有關 CSSF和如何聯絡它們的詳細資訊，請參閱 CSSF 網站上的客戶投訴。

- 線上爭議解決

  如果您在歐盟線上開立帳戶，也可以選擇CSSF使用線上爭議解決平台將投訴提交至 。此選項可供使用，因為 Amazon Payments Europe S.C.A. 提供金融服務，而 CSSF是負責其授權的 授權機構。如需詳細資訊，請參閱歐洲委員會網站上的線上爭議解決平台。

AWS Marketplace 賣方的其他工具

AWS Marketplace 為您提供賣方工具，可用來銷售和管理產品。您可以使用這些工具來深入了解您的客戶群，並協助更深入了解您的銷售。本主題提供這些工具的相關資訊，其中包含其他資源的連結。

AWS Marketplace 管理入口網站 是您銷售產品的主要工具 AWS Marketplace。您可以利用 中提供的 go-to-market活動，將產品管理為增量管道收入AWS Marketplace 管理入口網站。如需詳細資訊，請參閱AWS Marketplace 管理入口網站。

AWS Marketplace 提供下列其他賣方工具：

- 中的更多資源 AWS Marketplace 管理入口網站 – 如果您開啟 AWS Marketplace 管理入口網站並登入，您可以在 首頁的 Marketplace 資源區段中看到其他資源的連結。例如，若要在啟動前 90 天內和啟動後 90 天內取得產品行銷支援，您可以從 Marketplace 資源下的 AWS Marketplace 管理入口網站 首頁查看連結180 天 GTM 學院的 。
- AWS Marketplace Commerce Analytics Service – Commerce Analytics Service AWS Marketplace 可讓您透過程式設計方式存取產品和客戶資料 AWS Marketplace。註冊服務之後，您可以透過 AWS SDK 存取您的用量、訂閱和帳單報告。如需詳細資訊，請參閱使用 AWS Marketplace Commerce Analytics Service 存取產品和客戶資料。
- AWS Marketplace Field Demonstration Program (FDP) – FDP 允許 AWS 現場團隊 （內部核准 AWS 的員工） AWS Marketplace 透過 免費使用某些產品和解決方案。如需詳細資訊，請參閱AWS Marketplace 欄位示範計畫。
- 賣方報告、資料摘要和儀表板 – AWS Marketplace 提供工具，用於收集和分析產品銷售的相關資訊。如需詳細資訊，請參閱中的賣方報告、資料饋送和儀表板 AWS Marketplace。

如需詳細資訊，請參閱下列主題。

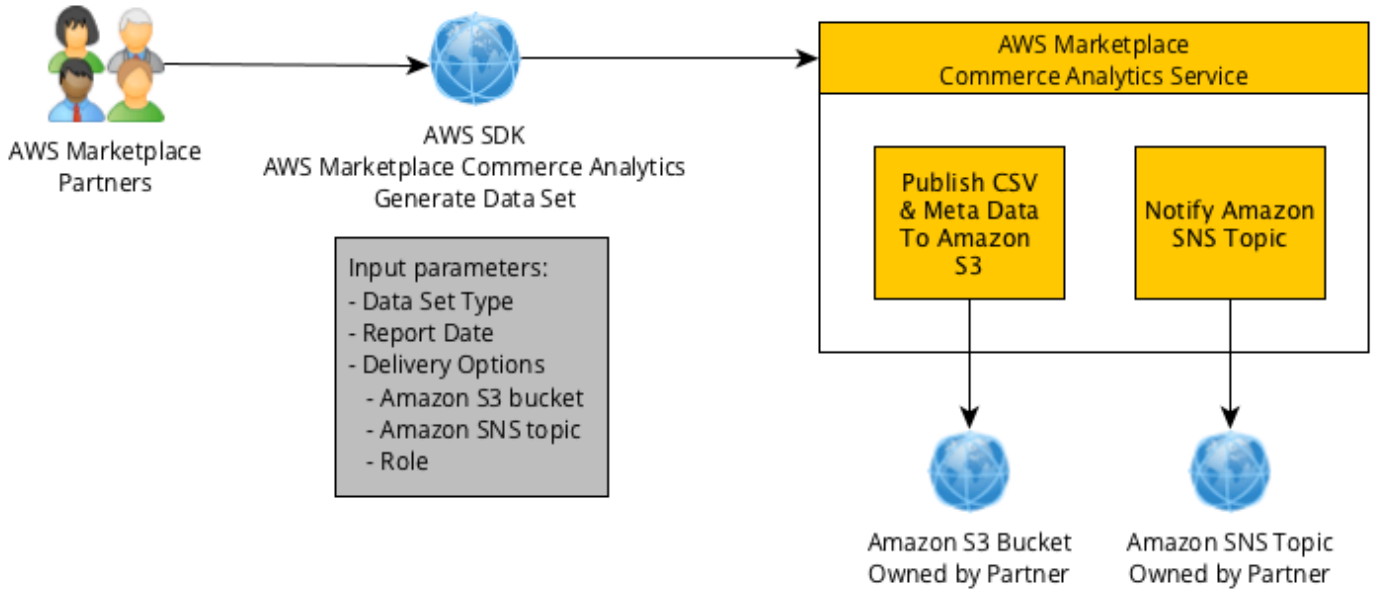
主題

- 使用 AWS Marketplace Commerce Analytics Service 存取產品和客戶資料
- AWS Marketplace 欄位示範計畫
- AWS 合作夥伴助理

使用 AWS Marketplace Commerce Analytics Service 存取產品和客戶資料

使用 AWS Marketplace Commerce Analytics Service，您可以透過 以程式設計方式存取產品和客戶資料 AWS Marketplace。註冊服務之後，您可以透過 AWS SDKs 存取您的用量、訂閱和帳單報告。您使用 SDK 工具請求的資料會以資料集 AWS 帳戶 的形式交付至您的 。大多數資料集對應到與 上可用文字型報告相同的資料AWS Marketplace 管理入口網站。您可以請求特定日期的資料集，並將資料交付至提供的 Amazon S3 儲存貯體。您透過 Amazon Simple Notification Service (Amazon SNS) 收到資料交付通知。本主題提供使用 Commerce Analytics Service AWS Marketplace 的條款與條件。

下列視覺化顯示 Commerce Analytics Service 如何存取 中的產品和客戶資料，並將其做為資料集 AWS Marketplace 交付至 Amazon S3 儲存貯體，並透過 Amazon SNS 啟動通知。



條款與條件

這些 AWS Marketplace Commerce Analytics 服務條款與條件 （以下「CAS 條款」) 包含您使用和存取 AWS Marketplace Commerce Analytics 服務 (「CA 服務」) 的特定條款與條件，自您按一下這些 CAS 條款所呈現的「我接受」按鈕或核取方塊當日起生效，或者，如果您先前使用任何 CA 服務方案時生效。這些 CAS 條款是您與 Amazon Web Services， Inc. 之間 AWS Marketplace 賣方條款與條件 (「AWS Marketplace 賣方條款」) 的附錄。 (「AWS」、「我們」、「我們」或「我們的」)，其條款在此併入本文。如果這些 CAS 條款與 AWS Marketplace 賣方條款之間發生衝突，則適用這些 CAS 條款的條款和條件，但僅限於此類衝突的範圍內，且僅限於您使用 CA 服務。此處使用但未在此定義的大寫詞彙應具有 AWS Marketplace 賣方條款中規定的含義。

1. CA 服務和 CAS 資料。若要符合 CA 服務的存取資格，您必須是受現有 AWS Marketplace 賣方條款約束的 AWS Marketplace 賣方。您接收或可存取的 CA 服務相關資訊和資料 (「CAS 資料」) 構成訂閱者資訊，並受 AWS Marketplace 賣方條款中規定的限制和義務約束。您可以機密使用 CAS 資料來改善和鎖定與您的 AWS Marketplace 內容相關的行銷和其他促銷活動，但前提是您不會 (a) 向任何第三方揭露 CAS 資料； (b) 以任何不符合適用隱私權政策或法律的方式使用任何 CAS 資料； (c) 聯絡訂閱者以影響他們，在 之外進行替代購買 AWS Marketplace； (d) 詆毀我們， 我們的附屬公司、 或其任何個別產品； 或 (e) 根據預期收件人為 AWS Marketplace 訂閱者，以任何類型的目標通訊。

2. CA 服務限制和安全性。您只能透過 CA 服務文件中描述的方式存取（或嘗試存取）CA 服務。使用 CA 服務時，您不會扭曲或遮罩您的身分或用戶端的身分。我們保留自行決定設定和強制執行 CA 服務使用限制的權利，包括但不限於在任何期間允許存取 CA 服務的連線、呼叫和伺服器數量。您同意且不會嘗試規避此類限制。如果我們認為您可能違反這些 CAS 條款或濫用 CA 服務，我們保留限制、暫停或終止您存取 CA 服務的權利。
3. CA 服務登入資料機密性和安全性。CA 服務登入資料（例如密碼、金鑰和用戶端 IDs) 旨在供您用來識別 API 用戶端。您全權負責將登入資料保密，並將採取所有合理措施來避免公開、傳播或未經授權使用此類登入資料，至少包括您為了保護自身類似性質的機密資訊而採取的措施。CA 服務登入資料可能不會內嵌在開放原始碼專案中。您需自行負責使用您的登入資料存取 CA 服務的任何和所有權限。
4. 修改。我們可能會隨時修改這些 CAS 條款，方法是在 AWS 網站上張貼修訂版本，或根據 AWS Marketplace 賣方條款為您提供通知。修改後的條款將在張貼時生效，或者，如果我們透過電子郵件通知您，如電子郵件訊息中所述。在這些 CAS 條款的任何修改生效日期之後繼續使用或存取 CA 服務，即表示您同意受到修改後條款的約束。
5. 終止。這些 CAS 條款和此處授予的 CAS 資料使用權利將在 AWS Marketplace 賣方條款因任何原因終止時終止，無論是否通知您。此外，我們可能會隨時基於任何原因或無任何原因停止提供 CA 服務或終止您對 CA 服務的存取權。

開始使用

如需 AWS Marketplace Commerce Analytics Service 的詳細資訊，包括加入、技術實作和故障診斷資訊，請參閱下列主題。

主題

- 加入 AWS Marketplace Commerce Analytics Service
- 搭配 AWS MarketplaceAWS CLI 和 使用 Commerce Analytics Service 適用於 Java 的 AWS SDK
- 使用 Commerce Analytics Service AWS Marketplace 產生資料集
- 對 AWS Marketplace Commerce Analytics Service 進行故障診斷

加入 AWS Marketplace Commerce Analytics Service

透過 AWS Marketplace Commerce Analytics Service，您可以透過程式設計方式存取產品和客戶資料 AWS Marketplace。若要開始使用 AWS Marketplace Commerce Analytics Service，您必須設定 AWS 帳戶 和 AWS 服務 以使用 AWS Marketplace Commerce Analytics Service。這些章節說明如何設定 AWS 帳戶 和 AWS 服務 以使用 AWS Marketplace Commerce Analytics Service。

使用 AWS Marketplace Commerce Analytics 服務

- 步驟 1： AWS 帳戶 使用 許可設定您的
- 步驟 2：建立目的地 Amazon S3 儲存貯體
- 步驟 3：設定回應通知的 Amazon SNS 主題
- 步驟 4：註冊 Commerce Analytics Service 計畫
- 步驟 5：驗證您的組態

步驟 1： AWS 帳戶 使用 許可設定您的

AWS Marketplace 強烈建議使用 AWS Identity and Access Management (IAM) 角色登入 ， AWS Marketplace 管理入口網站 而不是使用您的根帳戶登入資料。如需 AWS Marketplace Commerce Analytics Service 許可的特定 IAM 許可，the section called “賣方的 AWS Marketplace 政策和許可”請參閱 。為存取您帳戶的人建立個別 使用者，即可將一組唯一的安全登入資料授予每個 使用者。您也可以將不同的許可授予每個 使用者。如有需要，您隨時可以變更或撤銷 使用者的許可。

步驟 2：建立目的地 Amazon S3 儲存貯體

Commerce Analytics Service 會將您請求的資料交付至您指定的 Amazon S3 儲存貯體。如果您已有要使用的 Amazon S3 儲存貯體，請繼續下一個步驟。

如果您沒有 Amazon S3 儲存貯體，或想要特別為此資料建立 Amazon S3 儲存貯體，請參閱如何建立 Amazon S3 儲存貯體。

步驟 3：設定回應通知的 Amazon SNS 主題

Commerce Analytics Service 使用 Amazon SNS 傳送回應通知。服務會將訊息發佈至此主題，以在資料集可用或發生錯誤時通知您。如果您已有用於此目的的 Amazon SNS 主題，請繼續下一個步驟。

如果您沒有為此服務設定 Amazon SNS 主題，請立即設定。如需說明，請參閱建立主題。

為您建立的主題記錄 Amazon Resource Name (ARN) 主題，因為呼叫 服務需要 ARN。

步驟 4：註冊 Commerce Analytics Service 計畫

在您使用儲存貯體主題和名稱的 ARN 設定服務之後，Commerce Analytics Service 會存取 Amazon S3 儲存貯體和 Amazon SNS 主題。

啟用存取

1. AWS Marketplace 管理入口網站 使用 AWS 帳戶 您用來管理 AWS Marketplace 產品的 登入 。

2. 確保您擁有註冊 AWS Marketplace Commerce Analytics Service 所需的 IAM 許可。
3. 導覽至 Commerce Analytics Service 註冊頁面。
4. 輸入 Amazon S3 儲存貯體名稱和 Amazon SNS 主題 ARN，然後選擇註冊。
5. 在許可頁面上，選擇允許。
6. 在 上 AWS Marketplace 管理入口網站，在成功訊息中記錄角色名稱 ARN。您需要 ARN 才能呼叫服務。

> **Note**
>
> 加入 Commerce Analytics Service 會在您的 中建立 IAM 角色 AWS 帳戶。IAM 角色允許 AWS Marketplace 寫入 Amazon S3 儲存貯體，並將通知發佈至 Amazon SNS 主題。 AWS Marketplace 會使用帳戶 452565589796 使用此 IAM 角色執行這些相關動作。

步驟 5：驗證您的組態

最後一個步驟是驗證您的組態是否如預期般運作。

若要測試組態

1. 下載、安裝和設定 AWS 命令列界面 (AWS CLI)。
2. 使用 AWS CLI執行此命令。

```
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}"
```

- 對於 `--data-set-publication-date`，使用 ISO-8601 格式將 取代`{TODAY'S DATE}`為目前日期，`YYYY-MM-DDT00:00:00Z`其中 `YYYY`是四位數年份， `MM` 是兩位數月份，而 `DD`是兩位數日期。
- 對於 `--role-name-arn`，將 `{YOUR-ROLE-NAME-ARN}`取代為您從 中註冊程序取得之角色的 ARN步驟 4：註冊 Commerce Analytics Service 計畫。

- 對於 --destination-s3-bucket-name，請以您在 中建立的 Amazon S3 儲存貯體名稱取代 {amzn-s3-demo-bucket}步驟 2：建立目的地 Amazon S3 儲存貯體。
- 對於 –sns-topic-arn，請將 {YOUR-SNS-TOPIC-ARN} 取代為您在 中建立的 Amazon SNS 主題步驟 3：設定回應通知的 Amazon SNS 主題。

如果您收到包含來自 服務的 dataSetRequestId 回應的回應，表示您已完成加入程序。成功的回應如下所示：

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

搭配 AWS MarketplaceAWS CLI 和 使用 Commerce Analytics Service 適用於 Java 的 AWS SDK

透過 AWS Marketplace Commerce Analytics Service，您可以透過程式設計方式存取產品和客戶資料 AWS Marketplace。Commerce Analytics Service AWS Marketplace 透過 AWS 開發套件提供。您可以使用 AWS CLI和 適用於 Java 的 AWS SDK與 Commerce Analytics Service 互動。這些章節說明如何使用適用於 Java 的 AWS CLI 和 開發套件來實作 Commerce Analytics Service。

主題

- Commerce Analytics Service 的 IAM 政策
- 使用 提出請求 AWS CLI
- 使用 提出請求 適用於 Java 的 AWS SDK

Commerce Analytics Service 的 IAM 政策

若要允許您的使用者使用 Commerce Analytics Service，需要下列許可。

使用下列 IAM 許可政策註冊 AWS Marketplace Commerce Analytics Service。

JSON

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

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

使用以下 IAM 許可政策，允許使用者向 AWS Marketplace Commerce Analytics Service 提出請求。

JSON

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

如需詳細資訊，請參閱《IAM 使用者指南》中的在 IAM 主控台中建立政策。

使用 提出請求 AWS CLI

若要開始使用，請下載 AWS CLI。下列 AWS CLI 範例會針對 2017 年 10 月 1 日的每小時/每月訂閱資料集提出請求。此資料集會使用字首 demo-prefix 發佈至示範儲存貯體 Amazon S3 儲存貯體，並將通知訊息傳送至示範主題 Amazon SNS 主題。

```
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"
```

此請求會傳回每個請求的唯一識別符。您可以使用此識別符，將請求與發佈至 Amazon SNS 主題的通知建立關聯。下列範例是此識別符的範例。

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

使用 提出請求 適用於 Java 的 AWS SDK

若要開始，請下載 AWS Java 開發套件。下列 適用於 Java 的 AWS SDK 範例會針對 2015 年 10 月 1 日的每小時/每月訂閱資料集提出請求。此資料集會使用字首 demo-prefix 發佈至示範儲存貯體 Amazon S3 儲存貯體，並將通知訊息傳送至示範主題 Amazon SNS 主題。

```
/*
* 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 ### Java # AWS SDK.
* <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);
}
}
```

您應該預期類似此範例的結果。

```
================================================================
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
```

使用 Commerce Analytics Service AWS Marketplace 產生資料集

註冊 AWS Marketplace Commerce Analytics Service 之後，您可以透過 AWS CLI 和 存取您的用量、訂閱和帳單報告 適用於 Java 的 AWS SDK。您使用 SDK 工具請求的資料會以資料集 AWS 帳戶 的形式交付至您的 。大多數資料集對應到與 上可用的文字型報告相同的資料AWS Marketplace 管理入口網站。如需詳細資訊，請參閱使用 AWS Marketplace Commerce Analytics Service 存取產品和客戶資料。這些區段提供有關 Commerce Analytics Service 產生之資料集的相關參數、回應和輸出的資訊。

主題

- GenerateDataSet 參數
- 回應
- 輸出

`GenerateDataSet` 參數

AWS Marketplace Commerce Analytics Service 公開了一種方法 `GenerateDataSet`，您可以用來請求將資料集發佈到您的 Amazon S3 儲存貯體。下表列出 的參數`GenerateDataSet`。

資料集參數

欄位	Description
資料集類型	此資料集將作為請求的結果傳回。
資料集發佈日期	資料集發佈的日期。 對於每日資料集，請提供所需日期的日層級精細程度。 對於每月資料集，請提供所需月份的月份層級精細程度日期。系統會忽略日值。
角色名稱 ARN	角色的 ARN 與連接的許可政策，該政策可讓 服務存取您的 資源。
目的地 Amazon S3 儲存貯體名稱	目的地 Amazon S3 儲存貯體的名稱（易記的名稱，而非 ARN)。您的資料集會發佈至此位置。
目的地 Amazon S3 字首	（選用） 已發佈資料集的 Amazon S3 字首，類似於標準檔案系統中的目錄路徑。 例如，如果指定儲存貯體名稱`mybucket`和字首 `myprefix/mydatasets` ，輸出檔案會發佈至 `s3://amzn-s3-demo-bucket/myprefix/mydatasets/outputfile` 。 如果字首目錄結構不存在，則會建立。 如果未提供字首，資料集會發佈至 Amazon S3 儲存貯體根目錄。
SNS Topic ARN (SNS 主題 ARN)	發佈資料集或發生錯誤時通知的 Amazon SNS 主題 ARN。

回應

Commerce Analytics Service AWS Marketplace 會傳回兩個回應。第一個是同步，它會立即傳回，第二個是非同步，它會使用 Amazon SNS 傳回。同步回應類似於此範例。

資料集參數

欄位	Description
資料集請求 ID	代表服務特定請求的唯一識別符。此識別符可用來將請求與 Amazon SNS 主題上的通知建立關聯。

非同步回應會以 JSON 格式的文件發佈到您的 Amazon SNS 主題，與此範例類似。

資料集參數

欄位	Description
資料集 S3 位置	已交付資料集的儲存貯體名稱和金鑰。
資料集中繼資料 S3 位置	已交付資料集中繼資料檔案的儲存貯體名稱和金鑰。
資料集請求 ID	代表服務特定請求的唯一識別符。此識別符可用來將請求與 Amazon SNS 主題上的通知建立關聯。
成功	如果操作成功，則為「True」；如果失敗，則為「false」。
訊息	（選用）如果發生錯誤（例如，「成功」是「錯誤」），此訊息會包含失敗的相關資訊。

範例 JSON 格式的非同步回應

```
{
  "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
 }
```

輸出

請求成功後，請求的資料集會以 .csv 檔案的形式交付到您的 Amazon S3 儲存貯體。JSON 格式中繼資料檔案會發佈到與資料集檔案相同的位置。中繼資料檔案提供有關資料集和原始請求參數的實用資訊。中繼資料檔案與資料集檔案的名稱相同，但結尾為副檔名 .meta.json。下表列出 .csv 檔案中的中繼資料欄位。

中繼資料欄位

欄位	Description
資料集請求 ID	代表服務特定請求的唯一識別符。此識別符可用來將請求與 Amazon SNS 主題上的通知建立關聯。
資料集涵蓋範圍	定義資料涵蓋範圍的開始日期/時間和結束日期/時間。這些日期採用 ISO 8601 格式。
資料集請求參數	`GenerateDataSet` 方法的原始請求參數。
資料集 S3 位置	已交付資料集的儲存貯體名稱和金鑰。
資料集中繼資料 S3 位置	已交付資料集中繼資料檔案的儲存貯體名稱和金鑰。

以下是 JSON 格式中繼資料內容的範例。

```
{
"dataSetRequestId": "43d7137b-8a94-4042-a09d-c41e87f371c1",
"dataSetCoverageRange": {
"startDateTime": "2014-06-08T00:00:00.000Z",
"endDateTime": "2014-06-08T23:59:59.000Z"
},
"dataSetRequestParameters": {
"sellerAccountId": "123412341234",
"dataSetType": "customer_subscriber_hourly_monthly_subscriptions",
"dataSetPublicationDate": "2014-06-09T00:00:00.000Z",
"roleNameArn": "arn:aws:iam::123412341234:role/MarketplaceCommerceAnalyticsRole",
"destinationS3BucketName": "demo-bucket",
"destinationS3Prefix": "demo_prefix/customer_subscriber_hourly_monthly_subscriptions",
"snsTopicArn": "arn:aws:sns:us-west-2:123412341234:demo-topic"
},
"dataSetS3Location": {
"bucketName": "demo-bucket",
"key": "demo_prefix/customer_subscriber_hourly_monthly_subscriptions_2014-06-09.csv"
},
"dataSetMetaDataS3Location": {
"bucketName": "demo-bucket",
"key": "demo_prefix/
customer_subscriber_hourly_monthly_subscriptions_2014-06-09.meta.json"
}
}
```

如需可用資料集的完整清單，包括可用性日期，請參閱 AWS SDK 文件。

對 AWS Marketplace Commerce Analytics Service 進行故障診斷

您可以對 Commerce Analytics Service AWS Marketplace 的問題進行疑難排解，該服務以程式設計方式提供產品和客戶資料 AWS Marketplace。當您遇到錯誤或其他組態問題時，您可能需要對 Commerce Analytics Service 進行故障診斷。以下各節會引導您完成疑難排解程序，涵蓋診斷和解決 Commerce Analytics Service 常見問題的步驟。

由於允許清單問題，我無法存取服務。

如果您尚未在 上註冊為賣方 AWS Marketplace，請造訪 AWS Marketplace 管理入口網站 進行註冊。如果您已註冊為 上的賣方 AWS Marketplace，請聯絡AWS Marketplace 賣方營運團隊。

我無法為過去的日期請求資料集，即使軟體開發套件文件指出該日期應該可用。

雖然資料集列出為過去特定日期可用，但我們只有自您加入以來的資料 AWS Marketplace。如果您認為這是錯誤的，請聯絡AWS Marketplace 賣方營運團隊。

當我呼叫 服務時，會收到錯誤訊息「無法連線至端點 URL：https：//https://marketplacecommerceanalytics.eu-central-1.amazonaws.com/」

Commerce Analytics Service AWS Marketplace 僅適用於美國東部（維吉尼亞北部）區域。您必須對 Commerce Analytics Service 對`us-east-1`端點進行所有呼叫。

如果您使用的是 AWS CLI，請將「`--region`旗標新增至每個呼叫，並將 指定 AWS 區域 為 `us-east-1`，如下列範例所示。

```
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
```

我想要使用與我完成加入程序時選取的 Amazon S3 儲存貯體或 Amazon SNS 主題不同的 Amazon S3 儲存貯體或主題。

註冊 AWS Marketplace Commerce Analytics Service 時，您指定了 Amazon S3 儲存貯體和 Amazon SNS 主題。加入程序會設定您的 IAM 許可，僅允許服務存取這些特定資源。若要使用不同的資源，您需要修改 IAM 政策：

1. 登入 AWS Management Console 並開啟位於 https：//https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 選擇 IAM 主控台左側的角色。
3. 選擇 MarketplaceCommerceAnalyticsRole。
4. 如果尚未展開，請展開內嵌角色區段。
5. 找到名稱開頭為 的政策，oneClick_MarketplaceCommerceAnalyticsRole然後選擇編輯政策。

6. 在政策文件中，找到指定與您要修改之服務相關動作的 區段。例如，若要變更 Amazon S3 儲存貯體，請尋找包含以 s3 開頭之動作的 區段： ，並變更其各自的資源選擇以指定新的 Amazon S3 儲存貯體。

如需 IAM 政策的詳細資訊，請參閱下列指南：https：//https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

當我呼叫 **GenerateDataSet**動作時收到**AccessDeniedException**錯誤

如果您的使用者沒有呼叫 所需的許可，就會發生這種情況GenerateDataSet。下列程序概述使用 IAM 主控台建立具有這些許可的 IAM 政策所需的步驟，並將許可新增至您的使用者、群組或角色。

若要使用 JSON 政策編輯器來建立政策

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在左側的導覽窗格中，選擇 Policies (政策)。

   如果這是您第一次選擇 Policies (政策)，將會顯示 Welcome to Managed Policies (歡迎使用受管政策) 頁面。選擇 Get Started (開始使用)。
3. 在頁面頂端，選擇 Create policy (建立政策)。
4. 在政策編輯器中，選擇 JSON 選項。
5. 輸入下列 JSON 政策文件：

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

6. 選擇下一步。

> **Note**
>
> 您可以隨時切換視覺化與 JSON 編輯器選項。不過，如果您進行變更或在視覺化編輯器中選擇下一步，IAM 就可能會調整您的政策結構，以便針對視覺化編輯器進行最佳化。如需詳細資訊，請參閱 IAM 使用者指南中的調整政策結構。

7. 在檢視與建立頁面上，為您正在建立的政策輸入政策名稱與描述 (選用)。檢視此政策中定義的許可，來查看您的政策所授予的許可。
8. 選擇 Create policy (建立政策) 儲存您的新政策。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循「IAM 使用者指南」的為第三方身分提供者 (聯合) 建立角色中的指示。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照「IAM 使用者指南」的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台) 中的指示。

我的問題未列在此處。

請聯絡AWS Marketplace 賣方營運團隊。

AWS Marketplace 欄位示範計畫

AWS Marketplace Field Demonstration Program (FDP) 允許 AWS 現場團隊 （ 內部核准 AWS 的員工 ） AWS Marketplace 免費透過 使用某些產品和解決方案。

核准的 AWS 員工範例可能包括解決方案架構師和銷售和行銷專業人員。FDP 可讓這些員工示範產品功能，以教育客戶工作負載並納入客戶工作負載。

支援下列產品類型：

- Amazon Machine Images (AMI)

- 容器
- 機器學習演算法和模型套件 (SageMaker AI)
- 資料集 (AWS Data Exchange)

> **Note**
>
> 對於 AWS Data Exchange 產品，FDP 僅適用於公開優惠為 0 美元（免費）的產品。對於已啟用訂閱驗證的 AWS Data Exchange 產品，供應商需要核准訂閱請求。如需訂閱驗證的詳細資訊，請參閱《AWS Data Exchange 使用者指南》中的訂閱者的訂閱驗證。

當您註冊為 AWS Marketplace 賣方時，您會自動註冊 FDP 計畫。若要選擇退出，請向AWS Marketplace 賣方營運團隊提交支援請求。

若要檢視此計畫下產品用量的相關資訊，請參閱每月計費營收報告AWS 的欄位示範用量一節。

AWS 合作夥伴助理

AWS Partner Assistant 是 的生成式 AI AWS Partner支援聊天機器人。它可以從 Partner Central 和 AWS Marketplace 管理入口網站 (AMMP) 存取，並使用 Amazon Q Business，這是根據註冊合作夥伴獨有的文件進行訓練。

> **Note**
>
> 若要存取 AWS 合作夥伴助理，合作夥伴必須將其 AWS Partner Central 帳戶與其 AWS 帳戶連結。如需詳細資訊，請參閱AWS 《Partner Central 入門指南》中的將 Partner Central 帳戶與 AWS Marketplace 賣方帳戶連結。 AWS

聊天機器人提供 AWS 與服務、 AWS Marketplace 賣方指引和 AWS 合作夥伴網路獎勵、利益和計劃相關的各種主題的即時答案和文件摘要。如果聊天機器人無法提供回應，則建議您如何向支援團隊提交票證。您也可以使用拇指向上和拇指向下按鈕，在聊天機器人內提供意見回饋，這有助於改善聊天機器人的準確性和品質。

以下是與 AWS 合作夥伴助理的典型互動：

1. 提出問題：從 AWS Partner Central 或 AMMP 開啟聊天機器人，並使用日常語言提出問題。
2. AWS 合作夥伴助理處理請求：聊天機器人使用自然語言處理 (NLP) 來解譯您的查詢。

3. AWS 合作夥伴助理會產生回應：聊天機器人會傳回相關的回應，其中包含其他資源。
4. 提供意見回饋：聊天機器人會提示您提供意見回饋。

優點和功能

AWS 合作夥伴助理提供自助式選項，可改善對資訊的存取。

- 立即回答：合作夥伴通常必須 AWS 直接聯絡 尋求協助，但 AWS 合作夥伴助理始終可用。
- 準確的答案：合作夥伴會收到實際且及時的資訊，且回應變異性最低。
- 自助式支援：減少尋找相關資源所需的時間。
- 進階回應：提供自然、直覺、類似人類的回應。
- 內容理解：提供有關複雜主題的其他資訊。
- 自我學習系統：透過從互動中學習、隨著時間的推移微調回應，以及擴展其知識庫來持續改進。
- 可擴展支援：能夠同時處理大量查詢。

如何使用 AWS 合作夥伴助理

1. 登入 AWS Partner Central 或 AMMP。
2. 存取聊天：
   a. 選擇右下角的紫色浮動圖示。
   b. 選擇其中一個建議的提示，或開始在文字欄位中輸入您的問題。
   c. Chatbot 會以來源和連結回應。
3. 從 Partner Central 存取聊天機器人：
   a. 選擇支援功能表。
   b. 從下拉式清單中選取與 AWS 合作夥伴助理的即時聊天。
   c. （選用）從支援票證存取聊天機器人。在提交支援票證之前，請選擇詢問 AWS 合作夥伴助理以取得立即的答案。
4. 從 AMMP 存取聊天機器人：
   a. 在聯絡我們頁面中，選擇詢問 AWS 合作夥伴助理。
   b. （選用）從支援票證存取聊天機器人。在提交支援票證之前，請從聯絡我們頁面選擇提交支援請求。選擇詢問 AWS 合作夥伴助理以取得立即的答案。
5. （選用）提供意見回饋：
   a. 在每次回應之後，您可以選擇提供意見回饋：如果回應符合您的需求，請選擇大拇指。

b. 如果回應沒有幫助，請選擇拇指向下。
c. 如果您選擇拇指向下，請在文字方塊中提供更多詳細資訊，或從挑選清單中選擇最能描述回應為何沒有幫助的選項。
d. 透過選擇聊天機器人視窗左下角的意見回饋來評價您的體驗：選擇意見回饋以開啟具有評分比例的快顯視窗。使用文字欄位進行任何其他評論或意見回饋。

合作夥伴助理常見問答集

以下是 AWS 合作夥伴助理的常見問題。

什麼是 AWS 合作夥伴助理？

AWS Partner Assistant 是採用 AI 技術的生成式聊天機器人，可協助合作夥伴尋找資訊和回答問題。

如何存取 AWS 合作夥伴助理？

AWS 合作夥伴助理可供 AWS 合作夥伴和 Marketplace 賣方使用，但不適用於 Marketplace 買方。您可以透過 AWS Partner Central 或 AWS Marketplace Management Portal (AMMP) 從任何頁面存取聊天機器人。

AWS 合作夥伴助理是否隨時可用？

是。

Partner Assistant AWS 支援哪些語言？

目前， AWS 合作夥伴助理僅支援英文，但在未來版本中將支援其他語言。

AWS 合作夥伴助理可以做什麼？

聊天機器人提供有關 AWS 服務、 AWS Marketplace 賣方指引和 AWS Partner 計劃、獎勵和利益的即時答案和文件摘要。Chatbot 使用合作夥伴獨有的已發佈內容程式庫。

我可以詢問 AWS 合作夥伴助理什麼？

聊天機器人提供有關 AWS 服務、 AWS Market Marketplace 賣方指引，以及 AWS 合作夥伴網路計劃、獎勵和利益的即時答案和文件摘要。Chatbot 使用合作夥伴獨有的已發佈內容程式庫。

我不應該詢問 AWS 合作夥伴助理什麼問題？

對於需要深入疑難排解、帳戶驗證或存取敏感資料的問題，請直接聯絡 AWS 客戶支援。

AWS 合作夥伴助理可以提供個人化的回應嗎？

目前，AWS 合作夥伴助理僅提供一般資訊和指導。它無法產生創意內容或執行任務。如果 AWS 合作夥伴助理無法充分回答您的問題，請提交支援票證或提交意見回饋。以下是其他範例：

執行	不要
在哪裡可以找到我的方案續約日期？	我的方案續約日期是什麼時候？
如何註冊路徑？	在軟體路徑中註冊我們。
如何申請 MSP 計劃？	我可以申請 MSP 計劃嗎？

> **Note**
>
> 對於帳戶相關問題，合作夥伴必須使用現有的AWS 支援管道。

合作夥伴可以使用 AWS 合作夥伴助理來變更其帳戶或服務嗎？

否，但它可以引導您如何進行這些變更，或引導您支援團隊。

合作夥伴助理如何處理 AWS 機密資訊？

AWS Partner Assistant 旨在保護使用者隱私權和資料，但不建議共用敏感資訊。

如果 AWS 合作夥伴助理不了解我的問題，會發生什麼情況？

如果 AWS 合作夥伴助理無法回答您的問題，請嘗試重新表達。或者，它可以引導您到支援團隊尋求協助。在此案例中，請提交拇指向下意見回饋和其他內容，這將有助於我們改善聊天機器人的效能。或者，您可以從常見類別的選擇清單中提供結構化意見回饋。

如果 AWS 合作夥伴助理沒有回應，該怎麼辦？

請嘗試重新整理頁面或重新啟動應用程式。如果問題仍然存在，請向我們的支援團隊回報問題。

如果 AWS Partner Assistant 提供不正確的資訊，該怎麼辦？

如果您認為 AWS Partner Assistant 提供了不正確的資訊，請使用拇指向下選項在工作階段結束時提供意見回饋。

如果我不想使用，是否可以關閉 AWS 合作夥伴助理？

AWS 合作夥伴助理是選用功能。如果您不想使用它，請聯絡我們的支援團隊。

準備您的 產品 AWS Marketplace

下列主題說明如何準備要透過 銷售的軟體產品 AWS Marketplace。準備您的產品涉及設定套件、設定定價方案、決定要列出產品的類別，以及新增關鍵字，讓您的產品出現在相關搜尋中。如果您是獨立軟體廠商 (ISV)、管道合作夥伴、受管服務供應商 (MSP) 或個人，而這些個人提供適用於 AWS 產品和服務的內容，您可以使用此程序來準備發佈產品 AWS Marketplace。

主題

- 的產品交付 AWS Marketplace
- 在 上執行之產品的指定 AWS
- 的產品定價 AWS Marketplace
- 您 AWS Marketplace 產品的區域和國家/地區
- 翻譯和語言
- 在 中使用標準化合約 AWS Marketplace
- 提供 AWS Marketplace 產品的中繼資料
- 建立 的 AMI 和容器產品使用說明 AWS Marketplace
- 最佳化您的 AWS Marketplace 產品以進行搜尋
- 使用促銷媒體增強您的 AWS Marketplace 產品

的產品交付 AWS Marketplace

每個產品交付方法都有數種封裝、定價和交付選項。在您註冊支援它的程式 AWS Marketplace 之前，您無法以 上的賣方身分使用某些方法。

您可以使用標準清單價格和最終使用者授權合約 (EULA) 來建立產品。您也可以使用自訂定價和 EULAs 為個別客戶建立私有優惠。如果您需要對合約條款進行其他變更，您可以與 AWS Marketplace 團隊合作來建立自訂私有優惠。為了簡化採購程序，您可以針對公開產品清單和私有優惠使用標準化授權條款。

> **Note**
>
> AWS 為特定賣方提供提供引導式示範的選項 AWS Marketplace。如果您是符合 APN Customer Engagements (ACE) 資格的 AWS 合作夥伴網路 (APN) 合作夥伴，並且想要提供此選項給買方，請參閱 新增私有優惠和示範請求按鈕 以取得更多資訊。

下表列出您可以用來交付軟體產品的方法，以及 AWS Marketplace 買方用來在 AWS Marketplace 主控台中尋找每種交付方法的篩選條件。

產品交付方法	主控台上的交付方法篩選條件	描述
單一 AMI	Amazon Machine Image (AMI)	您可以為產品提供單一自訂 Amazon Machine Image (AMI)。AMI 提供啟動 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體所需的資訊。 買方可以使用單一 AMI，在您的產品已安裝且可供使用的情況下建立 Amazon EC2 執行個體。 如需詳細資訊，請參閱中的 AMI 型產品 AWS Marketplace。
使用 AWS CloudFormation 範本交付AMIs	CloudFormation 範本	您可以使用 CloudFormation 範本列出交付給 AWS Marketplace 買方的 AMI 型產品。 如需以 CloudFormation 範本形式交付 AMIs 的詳細資訊，請參閱使用 的 AMI 型交付 AWS CloudFormation。 如需 CloudFormation 範本的詳細資訊，請參閱AWS CloudFormation 《使用者指南》中的AWS CloudFormation 概念。
EC2 Image Builder 元件型產品	EC2 Image Builder 元件	您可以使用 EC2 Image Builder 元件列出交付給 AWS Marketplace 買方的產品。

產品交付方法	主控台上的交付方法篩選條件	描述
		如需使用 EC2 Image Builder 元件的詳細資訊，請參閱《AWS Marketplace API 參考》中的使用 EC2 Image Builder 元件產品。
容器型產品或應用程式	容器	您可以交付容器映像中封裝的產品。容器產品由選項組成，這些選項是一組可一起運作的容器映像和部署範本。 如需詳細資訊，請參閱上的容器型產品 AWS Marketplace。
資料產品	AWS Data Exchange	您可以使用 AWS Data Exchange 來建立資料產品。 如需有關透過 AWS Data Exchange 發佈和管理資料產品和優惠的資訊，請參閱《AWS Data Exchange 使用者指南》中的在 AWS Data Exchange 上提供資料產品。

產品交付方法	主控台上的交付方法篩選條件	描述
機器學習演算法和模型套件	SageMaker 模型	您可以使用 Amazon SageMaker AI 建立演算法或模型套件，然後將其發佈到其中 AWS Marketplace。 如需交付機器學習演算法和模型套件的詳細資訊，請參閱 機器學習產品。 如需 SageMaker AI 的相關資訊，請參閱《Amazon SageMaker AI 開發人員指南》中的什麼是 SageMaker AI？。Amazon SageMaker
軟體即服務 (SaaS)	SaaS	您可以為 SaaS 產品提供以訂閱為基礎、以合約為基礎或與耗用量定價模式簽訂的合約。 如需詳細資訊，請參閱中的 SaaS 型產品 AWS Marketplace。
專業服務	專業服務	您可以提供專業服務來支援或使用其他 AWS Marketplace 產品。

在 上執行之產品的指定 AWS

> **Note**
>
> 下列指定將於 2025 年 5 月 1 日生效。

部署在 上的產品會在 AWS Marketplace 搜尋結果及其產品詳細資訊頁面中 AWS 收到特殊指定。下表列出可接收指定項目的產品交付類型。

產品交付方法	指定為部署於 AWS
以 AMI 為基礎的	是
容器型	是，但部署在 Amazon EKS Anywhere 上的產品除外。如需啟用此功能之容器產品的詳細資訊，請參閱本指南將 AWS Marketplace for Containers Anywhere 與 License Manager 整合稍後的 。
SageMaker AI 演算法和模型套件型	是
以 SaaS 為基礎的	僅適用於完全執行的產品 AWS。如需詳細資訊，請參閱本指南架構準則稍後的 。
資料型	是
專業服務型	否

的產品定價 AWS Marketplace

身為 AWS Marketplace 賣方，您可以選擇您產品的定價模式。對於付費產品， 會向客戶 AWS Marketplace 收取軟體費用。所有定價均以美元 (USD) 為基礎。免費或開放原始碼軟體免費提供給客戶。本主題提供有關 中軟體產品的定價模型和定價的資訊 AWS Marketplace。

如需退款的詳細資訊，請參閱 中的產品退款 AWS Marketplace。

主題

- 定價模型
- 變更定價模型
- 變更價格
- 私有優惠
- 中的產品退款 AWS Marketplace

定價模型

下列主題提供 中可用定價模型的一般資訊 AWS Marketplace。

主題

- 年度定價
- 用量定價
- 合約定價
- 使用您自己的授權定價

如需特定產品交付方法定價模型的相關資訊，請參閱：

- 的 AMI 產品定價 AWS Marketplace
- 的容器產品定價 AWS Marketplace
- 的機器學習產品定價 AWS Marketplace
- 中的 SaaS 產品定價 AWS Marketplace
- 在 中為專業服務產品定價 AWS Marketplace

年度定價

年度定價模式可讓您提供產品給可購買 12 個月訂閱的客戶。例如，與長時間每小時執行相同的產品相比，訂閱定價最多可節省 40%。客戶會在訂閱時收到合約完整金額的發票。如需如何向客戶呈現年度訂閱的詳細資訊，請參閱付費容器產品的 AMI 訂閱或定價模型。 https://docs.aws.amazon.com/marketplace/latest/buyerguide/buyer-what-is-aws-marketplace-for-containers.html#what-is-aws-marketplace-for-containers-pricing

使用年度訂閱時的考量如下：

- 依執行個體類型定義年度定價。所有 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體類型都可以相同，或每個執行個體類型都可以不同。
- 所有年度執行個體類型也必須定義每小時執行個體類型。 AWS Marketplace 不提供年度限定定價或沒有相同產品的每小時。對於任何提供年度定價的產品，也需要指定每小時定價。
- 如果每小時價格也為 0 USD，且已定義其他非 0 USD 的年度執行個體類型，則特定執行個體類型允許 0 USD 的年度價格。
- 在年度訂閱期間結束時，會開始以每小時價格向客戶收費。

- 如果客戶購買 X 年度訂閱，但在 Y 執行個體上執行 Y 軟體，則客戶需支付年度訂閱未涵蓋之 (Y-X) 執行個體的每小時軟體價格。因此，所有年度定價執行個體類型都必須包含每小時費率。
- 使用賣方私有優惠，您可以提供多年（最多 3 年）或具有預付付款的自訂持續時間 AMI，或彈性的付款排程。如需多年期和自訂持續時間合約的詳細資訊，請參閱 為您的 AWS Marketplace 產品準備私有優惠和 私有優惠分期方案。

如果您在 中提供年度產品 AWS Marketplace，即表示您同意年度產品的特定退款政策，位於 中的檔案上傳者文件區段AWS Marketplace 管理入口網站。

價格變更

您可以每 90 天將每年價格 ($ 值，例如每年 1，000 美元變更為每年 1，200 美元）。不過，您必須提前 90 天通知現有客戶年度定價。新價格將適用於新的訂閱，但不會影響現有的訂閱。

> **Note**
>
> 在 90 天的通知期間，您無法更新支援的執行個體類型。

價格變更只有在價格在自動續約日期前至少 90 天變更時，才會對自動續約生效。客戶將在自動續約之前收到包含新價格的電子郵件訊息。

最終使用者授權合約

您在產品詳細資訊頁面上提供的 EULA 涵蓋 AWS 了客戶在年度訂閱下使用軟體 12 個月的情況 AWS Marketplace。

用量定價

用量定價模式，也稱為隨需付費定價，可讓您將產品提供給僅支付其使用量的客戶。

身為賣方，您可以選擇下列其中一個用量類別：

- 使用者
- 主機
- 頻寬
- 資料

- 層
- 單位（適用於自訂類別）

您也可以為產品定義最多 24 個維度。當軟體呼叫 API 時，會測量並報告費用。我們建議賣方根據其使用案例，將 API 設定為每小時呼叫一次做為最佳實務。所有用量都會每月計算，並使用與現有 AWS Marketplace 軟體相同的機制每月計費。

使用 AWS Marketplace Metering Service，您可以處理數個新的定價案例。

Example 依主機收費

如果您的軟體監控主機，您可以針對每個受監控的主機收取費用，並根據主機大小設定不同的定價。

Example 依使用者收費

如果您的軟體允許整個組織的多個使用者，您可以依使用者收費。每小時會向客戶收取佈建使用者總數的費用。

> **Note**
>
> 在產品載入表單 (PLF) 中，相關欄前面加上 "FCP"（彈性消費定價）。例如：FCP 類別（自訂定價類別）。

對於 AWS Marketplace Metering Service 產品，請注意下列事項：

- 如果您的軟體已開啟 AWS Marketplace，您將需要建立產品以啟用替代用量維度。您無法轉換標準產品以使用 AWS Marketplace Metering Service。發佈新產品後，您可以移除舊產品或將兩者保留在網站上。
- AWS Marketplace Metering Service 需要您的軟體每小時報告用量，並記錄該小時的客戶用量。如果傳輸或接收計量服務記錄失敗， AWS 將無法針對此類使用計費。您有責任確保成功接收計量記錄。
- 使用 AWS Marketplace Metering Service 的產品不支援 1-Click。買方必須使用具有特定許可的 AWS Identity and Access Management (IAM) 角色啟動您的軟體，並擁有網際網路閘道。
- 免費試用和年度定價與 AWS Marketplace Metering Service 不相容。
- 不支援變更維度（使用者、主機、頻寬和資料）或維度名稱。您需要建立新的產品。

合約定價

使用合約定價模式，您可以向客戶提供預付定價，讓他們能夠購買 1 個月、12 個月、24 個月或 36 個月的授權。

合約定價適用於下列產品：

- 單一 AMI 型產品和具有 AWS CloudFormation 範本型產品的 AMI。如需詳細資訊，請參閱上的 AMI 產品的合約定價 AWS Marketplace
- 容器型產品。如需詳細資訊，請參閱容器產品的合約定價。
- 軟體即服務 (SaaS) 型產品。如需詳細資訊，請參閱軟體 SaaS 合約定價。

> **Note**
>
> AMI 和容器型產品的合約定價僅適用於新產品。
> 如果您有現有的 AMI 或容器型產品，並想要使用合約定價，請建立新的清單，然後使用產品載入表單 (PLF) 來套用合約定價模型，以新增不同的維度、整合 AMI 或容器型產品與 AWS License Manager，然後發佈 AMI 或容器型產品。
> 當客戶購買具有合約定價的產品時， 會在客戶 AWS Marketplace 中建立授權 AWS 帳戶 ，您的軟體可以使用 License Manager API 檢查該授權。客戶需要 IAM 角色才能啟動 AMI 或容器型產品的執行個體。

使用您自己的授權定價

自帶授權 (BYOL) 產品不收取任何服務費用 AWS Marketplace。

為了實現客戶選擇的承諾，我們要求所有 BYOL 產品也都有付費選項。因此，沒有現有授權的客戶可以選擇購買和使用產品。

對於 BYOL 產品，我們了解線上購買軟體不同於某些公司執行業務的方式。因此，在啟動後的前 90 天，我們將放寬此軟體隨附可在 上購買的版本的需求 AWS Marketplace。在此期間， AWS Marketplace 帳戶管理團隊將與您一起解決挑戰。團隊可協助您判斷軟體是否可以及如何可供購買 AWS Marketplace。

變更定價模型

定價模型的變更必須由 審核和核准 AWS Marketplace ，以確保正面的客戶體驗並降低對各方的風險。聯絡AWS Marketplace 賣方營運團隊，討論您想要進行的定價模型變更。

定價模型變更的所有請求可能需要 30-90 天來處理和檢閱。

> **Note**
>
> 請注意，SaaS 產品不支援定價模型變更。如需 SaaS 產品定價的詳細資訊，請參閱 中的 SaaS 產品定價 AWS Marketplace。

變更價格

您可以透過 更新價格和中繼資料 AWS Marketplace 管理入口網站。

變更價格

1. 登入 AWS Marketplace 管理入口網站。
2. 在產品索引標籤中，您可以使用您建立的目前產品清單。在這裡編輯您的產品清單或請求變更。

> **Note**
>
> 對於新訂閱者，價格變更會立即生效。對於公有優惠的現有訂閱者，價格變更會在 90 天期間的下個月第一天生效。例如，如果您在 3 月 16 日傳送價格變更通知，價格變更會在 90 天期間的下個月第一天發生，也就是 7 月 1 日。

私有優惠

在 AWS Marketplace 賣方私人優惠計劃中， AWS Marketplace 賣方可以與個別 AWS Marketplace 客戶（買方）協商自訂定價和 EULAs。如需詳細資訊，請參閱為您的 AWS Marketplace 產品準備私有優惠。

中的產品退款 AWS Marketplace

> **Note**
>
> 如果您透過 Amazon Web Services EMEA 運算子交易，則只有經過驗證的使用者才能處理退款、修改了解您的客戶 (KYC) 資訊，以及變更銀行帳戶詳細資訊等財務資訊。已驗證的使用者是完成驗證的經 CSV 驗證的使用者和次要使用者。如需詳細資訊，請參閱完成了解您的客戶程序。

無論定價模式為何 AWS Marketplace， 中的所有付費產品都必須有軟體費用的聲明退款政策。退款政策必須包含退款條款，以及聯絡賣方要求退款的方法。

身為賣方，退款政策的詳細資訊由您決定。不過，我們建議您為買方提供某種形式的產品使用退款。您必須遵守張貼的退款政策。本主題提供有關 AWS Marketplace 產品退款請求類型、相關政策和核准程序，以及如何向買方提交退款請求的資訊。

主題

- AWS Marketplace 產品的退款請求類型
- AWS Marketplace 產品退款政策和核准
- 請求產品退款
- 請求大量退款

AWS Marketplace 產品的退款請求類型

買方可以請求不同類型的 AWS Marketplace 產品退款。對於 銷售 AWS Marketplace 的產品 AWS，請參閱退款政策頁面，然後使用 提交聯絡支援表單 AWS Support Center Console。如果買方直接向請求軟體退款 AWS，我們會指示他們使用您所張貼的有問題產品的支援聯絡資訊來聯絡賣方。任何 AWS 基礎設施費用的退款均由 自行決定， AWS 且與軟體退款無關。

對於第三方銷售的產品，買方必須在產品詳細資訊頁面上檢視退款政策。 AWS Marketplace 訂閱的軟體費用會支付給產品的賣方，且必須直接向賣方申請退款。每個 AWS Marketplace 賣方都必須在其 AWS Marketplace 頁面上包含退款政策。

AWS Marketplace 產品退款政策和核准

以下清單說明 AWS Marketplace 退款政策，以及是否需要您的核准：

- 免費試用

  如果您將軟體列為免費試用產品， AWS 可以在從免費試用轉換到付費訂閱後的七天內，代表您發出軟體費用的退款。與免費試用轉換相關的退款不需要您採取任何動作。啟用產品的免費試用即表示您同意此政策。

- 私有優惠

  所有私有優惠的退款都必須經過您的授權， AWS 才能處理。

- 軟體計量退款

如果您使用 AWS Marketplace Metering Service 來計量軟體的用量， AWS 可以代表您針對軟體計量錯誤所產生的軟體費用發出退款。如果這些錯誤在多個買方之間很常見， AWS 保留為每個買方決定適當退款的權利，並將其直接套用至每個買方。與 相關的退款 AWS Marketplace Metering Service 必須向賣方確認一次，但不需要賣方確認每個個別退款。透過 AWS Marketplace Metering Service 搭配產品使用 ，即表示您同意此政策。

- 購買後 48 小時內取消訂閱

  如果買方在非私有優惠購買後 48 小時內取消其訂閱， AWS 會發出完整退款 （取消並退款 100%)。與購買後 48 小時內取消相關的退款不需要採取任何動作。48 小時後，此類買方請求由您自行決定。在 上列出您的產品 AWS Marketplace，即表示您同意此政策。

- 訂閱升級

  如果買方以更昂貴的訂閱或同等價值的訂閱取代現有的非私有優惠訂閱， AWS 可以代表您為較低階訂閱發出退款。這是買方的兩個步驟：購買新訂閱，然後請求取消舊訂閱並退款。

- 訂閱降級

  所有降級訂閱退款請求都必須經過您的授權， AWS 才能處理。

所有 AWS 授權的退款都會自動處理，您不需要採取任何動作。

請求產品退款

使用下列程序請求外部買方或內部測試帳戶的軟體退款。

申請軟體退款

1. 收集下列資訊：

   - 用來訂閱產品的買方 AWS 帳戶 ID (12 位數）。如果買方是其組織的付款人，請取得用於訂閱的連結帳戶 AWS 帳戶 ID。
   - 與其相關聯的買方電子郵件地址 AWS 帳戶。
   - 賣方 AWS 帳戶 ID (12 位數）。
   - 產品 ID。您可以在協議和續約儀表板或帳單收入儀表板中找到產品 ID。
   - 優惠 ID。您可以在 的協議和續約儀表板或帳單收入儀表板、優惠索引標籤或協議索引標籤中找到優惠 IDAWS Marketplace 管理入口網站。
   - 帳單期間。

- 來自帳單收入儀表板的發票號碼。
- 稅前退款金額。

2. 登入 AWS Marketplace 管理入口網站。
3. 選擇支援。
4. 選擇請求退款。
5. 輸入訂閱者的 AWS 帳戶 號碼、產品 ID 和計費期間。
6. 在其他註解 （選用） 欄位中，輸入請求的簡短描述，包括產品 ID、退款類型和發票號碼 （如果有的話）。例如：

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

7. 若要提交年度訂閱的退款、升級或降級請求，請執行下列動作：

   a. 確認買方已使用協議和續約儀表板或帳單收入儀表板購買年度訂閱。

   b. 在其他註解 （選用） 欄位中，輸入訂閱取消日期，並指定買方是否請求退款、升級或降級。

8. 提交表格。我們將收到通知，並開始處理退款並核發給買方。
9. 若要管理退款請求， AWS Marketplace 買方支援團隊將在支援 中心主控台中建立對應的支援案例。退款的狀態會顯示在支援案例的主旨行中。狀態可以是下列其中一項：

   - 已完成 – 已處理退款，不需要進一步的動作。
   - 待定 – 退款會在目前帳單週期結束後處理。
   - 必要動作 – 無法處理請求，我們需要您提供其他資訊。您可以直接回應支援案例；不過，您也需要提交新的退款申請表。

   退款會在 24-48 小時內出現在買方帳戶中。不過，資金最多可能需要五個工作天才會出現在買方的金融帳戶中。

請求大量退款

下列步驟說明如何建立涉及超過 20 張發票或超過 20 個帳戶的退款請求，此程序可簡化大規模的退款請求，並確保您提供必要的資訊。

申請大量退款

1. 收集下列必要資訊：

- 賣方的 AWS 帳戶 ID (12 位數)
- 所有 AWS 帳戶 IDs的清單 （每個 12 位數 ）。您可以輸入多個買方，或輸入多個計費期間的單一買方。
- 所有產品 IDs
- 適用情況下IDs
- 所有發票 IDs
- 每個發票的計費期間
- AWS Marketplace 退款票證參考 ID

2. 使用下列資料欄將必要資訊編譯至 CSV 試算表：

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

3. 以根使用者身分登入 支援 中心主控台。

> **Note**
>
> 除非您以根使用者身分登入，否則無法完成這些步驟。

4. 建立帳戶和帳單的支援案例，然後選取 AWS Marketplace。
5. 在主旨行中，輸入 **`Request Bulk Refund`**。
6. 在其他註解下，輸入下列內容：

- 受影響的帳戶總數
- 發票總數
- 總退款金額
- 退款的業務理由
- AWS Marketplace 退款票證參考 ID

7. 連接 CSV 檔案並建立支援案例。

8. 支援案例會路由到下一個可用的客服人員以尋求協助。

您 AWS Marketplace 產品的區域和國家/地區

當您在 中建立產品時 AWS Marketplace，您可以選擇可用的 AWS 區域。您也可以選擇買方可以購買您產品的國家/地區。這兩個屬性類似，但不同。例如，買方可能位於美國並從中購買，但他們可能打算在歐洲（法蘭克福）區域安裝您的產品。為了讓此買方購買您的產品，您必須在您的國家/地區清單中同時包含美國，並在您的區域清單中包含歐洲（法蘭克福）區域。您可以使用此主題來進一步了解您 AWS Marketplace 產品的區域和國家/地區。

AWS 區域

建立或編輯伺服器或機器學習產品資訊時，您可以將產品限制在使用者可安裝和使用產品的特定 AWS 區域。

對於伺服器產品，包括 Amazon Machine Image (AMI)、容器和 AWS CloudFormation型產品，您可以選取產品可用的特定區域。您也可以選擇在新的美國區域、非美國區域或所有區域可用時自動提供您的產品。

對於機器學習產品，您可以選取特定區域，或在未來區域可用時選取所有區域。

如需 AWS 區域的詳細資訊，請參閱《 AWS 一般參考》中的AWS 服務端點。

國家/地區

根據預設，您的產品可供所有 AWS Marketplace 可用國家/地區的買方使用。對於新的和現有的伺服器和軟體即服務 (SaaS) 產品，您可以控制特定國家/地區的產品可用性，用於稅務、合規、支援或行銷目的。

此功能有一些例外狀況：

- 先前的購買 – 使用新的國家/地區清單更新您的產品後，已訂閱您產品的買方在其訂閱作用中仍可存取。
- 私有優惠 – 當您將產品限制在特定國家/地區的買方時，不會限制私有優惠。當您為特定買方建立私有優惠時，該買方仍可使用該優惠，即使他們位於您指定的國家/地區中未包含的國家/地區。

> **Note**
>
> 客戶資格是在 AWS 連結的帳戶層級決定。如需詳細資訊，請參閱如何 AWS 判斷您帳戶的位置？
>
> 共用權利的客戶只能在您允許的區域中啟用權利。如需管理權利的詳細資訊，請參閱《 AWS Marketplace 買方指南》中的在組織中共用訂閱。

翻譯和語言

根據預設， 會將您的產品清單和優惠 AWS Marketplace 轉換為下列語言：

- 法文 (fr-FR)
- 日文 (ja-JP)
- 韓文 (ko-KR)
- 西班牙文 (es-ES)

AWS Marketplace 也提供一組範例標準合約，僅供這些語言的參考。合約不具法律約束力。如需合約的詳細資訊，請參閱《 AWS Marketplace 買方指南》中的支援的語言。

下列主題說明翻譯的運作方式、如何下載翻譯的標準合約，以及如何選擇退出翻譯。

主題

- 開始使用
- 選擇退出翻譯

開始使用

請記住下列有關 AWS Marketplace 翻譯的事項：

- 為了協助將您的觸角擴展到全球市場， AWS Marketplace 會翻譯您提供的英文內容，無需您額外努力。如果您有翻譯方面的問題，請使用 AWS Marketplace 管理入口網站中的聯絡我們頁面。選擇產品清單和當地語言選擇退出類別，並提供有關問題的意見回饋。

  AWS Marketplace 可以進行有限的變更，例如不翻譯技術術語或產品名稱。
- 您可以選擇不翻譯您的清單。如果您這麼做，買方會以英文查看內容，並以他們偏好的語言查看任何相關 AWS Marketplace 內容。如需選擇退出的詳細資訊，請參閱選擇退出翻譯下列內容。

- 您必須以英文提交您的賣方註冊和產品清單資訊。
- 您可以使用支援的語言提交 EULAs。相關產品清單必須是地理目標，且語言必須是國家的主要語言。

  AWS Marketplace 提供一組標準範例合約，僅供參考，翻譯成上述語言。您可以在 AWS Marketplace 買方指南中，從支援的語言下載合約。
- 根據最佳實務，當您發佈新產品時，請檢閱產品處於有限狀態的翻譯。在公開產品之前，若有任何問題，請聯絡我們。
- 轉換程序會在您將產品移至有限狀態後開始。買方會看到英文文字，直到翻譯可用為止。

選擇退出翻譯

下列步驟說明如何選擇退出翻譯您的產品清單。您可以選擇不使用部分或全部語言。例如，您只能將清單翻譯為法文和日文。

選擇退出需要 3 到 5 天才會生效。

選擇退出

1. 前往 AWS Marketplace 管理入口網站，然後選擇聯絡我們。
2. 依列出的順序選取下列類別：
   - 商業市場
   - 產品清單
   - 當地語言選擇退出
3. 包含清單 ID 和您要選擇退出的語言。
4. 最佳實務是協助您的買方了解為什麼他們看到英文文字。例如：
   a. 選擇退出所有翻譯時，請將 新增至產品名稱的**English**結尾。
   b. 選擇退出一或多個翻譯時，請在產品描述結尾新增一些說明文字。例如：**We have declined automatic translation services for this product**。
5. 提交您的請求。

在 中使用標準化合約 AWS Marketplace

當您以賣方身分準備產品時，您需要判斷哪些最終使用者授權合約 (EULA) 將控管產品的使用。 AWS Marketplace 您可以使用下列項目：

- 您自己的 EULA。
- 標準 AWS Marketplace (SCMP) 合約範本，可協助簡化採購工作流程和加速交易。您也可以使用 SCMP 的聯邦增補合約，協助加速美國政府的採購。
- AWS Marketplace (RCMP) 的經銷商合約，這是 ISVs可以使用的標準化經銷商合約範本。

下列主題說明如何使用標準化 AWS Marketplace 合約範本。

主題

- 的標準合約 AWS Marketplace
- 的經銷商合約 AWS Marketplace

的標準合約 AWS Marketplace

AWS Marketplace 與買方和賣方社群合作開發了適用於 AWS Marketplace (SCMP) 的標準合約。SCMP 管理用量，並定義數位解決方案買方和賣方的義務。數位解決方案的範例包括伺服器軟體、軟體即服務 (SaaS)，以及人工智慧和機器學習 (AI/ML) 演算法）。

SCMP 會主動定義重要契約子句的共同基礎，例如使用、保證、賠償和管理法律。賣方可以提供 SCMP 條款做為自助式交易的 EULA，買方可以在其中搜尋、購買和快速部署解決方案。對於私有優惠，買方可以向賣方請求 SCMP 範本，並且可以修改條款，以滿足雙方同意的自訂交易要求。

您也可以將下列選用附錄與自助式或私有優惠的 SCMP 搭配使用：

- 增強型安全增補 – 支援資料安全需求提高的交易。
- HIPAA 商業夥伴增補合約 – 支援符合 1996 年健康保險流通與責任法案 (HIPAA) 合規要求的交易。
- 聯邦增補合約 – 支援涉及美國政府的軟體購買。

> **免責聲明**
>
> EULA 為您和買方之間。使用 SCMP 做為您的 EULA 由您自行決定。透過將 SCMP 套用至您的產品清單，您可以選擇加入 SCMP 計畫。在此計劃下， AWS 可能會定期更新 SCMP 範

本，並可能會更新含有目前版本之條款的產品清單。您可以隨時將 SCMP 範本取代為您自己的 EULA，以退出 SCMP 程式。

SCMP 入門

您可以將 EULA 更新為 SCMP，並將其提供給全新和現有單一 Amazon Machine Instance (AMI) 和軟體即服務 (SaaS) 產品的買方。您用來請求更新的程序取決於產品是否透過自助服務清單 (SSL) 列出。

如果您有任何問題，請傳送電子郵件訊息給 AWS Marketplace 的標準合約團隊aws-mp-standardcontract@amazon.com。

若要將 EULA 更新為透過 SSL 列出的 AMI 和 SaaS 產品的 SCMP

1. 檢閱標準合約條款 AWS Marketplace。
2. 登入 AWS Marketplace 管理入口網站。
3. 針對產品，選擇產品類型，然後選取您要更新合約的產品清單。
4. 選擇請求變更，然後選擇更新區域和定價。
5. 如果尚未選取，請選擇 EULA 的標準合約 AWS Marketplace。
6. 在備註和通知中，輸入 **`Please update this product to SCMP V2`**。
7. 檢閱變更，然後選擇提交以供檢閱。

將 EULA 更新為未透過 SSL 列出的 AMI 和 SaaS 產品的 SCMP

1. 開啟 上的聯絡我們頁面 AWS Marketplace 管理入口網站。
2. 登入您的 AWS Marketplace 賣方帳戶。
3. 如下所示完成表格：
   - 針對您的問題主題，選擇商業市場。
   - 針對 類別，選擇產品清單。
   - 針對子類別，選擇標準合約請求。
   - 在描述中，輸入**`Please update these products to SCMP V2`**並提供您要使用 SCMP 更新的產品標題和 IDs。

的經銷商合約 AWS Marketplace

AWS Marketplace (RCMP) 的經銷商合約是一種標準化經銷商合約範本，ISVs可在授權通路合作夥伴將 ISV 產品轉售給 AWS Marketplace 買方時使用。當 ISVs和管道合作夥伴建立經銷商關係和/或為其管道合作夥伴私有優惠 (CPPO) 使用經銷商條款時，合約有助於減少法律合約審查的備援，並加速上市時間。建立 CPPO 時，ISVs可以將合約上傳至機會（轉售授權），然後管道合作夥伴可以檢視和接受合約。 AWS Marketplace 買方無法檢視 RCMP。

> **免責聲明**
>
> RCMP 是 ISVs的選用合約。如果 ISVs決定透過通路合作夥伴轉售其產品，則在建立機會時，他們可以連接 RCMP 或自己的自訂合約條款 - 現有或預先交涉。

RCMP 入門

本節說明如何檢閱詞彙並使用 RCMP。

在建立 AWS Marketplace 機會時使用 RCMP

1. 檢閱經銷商合約 AWS Marketplace的條款。
2. 參考 RCMP 指南時，請遵循為頻道合作夥伴建立轉售機會中的步驟。

提供 AWS Marketplace 產品的中繼資料

身為賣方，當您將產品新增至 時 AWS Marketplace，您會指定產品中繼資料。產品中繼資料包括您用來描述客戶產品的名稱、描述、類別和關鍵字。 僅會為了品質保證和錯誤修正而 AWS Marketplace 修訂產品中繼資料。本主題提供您身為 AWS Marketplace 賣方，可用來有效命名、描述和分類產品的資訊和最佳實務。

> **Note**
>
> 定價、用量、類似產品或產品比較等所有產品詳細資訊頁面標籤和區段都是動態的，並由 管理 AWS Marketplace。賣方無法自訂產品詳細資訊頁面來新增或移除區段。

命名和描述您的產品

買方可以看到您提供的產品相關資訊。確保潛在買方有足夠的資訊來做出購買產品的明智決策。

建立產品名稱

建立產品名稱時，請記住下列準則：

- 使用標題大小寫 （ 大寫每個重要字詞的第一個字母 ）
- 確保買方可以僅透過名稱來識別產品
- 使用品牌或製造商的名稱
- 避免描述性資料或 Hyperbole

產品名稱範例：智慧型 Solution Load Balancer - Premium Edition。

支援的字元

中繼資料欄位支援的字元為：

- ASCII 字元代碼 0 到 126
  - 如需詳細資訊，請參閱標準 ASCII 資料表。
- 著作權 ©
- 已註冊的 ®
- 商標 –
- 貨幣符號 ¢ £ ¤ ¥

撰寫產品描述

產品描述會列出產品的功能、優點和用量。它也可以提供其他相關的特定產品資訊。描述長度最多可達 350 個字元。

當您撰寫產品描述時，請記住下列準則：

- 避免不必要的大寫
- 避免不必要的標點符號
- 請勿包含重新導向資訊

- 檢查拼字和文法
- 僅包含重要且有用的資訊
- 避免描述性資料和 Hyperbole

範例產品解決方案：智慧型解決方案會自動將傳入的應用程式流量分散到多個 Amazon EC2 執行個體。它可讓您在應用程式中達到更高的容錯能力，提供回應傳入應用程式流量所需的負載平衡容量。智慧解決方案會偵測集區中運作狀態不佳的執行個體，並自動將流量重新路由至運作狀態良好的執行個體，直到還原運作狀態不佳的執行個體為止。您可以在單一 AWS 可用區域中或跨多個可用區域啟用智慧型解決方案，以確保更一致的應用程式效能。

撰寫產品重點

產品資訊頁面最多可顯示三個產品反白項目符號點。使用這些項目符號點來簡短描述產品的主要銷售點。

產品重點範例：投影成本：使用智慧型解決方案時，您只需按使用量付費。系統會針對 Smart Solution 正在執行的每個小時或部分小時向您收費。

撰寫支援資訊

客戶必須能夠輕鬆取得問題的協助，例如使用 服務、疑難排解和請求退款（如適用）。您必須在履行登陸頁面上指定支援聯絡選項，例如電子郵件、電話號碼或支援 Web 表單連結。

選擇類別和關鍵字

當您列出產品時，最多可以選擇三個軟體類別和對應的產品子類別。這有助於買方在瀏覽或搜尋產品時探索您的產品 AWS Marketplace。僅選擇與您的產品相關的類別；在大多數情況下，僅適用一個類別。產品載入表單和產品索引標籤都包含完整的類別清單。

類別與關鍵字不同。可用的類別和子類別是預先定義的 AWS Marketplace，您可以在產品請求程序期間從清單中選擇哪些類別和子類別適用於您的產品。關鍵字未預先定義，但會在程序期間建立。您不需要將 類別新增為關鍵字。

建立搜尋關鍵字

在產品請求程序期間，您最多可以輸入三個關鍵字（單字或片語），以協助買方透過網站搜尋探索您的產品。關鍵字欄位最多可包含 250 個字元。

下列秘訣可協助您建立一組相關的搜尋關鍵字：

- 使用相關術語。
- 請勿使用其他賣方發佈的產品名稱，或使用其他賣方的名稱。
- 從買方的詞彙中選擇關鍵字，也就是買方在考慮您的產品類型時可能使用的字詞和片語。
- 根據產品中的特定功能建立關鍵字。
- 請勿使用產品標題做為關鍵字。產品標題已在搜尋中編製索引。

> **Note**
>
> 關鍵字與軟體類別不同。關鍵字是與您產品相關的更具體術語。

建立 的 AMI 和容器產品使用說明 AWS Marketplace

身為 AWS Marketplace 賣方，您會以不同的方式將產品交付給買方。使用 Amazon Machine Image (AMIs) 和容器映像交付 時，您必須負責為其撰寫使用說明。下列各節提供撰寫這些產品類型使用說明的要求和建議。

如需詳細資訊，請參閱中的 AMI 型產品 AWS Marketplace。

主題

- 需求
- 撰寫版本備註
- 撰寫使用說明
- 撰寫升級指示
- 撰寫 CloudFormation 交付指示

需求

為產品建立使用說明時，您必須包含下列資訊：

- 客戶儲存的所有敏感資訊的位置。
- 說明所有資料加密組態。如果您的應用程式使用任何加密技術，請提供使用者如何與您的應用程式互動以解密必要資料的詳細說明。
- 如果您的產品包含密碼編譯材料，您必須在 使用說明中包含輪換要求。如需使用登入資料和密碼編譯金鑰的清單the section called “以 AMI 為基礎的產品需求”基本需求，請參閱 。

- 如果產品中的任何資料存放區為專屬，請提供組態、備份和復原的step-by-step說明。
- 如何評估和監控應用程式運作狀態和適當功能的Step-by-step說明。例如：
  - 導覽至 Amazon EC2 主控台，並驗證您位於正確的區域。
  - 選擇執行個體，然後選取您啟動的執行個體。
  - 選取要顯示中繼資料頁面的伺服器，然後選擇頁面底部的狀態檢查索引標籤，以檢閱您的狀態檢查是否通過或失敗。
- 管理 AWS 服務配額的規範性指引。如需詳細資訊，請參閱 AWS 一般參考指南。
- 定價明細，包括超過標準配額的執行 AWS 資源成本。這可以包含在產品使用說明中，或連結到包含有關管理和請求增加服務配額的詳細資訊的文件。

撰寫版本備註

每次更新產品時，您必須提供版本備註中變更的說明。版本備註應包含特定資訊，以協助使用者決定是否安裝更新。使用清晰的標籤進行更新，例如「關鍵」進行安全性更新，或「重要」或「選用」進行其他類型的更新。

撰寫使用說明

提供使用說明，以協助確保買方可以成功設定和執行軟體。您提供的使用說明會在組態程序期間顯示。

若要撰寫有效的使用說明，請遵循下列準則：

- 假設使用者感興趣但沒有資訊。
- 提供使用者啟動和使用您的產品所需的所有資訊，包括任何組態設定和特殊步驟。

範例使用說明：

1. 使用 1-Click 啟動產品。
2. 使用 Web 瀏覽器在 存取應用程式`https://<EC2_Instance_Public_DNS>/index.html`。
3. 使用下列登入資料登入：
   - 使用者名稱： `user`
   - 密碼：執行個體 ID (`instance_id`)

撰寫升級指示

提供買方如何從舊版產品升級的詳細資訊。包含有關如何在建立另一個執行個體時保留資料和設定的資訊。如果沒有升級路徑，請編輯此欄位以特別提及。

升級指示範例：

1. 執行 ****，然後執行 ****。
2. 透過執行 ***，檢查專案使用的所有外掛程式是否與版本 *.* 相容。如果不相容，請執行 ***。
3. 透過執行 *** 來備份您的資料。

撰寫 CloudFormation 交付指示

使用 CloudFormation 交付時，您還必須包含下列項目：

- 範本建立的每個 AWS Identity and Access Management (IAM) 角色和 IAM AWS CloudFormation 政策的目的
- AWS CloudFormation 範本所建立之每個金鑰的用途和位置
- 部署中涉及多個元素的網路組態詳細資訊
- 有關如何啟動應用程式的詳細指南，以及如何設定它們在部署包含多個 AWS 資源時進行通訊
- 定價明細，包含執行超過標準限制之 AWS 資源的成本。提供管理 AWS 服務限制的規範性指引。
- 所有資料加密組態。例如：Amazon S3 伺服器端加密、Amazon Elastic Block Store (Amazon EBS) 加密、Linux Unified Key Setup (LUKS) 等）

最佳化您的 AWS Marketplace 產品以進行搜尋

身為 AWS Marketplace 賣方，您可以最佳化產品以進行搜尋，讓買方更容易找到它們。搜尋是買方旅程中的重要工具，可讓客戶找到符合其獨特需求的最佳產品。對於 AWS Marketplace 客戶，搜尋發生在兩個主要位置：搜尋引擎（例如 Google 或 Bing) 和其中的搜尋函數 AWS Marketplace。本主題提供如何最佳化兩個位置清單的資訊。

主題

- 搜尋引擎最佳化
- AWS Marketplace 搜尋

搜尋引擎最佳化

最佳化您的產品詳細資訊頁面以針對相關關鍵字進行較高排名，對於透過搜尋引擎將更獨特的訪客導向您的產品詳細資訊頁面至關重要。

有三個主要頁面元素，對於改善對任何網頁的有機搜尋具有基礎和影響，包括產品詳細資訊頁面：關鍵字、標題標籤和 H1 標題標籤。

關鍵字

關鍵字是搜尋引擎最佳化的中心元素，因為它們會將主題分割為聚焦查詢，以驅動搜尋引擎結果。識別您頁面最相關的關鍵字的程序涉及關鍵字研究。搜尋引擎最佳化工具可以提供寶貴的資訊，例如關鍵字搜尋磁碟區（每個月在 Google 上搜尋多少次關鍵字）、目前的排名、搜尋趨勢、關鍵字競爭力和相關關鍵字。從這項研究，您可以識別主要和次要關鍵字。

您的主要搜尋引擎最佳化關鍵字應該是代表頁面主要主題的唯一單字或片語。此主要關鍵字應自然編入您的產品標題、簡短描述和反白部分。次要關鍵字應該是高度相關的詞彙，可在其餘頁面內容中找到。

標題標籤

出現在搜尋引擎結果頁面的標題標籤，以及瀏覽器視窗或索引標籤中頁面的標題，會通知讀者和搜尋引擎網路爬蟲程式有關頁面的內容。對於 AWS Marketplace 產品詳細資訊頁面，產品標題做為標題標籤，因此使用搜尋引擎最佳化關鍵字最佳化您的產品標題，以提高排名潛力非常重要。為了提高在搜尋引擎結果頁面中達到高排名的可能性，請在標題標籤中整合您的品牌名稱、產品名稱和相關關鍵字。

H1 標題標籤

H1 標題標籤有三個角色：

- 它們可協助訪客掃描頁面內容，以取得他們所需的資訊。
- 它們為視覺障礙訪客增加可存取性，這些訪客使用螢幕閱讀器來了解頁面的內容。
- 它們提供頁面標題中找到的關鍵字，如果後續頁面內容支援，則會收到額外的搜尋引擎最佳化相關性權重。

AWS Marketplace 搜尋

AWS Marketplace 該網站使用與整個產業使用的搜尋最佳化技術類似的搜尋查詢結果進行排名。透過了解 AWS Marketplace 排名和傳回搜尋結果的方式，您可以為 AWS Marketplace 搜尋引擎建立最佳化的產品詳細資訊。我們建議您在建立產品詳細資訊頁面時，將本指南納入考量。

關鍵字

在產品建立過程中，您最多可以提交三個關鍵字（單字或片語），以協助客戶透過網站搜尋探索您的產品。關鍵字文字方塊最多可包含 250 個字元。

使用以下提示來建立搜尋關鍵字：

- 使用相關的詞彙，讓客戶可以輕鬆找到您的產品。
- 從客戶的詞彙中選擇關鍵字，也就是他們在考慮您的產品類型時可能使用的字詞和片語。
- 根據產品中的特定功能建立關鍵字。
- 請勿在您提交的詞彙中包含產品標題。產品標題已在搜尋中編製索引。

> **Note**
>
> 關鍵字與軟體類別不同。關鍵字是與您產品相關的更具體詞彙。

您可以在建立產品後編輯關鍵字，方法是編輯產品的中繼資料。對於您在 中使用產品索引標籤建立的產品 AWS Marketplace 管理入口網站，您也可以使用產品索引標籤進行變更。如需詳細資訊，請參閱產品變更和更新。

AWS Marketplace 賣方操作團隊可協助重新導向具有類似聲音單字或含義相似單字的查詢。例如，當客戶在您預期他們搜尋汽車時搜尋汽車。

軟體類別

當您列出產品時，最多可以選擇三個軟體類別和對應的產品子類別。這有助於客戶在瀏覽或搜尋產品時探索您的產品 AWS Marketplace。僅選擇與您的產品相關的類別。在大多數情況下，只會套用一個類別。產品載入表單和產品頁面都包含完整的類別清單。

> **Note**
>
> 類別與關鍵字不同。可用的類別和子類別已預先定義 AWS Marketplace。您可以從清單中選擇哪些適用於您的產品。關鍵字未預先定義，但會在程序期間建立。

反白顯示區段

產品詳細資訊頁面最多顯示三個產品亮點做為項目符號點。客戶可以依重點搜尋產品，因此請在建立產品時包含重點。重點應以簡短且資訊豐富的語言描述產品的主要銷售點。

Example 重點標示

- 投影成本：使用 AnyCompany 的產品時，您只需支付使用量的費用。系統會針對其執行的每個小時或部分小時向您收費。

簡短描述

產品描述會列出產品的功能、優點和使用說明，以及其他相關和特定的產品資訊。建立產品描述時，請記住下列指導方針：

- 避免不必要的大寫和標點符號
- 請勿包含重新導向資訊
- 檢查拼寫和文法
- 僅包含重要且有用的資訊

Example 簡短描述

AnyCompany 的產品會自動將傳入的應用程式流量分散到多個 Amazon EC2 執行個體。它可讓您順暢地提供回應傳入應用程式流量所需的負載平衡容量，以改善應用程式的容錯能力。AnyCompany 的產品會偵測集區中運作狀態不佳的執行個體，並自動將流量重新路由至運作狀態良好的執行個體，直到還原運作狀態不佳的執行個體為止。客戶可以在單一 AWS 可用區域中或跨多個可用區域啟用它，以啟用更一致的應用程式效能。

使用促銷媒體增強您的 AWS Marketplace 產品

身為 AWS Marketplace 賣方，您可以使用促銷媒體增強您的產品清單，協助客戶更好地探索和評估您的產品。促銷媒體是影片和影像，會顯示在您的產品頁面上，讓客戶能夠輕鬆了解您的產品。下列各節提供促銷媒體的最佳實務，以及在 () 中新增和管理促銷媒體的 AWS Marketplace Management Portal 秘訣AMMP。

主題

- 促銷媒體的最佳實務
- 新增和管理促銷媒體的提示

促銷媒體的最佳實務

最多支援 5 個影片和 10 個影像。每個促銷媒體項目都應包含必要的標題和選用的描述。描述用作媒體的替代文字，強烈建議您改善視覺化可存取性和搜尋引擎最佳化。

> **Note**
>
> 您必須擁有適當的權利和許可，才能上傳或新增任何促銷媒體。新增到您產品的媒體將公開提供給瀏覽的所有使用者 AWS Marketplace。

影片

此資產類型可讓您透過簡潔的概觀介紹您的產品和公司。您也可以包含內容，例如客戶面試、引號、相關利益和資料點。此外，影片是透過錄製的示範或逐步解說展示產品的好方法，尤其是關鍵功能或使用案例。我們建議影片的持續時間為 2-5 分鐘，因為較短的影片更具影響力。

規格：

- 我們支援透過AMMP或透過可公開存取S3的連結直接上傳影片。
- 建議的影片解析度為 1080p (1920x1080 像素），具有偏好的 16：9 長寬比。
- 影片格式必須為 .mp4，檔案大小上限為 50MB。
- （選用）可以新增封面影像來取代自動產生的封面。
  - 封面影像的解析度應為 500x281 像素 (16：9 長寬比）。
  - 影像格式必須是 .png（偏好）、.jpg 或 .svg，而且沒有透明度。

> **Note**
>
> 外部託管影片不受直接支援。在新增期間， AWS Marketplace 會快取公有S3連結，任何新增後變更在再次新增媒體之前都不會反映。

支援有限功能的 YouTube 影片：

- 如果產品影片 URL 欄位（與促銷媒體分開欄位）包含 YouTube 影片的直接連結，如果為影片啟用內嵌，則影片會直接內嵌為最後一個促銷媒體項目。
- 僅支援一個產品影片 URL（即僅支援一個 YouTube 影片），不支援重新排序 YouTube 內嵌媒體。

映像

此資產類型讓您有機會透過螢幕擷取畫面和關鍵概念標註來提升您的產品。影像非常適合透過簡潔圖表或流程圖來說明複雜的產品功能。此外，影像可用來說明無法透過定價維度描述定址的定價方案和分組。

規格：

- 我們支援透過 AMMP或透過可公開存取S3的連結直接上傳映像。
- 建議的影像解析度為 780x439 像素，具有偏好的 16：9 長寬比。最多 3480x3480 像素。
- 影像格式必須是 .png （偏好）、.jpg 或 .svg，而且沒有透明度。
- 縮圖會從原始來源影像自動產生。

新增和管理促銷媒體的提示

新增媒體

- 您可以登入 AWS Marketplace Management Portal並建立新產品或更新現有產品來新增促銷媒體。影像和影片區段用於新增促銷媒體，位於產品資訊區域中。
- 雖然 APIl也可以用來新增促銷媒體，但僅支援透過 AMMP 直接上傳。API 和 都支援公有S3連結 AMMP。
- 依您希望媒體顯示的順序新增促銷媒體。第一個項目將是產品頁面上的精選媒體。
- 使用影像資產類型時，無法使用選用的封面影像。新增後，影像會自動產生縮圖。

更新或移除媒體

- 將促銷媒體新增至產品後，即可視需要編輯產品來調整標題、描述和涵蓋範圍 （如果有的話）。
- 在提交之前，按一下 AMMP 中該資產的刪除，即可移除促銷媒體。

訂購媒體

- 根據預設，新的促銷媒體資產會新增至資產訂單的結尾。
- 若要重新排序媒體，請刪除不按順序排列的資產，並以正確的順序再次新增它們以供提交。

為您的 AWS Marketplace 產品準備私有優惠

私有優惠是用於購買產品的協商條款 AWS Marketplace。這可能涉及自訂定價計劃、最終使用者授權合約 (EULA) 或自訂解決方案。賣方和買方在遞交與公開優惠不同的私有優惠之前進行協商。您可以建立多個私有優惠並將其擴展至單一買方。您延長私有優惠的買方可以選擇私有優惠和公有優惠。買方在任何指定時間只能訂閱一個優惠。他們無法同時訂閱私有優惠和公開優惠。本主題提供私有優惠的運作方式相關資訊，包括特殊考量、買方體驗和賣方報告。

> **Note**
>
> AWS 在 上為買方提供唯一或企業使用案例 AWS Marketplace ，以直接從產品詳細資訊頁面為您的產品請求私有優惠。如果您是符合 APN Customer Engagements (ACE) 資格的 AWS 合作夥伴網路 (APN) 合作夥伴，並且想要提供此選項給買方，請參閱 新增私有優惠和示範請求按鈕 以取得更多資訊。

主題

- 私有優惠的運作方式
- 私有優惠考量事項
- 買方的私有優惠體驗
- 報告私有優惠
- AWS Marketplace 私有優惠支援的產品類型
- 建立和管理私有優惠
- 以 AWS Marketplace 頻道合作夥伴身分建立私有優惠
- 私有優惠分期方案
- 為私有優惠建立未來日期的協議
- 私有優惠常見問答集

私有優惠的運作方式

您可以從 中的優惠頁面建立和管理私有優惠AWS Marketplace 管理入口網站。您可以為優惠指定產品，以產生唯一的 ID 和 URL。您將為私有優惠建立定價計劃、新增法律條款和銷售文件，並將優惠擴展至特定買方 AWS 帳戶。只有您為其建立優惠的帳戶才能看見優惠。

在您建立私有優惠並通知潛在買方之後，他們可以檢視並接受優惠。若要檢視優惠，買方必須登入 AWS 帳戶 收到優惠的 。

> **Note**
>
> 買方無法檢視優惠，除非您將其擴展到其連結帳戶或其管理帳戶。您無法在優惠中提供服務限制，因此除非產品有限制，否則買方可以按照他們想要的協議價格使用您的產品。

如需建立私有優惠的詳細資訊，請參閱建立和管理私有優惠。

私有優惠會在賣方報告中追蹤。如需詳細資訊，請參閱報告私有優惠和賣方報告指南。

私有優惠考量事項

使用私有優惠時，請考慮下列事項：

- 當您新增對新執行個體類型的支援，或者 AWS 區域，已為您的產品訂閱私有優惠的客戶將無法自動存取新新增的執行個體或區域。您必須使用您希望客戶存取的執行個體和區域建立另一個私有優惠。接受新優惠後，客戶即可存取新增的執行個體和區域。未來訂閱您產品的客戶也可以存取他們，只要他們包含在私有優惠中。如需如何建立新的私有優惠的詳細資訊，請參閱 在 中修訂協議 AWS Marketplace。
- 您無法為第二方建立私有優惠、每月 Amazon Machine Image (AMI) 或使用 AWS CloudFormation 產品的multi-AMI-based，或限制客戶用量。
- 對於具有分期付款計劃的私有優惠，隨著時間的推移，可以將預付款分成多次付款。如需詳細資訊，請參閱私有優惠分期方案。
- 如果私有優惠的買方帳戶是透過私有市集進行管理，您必須在優惠中同時包含買方帳戶和包含其私有市集管理員的帳戶。
- 私有優惠不支援自帶授權 (BYOL) 模型。
- 當您在私有優惠中建立具有唯一協商合約條款的私有優惠時，請使用自訂 EULA 選項。您最多可以連接五個文件。
- 對於軟體即服務 (SaaS) 合約和使用產品的 SaaS 合約，您可以針對買方接受私有優惠時所簽訂的協議提供升級和續約。例如，您可以這樣做來授予新權利、提供定價折扣、調整付款排程，或變更最終使用者授權合約 (EULA) 以使用標準化授權條款。如需詳細資訊，請參閱在 中修訂協議 AWS Marketplace。

買方的私有優惠體驗

當買方導覽至您產品的訂閱頁面時，橫幅會指出有可用的私有優惠。買方接受優惠後，會使用與所有 AWS Marketplace 交易相同的入口網站工具，為購買開立發票。接受的優惠會成為協議。買方可以在 的管理訂閱區段中找到協議詳細資訊 AWS Management Console，賣方可以在 的協議索引標籤中找到詳細資訊 AWS Marketplace 管理入口網站。

AWS Marketplace 買方可以針對私有優惠存取第三方貸款。如需詳細資訊，請參閱 中的客戶貸款。AWS Marketplace

> **Note**
>
> 優惠只能在過期日期之前接受。如果優惠過期，則會移至已接受和過期的優惠索引標籤。

檢視並接受私有優惠	買方可以
從 AWS Marketplace 主控台	在 AWS Marketplace 主控台中導覽至私有優惠，然後從可用優惠索引標籤中選取優惠 ID。 如需私有優惠買方體驗的詳細資訊，請參閱《 AWS Marketplace 買方指南》中的私有優惠。
使用賣方提供的連結	依照賣方傳送的連結，直接存取私有優惠。 如需詳細資訊，請參閱傳送私有優惠給買方。
從您的產品頁面	導覽至產品的產品頁面，然後選擇橫幅中的連結以檢視私有優惠。 如需私有優惠買方體驗的詳細資訊，請參閱AWS Marketplace《買方指南》中的私有優惠。

報告私有優惠

私有優惠會出現在現有的賣方報告以及與優惠相關的報告中。是每月每月計費營收報告產生的，並提供可見性和優惠 ID 資訊。為買方產生發票時，發票會出現在涵蓋適當計費期間的報告中。如需詳細資訊，請參閱賣方儀表板。

優惠 ID 欄位包含為私有優惠產生的唯一優惠 ID。除非報告項目是用於私有優惠，否則為空白。優惠可見性欄位指出報告項目是公有或私有優惠。對於所有私有優惠，項目會標記為私有。

AWS Marketplace 私有優惠支援的產品類型

身為 AWS Marketplace 賣方，您可以使用私有優惠來銷售下列產品類型：Amazon Machine Image (AMIs)、容器、專業服務、機器學習 (ML) 和軟體即服務 (SaaS) 產品。私有優惠是用於從中購買產品的協商條款 AWS Marketplace。本主題提供有關 AMI、容器、SaaS、ML 和專業服務產品的私有優惠資訊。

如需私有優惠的詳細資訊，請參閱 為您的 AWS Marketplace 產品準備私有優惠。

主題

- AMI 產品的私有優惠
- 容器產品的私有優惠
- 專業服務產品的私有優惠
- SaaS 產品的私有優惠
- ML 產品的私有優惠

AMI 產品的私有優惠

您可以為 AMI 產品提供私有優惠定價。

優惠可以是下列項目的任何自訂持續時間：

- AMI 每小時或 AMI 每小時，提供年度私有優惠：最長 3 年 (1，095 天）。只有具有年度私有優惠或 AMI 合約的 AMI 每小時支援彈性付款排程。
- AMI 合約私有優惠：最長 5 年 (60 個月）

  對於 AMI 合約，私有優惠不會監控用量。

  買方可以隨時手動升級至新的合約層級。不過，獨立軟體廠商 (ISV) 可以定義合約方案、強制執行服務限制，並建議買方手動升級更多單位的合約。目前只有非分層定價型合約支援升級。私有優惠的合約持續時間可以符合公開產品清單，也可以是以月為單位的自訂持續時間（最多 60 個）。

授權權利從買方接受私有優惠的日期開始。

對於具有彈性付款排程的 AMI 私有優惠，您可以在合約期間設定合約中同意的年度執行個體類型數目。

> **Note**
>
> 私有優惠不適用於每月帳單合約。

容器產品的私有優惠

您可以為容器型產品合約提供私有優惠定價。

優惠可以是下列項目的任何自訂持續時間：

- 每小時容器或每小時容器，提供長期的私有優惠 – 最長 3 年 (1，095 天）。只有具有長期私有優惠或容器合約的每小時容器支援靈活的付款排程。
- 容器合約私有優惠 – 最長 5 年 (60 個月）

  對於容器合約，私有優惠不會監控用量。只有在您使用非分層定價時，才能升級容器合約。

  買方可以隨時手動升級至新的合約層級。不過，獨立軟體廠商 (ISV) 會定義合約層、強制執行服務限制，並建議買方手動升級更多單位的合約。目前只有非分層定價型合約支援升級。私有優惠的合約持續時間可以符合公開產品清單，也可以是以月為單位的自訂持續時間（最多 60 個月）。

授權權利從買方接受私有優惠的日期開始。對於具有彈性付款排程的容器私有優惠，您可以在合約期間設定合約中同意的單位數量。如果買方使用更多，您也可以為這些相同單位定義自訂的每小時價格。

> **Note**
>
> 私有優惠不適用於每月帳單合約。

專業服務產品的私有優惠

所有專業服務產品產品均透過私有優惠完成。如需詳細資訊，請參閱建立私有優惠。

SaaS 產品的私有優惠

軟體即服務 (SaaS) 私有優惠產品無法根據時間變更指定定價方案的定價等級。例如，優惠三個月內無法收取每小時 0.80 USD，之後會將相同定價方案的定價變更為每小時 0.60 USD。對於 SaaS 合約，私有優惠不會監控用量。

買方可以隨時手動升級至新的合約層級。不過，獨立軟體廠商 (ISV) 會定義合約方案、強制執行服務限制，並建議買方視需要手動升級至更高的合約方案。私有優惠的合約持續時間可以符合公開產品清單，也可以是以月為單位的自訂持續時間 （ 最多 60 個月 ）。

ML 產品的私有優惠

Machine Learning (ML) 私有優惠產品為特定買方提供不同於公開顯示價格的價格。您與私有優惠買方之間的一組條款和協議，可能與公有優惠或其他私有優惠中的條款和協議不同。

Private 以下列幾種方式之一提供工作：

- 每小時 – 私有優惠可以是每小時費率，與公開顯示的每小時費率不同。此每小時費率是永久的，因為機器學習產品的私有優惠不會過期。如果未來需要價格變更，買方必須切換到新的私有優惠。產品的現有執行中執行個體或端點會自動依新接受優惠中設定的每小時費率計費。確保在私有優惠中的任何合約元件到期後，將其設定為產品的每小時費率。將此每小時費率設定為 0 美元，可讓買方無限期地使用產品，無需支付軟體費用。
- 每個推論：如果您已設定產品部署為端點時的推論定價，則私有優惠的推論速率可能會與公開顯示的推論速率不同。
- 合約 – 私有優惠可以是指定天數的固定預付費用合約。買方在整個合約期間可以使用不限數量的執行個體。在合約結束時，任何繼續執行的執行個體都會以您在私有優惠中設定的每小時費率計費。例如，您可以建立 365 天無限制使用之固定預付費用的合約。您也可以設定私有優惠的每小時費率。當買方接受此私有優惠時，他們會支付該預付費用。當合約結束時，任何仍在執行的執行個體都會以該小時費率計費。如果您要提供免費私有試用，請確定您在免費試用期結束後設定正確的每小時費率，以避免免費永久授權。

您可以建立多個私有優惠並將其擴展至單一買方。您擴展私有優惠的買方可以選擇在私有優惠和公有優惠之間進行選擇。買方在任何指定時間只能訂閱一項優惠。他們無法同時訂閱私有優惠和公開優惠。

若要為 SageMaker 產品的特定買方建立私有優惠，請參閱 the section called “建立私有優惠”。

建立和管理私有優惠

身為 AWS Marketplace 賣方，您可以建立和管理私有優惠。私有優惠是用來從中購買產品的協商條款 AWS Marketplace。這可能涉及自訂定價計劃、最終使用者授權合約 (EULA) 或自訂解決方案。下列各節說明如何建立和管理私有優惠。

> **Note**
>
> 若要符合發行 Private Offer 的資格，您必須至少有一個作用中的公有清單。如果您有公開清單，而且無法存取 Private Offers 索引標籤，請參閱 IAM 許可或聯絡 AWS Marketplace 支援。

主題

- 啟動新的私有優惠
- 了解優惠狀態
- 起草和發佈私有優惠
- 新增私有優惠和示範請求按鈕
- 傳送私有優惠給買方
- 複製您的私有優惠
- 下載優惠詳細資訊
- 儲存您的私有優惠進度
- 更新私有優惠的過期時間
- 取消私有優惠

啟動新的私有優惠

使用下列程序建立優惠，並使用 CreateO ffer API 變更請求產生優惠 ID。它會以草稿狀態建立空白優惠。

開始建立新的私有優惠

1. 登入 AWS Marketplace 管理入口網站，然後選擇優惠。
2. 在優惠頁面上，選擇建立優惠。

3. 在建立優惠頁面上，選擇您要從中建立私有優惠的產品類型和產品。處理最多需要 30 秒。在處理完成之前，請勿關閉或重新整理頁面。

> **Note**
>
> 建立優惠後，您將無法變更產品類型和產品。如需每個產品類型之私有優惠的詳細資訊，請參閱支援的產品類型。
> 如果您是 AWS Marketplace 通路合作夥伴，請選擇您要為自己的產品建立優惠，還是從轉售授權建立 AWS Marketplace 通路合作夥伴私有優惠 (CPPO)。如果是 CPPO，請選擇獨立軟體廠商 (ISV)、產品和授權。

4. 選擇繼續以提供詳細資訊。將開啟step-by-step體驗，以便您可以繼續建立您的私有優惠。

了解優惠狀態

優惠有三種狀態之一，視生命週期而定：

- 草稿 – 優惠不完整，您仍在準備中。處於草稿狀態的私有優惠不受保留排程約束。必須完成並提交所有必要的詳細資訊，才能發佈優惠並將其延伸至您的買方。
- 作用中 – 優惠已發佈並延伸給買方。優惠尚未過期，因此買方可以訂閱優惠。
- 已過期 – 優惠已發佈並延伸至買方。優惠已過期，因此買方無法訂閱優惠。您可以更新過期日期，讓您的買方有更多時間接受優惠。若要更新優惠過期，請參閱更新私有優惠過期。

> **Note**
>
> 接受優惠後，它會在協議索引標籤中顯示為協議。優惠的狀態不會變更。

起草和發佈私有優惠

使用下列程序來草擬和發佈您的私有優惠。

草擬和發佈您的私有優惠

1. 在提供優惠資訊頁面上，提供優惠名稱、優惠詳細資訊、續約類型和優惠過期日期。如果這是續約優惠，您必須在 上為續約選擇現有客戶 AWS Marketplace，以續約在 中建立的現有協議 AWS Marketplace，或為遷移現有客戶至 的續約選擇現有 AWS Marketplace客戶 AWS Marketplace。

> **Note**
>
> 優惠過期日期是優惠變成 null 和 void 的日期。在此日期 23：59：59 UTC 之後，買方將無法檢視和接受此私有優惠。

2. 選擇下一步。
3. 在設定優惠定價和持續時間頁面上，選擇定價模式、合約或用量持續時間、定價、貨幣和付款排程。如需具有分期付款計劃的定價模型，請參閱分期付款計劃。

> **Note**
>
> 非 USD 私有優惠僅限於合約定價產品。此外，請確定您已設定非 USD 支出偏好設定。如需詳細資訊，請參閱the section called “支出偏好設定”。
> 所有具有消費定價的公開優惠和私有優惠只能以 USD 建立。

4. 在新增買方頁面上，為您要擴展私有優惠的每個 AWS Marketplace 買方提供 AWS 帳戶 ID。每個選取的買方都必須在支援所選優惠貨幣 AWS 區域 的 AWS 帳戶 中擁有 。若要新增另一個 AWS 帳戶 ID，請選擇新增另一個買方。每個私有優惠最多可新增 24 位買方。
5. 選擇下一步。
6. 在設定法律條款和優惠文件頁面上，選擇下列其中一個選項：
   - 公開優惠最終使用者授權合約 (EULA) – 使用公開優惠中的 EULA。
   - AWS Marketplace (SCMP) 的標準合約 – 使用 提供的標準合約 AWS Marketplace。
   - 自訂法律條款 – 上傳最多五個與私有優惠相關的檔案，包括法律條款、工作聲明、物料清單、定價表或其他附錄。建立優惠時，這些檔案會合併為一份文件。
7. 在檢閱和建立頁面上，檢閱私有優惠的詳細資訊。檢閱並確認後，請選擇建立優惠以發佈優惠，並將其延伸到您選擇的買方。優惠發佈包含對 AWS Marketplace 目錄 API 的請求，因此最多可能需要一小時的時間來驗證和處理優惠。您可以在請求頁面上檢視此請求。

> **Note**
>
> 只有在請求成功時，才會發佈和延長優惠。如果請求失敗，將不會擴展到客戶。失敗表示發生系統錯誤或您必須在重新提交之前更正錯誤。

新增私有優惠和示範請求按鈕

賣方可以將call-to-action按鈕新增至其產品詳細資訊頁面。這些按鈕可讓買方請求私有優惠和引導式產品示範。您可以將一個或兩個按鈕新增至您的產品詳細資訊頁面。

您可以搭配下列產品類型使用按鈕：

- Amazon Machine Image
- 軟體即服務 (SaaS)
- 容器
- CloudFormation 範本

若要使用按鈕，您必須屬於 APN Customer Engagements Program (ACE)。當買方請求優惠或示範時，他們會將他們的聯絡資料和請求詳細資訊輸入表單。然後， AWS 需求產生團隊會符合資格請求，並透過 Partner Central 中的 ACE 將這些合格請求轉移給您，做為 AWS 原始機會。然後，您追蹤客戶以討論優惠詳細資訊或安排引導式示範。如需 ACE 的詳細資訊，請參閱 APN Customer Engagements Program 網站，以及 APN Customer Engagement (ACE) FAQs中的潛在客戶和機會。

下列主題中的步驟說明如何將按鈕新增至您的產品詳細資訊頁面。

主題

- 按鈕先決條件
- 啟用按鈕

按鈕先決條件

您必須先具備下列先決條件，才能將call-to-action按鈕新增至您的產品詳細資訊頁面：

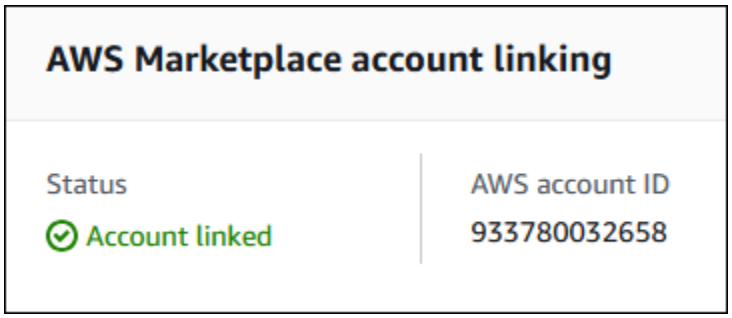
- 請確定您可以在 AWS Partner Central 中收到 AWS 推薦的潛在客戶和機會。如需詳細資訊，請參閱 APN Customer Engagements Program 網站。

  > **Note**
  >
  > 在您註冊 ACE 計劃之後，狀態更新會每兩週更新一次。私有優惠和示範請求按鈕只會在狀態更新完成後顯示。如果您選擇這些選項並收到訊息，您的存取權將等待下一次兩週更新。

- 連結您的 AWS Partner Central 和 AWS Marketplace 帳戶。若要這樣做，您必須：

- 建立 `CreatePartnerCentralCloudAdminRole` IAM：政策。如需詳細資訊，請參閱《 AWS Partner Central 入門指南》中的帳戶連結的先決條件。
- 連結您的 AWS Partner Central 和 AWS Marketplace 帳戶。如需詳細資訊，請參閱《 AWS Partner Central 入門指南》中的將 Partner Central 帳戶連結至 AWS Marketplace 您的帳戶。 AWS

  連結 AWS Partner Central 和 AWS Marketplace 帳戶後，您的 Partner Central 首頁會顯示下列狀態訊息：



如需詳細資訊，請登入 Partner Central 並參閱以下內容：

- AWS Partner Central & Marketplace 帳戶連結指南
- AWS Partner 和 Marketplace 帳戶連結示範影片

> **Note**
>
> 您必須登入才能使用這些資源。

啟用按鈕

一旦您符合 ACE 接收 AWS 推薦的資格，您可以使用 AWS Marketplace 管理入口網站來啟用一個或兩個call-to-action按鈕。

視您是否建立新的產品清單或更新目前的清單而定，您可以遵循個別的程序來啟用按鈕。

啟用新產品的按鈕

1. 使用 AWS Marketplace 管理入口網站建立下列類型的產品，並將其設為公開：
   - AMI

- SaaS
- 容器
- 雲端前端範本

2. 在建立產品的過程中，在引導式示範和私有優惠請求下，選擇為買方啟用引導式示範請求和為買方啟用私有優惠請求的任意組合。

> **Note**
>
> 這些按鈕只會在您公開產品後，出現在您私有優惠的產品詳細資訊頁面上。

啟用現有產品的按鈕

1. 在 AWS Marketplace 管理入口網站的產品索引標籤中，選取您要變更的產品。
2. 開啟請求變更清單，然後選擇更新產品資訊。
3. 選擇為買方啟用引導式示範請求和為買方啟用私有優惠請求的任意組合。

這些按鈕只會在您儲存變更後顯示在產品詳細資訊頁面上。

傳送私有優惠給買方

發佈私有優惠之後，買方可以透過導覽至 中私有優惠頁面上的可用私有優惠索引標籤來檢視該優惠 AWS Marketplace 管理入口網站。在可用私有優惠索引標籤上，買方可以在記錄賣方欄中查看 AWS Marketplace 由頻道合作夥伴延伸的優惠。 獨立軟體廠商 (ISV) 會顯示在發佈者欄中。買方可以在其優惠清單中選擇適當的優惠 ID，以導覽至私有優惠。

買方可以在已接受或過期的優惠索引標籤上檢視已接受或已過期的優惠 IDs。

發佈私有優惠之後，您可以將 URL 傳送給買方，前往優惠的履行頁面。

傳送私有優惠給您的買方

1. 登入 AWS Marketplace 管理入口網站，然後選擇優惠。
2. 選取優惠旁的選項按鈕。
3. 選擇動作，然後複製優惠 URL。
4. 將 URL 傳送給您的買方。

複製您的私有優惠

您可以複製私有優惠，包括 AWS Marketplace 頻道合作夥伴私有優惠。使用複製來使用範本建立新的優惠，或更新和取代現有的優惠。

複製私有優惠

1. 登入 AWS Marketplace 管理入口網站，然後選擇優惠。
2. 在優惠表格中，選取您要複製優惠旁的選項。
3. 選擇複製優惠。
4. 新的優惠建立體驗將會開啟，其中包含所選優惠中預先填入的資訊。視需要檢閱和修改優惠詳細資訊。
5. （選用）如果您要複製以取代現有的優惠，請選取取消現有的優惠。選取時，原始優惠會自動過期，且發佈此新優惠時，買方無法存取。這只會影響優惠的可存取性，如果買方已接受原始優惠，則不會影響任何現有的訂閱。
6. 選擇複製私有優惠。這將發佈優惠，並將其擴展到您先前選取的買方。

下載優惠詳細資訊

使用下列程序在 .pdf 檔案中下載優惠詳細資訊。

下載優惠詳細資訊

1. 登入 AWS Marketplace 管理入口網站，然後選擇優惠。
2. 在優惠表格中，選取優惠旁的選項，然後選擇檢視詳細資訊。或者，您可以在優惠 ID 欄中選擇優惠的連結。
3. 在優惠詳細資訊頁面上，選擇下載 PDF。

儲存您的私有優惠進度

使用下列程序來儲存您的進度並稍後繼續。

儲存和繼續您的工作

1. 在任何完成的步驟中，選擇儲存並結束。在對話方塊中，確認您以草稿狀態儲存內容，並檢閱任何驗證錯誤。如果有任何驗證錯誤或缺少詳細資訊，您可以選擇修正以前往步驟並解決問題。當您準備好時，請選擇儲存並結束以儲存您的變更。

在您儲存並結束之後，請求正在處理中。完成處理可能需要幾分鐘或幾小時。在請求成功之前，您無法繼續這些步驟或修改請求。請求成功後，您已完成儲存。如果請求失敗，則表示發生系統錯誤或您必須在重新提交之前更正錯誤。

2. 若要繼續處理優惠，請開啟優惠頁面，選擇您的優惠，然後選擇繼續優惠建立。
3. 完成後，您可以選擇儲存並結束以儲存進度，或建立優惠以發佈私有優惠，並將私有優惠延伸到您選取的買方。

更新私有優惠的過期時間

使用下列程序來更新私有優惠的過期日期。

更新私有優惠的過期日期

1. 登入 AWS Marketplace 管理入口網站，然後選擇優惠。
2. 在優惠頁面上，選擇您要更新的優惠。
3. 選擇編輯。
4. 提供新的優惠過期日期。
5. 選擇提交。

更新完成後，優惠將變更為作用中狀態，您的買方可以接受優惠。

取消私有優惠

使用下列程序取消私有優惠。

1. 登入 AWS Marketplace 管理入口網站，然後選擇優惠。
2. 在優惠頁面上，選擇您要更新的優惠。

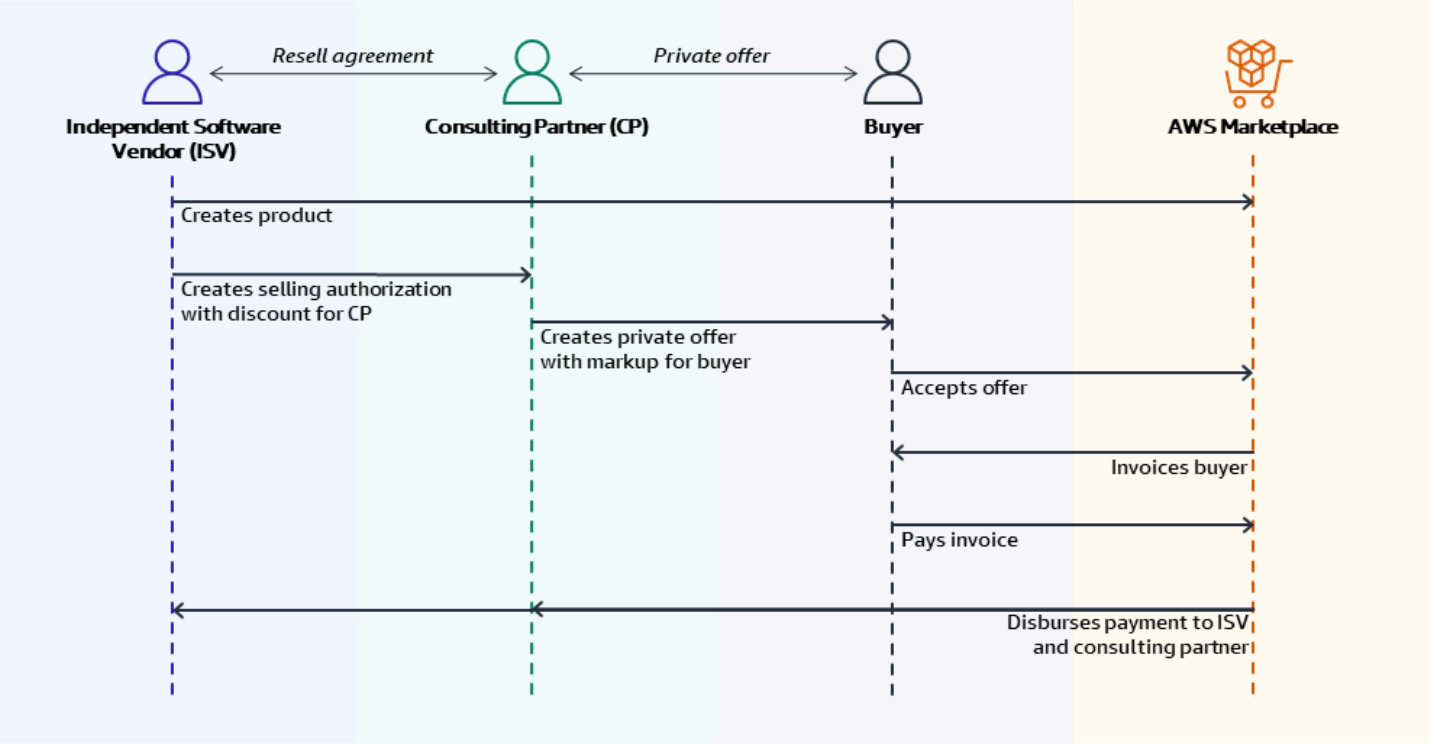
> **Note**
>
> 取消優惠將修改優惠過期日期，因此延長優惠的買方，優惠會顯示為過期。

3. 選擇動作，然後選擇取消優惠。

以 AWS Marketplace 頻道合作夥伴身分建立私有優惠

AWS Marketplace 通路合作夥伴私有優惠讓通路合作夥伴有機會在其中轉售獨立軟體廠商 (ISVs產品 AWS Marketplace。 AWS Marketplace 通路合作夥伴和 ISV 建立了協議來轉售一個或多個 ISV 產品，然後通路合作夥伴會將該產品的私有優惠提供給買方。

下圖顯示 ISV、頻道合作夥伴和買方之間的這種關係。



> **Note**
>
> 如需為頻道合作夥伴建立銷售授權做為 ISV 的詳細資訊，請參閱 建立 AWS Marketplace 頻道合作夥伴的銷售授權做為 ISV。

每個 AWS Marketplace 通路合作夥伴私有優惠僅供單一買方查看，具有自訂定價和獨特的商業條款，以滿足買方的需求。建立私有優惠時，您可以從 ISV 設定的批發成本開始。然後，標記該價格以建立買方的優惠價格。

> **Note**
>
> 建立私有優惠時，頻道合作夥伴必須使用 ISV 在銷售授權中定義的貨幣。

您可以使用下列其中一種方式來判斷批發成本：

- 經常性折扣 – ISV 授權 AWS Marketplace 通路合作夥伴以同意折扣的價格，透過經常性銷售授權來轉售其產品或產品。 AWS Marketplace 通路合作夥伴可以利用此折扣繼續銷售產品，而無需進一步與 ISV 協商價格。此折扣可以設定為持續到指定的日期，或無限期地持續到 ISV 或頻道合作夥伴結束為止。
- 非經常性折扣 – ISV 提供給 AWS Marketplace 頻道合作夥伴的銷售授權是一次性折扣，僅供特定買方使用。

在這兩種情況下，買方支付私有優惠後， AWS Marketplace 會使用標準程序，根據同意的定價將資金分配到 AWS Marketplace 頻道合作夥伴和 ISV。

> **Tip**
>
> ISVs和頻道合作夥伴可以使用 上的合作夥伴選單AWS Marketplace 管理入口網站來檢視銷售授權。

如需建立私有優惠的詳細說明，請參閱AWS Marketplace 頻道合作夥伴私有優惠 – 建立優惠。

如需私有優惠第三方貸款的相關資訊，請參閱 中的客戶貸款。 AWS Marketplace

其他資訊

如需其他資訊和問題，我們鼓勵 ISVs和管道合作夥伴與 AWS Marketplace 管道團隊聯絡。如果您不知道要特別聯絡誰，請傳送電子郵件訊息到 aws-mp-channel@amazon.com，團隊成員將在一個工作天內回覆您。

建立 AWS Marketplace 頻道合作夥伴的銷售授權做為 ISV

身為獨立軟體廠商 (ISV)，您可以為該合作夥伴建立銷售授權，授權 AWS Marketplace 通路合作夥伴轉售您的產品。通路合作夥伴可以使用銷售授權，為最終買方建立通路合作夥伴私有優惠 (CPPO)。支援的產品類型包括：

- 以 AMI 為基礎的產品
- 容器型產品
- 以 SaaS 為基礎的產品

- 專業服務產品

下列程序概述 ISVs如何為 AWS Marketplace 頻道合作夥伴建立銷售授權。若要使用此功能，您必須具有使用 中銷售授權索引標籤的許可 AWS Marketplace 管理入口網站。如需詳細資訊，請參閱AWS Marketplace 賣方的政策。

建立銷售授權

1. AWS Marketplace 管理入口網站 使用您的 AWS Marketplace 賣方帳戶登入 。

   > **Tip**
   >
   > 在使用您的 AWS Marketplace 賣方帳戶登入 AWS 帳戶 之前，請確定您已從另一個 登出。

2. 選擇銷售授權索引標籤，然後選擇建立銷售授權。
3. 在建立銷售授權頁面上，輸入下列詳細資訊：

   - 針對銷售授權名稱，輸入授權的名稱。

     > **Note**
     >
     > 您在銷售授權名稱中輸入的資訊，管道合作夥伴會在其銷售報告中看到。

   - 對於經銷商，請從下拉式清單中選擇您要授權的 AWS Marketplace 頻道合作夥伴 （經銷商）。您可以依名稱或帳戶 ID 選取經銷商。
   - 針對產品類型，選取產品類型，然後選取您要為其建立銷售授權的其中一個產品。
   - 選擇繼續以授權詳細資訊。
4. 在指定詳細資訊頁面上，輸入下列詳細資訊：

   - 針對銷售授權描述，輸入描述。

     > **Note**
     >
     > 您在銷售授權描述中輸入的資訊，管道合作夥伴會在其銷售報告中看到。

   - 對於續約，請指出此授權是否旨在續約相同產品的現有客戶現有的付費訂閱。
   - （選用）設定一或多個買方帳戶 IDs，以指定銷售授權僅適用於這些買方。

- 您可以選擇儲存並結束，或選擇下一步。在程序中的任何步驟選擇儲存並退出，會將您的銷售授權儲存為草稿。選擇下一步會帶您前往選取持續時間和定價頁面。

5. 在選取持續時間和定價頁面上，輸入下列詳細資訊：

- 針對定價模型，選擇下列其中一個選項：
  - 分期付款計劃的合約定價 - 您可以為每個維度類型指定固定數量、超額的每小時費率，以及分期付款計劃。超額按每小時費率計費，並單獨計費。
  - 含預付款的合約定價 - 您可以指定每個維度類型的合約價格，以及額外用量的每小時費率。買方可以選擇為每個維度類型選擇以合約價格遞交的數量，並在接受時收到完整金額的發票。其他用量會以每小時費率計費，並另外計費。
  - 用量定價 - 您可以為每個維度類型指定每小時費率。買方會根據其用量的每小時費率付費。
- 針對貨幣，選擇銷售授權的貨幣。

> **Note**
>
> 非 USD 貨幣僅適用於合約定價優惠。

- 針對持續時間，選取銷售授權的持續時間。

> **Note**
>
> 經銷商的開始日期必須早於製造商在轉售授權中列出的日期。

- 產品維度區段顯示產品中目前可用的合約維度。您可以選擇要包含在銷售授權中的維度，並提供數量或價格。使用分期方案的合約定價可讓您輸入每個維度的數量，使用預付付款的合約定價可讓您輸入每個維度的價格。
- 選擇管理自訂維度，將維度新增至此產品中可用於未來優惠和銷售授權的產品。無法移除新增至產品的自訂維度。1 個產品中最多可有 100 個維度。
- 對於每個用量維度的價格，請為每個維度類型指定每小時費率。買方會根據其用量的每小時費率付費。根據預設，公開優惠價格會填入此處。
- 對於買方分期付款計劃，輸入合約總計，並根據所需的頻率產生分期付款計劃。這對於具有分期付款計劃的合約定價是必要的。
- 針對每個執行個體類型的定價，設定每個執行個體類型的使用價格。選用的定價工具可讓您套用折扣到公有價格，或套用相同的價格到所有執行個體，以大量更新價格。
- 針對銷售授權可用性，請選擇下列其中一個選項，以限制建立多少個私有優惠的可用性，或直到使用此銷售授權建立私有優惠的特定時間為止：

- 單次使用 – 允許經銷商在指定的結束日期之前建立單一優惠
- 特定持續時間 – 允許經銷商在指定的結束日期之前建立多個優惠
- 無設定時間持續時間 – 允許經銷商建立多個優惠，直到手動停用銷售授權為止

- 您可以選擇儲存並結束、上一個或下一個。選擇下一步會帶您前往設定法律條款頁面。

6. 在設定法律術語頁面上，輸入下列詳細資訊：

   - 針對最終使用者授權合約，選取公有 EULA 或標準合約 for AWS Marketplace (SCMP) 或上傳自訂 EULA。

     > **Note**
     >
     > 專業服務賣方僅支援自訂 EULAs。

   - （選用）選取 AWS Marketplace (RCMP) 的經銷商合約，或上傳要包含在銷售授權中的自訂合約。
   - 您可以選擇儲存並結束、上一個或下一個。選擇下一步會帶您前往檢閱和建立頁面。

7. 在檢閱和建立頁面上，確定所有資訊都是正確的。一旦發佈銷售授權，就無法修改。
8. 選擇建立銷售授權，將銷售授權發佈給經銷商。

管理銷售授權

- 銷售授權建立的資料表已更新，以顯示相關的銷售授權詳細資訊，包括銷售授權名稱、產品名稱、經銷商名稱、建立日期、過期日期和狀態。
- 建立銷售授權後，您無法延長過期日期或修改任何其他詳細資訊。
- 如果您不再希望經銷商使用它，您可以停用銷售授權。當您停用銷售授權時，無法使用該銷售授權建立新的優惠。任何已建立的優惠都不會受到影響。
- 您也可以透過選取銷售授權並選擇複製來複製銷售授權。這會將值預先填入所有欄位，並允許編輯。

銷售授權狀態和動作

下列清單說明銷售授權狀態值及其意義：

- 草稿 - 銷售授權已建立但尚未啟用。
- 已授權 - 銷售授權處於作用中狀態，可用於建立頻道合作夥伴私人優惠 (CPPOs)。尚未建立 CPPOs。

- 已授權（可重複使用）- 銷售授權已用於建立至少一個 CPPO，並可用於建立其他 CPPOs。此狀態常見於具有特定持續時間或未設定持續時間的授權。
- 已授權（取用）- 銷售授權已充分利用，無法用於建立其他 CPPOs。這通常發生在經銷商建立私有優惠之後的單次使用授權。
- 已過期 - 銷售授權的可用性結束日期已過。它無法再用於建立 CPPOs。
- 已停用 - 獨立軟體廠商 (ISV) 已手動停用授權。它無法再用於建立 CPPOs。此狀態在 API 中稱為「限制」。

> **Note**
>
> 系統會根據數個因素決定狀態，包括銷售授權的內部狀態、可用性結束日期，並提供延長狀態。狀態值是根據優先順序指派，以解決潛在的重疊。

私有優惠分期方案

身為 AWS Marketplace 賣方，您可以使用分期付款計劃（也稱為彈性付款排程），透過自訂付款排程來擴展私有優惠。分期付款計劃適用於特定產品和定價類型的私有優惠。如需詳細資訊，請參閱符合私有優惠資格的產品類型。付款排程可以分散在接受的合約期間內，買方會定期分期付款。

對於多年期和自訂期間 Amazon Machine Image (AMI) 產品，請設定優惠中包含的每個執行個體類型的執行個體數量，以及任何其他啟動執行個體的每小時定價。買方啟動指定數量的執行個體後，任何啟動的額外執行個體都會以私有優惠中指定的每小時費率計費。

您無法在已延長至買方並由買方訂閱的私有優惠上修改付款排程。若要進行變更，您必須建立新的優惠。

為私有優惠建立分期付款計劃

建立私有優惠時，您可以使用分期付款計劃設定自訂付款排程。

為私有優惠建立分期付款計劃

1. 在設定優惠定價和持續時間頁面上，針對產品定價，選擇具有分期方案的合約定價。
2. 選擇此優惠的合約期間，並指定優惠詳細資訊。如需詳細資訊，請參閱為您的 AWS Marketplace 產品準備私有優惠。
3. 在買方分期付款計劃下，輸入下列參數：

- 合約總計
- （選用）如果您想要第一個付款與其他付款不同，請輸入初始付款。餘額將在後續付款之間平均分配。
- Frequency (頻率)

  選擇每月、每季、每年或自訂。如果您選擇自訂，也請輸入分期數。

  您最多可以新增 60 個付款。您也可以選擇對每個付款明細項目進行調整。每次調整付款明細項目時，買方的應付總額都會更新。
- 第一個發票日期

4. 選擇產生分期付款計劃。如果發票日期超過合約期限，您會收到錯誤訊息。
5. 驗證所有發票金額和日期後，請確認買方應付的總金額符合您希望買方在私有優惠期間支付的總價。若要完成建立私有優惠，請完成 中的其餘步驟the section called “起草和發佈私有優惠”。

買方接受私有優惠後，將於 UTC 00：00 依您在付款排程中定義的發票日期開立發票。在收到買方的付款後，您 AWS Marketplace 會收到每張發票的付款。優惠接受日期之前只能有一個發票日期。如果在付款排程中的第一個發票日期之後接受私有優惠，則第一個發票會在接受優惠後立即產生。您的買方訂閱後，他們可以按照排程和 AWS 發票查看所有付款，協助他們追蹤支出。

分期付款計劃報告

分期付款計劃的私有優惠報告會顯示在每月計費營收報告的第 4 節。如需詳細資訊，請參閱 第 4 節：具有彈性付款排程的合約。

為私有優惠建立未來日期的協議

身為 AWS Marketplace 賣方，您可以使用未來日期的協議來銷售買方將在預定未來日期收到的產品。在一般 AWS Marketplace 交易中，買方會在接受優惠或建立協議後立即收到產品授權或權利。相反地，與未來日期協議 (FDA) 相比，買方會在預定的未來日期收到產品授權或權利。FDA 可用於設定與買方的現有交易續約。適用於合約和具有耗用 (CCP) 定價的合約的軟體即服務 (SaaS) 產品支援 FDA，無需彈性付款。下列各節提供未來日期協議的使用資訊。

使用 FDA，您可以在買方選擇時關閉與買方的交易，而不是買方想要開始產品使用時。您可以使用 FDA 依存地對 上的交易執行下列動作 AWS Marketplace：

- 根據銷售需求預訂（買方接受優惠）交易。

- 根據您的財務或會計需求向買方收費。
- 根據買方需求，提供買方對產品的存取權，例如啟用授權或權利。

買方可以接受 FDA 優惠，即使其目前協議處於作用中狀態，如果：

- FDA 開始日期會在其現有協議結束後發生。
- 在接受 FDA 之前，會停用自動續約。

> **Important**
>
> 一旦接受 FDA，就無法重新啟用自動續約。

- 服務日期不會與任何其他接受FDAs 重疊。

主題

- 未來資料協議的考量事項
- 建立未來日期的協議
- 將分期付款計劃與未來日期的協議搭配使用
- 接收未來日期協議的通知
- 使用未來日期的協議與通路合作夥伴私有優惠的轉售

未來資料協議的考量事項

當您使用未來日期的協議時，請記住下列日期：

協議簽署日期

買方接受優惠的日期，以及協議的建立時間。

協議開始日期

啟動買方對產品的授權或權利，買方可以開始使用產品的日期。

協議結束日期

協議結束日期。協議和買方的授權或權利在此日期過期。

> **Note**
>
> 最終使用者授權合約、訂單表單或與訂閱者簽訂的其他合約中指定的合約期限，將控制與訂閱者 AWS Marketplace 發票中指定的期限是否有衝突。

建立未來日期的協議

記錄賣方會在產生具有未來開始日期的私有優惠時設定協議開始日期。買方無法變更開始日期，但可以在接受私有優惠之前檢閱開始日期 AWS Marketplace。

建立未來開始日期的私有優惠

1. 建立私有優惠時，請選擇合約期間下的未來日期開始。
2. 在服務日期區段中，輸入服務開始日期和服務結束日期。當買方接受優惠時，您在此處選擇的服務開始日期將是您未來日期協議的協議開始日期。

> **Note**
>
> 若要使用 FDA 進行續約，服務開始日期必須是您想要續約之合約結束日期的一或多天後。例如，如果協議結束日期為 12/31/2024」，您應該將服務開始日期設定為「1/1/2025」。
> 賣方可以選擇服務開始日期，最長可達未來 3 年。

將分期付款計劃與未來日期的協議搭配使用

搭配 FDA 使用分期付款計劃，您可以設定購買款項，以便在協議簽署日期和協議結束日期之間的任何時間發生。這包括協議開始日期前後的付款。

記錄的賣方會選擇私有優惠付款日期和金額。如需設定分期方案的詳細資訊，請參閱 the section called “ 為私有優惠建立分期付款計劃”。

接收未來日期協議的通知

您會收到指定根帳戶的電子郵件通知，告知您未來簽訂的協議採取的下列動作：

- 優惠接受/協議建立 （ 協議簽署日期 ）

- 授權或權利啟用時（協議開始日期）
- 協議提前 30、60 或 90 天到期的提醒
- 協議過期（協議結束日期）
- 協議修訂或取代時

> **Note**
>
> SaaS 的所有現有 Amazon Simple Notification Service (Amazon SNS) 通知也適用於 FDA。對於 FDAs，兩個 Amazon SNS 主題都會在協議開始日期（而非協議簽署日期）啟動。如需詳細資訊，請參閱the section called “SaaS 產品的 Amazon SNS 通知”。

使用未來日期的協議與通路合作夥伴私有優惠的轉售

製造商和經銷商可以針對 AWS Marketplace 通路合作夥伴私有優惠使用未來日期的協議。

身為製造商：

- 與標準 AWS Marketplace 頻道合作夥伴私人優惠 (CPPOs) 類似，製造商必須透過將轉售授權延伸至頻道合作夥伴，授權 AWS Marketplace 頻道合作夥伴建立具有未來開始日期的 CPPOs。

  若要了解如何建立轉售授權，請遵循the section called “建立銷售授權做為 ISV”頁面上的步驟。
- 建立轉售授權時，製造商可以選擇指定允許的服務開始日期上限。這將是頻道合作夥伴在 AWS Marketplace 建立對應的 AWS Marketplace 頻道合作夥伴私有優惠時可指定的服務開始日期上限。

  > **Note**
  >
  > 如果製造商未指定日期上限，則 AWS Marketplace 通路合作夥伴可以指定未來最多 3 年的任何服務日期。

身為經銷商：

- 對於經銷商和通路合作夥伴，建立未來日期的通路合作夥伴私有優惠和一般未來日期的私有優惠的步驟相同，具有一個關鍵差異。經銷商可以指定的協議開始日期必須早於製造商在轉售授權中允許的最大服務開始日期。
- 若要了解如何建立頻道合作夥伴私有優惠，請參閱 the section called “管道合作夥伴私有優惠”。

私有優惠常見問答集

此常見問答集會回答有關建立、管理和疑難排解私有優惠的常見問題。

主題

- 什麼是 Private Offer Success Team (POST)，賣方如何聯絡他們？
- 賣方如何在存取優惠索引標籤時解決錯誤？
- 什麼是分期付款計劃或彈性付款排程？
- 如何協助在存取私有優惠時收到 404 錯誤的買方？
- 為什麼客戶在嘗試接受私有優惠時看到「您已經有作用中的合約」錯誤？
- 賣方或買方是否可以取消私有優惠？
- 如何申請退款或合約取消？
- 何時開立買方發票？
- 接受優惠後，賣方應採取哪些步驟？
- 如何 AWS 支付賣方和合作夥伴的費用？
- 如何 AWS 評估稅金？
- 資源和支援

什麼是 Private Offer Success Team (POST)，賣方如何聯絡他們？

POST 可讓外部對象了解 AWS Marketplace Private Offer 體驗。賣方可以在 AWS Marketplace 管理入口網站中使用支援表單。如需支援表單的協助，請下載並參閱 Private Offer Support Form Guide PDF。

賣方如何在存取優惠索引標籤時解決錯誤？

如果您在 AWS Marketplace 管理入口網站中選擇優惠索引標籤時發生錯誤，請確定您符合下列先決條件：

建立私有優惠以直接銷售軟體或服務的必要條件

1. 確認 AWS 您的帳戶具有適當的 Identity and Access Management (IAM) 政策。如需必要政策的詳細資訊，請參閱私有優惠的 IAM 政策。

> **Note**
>
> 如果您需要協助修改 IAM 政策或許可，請聯絡您的內部 AWS 管理員。 AWS 無法協助 IAM 政策或許可，因為存取權是由客戶根據共同責任模型管理。

2. 至少有一個非零的公開上市產品。您可以透過搜尋您的廠商來驗證這一點 AWS Marketplace。如果沒有產品出現，您可能沒有公開列出產品，或者您的清單有限。

建立頻道合作夥伴私有優惠 (CPPOs) 以轉售軟體或服務的必要條件

- 符合地理先決條件，以成為 中的「賣方」 AWS Marketplace：
  - 使用合格司法管轄區的法人實體，或在其中一個區域中註冊的商業實體。
  - 提供銀行帳戶的必要資訊。您可以提供一或多個銀行帳戶，包括：
    - 美國自動清算中心 (ACH) 帳戶
    - 來自合格司法管轄區的全球銀行間金融電信協會 (SWIFT) 銀行帳戶
    - Hyperwallet 帳戶

什麼是分期付款計劃或彈性付款排程？

分期付款計劃也稱為彈性付款排程 (FPS)，可讓您使用自訂付款排程來擴展私有優惠。這些計劃適用於特定產品和定價類型的私有優惠。如需詳細資訊，請參閱符合私有優惠資格的產品類型。

付款排程可以分散在接受的合約期間內，買方會定期分期付款。訂閱後，您的客戶可以在排程和 AWS 發票上檢視所有付款，協助他們追蹤支出。

分期付款計劃和 FPSs賣方指定：

- 每個維度或每個執行個體類型的單位數量
- 合約的付款條件 （ 預付、延遲或多張發票 ）

建立分期付款計劃

> **Note**
>
> 在軟體轉售案例中，獨立軟體廠商 (ISV) 會決定分期付款計劃。

建立分期付款計劃

1. 在設定優惠定價和持續時間頁面上，針對產品定價，選擇具有分期方案的合約定價。
2. 選擇合約持續時間，並指定優惠詳細資訊。
3. 在買方分期付款計劃下，輸入所需的參數：
   - 對於接受時的預付發票，請輸入美元金額，並將發票日期設定為您建立私有優惠的日期。
   - 對於延遲發票，請輸入美元金額並設定未來的發票日期。
   - 針對分期發票，選擇新增付款以輸入包含金額和發票日期的多個付款明細項目。

如需詳細資訊，請參閱建立分期方案。您也可以在 Installment Plans 上觀看影片教學課程。

設定固定 SKU 或執行個體類型

在優惠建立頁面的維度區段中：

1. 選擇「買方可以選擇一個或多個提供的選項」。
2. 輸入不同的維度以及每個維度的數量。
3. 選擇「新增維度」以包含多個維度，或選擇「建立並新增維度」以為此私有優惠建立自訂維度。

如何協助在存取私有優惠時收到 404 錯誤的買方？

以下是 404 個錯誤的常見原因及其解決方法：

不正確的帳戶關聯

1. 要求買方存取其AWS 主控台中的 Private Offers 索引標籤。
2. 如果在可用私有優惠或已接受和過期優惠索引標籤中看不到優惠：
   - 確認買方已登入正確的帳戶。他們可以在 AWS 主控台的右上角確認其帳戶 ID。

- 如果登入正確的帳戶，請確保私有優惠適用於 Amazon Machine Image (AMI) 有限清單 （請參閱下方的 AMI 有限清單區段）。

解決方案：買方需要登入正確的帳戶，或者您需要向正確的 AWS 帳戶 ID 發出私有優惠。

已過期的私有優惠

- 如果優惠出現在已接受和過期的優惠索引標籤中，則表示優惠已過期。

解決方案：延長過期日期或發出新的私有優惠。

AMI 有限清單產品

- 如果任一標籤都看不到優惠，且買方已登入正確的帳戶，則產品可能處於有限清單狀態。

解決方法：

- 對於單一 AMI 有限清單：遵循允許列出買方帳戶的指示，允許列出買方的帳戶。
- 如需其他伺服器類型限制清單：請聯絡 Private Offer Success Team (POST) 以允許列出買方帳戶。

> **Note**
>
> 允許清單是每個產品的一次性活動。

私有市場限制

- 如果買方有私有市集，他們可能會看到錯誤，指出其私有目錄中沒有該產品。

解決方案：買方需要將產品新增至其允許清單。請參閱將產品新增至私有市集。

如果問題仍然存在，請要求買方嘗試下列動作：

- 登出並返回
- 清除其瀏覽器快取
- 刪除 Cookie
- 登入 incognito 視窗

- 使用不同的瀏覽器 （ 非 Internet Explorer)

為什麼客戶在嘗試接受私有優惠時看到「您已經有作用中的合約」錯誤？

當買方帳戶已經有產品的作用中訂閱時，就會發生此錯誤。解析度取決於產品類型：

軟體即服務 (SaaS) 合約

每個買方帳戶只能有一個有效的 SaaS 合約訂閱，或是與消耗產品的合約。若要更新或展開作用中訂閱：

1. 從 AWS Marketplace 管理入口網站 (AMMP) 中的協議索引標籤建立以協議為基礎的優惠。
2. 在以協議為基礎的優惠中包含來自原始優惠的任何待定付款期限，因為它將覆寫尚未開立發票的剩餘付款。

建立以協議為基礎的優惠

對於 Marketplace Private Offer (MPPO)：

1. 導覽至 AMMP 中的協議索引標籤。
2. 搜尋買方的帳戶 ID。
3. 選取買方目前的協議。
4. 選擇「建立以協議為基礎的優惠」。

對於頻道合作夥伴私人優惠 (CPPO)：

1. ISV 向頻道合作夥伴授予銷售授權，其中包含來自其合作夥伴索引標籤的新定價，包括授權單位數量、CP 的付款條件，以及合約持續時間。
2. 頻道合作夥伴導覽至 AMMP 中的協議索引標籤。
3. 搜尋買方的帳戶 ID。
4. 選取買方目前的協議。
5. 選擇建立以協議為基礎的優惠。
6. 開啟銷售授權清單，然後從步驟 1 選取新機會。這可讓頻道合作夥伴輸入以協議為基礎的優惠條款。

如需詳細資訊，請參閱建立以協議為基礎的優惠。

SaaS 合約和具有消費定價 CCPs) 的合約的其他選項：

- 針對以不同買方帳戶為目標的相同產品 ID，建立新的私有優惠。
- 為相同的買方帳戶發行不同產品 ID 的新私有優惠。
- 請求取消作用中合約，然後為相同的產品 ID 和買方 ID 發出新的私有優惠。

對於 AMI，每小時和每年

每個買方帳戶只能有一個作用中的訂閱。若要更新或展開：

1. 從優惠索引標籤發出新的優惠。
2. 當買方接受時，新條款會覆寫先前的合約條款。
3. 除非取消，否則上一個優惠付款排程的待定費用將繼續。

對於 AMI 合約和專業服務

必須先取消買方目前的訂閱，才能接受新的私有優惠。使用退款/取消表單來啟動取消請求。

賣方或買方是否可以取消私有優惠？

- 如果買方尚未訂閱：賣方可以透過導覽至 AWS Marketplace 管理入口網站中的優惠索引標籤，選取優惠，然後選擇取消來取消。
- 如果買方已訂閱：記錄賣方必須使用 AWS Marketplace 管理入口網站中的退款/取消表單啟動取消請求。如需詳細說明，請參閱此影片教學課程。

如何申請退款或合約取消？

退款和合約取消由 AWS 客戶服務團隊處理。

Sellers—適用於 Marketplace 的 ISVs私有優惠、適用於頻道合作夥伴私有優惠的頻道合作夥伴—必須使用退款/取消表單來啟動退款或取消。

此影片說明整個程序。

賣方必須輸入下列資料。

- 買方帳戶 ID：此資訊可在優惠詳細資訊或帳單營收儀表板中找到。這必須是訂閱者帳戶 ID。
- 賣方帳戶 ID：這是用來建立私有優惠的賣方 AWS 帳戶 ID。

- 產品 ID：您可以在優惠詳細資訊或舊版產品 ID 欄中的 計費收入儀表板中找到此資訊。
- 帳單日期：您可以在優惠詳細資訊或用量開始期間欄中的 計費收入儀表板中找到此資訊。
- 退款金額：如果不需要退款，賣方可以將此設定為 0 美元。
- 其他詳細資訊：請參閱下列備註。

> **Important**
>
> - 對於包含合約取消的請求，請包含下列文字"
>
> 請取消帳戶 X 的 offer-X 訂閱。
>
> 對於退款，在本節中指定買方的 AWS 發票 ID 有幫助，但不是強制性的。
> - 儲存提交後提供的參考 ID，以供日後進行後續追蹤時使用。

提交請求後，請檢查AWS 支援主控台是否有狀態更新。

何時開立買方發票？

- 接受時：發票會在訂閱後立即在帳單主控台中建立。
- 彈性付款排程：發票是根據賣方和買方交涉的自訂付款排程。
- AWS 合併帳單 （ 每月 2/3 日 ）：可包含公開優惠購買和含計量的產品。

接受優惠後，賣方應採取哪些步驟？

1. 追蹤優惠接受。
2. 對於財務團隊：標準實務是讓賣方透過建立開放的應付款項並禁止為客戶建立 AWS Marketplace 訂單發票來追蹤購買。

如何 AWS 支付賣方和合作夥伴的費用？

- 只有在成功向訂閱者收取款項之後，才會啟動付款。
- 預設支出節奏上的支出是每月 7 號和 10 號之間的每月。或者，ISV 和頻道合作夥伴可以選取每日或每月的支出排程。如果 ISV 或頻道合作夥伴選取每月選項，他們可以選取想要收到其支出的月份日期。

- 支出會存入 ISV 或頻道合作夥伴帳戶的美國銀行帳戶，減去列示費用。資金可能需要 1 到 3 個工作天才能送達一般銀行 ACH SLAs。如果您沒有美國銀行帳戶，您可以使用 Hyperwallet 將 Amazon 銷售的支出匯入存款帳戶，並以您的當地貨幣將其直接轉移到當地銀行。

如何 AWS 評估稅金？

AWS Marketplace 會根據下列項目收取稅金：

- 產品訂閱者的稅務地址
- 產品類型
- Marketplace 協調人員法律

Marketplace 協調人員：要求 Marketplace Operator 向稅務機關收取、收取和匯款稅金。

非市場協調人員：責任取決於賣方。

如需稅務義務的詳細資訊，請參閱 AWS Marketplace – 賣方的稅務說明。

資源和支援

如果您有特定請求，請透過 AWS Marketplace 管理入口網站聯絡下列其中一個 AWS Marketplace 團隊。

Private Offers Success Team (POST)： 團隊支援賣方 (ISV 和通路合作夥伴） 和買方提供私有優惠操作支援和支援。透過支援表單聯絡我們。

廠商財務成功團隊 (VFS)： 團隊改善了影響廠商財務現金應用程式、對帳和相關報告的程序，並透過有關 MPPOs 和 CPPOs 非公開優惠的財務特定問題來支援加入賣方 (ISV 和通路合作夥伴）。透過支援表單聯絡我們。

Managed Catalog Operations (MCO)： 團隊負責將 AWS Marketplace 軟體的第三方賣方加入平台、審查和處理其軟體產品，以確保政策合規和買方體驗，以及管理與賣方的營運關係。透過支援表單聯絡我們。

如果您不熟悉私有優惠程序，請使用此影片庫來協助您開始使用介紹、概觀和常見問題的答案。您也可以在連結影片或AWS Marketplace 廠商財務成功 PDF 中找到有關買方發票、支出或AWS 稅務規則問題的答案。

使用協議

AWS Marketplace 賣方協議是賣方與買方之間銷售產品的合約 AWS Marketplace。當買方接受優惠時，即會形成協議，該優惠可以是公有或私有。本節中的主題說明協議類型、狀態，以及如何變更或修改協議。

主題

- 協議類型
- 使用協議
- 在 中修訂協議 AWS Marketplace

協議類型

AWS Marketplace 提供下列類型的協議：

- 公開優惠協議 – 從賣方購買公開上市產品時，買方接受的標準、未交涉法律協議 AWS Marketplace。
- 私有優惠協議 – 買方接受的私有優惠會成為協議。如需詳細資訊，請參閱 建立和管理私有優惠。
- 未來日期協議 – 買方會在預定的未來日期收到產品授權或權利。如需詳細資訊，請參閱 為私有優惠建立未來日期的協議。

使用協議

您可以從 中的協議頁面檢視和管理協議AWS Marketplace 管理入口網站。在協議頁面上，協議表格會顯示您銷售產品的合約 AWS Marketplace。在協議表格中，協議可以有下列其中一種狀態：

- 作用中 – 協議的條款為作用中。
- 過期 – 合約將在 90 天內過期。
- 已過期 – 協議於定義的合約結束日期結束。
- 已續約 – 協議已續約，建立具有更新條款的新協議。
- 已取消 – 接受者提早結束協議。
- 已取代 – 已使用協議替代優惠取代協議。
- 已終止 – 協議在原始合約結束日期 AWS 之前因付款失敗等原因而由 終止。
- 已封存 – 協議在沒有指定原因的情況下結束。

尋找協議

若要尋找協議，請設定篩選條件。選擇尋找協議搜尋列，然後選擇篩選條件屬性和值。例如，若要尋找特定協議，請選擇協議 ID，然後選擇 ID。若要儲存篩選條件，請針對清除篩選條件，選擇另存為新的篩選條件集。

排序協議

資料表一開始會依協議結束於 (UTC) 欄排序。它會以遞減順序顯示協議，其中最接近未來的協議會出現在清單頂端。您可以選擇協議結束於 (UTC) 欄標題，以變更排序順序。

檢視協議詳細資訊

若要檢視協議詳細資訊，請選擇協議 ID 欄中的連結。或者，您可以選擇協議旁的選項，然後選擇檢視詳細資訊。

> **Note**
>
> 若要修改協議，請參閱 在 中修訂協議 AWS Marketplace。

下載協議詳細資訊

使用下列程序在 .pdf 檔案中下載協議詳細資訊。

下載協議詳細資訊

1. 登入 AWS Marketplace 管理入口網站，然後選擇協議。
2. 在協議表格中，選取協議旁的選項，然後選擇檢視詳細資訊。或者，您可以在協議 ID 欄中選擇協議的連結。
3. 在協議詳細資訊頁面上，選擇下載 PDF。

在 中修訂協議 AWS Marketplace

AWS Marketplace 賣方、通路合作夥伴和 ISVs可以提供升級、續約和修訂，以取代作用中的協議。例如，您可以授予新權利、提供定價折扣、調整付款排程，或變更最終使用者授權合約 (EULA) 以使用標準化授權條款。您也可以變更單位數量和付款排程，並新增自訂結束日期。

軟體即服務 (SaaS) 合約和含消費產品的 SaaS 合約支援公有和私有優惠修訂。所有 AWS Marketplace 賣方都可以升級、續約或修改這些產品類型的私有優惠，包括獨立軟體廠商 (ISVs) 和管道合作夥伴。下列各節提供程序的相關資訊。

優惠與協議之間的差異在於買方是否接受其條款：

- 優惠是買方使用產品的一組條款。優惠可以是公有或私有。
- 協議是買方接受的優惠。協議包括賣方使用公有或私有優惠提供的購買和免費產品。

您無法修改協議，以指定與原始協議記錄賣方不同的記錄賣方。若要使用此功能，您必須具有使用中協議索引標籤的許可 AWS Marketplace 管理入口網站。如需相關資訊，請參閱AWS Marketplace 賣方的許可。

主題

- 公有和私有優惠修訂支援的產品類型
- 建立公有和私有優惠升級、續約和修訂
- 升級、續約和修訂的報告

公有和私有優惠修訂支援的產品類型

只有下列產品類型支援優惠修訂：

- SaaS 合約
- 使用 的 SaaS 合約

您可以在 的協議索引標籤上查看下列其他產品類型 AWS Marketplace 管理入口網站。不過，這些產品類型不支援修訂：

- SaaS 用量型產品
- 以 AMI 為基礎的產品
- 容器型產品
- 伺服器合約
- 專業服務產品

建立公有和私有優惠升級、續約和修訂

您可以使用 AWS Marketplace 管理入口網站 下列程序，從 建立優惠升級、續約和修訂。對於頻道合作夥伴私有優惠，頻道合作夥伴在建立修訂時必須使用銷售授權中定義的貨幣。

> **Note**
>
> - 修改後的優惠不能有未來的協議開始日期。修訂後僅提供目標有效協議，一旦接受，修訂後的協議會立即變成有效，使得未來無法交往。
> - 如果您修改已接受的公開優惠，它會成為私有優惠，不再自動續約。若要維持自動續約，買方必須訂閱公開優惠。如需詳細資訊，請參閱 私有優惠的運作方式。

建立優惠升級、續約和修訂

1. 登入 AWS Marketplace 管理入口網站並選擇協議。
2. 在協議頁面上，選擇協議旁的核取方塊，然後選擇檢視詳細資訊。
3. 在檢視協議頁面上，選擇修改協議。
4. 在修改協議詳細資訊頁面上，您也可以變更服務日期、產品維度、優惠貨幣 （適用於 AWS MP 直接 PO)、付款排程、用量維度、續約狀態、EULA 和優惠過期日期。
5. 檢閱優惠並選擇建立優惠。

> **Tip**
>
> 輸入描述性自訂優惠名稱可協助您在優惠頁面上區分作用中優惠。買方也可以看見自訂優惠名稱。
> AWS 建議使用自訂優惠名稱，其中包含任何其他識別詳細資訊，例如您自己的 IDs和採購訂單號碼。也建議使用高階描述，例如 **upgrade**或 **renewal**和自訂公司名稱。請勿使用任何個人身分識別資料 （例如，名字或姓氏、電話號碼或地址）。您最多可以為此欄位輸入 150 個字元。

有關已修訂優惠的詳細資訊

修改後的優惠會在大約 45 分鐘內顯示在 私有優惠 頁面上。若要檢視優惠，請登入 AWS Marketplace 管理入口網站 並選擇優惠。在私有優惠頁面上，買方可以選擇接受優惠或繼續原始協議。

如果買方接受公有或私有優惠升級或續約，新協議會立即生效，且協議會列在 中的協議頁面上 AWS Marketplace 管理入口網站。先前協議中任何剩餘的排程付款都會取消。買方接受修訂的方式與接受私有優惠的方式相同。如需私有優惠買方體驗的詳細資訊，請參閱《 AWS Marketplace 買方指南》中的私有優惠。

如果買方在到期前不接受公有或私有優惠升級或續約，原始協議將保持不變。

> **Note**
>
> 對於 SaaS 產品的 Amazon SNS 通知，當買方接受修訂`offer-identifier`時，`subscribe-success`訊息會與新的 一起傳送。

升級、續約和修訂的報告

公有和私有優惠升級和續約會顯示在現有的賣方報告以及與優惠相關的報告中。每天每日業務報告都會產生 每日客戶訂閱者報告和 。每月計費營收報告 是每月產生的。

在每日客戶訂閱者報告中，訂閱意圖欄位指出報告項目是否為新的私有優惠。上一個優惠 ID 欄位會顯示任何上述優惠的 ID。所有私有優惠都會在報告中標示為「私有」。

> **Important**
>
> 修訂會取代買方目前的訂閱。現有的發票保持不變。不過，修訂中的付款排程會取代先前訂閱的待處理發票。

中的 AMI 型產品 AWS Marketplace

身為 AWS Marketplace 賣方，您可以使用 Amazon Machine Image (AMIs) 將產品交付給買方。AMI 提供啟動 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體所需的資訊。您可以為您的產品建立自訂 AMI，而購買者可以使用它來建立 EC2 執行個體，其中產品已安裝並可供使用。本主題提供使用 AMIs交付 AWS Marketplace 產品的相關資訊。

當購買者使用您提供的 AMI 時，他們需要按照您為產品建立的定價和計量選項，支付他們建立的執行個體的費用。買方可以使用您的產品 AMI，方式與他們在其中使用其他 AMIs 的方式相同 AWS，包括製作新的自訂 AMI 版本。從 AMI 建立的 EC2 執行個體仍會根據 AMI 產品程式碼以您的產品計費。

AMI 型產品交付方法

您可以使用下列其中一種方式交付 AMI 型產品：

- 單一 AMI – 買方會選取並使用 AMI 做為 EC2 執行個體的範本。買方可以使用 Amazon Machine Image 交付方法篩選條件來尋找這些產品。如需詳細資訊，請參閱建立 AMI 型產品。
- AWS CloudFormation 範本 – 您可以建立範本，允許買方安裝具有不同角色的多個執行個體系統做為單一單位。買方可以使用 CloudFormation 交付方法篩選條件來尋找這些產品。如需詳細資訊，請參閱將 CloudFormation 範本新增至您的產品。

其他資源

如需 AMI 產品的詳細資訊，請參閱下列主題。

AWS Marketplace

- 的產品定價 AWS Marketplace
- 使用 設定 AMI 產品的自訂計量 AWS Marketplace Metering Service

AMI 型產品

- 了解 中的 AMI 型產品 AWS Marketplace
- 建立 AMI 型產品
- 以 AWS Marketplace 賣方身分管理 AMI 型產品
- 將 CloudFormation 範本新增至您的產品

- 建置 AMIs 以搭配 使用的最佳實務 AWS Marketplace
- 的 AMI 產品定價 AWS Marketplace
- 在 上接收 AMI 產品的 Amazon SNS 通知 AWS Marketplace
- 的 AMI 產品檢查清單 AWS Marketplace
- 的 AMI 型產品需求 AWS Marketplace

了解 中的 AMI 型產品 AWS Marketplace

身為 AWS Marketplace 賣方，您可以使用 Amazon Machine Image (AMIs) 將產品交付給買方。AMI 提供啟動 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體所需的資訊。以下各節說明使用 AMI 型產品的關鍵概念。

> **Note**
>
> 您只能在 AMI 產品中使用一個 AMI，但您可以將該 AMI 的版本新增至產品。

主題

- 產品生命週期
- AMI 產品代碼
- 變更請求
- 產品載入表單
- 年度協議修訂

產品生命週期

以 AMI 為基礎的產品包含一或多個版本的軟體，以及整體產品的中繼資料。當您建立產品時，您可以在 中設定其屬性， AWS Marketplace 包括產品名稱、描述和定價。您也可以判斷產品的適當類別並新增關鍵字，讓您的產品出現在相關搜尋中。

您也可以建立軟體的第一個版本。這可以只是 AMI，也可以包含 AWS CloudFormation 範本或軟體套件，供買方用來建立自己的 AMIs。如需詳細資訊，請參閱AMI 型產品交付方法。

對於付費產品，買方需支付已安裝執行個體的數量。若要測量軟體追蹤的不同維度，例如產品使用者數量，請將您的產品與 AWS Marketplace 計量服務整合。如需詳細資訊，請參閱使用 設定 AMI 產品的自訂計量 AWS Marketplace Metering Service。

當您建立產品和軟體的第一個版本時，最初會在有限範圍內發佈，因此只有您的帳戶可以存取它。當您準備好時，您可以將其發佈至 AWS Marketplace 目錄，以允許買方訂閱和購買您的產品。

您可以使用伺服器產品頁面來檢視您的產品清單。產品將具有下列其中一種狀態：

- 預備 – 您仍在為其新增資訊的不完整產品。在第一次儲存並退出建立自助式體驗時，成功的變更請求會建立未發佈的產品，其中包含您提交的完整步驟中的資訊。從此狀態，您可以繼續新增資訊至產品，或透過變更請求變更已提交的詳細資訊。
- 有限 – 產品提交至系統後即完成，並在系統中傳遞所有驗證。然後，產品會發佈至有限狀態。此時，產品有一個詳細資訊頁面，只有您的帳戶和您已允許列出的任何人才能存取。您可以透過詳細資訊頁面來測試您的產品。如有必要，如需詳細資訊和協助，請聯絡AWS Marketplace 賣方營運團隊。
- 公開 – 當您準備好發佈產品，以便買方可以檢視和訂閱產品時，您可以使用更新可見性變更請求。這會為 AWS Marketplace 賣方營運團隊啟動工作流程，以根據我們的政策來檢閱和稽核您的產品。一旦產品獲得核准並處理變更請求，產品的狀態就會從限制為公開。
- 受限 – 如果您想要停止新使用者訂閱您的產品，您可以使用更新可見性變更請求來限制產品。受限狀態表示現有使用者可以繼續使用產品。不過，大眾將無法再看到該產品，也無法供新使用者使用。

的 AMI 型產品的生命週期 AWS Marketplace 不會在您發佈第一個版本後結束。您應該將產品與新版本的軟體保持最新狀態，以及與基礎作業系統的安全修補程式保持同步。

作為完整 AMI 型產品生命週期的範例，假設賣方想要在其上銷售其 AMI 型產品 AWS Marketplace。以下是賣方如何隨時間建立和維護產品的方式：

1. 建立產品 – 賣方會建立產品，並將 1.0.0 版發佈至 AWS Marketplace。買方可以建立 1.0.0 版的執行個體並使用它。
2. 新增新版本 – 稍後，賣方會將新功能新增至產品，並新增包含該功能的新版本 1.1.0。買方仍然可以使用原始版本 1.0.0，也可以選擇新版本 1.1.0。

   > **Note**
   >
   > 與新產品不同，新版本會發佈至完全公有可用性。只有在產品整體為有限版本時，您才能在中測試它們， AWS Marketplace 而客戶不會看到它們。

3. 更新產品資訊 – 使用 1.1.0 版，賣方可透過使用描述功能的新反白文字更新產品資訊，讓買方了解新功能。

4. 新增次要版本 – 當賣方修正 1.1.0 版中的錯誤時，他們會透過新增 1.1.1 版來發行該版本。買方現在可以選擇使用 1.0.0、1.1.0 或 1.1.1 版。
5. 限制版本 – 賣方決定錯誤嚴重到不希望買方能夠使用 1.1.0 版，因此限制該版本。然後，沒有任何新客戶可以購買 1.1.0（他們只能選擇 1.0.0 或 1.1.1），雖然現有買方仍然可以存取它。
6. 更新版本資訊 – 為了協助這些現有的買方，賣方會更新 1.1.0 的版本資訊，建議 升級至 1.1.1 版。
7. 監控用量 – 當買方購買和使用產品時，賣方會使用 監控銷售額、用量和其他指標 AWS Marketplace 中的賣方報告、資料饋送和儀表板 AWS Marketplace。
8. 移除產品 – 當不再需要產品時，賣方會從中移除它 AWS Marketplace。

在此範例中，賣方在產品中建立了三種不同的 AMI 版本，但只有兩個可供新買方使用（在移除產品之前）。

若要修改版本或產品資訊，您可以在 中建立變更請求 AWS Marketplace 管理入口網站。

如需建立和管理 AMI 型產品步驟的詳細說明，請參閱 建立 AMI 型產品。

AMI 產品代碼

當您在 中建立產品時，系統會將唯一的產品代碼指派給您的產品 AWS Marketplace。該產品程式碼與您產品的 AMI 相關聯，用於追蹤產品的使用情況。當買方使用 軟體時，產品代碼會自動傳播。例如，客戶訂閱並啟動 AMI、設定 AMI，以及產生新的 AMI。新的 AMI 仍然包含原始產品代碼，因此正確的用量追蹤和許可仍然存在。

> **Note**
>
> 產品代碼與您產品的產品 ID 不同。中的每個產品 AWS Marketplace 都會獲指派唯一的產品 ID。產品 ID 用於在 AWS Marketplace 目錄、客戶帳單和賣方報告中識別您的產品。產品程式碼會附加至從 AMI 建立的執行個體，做為執行個體中繼資料。當具有該產品碼的 AMI 用於建立執行個體時，客戶會收到顯示相關聯產品 ID 的帳單。建立產品後，請在 AWS Marketplace 管理入口網站 頁面尋找產品代碼和產品 ID。

身為賣方，您的軟體可以在執行時間從執行個體中繼資料取得執行中 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的產品程式碼。您可以使用產品程式碼來提高安全性，例如在產品開始時驗證產品程式碼。在產品發佈至有限狀態進行測試之前，您無法對 AMI 的產品程式碼進行 API 呼叫。如需驗證產品程式碼的詳細資訊，請參閱 驗證您的軟體是否在 AMI 上執行 AWS Marketplace 。

變更請求

若要變更 中的產品或版本 AWS Marketplace，您可以透過 提交變更請求 AWS Marketplace 管理入口網站。變更請求會新增至佇列，而且可能需要幾分鐘到幾天的時間才能解決，具體取決於請求的類型。您可以在 中查看請求的狀態 AWS Marketplace 管理入口網站。

> **Note**
>
> 除了 之外 AWS Marketplace 管理入口網站，您也可以使用AWS Marketplace 目錄 API 建立變更請求。

您可以為 AMI 型產品請求的變更類型包括：

- 更新向買方顯示的產品資訊。
- 更新向買方顯示的版本資訊。
- 新增您產品的新版本。
- 限制 版本，讓新買家無法再存取該版本。
- 更新 AWS 區域 產品可用的 。
- 更新產品的定價和執行個體類型。
- 從中移除產品 AWS Marketplace。

> **Note**
>
> 有些變更請求要求您使用產品載入表單來建立請求。如需詳細資訊，請參閱產品載入表單。

更新變更請求

以更新開頭的變更請求將載入目前的詳細資訊。然後，您會進行更新，以覆寫現有的詳細資訊。

新增或限制變更請求

新增和限制請求對專門用於在每次請求成功後佈建的步驟和更新。在自助式體驗中選擇儲存並結束和提交動作後，請求就會成功。

例如，如果 AMI 資產在新增後佈建至執行個體和區域，則它們只能受到限制，而不是完全移除。這表示現有的訂閱者和使用者可以繼續使用產品，直到其訂閱或合約用盡。不過，任何新的訂閱者都無法新增至處於受限狀態的產品。

產品載入表單

一般而言，您可以使用 AWS Marketplace 管理入口網站 來建立或編輯產品。不過，有些操作要求您使用產品載入表單 (PLF)。

PLF 是試算表，其中包含產品的所有資訊。若要取得 PLF，您可以：

- 從 中的產品詳細資訊頁面下載現有產品的 PLF AWS Marketplace 管理入口網站。
- 為需要 PLF 的動作選取選單項目。例如，如果您建立新的每月計費伺服器產品，系統會提示您下載適當的 PLF。

  如果動作是對現有產品的編輯，則 PLF 會預先填入該產品的資訊，因此您只需變更要更新的詳細資訊。
- 如果您需要新的空白 PLF，請導覽至 AWS Marketplace 管理入口網站 檔案上傳頁面。此頁面包含您要建立之產品類型的 PLFs 連結。

完成 PLF 之後，請將其上傳至 AWS Marketplace 管理入口網站 檔案上傳頁面。PLF 本身在指示索引標籤中有更詳細的指示。

年度協議修訂

每小時（每年）計劃修訂可讓您和您的買方對現有計劃進行下列變更：

- 在 Amazon EC2 執行個體類型系列之間切換
- 在 Amazon EC2 執行個體類型大小之間切換
- 新增執行個體類型
- 增加協議中現有執行個體類型的數量

只要變更的按比例分配成本大於零（無法降低訂閱的值），買方就可以進行變更。新新增的 Amazon EC2 執行個體按比例分配的成本，是根據根據合約剩餘期間調整的執行個體類型的年度成本。切換執行個體類型時，移除的 Amazon EC2 執行個體類型的按比例分配成本會從新新增的 Amazon EC2 執行個體類型的按比例分配成本中扣除。

不需要採取其他動作，即可對 AMI 年度產品啟用修訂。所有以公開優惠所簽訂的協議，以及不使用分期付款計劃的私有優惠所簽訂的協議，都支援修訂。

您可以在下列儀表板上查看買方所做的修訂：

- 協議和續約精靈 – 修訂後的協議清單。
- 帳單收入儀表板 – 向客戶收取費用。
- 集合和支出儀表板 – 支出。

建立 AMI 型產品

Amazon Machine Image (AMI) 自助式體驗會在您建立產品清單和提出變更請求時為您提供指引。透過使用自助式體驗，您可以直接更新產品清單，減少 AWS Marketplace 賣方營運團隊處理所需的時間。自助服務體驗的許多步驟都與 中的目錄系統保持一致 AWS Marketplace，這有助於直接驗證，而不是等待 AWS Marketplace 賣方營運團隊的處理和驗證。本主題說明如何使用 AMI 自助服務體驗來建立單一 AMI 的產品清單。客戶使用 AMIs 來建立 Amazon EC2 執行個體，您的產品已安裝並設定完成。

主題

- 先決條件
- 了解自助式體驗
- 建立清單
- 其他資源

先決條件

建立 AMI 產品清單之前，您必須完成下列先決條件：

1. 有權存取 AWS Marketplace 管理入口網站。這是您用來註冊為賣方和管理您銷售產品的工具 AWS Marketplace。若要進一步了解如何存取 AWS Marketplace 管理入口網站，請參閱 賣方的 AWS Marketplace 政策和許可。
2. 註冊為賣方，如果您想要為您的產品收費，請提交您的稅務和銀行資訊。若要進一步了解如何成為賣方，請參閱以 AWS Marketplace 賣方身分開始使用。
3. 擁有您想要銷售的產品。對於 AMI 型產品，這通常表示您已建立或修改伺服器軟體，並建立 AMI 供客戶使用。若要進一步了解如何準備 AMI 以供 使用 AWS Marketplace，請參閱建置 AMIs 以搭配使用的最佳實務 AWS Marketplace。

了解自助式體驗

自助式體驗會引導您完成建立產品 AWS Marketplace。進行這些步驟時，您可以指定產品資訊和 AMI 部署設定 AWS 區域，例如執行個體類型和 AMI 詳細資訊。您也可以設定交易詳細資訊，包括定價、國家/地區可用性、EULA 和退款政策。您可以選擇指定 AWS 帳戶 IDs 清單，以在產品處於有限狀態時存取和測試產品。

開始之前，請檢閱自助式體驗的下列關鍵層面：

- 您只能在目前步驟中完成必要欄位後，才能前往下一個步驟。此要求是因為每個步驟結束時都有頁面層級驗證。您無法儲存或提交未完成的步驟。
- 如果您需要在完成程序中的所有步驟之前結束工作階段，您可以選擇儲存並退出，將完成的步驟提交到預備區域。
- 未完成且未通過驗證的步驟不會提交至系統。部分完成的步驟無效，無法儲存。
- 當您選擇儲存並結束時，儲存並結束對話方塊會顯示通過驗證檢查的步驟。您可以檢閱並選擇儲存直到最後完成且經過驗證的步驟。如果出現驗證錯誤或缺少詳細資訊，您可以選擇修正以返回該步驟。
- 在您儲存並結束之後，請求正在處理中。完成處理可能需要幾分鐘或數小時的時間。在請求成功之前，您無法繼續這些步驟或進行變更。在第一次儲存和結束時，請求會與您已完成的步驟平行建立產品。
  - 請求成功後，您已完成儲存。若要在產品概觀頁面上繼續變更，請選擇繼續產品建立或使用請求變更來更新您先前在上一個工作階段中提交的詳細資訊。當您繼續時，請注意您已完成的步驟會以綠色成功標籤標記。若要更新先前提交的步驟，請使用請求變更。先前的儲存和結束請求必須先完成，才能繼續此步驟。
- 完成所有步驟後，您可以選擇下一步以查看檢閱。選擇提交系統執行最終驗證的請求。收到成功回應後，產品會移至有限狀態。您可以在詳細資訊頁面上看到，允許清單上的任何人現在可以使用該產品。如果請求失敗，產品會保持在預備狀態，並要求您在重新提交之前進行更正。

建立清單

本節中的步驟說明如何建立單一 AMI 產品的清單。

> **Note**
>
> 您只能在目前步驟中完成必要欄位時，前往下一個步驟。您無法儲存或提交未完成的步驟。如果您需要在完成程序中的所有步驟之前結束工作階段，請選擇儲存並退出，將完成的步驟提交到預備區域。如需詳細資訊，請參閱了解自助式體驗。

建立單一 AMI 產品

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 從產品功能表中，選擇伺服器。或者，您可以直接前往伺服器產品頁面。
3. 從伺服器產品索引標籤中，選取建立伺服器產品，選取 Amazon Machine Image (AMI)，然後選取單一 AMI 產品的其中一個授權類型：
   - 自備授權 (BYOL) – 使用者從外部取得授權的產品 AWS Marketplace。它可以是付費或免費授權。
   - 免費 – 可供訂閱者免費使用的產品。（他們仍會支付任何相關聯 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或其他 AWS 資源的費用。）
   - 按小時或按小時每年付費 – 買方按小時或以年度合約按小時支付的產品。 會根據 AMI 上的產品代碼 AWS 進行計量。
   - 每月付費 – 買方按月計費的產品 AWS。
   - 付費用量 – 軟體會直接針對您提供的值，以及四個用量類別之一收費：使用者、資料、頻寬或主機。您最多可以為產品定義 24 個維度。客戶仍會產生所有費用。
   - 具有合約定價的 AMI – 具有 AWS CloudFormation 堆疊的單一 AMI 產品或單一 AMI，買方需預付費用。
4. 自助式體驗會引導您完成建立 AWS Marketplace 清單的步驟。您必須輸入產品資訊（中繼資料）、產品部署詳細資訊 (AWS 區域、執行個體和 AMI 詳細資訊），以及公開優惠詳細資訊（價格、EULA、依國家/地區的可用性、EULA、退款）。您可以選擇將帳戶新增至允許清單以測試產品。完成每個步驟以移至程序的下一個步驟。

> **Note**
>
> 如果您需要在完成程序中的所有步驟之前結束工作階段，您可以使用儲存和結束函數，將完成的步驟提交到預備區域。這會為您提供的資訊建立請求，以供驗證。處理請求時，您無法編輯產品。請求成功後，您可以選擇繼續建立產品，以繼續建立您的產品。
> 失敗的請求表示因為驗證錯誤而未更新產品。這會顯示在您產品的請求日誌中。您可以選取請求來檢視錯誤，使用動作下的複製到新的 來更正錯誤，然後重新提交請求。當您繼續執行步驟時，您可以在上一個工作階段中儲存的步驟之後繼續執行步驟。若要更新先前步驟，請前往產品概觀頁面並提交變更請求，以更新您先前提交的步驟。

5. 輸入所有變更請求步驟的必要資訊後，請選擇提交。此提交會建立對 AWS Marketplace 目錄系統的請求，以驗證資訊，並在驗證通過時將產品釋出至有限狀態。當請求正在處理時，您無法繼續編輯產品。請求成功後，產品會移至有限狀態。

   - 當您的產品最初發佈時，只有您的 AWS 帳戶（您用來建立產品的）和 AWS Marketplace 賣方營運團隊的測試帳戶才能存取。如果您從伺服器產品頁面檢視產品，您可以選擇在 上檢視 AWS Marketplace，以檢視 AWS Marketplace 買方將出現在 中的產品詳細資訊。其他 AWS Marketplace 使用者看不到此詳細資訊清單。
   - 此功能可讓您在公開發佈產品之前測試產品（甚至發佈多個版本進行測試）。

6. 在有限狀態下測試您的產品，並確保其遵循 AWS Marketplace AMI 型產品要求和產品檢查清單。然後，若要請求將您的產品發佈至 Public，請選擇更新可見性。 AWS Marketplace 賣方營運團隊必須先檢閱您的產品，才能核准產品進入公有。

> **Note**
>
> 產品驗證和發佈是手動程序，由 AWS Marketplace 賣方營運團隊處理。如果沒有錯誤，可能需要 7-10 個工作天才能發佈您的初始產品版本。如需計時的詳細資訊，請參閱計時和期望。

如需準備和提交單一 AMI 產品和產品資訊的詳細資訊，請參閱 其他資源。

其他資源

如需準備產品資訊並提交以供發佈的詳細資訊，請參閱下列資源：

- 準備您的 產品 AWS Marketplace
- 提交您的產品以在 上發佈 AWS Marketplace

如需準備單一 AMI 產品以提交至 的詳細資訊 AWS Marketplace，請參閱下列資源：

- 建置 AMIs 以搭配 使用的最佳實務 AWS Marketplace
- 的 AMI 產品檢查清單 AWS Marketplace
- 的 AMI 型產品需求 AWS Marketplace

將 CloudFormation 範本新增至您的產品

AWS Marketplace 賣方可以使用 AWS CloudFormation 範本列出交付給 AWS Marketplace 買方的 AMI 型產品。將 CloudFormation 範本新增至 AMI 型產品可讓您的買方部署解決方案，而無需手動設定資源和相依性。您可以使用 範本來定義 產品的叢集或分散式架構，或選取不同的 AMI 組合或產品組態。單一 AMI 解決方案最多可包含三個 CloudFormation 範本。

CloudFormation 範本可設定為提供具有相關聯組態檔案和 Lambda 函數的單一 Amazon Machine Image (AMI)。此外，您必須包含每個範本的架構圖。

主題

- 準備您的 CloudFormation 範本
- 架構圖
- 轉換現有產品的 CloudFormation 範本
- 新增無伺服器應用程式元件

準備您的 CloudFormation 範本

若要建置 CloudFormation 範本，您必須符合範本先決條件，並提供必要的輸入和安全性參數。提交 CloudFormation 範本時，請使用下列各節中的準則。

範本先決條件

- 確認範本已成功透過針對您的產品啟用的所有 AWS CloudFormation 主控台啟動。 AWS 區域 您可以使用 TaskCat 工具來測試您的範本。
- CloudFormation 範本中包含AMIs 必須是您要發佈產品的 AMI，或 AWS 受管 AMI，例如最新的 Amazon Linux 2。請勿包含您或任何其他第三方擁有和共用的任何社群 AMI 或 AMI。若要使用 AWS 受管 AMI，請在 AWS Systems Manager 參數存放區中使用公有參數，而非硬式編碼 AMI IDs。例如，在指定 AMI ID 的 CloudFormation 範本中，您可以使用動態參考 `ImageId: '{{resolve:ssm:/aws/service/ecs/optimized-ami/amazon-linux-2/recommended/image_id}}'`。
- 建置範本，使其不依賴特定可用區域 (AZ) 中的使用。並非所有客戶都可以存取所有 AZs，而且 AZs 會針對不同的帳戶進行不同的映射。
- 如果您使用 Auto Scaling 群組建置叢集解決方案，我們建議您考慮擴展事件。新節點應會自動加入執行中的叢集。
- 即使是單一節點產品，我們建議使用 Auto Scaling 群組。

- 如果您的解決方案涉及多個執行個體的叢集，如果您想要低網路延遲、高網路輸送量，或在執行個體之間同時使用置放群組，請考慮使用置放群組。
- 為了方便 AWS Marketplace 團隊審查和向客戶提供透明度，我們建議您在 UserData 區段中新增評論。

AMI 詳細資訊的要求

> **Note**
>
> 如果您在賣方入口網站的伺服器產品頁面上使用 CloudFormation 建立 Amazon Machine Image (AMI) 或 AMI，並收到下載產品載入表單的提示，請the section called “使用產品載入表單的 AMI 詳細資訊需求”改為參閱 。

指定將 AMI 部署到 EC2 執行個體的資源`ImageId`屬性時，例如 AWS::EC2::Instance、AWS::AutoScaling::LaunchConfiguration 和 AWS::EC2::LaunchTemplate 資源，您必須參考範本參數。參數類型必須是 `AWS::EC2::Image::Id`、`AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`或 `String`。

您可以將此範本參數命名為任何有效的參數名稱。 會將您的範本 AWS Marketplace 複製到自己的 Amazon S3 儲存貯體，並以 AWS Systems Manager 參數存放區參數取代指定的參數。 AWS Marketplace 也會更新描述和限制條件文字，讓部署範本的買方清楚正確的值。當買方部署您的範本時，該參數會解析為您已發佈產品的 AWS 區域特定 AMI ID。

下列範本範例說明使用內部函數 Ref 參考範本參數的 `ImageId` 屬性。

YAML 範例：

```
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example1234567890
Resources:
  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
```

JSON 範例：

```
{
  "Parameters": {
    "ImageId": {
      "Type": "AWS::EC2::Image::Id",
      "Default": "ami-example1234567890"
    }
  },
  "Resources": {
    "MyInstance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": {
          "Ref": "ImageId"
        }
      }
    }
  }
}
```

如果您要在巢狀堆疊內而非根堆疊內部署 EC2 執行個體，AMI ID 必須動態地從根堆疊繼承其值。編輯您的根和巢狀堆疊，以便在根堆疊中，設定範本參數的值會覆寫此巢狀堆疊中使用的 AMI ID。

使用產品載入表單的 AMI 詳細資訊需求

> **Note**
>
> 當您在賣方入口網站的伺服器產品頁面上使用 CloudFormation 建立 Amazon Machine Image (AMI) 或 AMI，且未立即提示您下載產品載入表單時，請參閱 the section called “AMI 詳細資訊的要求” 。

AMIs 必須位於每個區域的映射表中。 AWS Marketplace 團隊會在複製 AMI IDs 之後更新這些 ID。您的來源 AMI 必須位於 `us-east-1`區域。其他區域可以使用預留位置。

YAML 範例：

```
Mappings:
  RegionMap:
      us-east-1:
          ImageId: ami-0123456789abcdef0
      us-west-1:
```

```
          ImageId: ami-xxxxxxxxxxxxxxxxx
      eu-west-1:
          ImageId: ami-xxxxxxxxxxxxxxxxx
      ap-southeast-1:
          ImageId: ami-xxxxxxxxxxxxxxxxx
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap
        - RegionMap
        - !Ref AWS::Region
        - ImageId
```

巢狀堆疊範本的需求

> **Note**
>
> 本節僅適用於不使用產品載入表單的定價模型。對於使用產品載入表單的定價模型，巢狀堆疊TemplateURL屬性只允許固定字串。

如果您的範本包含巢狀堆疊，巢狀堆疊資源的 TemplateURL 屬性必須參考 Amazon S3 儲存貯體名稱、儲存貯體區域和 Amazon S3 物件金鑰字首的範本參數。儲存貯體名稱的參數名稱必須是 MPS3BucketName，儲存貯體區域必須是 MPS3BucketRegion，而物件金鑰字首必須是 MPS3KeyPrefix。

設定這些參數的預設值，以對應至存放巢狀範本的 Amazon S3 儲存貯體。所有巢狀範本都必須可公開存取。當您提交範本進行發佈時， 會將範本 AWS Marketplace 複製到自己的 Amazon S3 儲存貯體，並修改這三個參數的屬性，讓預設值和允許值設定為與存放複本的位置對應。 AWS Marketplace 也會更新描述和限制文字，讓部署範本的買方清楚正確的值。

如果您有多層巢狀堆疊，則必須設定建立其他巢狀堆疊的所有巢狀堆疊，以便 TemplateURL 屬性動態繼承根堆疊中的 Amazon S3 儲存貯體名稱、Amazon S3 儲存貯體區域和 Amazon S3 物件金鑰的值。編輯您的根和巢狀堆疊，以便在根堆疊中設定範本參數 MPS3BucketName、 和 的值MPS3BucketRegion，MPS3KeyPrefix覆寫在此巢狀堆疊中使用的 URL 中的個別值，以建立其他巢狀堆疊。

下列範本範例說明使用內部函數 Fn：：Sub 參考範本參數的 TemplateURL 屬性。

YAML 範例：

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: AWS Marketplace Parameters
        Parameters:
          - ImageId
          - MPS3BucketName
          - MPS3BucketRegion
          - MPS3KeyPrefix
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example1234567890
    Description: The AMI that will be used to launch EC2 resources.
  MPS3BucketName:
    Type: String
    Default: sellerbucket
    Description: Name of the S3 bucket for your copy of the nested templates.
  MPS3BucketRegion:
    Type: String
    Default: us-east-1
    Description: AWS Region where the S3 bucket for your copy of the nested templates
 is hosted.
  MPS3KeyPrefix:
    Type: String
    Default: sellerproductfolder/
    Description: S3 key prefix that is used to simulate a folder for your copy of the
 nested templates.
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml
```

JSON 範例：

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Metadata": {
        "AWS::CloudFormation::Interface": {
            "ParameterGroups": [
                {
                    "Label": {
                        "default": "AWS Marketplace Parameters"
                    },
                    "Parameters": [
                        "ImageId",
                        "MPS3BucketName",
                        "MPS3BucketRegion",
                        "MPS3KeyPrefix"
                    ]
                }
            ]
        }
    },
    "Parameters": {
        "ImageId": {
            "Type": "AWS::EC2::Image::Id",
            "Default": "ami-example1234567890",
            "Description": "The AMI that will be used to launch EC2 resources."
        },
        "MPS3BucketName": {
            "Type": "String",
            "Default": "sellerbucket",
            "Description": "Name of the S3 bucket for your copy of the nested
templates."
        },
        "MPS3BucketRegion": {
            "Type": "String",
            "Default": "us-east-1",
            "Description": "AWS Region where the S3 bucket for your copy of the nested
templates is hosted."
        },
        "MPS3KeyPrefix": {
            "Type": "String",
            "Default": "sellerproductfolder/",
            "Description": "S3 key prefix that is used to simulate a folder for your
copy of the nested templates."
        }
```

```
    },
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": {
                    "Ref": "ImageId"
                }
            }
        },
        "NestedStack": {
            "Type": "AWS::CloudFormation::Stack",
            "Properties": {
                "TemplateURL": {
                    "Fn::Sub": "https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml"
                }
            }
        }
    }
}
```

> **Note**
>
> AWS::CloudFormation::Interface 用於定義當買方部署您的範本時，如何在 AWS CloudFormation 主控台中對參數進行分組和排序。

範本輸入參數

- 範本的輸入參數不得包含 AWS Marketplace 客戶的 AWS 登入資料（例如密碼、公有金鑰、私有金鑰或憑證）。
- 對於密碼等敏感輸入參數，請選擇 NoEcho 屬性並啟用更強大的規則表達式。對於其他輸入參數，請設定最常見的輸入以及適當的協助程式文字。
- 在可用的情況下，對輸入使用 AWS CloudFormation 參數類型。
- 使用 `AWS::CloudFormation::Interface`將輸入參數分組和排序。
- 請勿為下列輸入參數設定任何預設值：

> **Note**
>
> 客戶必須提供這些做為輸入參數。

- 預設 CIDR 範圍，允許從公有網際網路傳入遠端存取連接埠
- 預設 CIDR 範圍，允許從公有網際網路傳入資料庫連線連接埠
- 使用者或資料庫的預設密碼

網路和安全性參數

- 確定預設 SSH 連接埠 (22) 或 RDP 連接埠 (3389) 未開啟至 0.0.0.0。
- 建議您使用適當的存取控制清單 (ACLs) 和安全群組來建置 VPC，而不是使用預設的虛擬私有雲端 (VPC)。
- 您的範本無法向使用者請求長期存取金鑰，也無法建立他們來存取 AWS 資源。如果您的 AMI 應用程式需要存取買方帳戶中 AWS 的服務，則必須使用 Amazon EC2 的 IAM 角色。
- 設定 IAM 角色和政策以授予最低權限，並僅在絕對必要時啟用寫入存取。例如，如果您的應用程式只需要 `S3:GET`、 `PUT`和 `DELETE`操作，請僅指定這些動作。`S3:*` 在這種情況下，我們不建議使用 。

收到您的範本後， 會 AWS Marketplace 驗證產品組態和資訊，並提供任何必要修訂的意見回饋。

架構圖

您必須為每個範本提供架構圖。若要進一步了解圖表，請參閱什麼是架構圖表？

圖表必須符合下列條件：

- 說明 上的標準部署 AWS。
- 在邏輯上描述資源的部署位置。例如，Amazon EC2 執行個體之類的資源位於正確的子網路中。
- 針對透過 AWS CloudFormation 範本 AWS 服務 部署的每個 ，使用最新的 AWS 產品圖示。若要下載目前一組架構圖示，請參閱AWS 架構圖示。
- 包含 AWS CloudFormation 範本部署之所有服務的中繼資料。
- 包含 AWS CloudFormation 範本部署的所有網路、VPCs 和子網路。

- 顯示整合點，包括第三方資產、APIs 和內部部署、混合資產。
- 圖表的大小必須為 1100 x 700 像素。維持原始圖表比例，而不進行拉伸或裁切。

轉換現有產品的 CloudFormation 範本

> **Note**
>
> 本節適用於具有具有 CloudFormation 產品之現有 AMI 的賣方，該產品使用產品載入表單發佈其範本，現在想要更新該範本而不使用產品載入表單。如果您要發佈新產品，請參閱準備 CloudFormation 範本。
> 如果您在賣方入口網站的伺服器產品頁面上使用 CloudFormation 建立 Amazon Machine Image (AMI) 或 AMI，並收到下載產品載入表單的提示，請參閱 the section called “使用產品載入表單的 AMI 詳細資訊需求”。

如果您想要使用自助式體驗來更新先前使用產品載入表單發佈的現有產品，您必須變更現有的 CloudFormation 範本。

下表說明使用產品載入表單與自助式體驗之間的差異：

	產品載入表單	自助式體驗
EC2 資源的**`ImageId`**屬性值	參考 AMI ID 的映射表。如需詳細資訊，請參閱the section called “使用產品載入表單的 AMI 詳細資訊需求”。	參考 AMI ID 的範本參數。如需詳細資訊，請參閱the section called “AMI 詳細資訊的要求”。
巢狀堆疊的**`TemplateURL`** 屬性值	必須是固定字串，且無法使用內部函數。	可以使用內部 函數來動態。必須參考一組範本參數。如需詳細資訊，請參閱the section called “巢狀堆疊範本的需求”

下列範例範本說明使用產品載入表單來發佈範本的現有產品範例。在此範例中，AMI ID 為 ，`ami-example123456`而巢狀範本位於位置 的賣方 S3 儲存貯體中`https://sellerbucket.s3.us-east-1.amazonaws.com/sellerproductfolder/nested-template.yaml`。

使用產品載入表單發佈的 YAML 範例：

```
AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-example123456
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap
        - RegionMap
        - !Ref AWS::Region
        - AMI
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: https://sellerbucket.s3.us-east-1.amazonaws.com/sellerproductfolder/
nested-template.yaml
```

使用產品載入表單發佈的 JSON 範例：

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Mappings": {
        "RegionMap": {
            "us-east-1": {
                "AMI": "ami-example123456"
            }
        }
    },
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": {
                    "Fn::FindInMap": [
                        "RegionMap",
                        {
                            "Ref": "AWS::Region"
                        },
                        "AMI"
                    ]
                }
```

```
            }
        },
        "NestedStack": {
            "Type": "AWS::CloudFormation::Stack",
            "Properties": {
                "TemplateURL": "https://sellerbucket.s3.us-east-1.amazonaws.com/
sellerproductfolder/nested-template.yaml"
            }
        }
    }
}
```

下列範本範例說明使用自助式體驗更新產品所需的變更。

以自助式體驗發佈的 YAML 範例：

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: AWS Marketplace Parameters
        Parameters:
          - ImageId
          - MPS3BucketName
          - MPS3BucketRegion
          - MPS3KeyPrefix
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example123456
    Description: The AMI that will be used to launch EC2 resources.
  MPS3BucketName:
    Type: String
    Default: sellerbucket
    Description: Name of the S3 bucket for your copy of the nested templates.
  MPS3BucketRegion:
    Type: String
    Default: us-east-1
    Description: AWS Region where the S3 bucket for your copy of the nested templates
 is hosted.
  MPS3KeyPrefix:
    Type: String
    Default: sellerproductfolder/
```

```
    Description: S3 key prefix that is used to simulate a folder for your copy of the
 nested templates.
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml
```

以自助式體驗發佈的 JSON 範例：

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Metadata": {
        "AWS::CloudFormation::Interface": {
            "ParameterGroups": [
                {
                    "Label": {
                        "default": "AWS Marketplace Parameters"
                    },
                    "Parameters": [
                        "ImageId",
                        "MPS3BucketName",
                        "MPS3BucketRegion",
                        "MPS3KeyPrefix"
                    ]
                }
            ]
        }
    },
    "Parameters": {
        "ImageId": {
            "Type": "AWS::EC2::Image::Id",
            "Default": "ami-example123456",
            "Description": "The AMI that will be used to launch EC2 resources."
        },
        "MPS3BucketName": {
            "Type": "String",
            "Default": "sellerbucket",
```

```
            "Description": "Name of the S3 bucket for your copy of the nested
 templates."
        },
        "MPS3BucketRegion": {
            "Type": "String",
            "Default": "us-east-1",
            "Description": "AWS Region where the S3 bucket for your copy of the nested
 templates is hosted."
        },
        "MPS3KeyPrefix": {
            "Type": "String",
            "Default": "sellerproductfolder/",
            "Description": "S3 key prefix that is used to simulate a folder for your
 copy of the nested templates."
        }
    },
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": {
                    "Ref": "ImageId"
                }
            }
        },
        "NestedStack": {
            "Type": "AWS::CloudFormation::Stack",
            "Properties": {
                "TemplateURL": {
                    "Fn::Sub": "https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml"
                }
            }
        }
    }
}
```

新增無伺服器應用程式元件

> **Important**
>
> AWS Marketplace 不再支援使用從中部署資源的 CloudFormation 範本發佈新產品 AWS Serverless Application Repository。賣方可以繼續使用從 部署資源的 CloudFormation 範本來發佈現有產品， AWS Serverless Application Repository 直到要宣布的未來日期為止。

您可以建立產品，其中包含使用 AWS CloudFormation 範本交付的 Amazon Machine Image (AMI)，並將無伺服器元件整合到產品中。例如，您可以使用設定為控制器伺服器的 AMI 建立產品，並將其做為 AWS CloudFormation 堆疊交付。用來建立堆疊的 AWS CloudFormation 範本可以包含 定義，以設定由伺服器中的事件觸發的 AWS Lambda 函數。當您使用此方法來設計產品時，您可以簡化架構，並讓買方更容易啟動。這種方法也可以讓您更輕鬆地更新產品。以下各節說明如何建立並提供此類產品。

如需為您的產品建立 AMI 的詳細資訊，請參閱 中的 AMI 型產品 AWS Marketplace。如需完成產品 AWS CloudFormation 範本的相關資訊，請參閱 將 CloudFormation 範本新增至您的產品。

當您定義無伺服器應用程式時，您可以使用存放在 中的 AWS Serverless Application Model (AWS SAM) 範本 AWS Serverless Application Repository。 AWS SAM 是用於建置無伺服器應用程式的開放原始碼架構。在部署期間， AWS Serverless Application Model 會將語法 AWS SAM 轉換並展開為 AWS CloudFormation 語法。 AWS Serverless Application Repository 是無伺服器應用程式的受管儲存庫。您可以儲存和共用可重複使用的應用程式，讓買方可以組合和部署無伺服器架構。

> **Note**
>
> - AWS Marketplace 在建立您的清單之前， 會檢閱並驗證您的產品。如果有在列出優惠之前必須解決的問題，我們會傳送電子郵件訊息給您。
> - 在履行訂閱的過程中，我們會將 AMIs、無伺服器應用程式和 AWS CloudFormation 範本複製到每個 中擁有 AWS Marketplace的儲存庫 AWS 區域。當買方訂閱您的產品時，我們會授予他們存取權，並在您更新軟體時通知他們。

主題

- 步驟 1：建立無伺服器應用程式
- 步驟 2：將您的應用程式發佈至儲存庫
- 步驟 3：建立 CloudFormation 範本

- 步驟 4：提交您的 CloudFormation 範本和組態檔案
- 步驟 5：更新您的 AWS Serverless Application Repository 應用程式許可
- 步驟 6：共用您的 AMI
- 步驟 7：使用 AMI 和無伺服器應用程式提交您的 CloudFormation 產品

步驟 1：建立無伺服器應用程式

您的第一步是封裝用於建立無伺服器應用程式的 AWS Lambda 函數。您的應用程式是 Lambda 函數、事件來源和其他資源的組合，這些資源可共同執行任務。無伺服器應用程式可以像一個 Lambda 函數一樣簡單，也可以包含具有其他資源的多個函數，例如 APIs、資料庫和事件來源映射。

使用 AWS SAM 為您的無伺服器應用程式定義模型。如需屬性名稱和類型的說明，請參閱 GitHub 上 AWS Labs 中的 AWS::Serverless::Application。以下是具有單一 Lambda 函數和 AWS Identity and Access Management (IAM) 角色的 AWS SAM 範本範例。

```
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"
```

步驟 2：將您的應用程式發佈至儲存庫

若要發佈應用程式，您需先上傳應用程式程式碼。將程式碼成品（例如 Lambda 函數、指令碼、組態檔案）存放在您帳戶擁有的 Amazon S3 儲存貯體中。當您上傳應用程式時，它最初會設定為私有，這表示它僅適用於 AWS 帳戶 建立它的 。您必須建立 IAM 政策，授予存取您上傳成品的 AWS Serverless Application Repository 許可。

將無伺服器應用程式發佈至無伺服器應用程式儲存庫

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選擇您用來封裝應用程式的 Amazon S3 儲存貯體。
3. 選擇許可索引標籤標籤。
4. 選擇 Bucket Policy (儲存貯體政策)。
5. 複製並貼上下列範例政策陳述式。

> **Note**
>
> 範例政策陳述式會產生錯誤，直到 `aws:SourceAccount`和 的值`Resource`在下列步驟中更新為止。

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service":  "serverlessrepo.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition" : {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

a. 將Resource屬性值中的 amzn-s3-demo-bucket 取代為儲存貯體的儲存貯體名稱。

b. 將 Condition元素中的 *123456789012* 取代為您的 AWS 帳戶 ID。Condition 元素可確保 AWS Serverless Application Repository 僅具有從指定 存取應用程式的許可 AWS 帳戶。

6. 選擇儲存。
7. 在 開啟 AWS Serverless Application Repository 主控台https://console.aws.amazon.com/serverlessrepo。
8. 在我的應用程式頁面上，選擇發佈應用程式。
9. 視需要完成必要欄位和任何選用欄位。必要欄位為：
   - Application name (應用程式名稱)
   - 作者
   - Description
   - 來源碼 URL
   - SAM 範本
10. 選擇發佈應用程式。

發佈應用程式的後續版本

1. 在 開啟 AWS Serverless Application Repository 主控台https://console.aws.amazon.com/serverlessrepo。
2. 在導覽窗格中，從我的應用程式選擇應用程式。
3. 選擇 Publish new version (發佈新版本)。

如需詳細資訊，請參閱使用 CLI AWS SAM 發佈無伺服器應用程式。

步驟 3：建立 CloudFormation 範本

若要建置 CloudFormation 範本，您必須符合範本先決條件，並提供必要的輸入和安全參數。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的範本結構。

在 CloudFormation 範本中，您可以參考無伺服器應用程式和 AMI。您也可以在根範本和巢狀範本中使用巢狀 CloudFormation 範本和參考無伺服器應用程式。若要參考無伺服器應用程式，請使用 AWS SAM 範本。您可以從 自動產生應用程式的 AWS SAM 範本 AWS Serverless Application Repository。以下是範例範本。

```
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 範本包含下列元素：

- `ApplicationID` – 應用程式的 Amazon Resource Name (ARN)。此資訊位於 的我的應用程式區段 AWS Serverless Application Repository。
- `SemanticVersion` – 無伺服器應用程式的版本。您可以從 的我的應用程式區段找到此項目 AWS Serverless Application Repository。
- `Parameter` （選用） – 應用程式參數。

> **Note**
>
> 對於 `ApplicationID`和 `SemanticVersion`，不支援內部函數。您必須硬式編碼這些字串。複製時，`ApplicationID`會更新 AWS Marketplace。

如果您打算參考 CloudFormation 範本中的組態和指令碼檔案，請使用下列格式。對於巢狀範本 (`AWS::Cloudformation::Stack`)，僅支援`TemplateURLs`沒有內部函數的 。記下範本中的`Parameters`內容。

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  Name: Seller test product
Parameters:
  CFTRefFilesBucket:
    Type: String
    Default: "seller-bucket"
  CFTRefFilesBucketKeyPrefix:
    Type: String
    Default: "cftsolutionFolder/additionCFfiles"
Resources:
  TestEc2:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        addCloudAccount:
          files:
```

```
            /etc/cfn/set-aia-settings.sh:
              source:
                Fn::Sub:
                - https://${CFTRefFilesBucket}.${S3Region}amazonaws.com/
${CFTRefFilesBucketKeyPrefix}/sampleScript.sh
                - S3Region:
                    !If
                    - GovCloudCondition
                    - s3-us-gov-west-1
                    - s3
              owner: root
              mode: '000700'
              authentication: Amazon S3AccessCreds
    ..
    ..
    ..
  SampleNestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: 'https://sellerbucket.s3.amazon.com/sellerproductfolder/
nestedCft.template'
      Parameters:
        SampleParameter: 'test'
Transform: AWS::Serverless-2016-10-31
```

步驟 4：提交您的 CloudFormation 範本和組態檔案

若要提交 CloudFormation 範本和組態和指令碼檔案，請授予 AWS Marketplace 許可來讀取存放這些檔案的 Amazon S3 儲存貯體。若要這樣做，請更新您的儲存貯體政策，以包含下列許可。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service":  "assets.marketplace.amazonaws.com"
            },
            "Action": ["s3:GetObject", "s3:ListBucket"],
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket",
```

```
                "arn:aws:s3:::amzn-s3-demo-bucket/*"]
        }
    ]
}
```

步驟 5：更新您的 AWS Serverless Application Repository 應用程式許可

若要提交您的 AWS Serverless Application Repository 應用程式 AWS Marketplace，您必須授予讀取應用程式的 AWS Marketplace 許可。若要這樣做，請將許可新增至與無伺服器應用程式相關聯的政策。有兩種方式可以更新您的應用程式政策：

- 前往 AWS Serverless Application Repository。從清單中選擇無伺服器應用程式。選取共用索引標籤，然後選擇建立陳述式。在陳述式組態頁面上，**assets.marketplace.amazonaws.com**在帳戶 ID 欄位中輸入下列服務主體 。然後選擇 Save (儲存)。
- 使用下列 AWS CLI 命令來更新您的應用程式政策。

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements Principals=assets.marketplace.amazonaws.com,Actions=Deploy
```

步驟 6：共用您的 AMI

建立並提交至 的所有 AMIs AWS Marketplace 都必須遵守所有產品政策。自助式 AMI 掃描可在 中使用 AWS Marketplace 管理入口網站。使用此功能，您可以啟動 AMIs的掃描。您會在單一位置快速收到掃描結果（通常不到一小時），並提供明確的意見回饋。成功掃描 AMI 之後，請上傳您的產品載入表單，提交 AMI 以供 AWS Marketplace 賣方營運團隊處理。

步驟 7：使用 AMI 和無伺服器應用程式提交您的 CloudFormation 產品

提交產品之前，請記住下列事項：

- 您必須為每個範本提供架構圖。圖表必須針對透過 CloudFormation 範本部署的每個 AWS 服務使用 AWS 產品圖示。此外，圖表必須包含 服務的中繼資料。若要下載我們的官方 AWS 架構圖示，請參閱AWS 架構圖示。
- 向買方顯示的每個範本的基礎設施成本估算是根據您使用 AWS 定價計算器提供的估算。在預估中，包含要部署做為範本一部分的服務清單，以及典型部署的預設值。

- 完成產品載入表單。您可以從 找到產品載入表單 AWS Marketplace 管理入口網站。單一 AMI 產品和多個 AMI 產品需要不同的產品載入表單。在產品載入表單中，您將為 CloudFormation 範本提供公有 URL。CloudFormation 範本必須以公有 URL 的形式提交。
- 使用 AWS Marketplace 管理入口網站 提交您的清單。從資產中，選擇檔案上傳，連接您的檔案，然後選擇上傳。在我們收到您的範本和中繼資料後， 會 AWS 開始處理您的請求。

提交清單後， 會 AWS Marketplace 檢閱並驗證產品載入表單。此外， AWS Marketplace 區域化 AMIs 和無伺服器應用程式，並代表您更新 AWS CloudFormation 範本的區域映射。如果發生任何問題， AWS Marketplace 賣方營運團隊會透過電子郵件與您聯絡。

以 AWS Marketplace 賣方身分管理 AMI 型產品

身為 AWS Marketplace 賣方，您可以管理和更新單一 Amazon Machine Image AMIs) 產品。AMI 型產品包含一組或多個版本的軟體，以及有關產品整體的中繼資料。當您建立產品時，您可以在 中設定其屬性 AWS Marketplace，包括您的產品名稱、描述和定價。您也可以判斷產品的適當類別並新增關鍵字，讓您的產品出現在相關搜尋中。建立單一 AMI 產品後，您可以提交變更請求，以對您的產品或版本進行變更。

您可以為 AMI 型產品請求的變更類型包括：

- 更新向買方顯示的產品資訊。
- 更新向買方顯示的版本資訊。
- 新增您產品的新版本。
- 限制 版本，讓新的買方無法再存取該版本。
- 更新 AWS 區域 產品可用的 。
- 更新產品的定價和執行個體類型。
- 從中移除產品 AWS Marketplace。

下列主題顯示 來管理和更新單一 AMI 產品。

主題

- 在 中建立 AMI 型產品的變更請求 AWS Marketplace
- 更新 AMI 型產品可見性
- 新增和限制 的 AMI 執行個體 AWS Marketplace

- 在 上管理 AMI 型產品的版本 AWS Marketplace
- 更新 上的 AMI 型產品資訊 AWS Marketplace
- 依 AWS 區域 和國家/地區管理 AMI 型產品可用性
- 更新以 AMI 為基礎的 AWS Marketplace 產品的最終使用者授權合約 (EULA)
- 在 上更新以 AMI 為基礎的產品的退款政策 AWS Marketplace
- 提供 AMI 的 AWS Marketplace 存取權
- 從 移除產品 AWS Marketplace
- 對 上變更請求的常見錯誤進行故障診斷 AWS Marketplace

在 中建立 AMI 型產品的變更請求 AWS Marketplace

若要變更 中的產品或版本 AWS Marketplace，您可以透過 提交變更請求 AWS Marketplace 管理入口網站。變更請求會新增至佇列，且可能需要幾分鐘到幾天的時間才能解決，視請求類型而定。您可以在 中查看請求的狀態 AWS Marketplace 管理入口網站。本主題提供您可用來在 中建立單一 AMI 產品變更請求的程序 AWS Marketplace，包括使用自助式體驗的選項。

您可以為下列情況建立變更請求：

- 您已儲存進行中的步驟，但尚未完成整個程序，同時使用自助式體驗來建立單一 AMI 產品清單。若要完成其餘步驟，請建立變更請求。
- 您想要修改處於有限或公有狀態之產品的產品資訊。若要更新資訊，您可以建立變更請求。如需您可以請求 AMI 型產品變更類型的詳細資訊，請參閱建立變更請求。

> **Note**
>
> 除了 之外 AWS Marketplace 管理入口網站，您也可以使用AWS Marketplace 目錄 API 建立變更請求。

主題

- 使用自助服務建立變更請求
- 建立變更請求
- 取得變更請求的狀態
- 其他資源

使用自助服務建立變更請求

若要修改版本或產品資訊，您可以在 中建立變更請求 AWS Marketplace 管理入口網站。變更請求是您用來變更產品的自助式清單的建置區塊。每次您選取儲存並退出步驟，或選取提交以進行任何更新時，您都會提出變更請求。您可以在 AWS Marketplace 管理入口網站 Request 索引標籤上找到您的請求。

使用自助服務建立變更請求

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，並登入您的賣方帳戶，然後前往伺服器產品頁面。
2. 在伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中選擇選項。
4. 在您提出變更請求後，系統有等待時間處理您的請求，反映在審核中。當請求完成時，將導致成功或失敗。

   - 提交請求後，它會開始處理這些狀態：檢閱中、準備變更和套用變更。
   - 成功表示已處理請求的變更，且變更會在系統中反映。
   - 失敗表示請求發生錯誤，因此不會處理變更。如果狀態為失敗，您可以選取請求，以尋找錯誤代碼，提供如何修正錯誤的建議。此時，您可以對錯誤進行故障診斷，並建立新的變更請求。若要讓程序更快，您可以選擇複製到新請求，以複製失敗請求的詳細資訊。然後，您可以進行調整並重新提交請求。

建立變更請求

> **Important**
>
> 在 2023 年 6 月 15 日， AWS Marketplace 將停止下列程序。2023 年 6 月 15 日之後，請使用 the section called “使用自助服務建立變更請求” 程序。

若要修改版本或產品資訊，您可以在 中建立變更請求 AWS Marketplace 管理入口網站。

建立變更請求

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，並登入您的賣方帳戶，然後前往伺服器產品頁面。

2. 在伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中選擇選項。

   對於大多數變更請求，您只需在使用者介面中填寫表單並提交。不過，對於某些變更，您必須下載、完成，然後上傳產品載入表單 (PLF)。這是試算表，其中包含表格供您填寫所需資訊。當您選擇其中一個變更請求時，系統會提示您為嘗試建立的請求下載正確的 PLF。PLF 會預先填入現有產品詳細資訊的資訊。您可以將完成的 PLF 上傳至 AWS Marketplace 管理入口網站 檔案上傳頁面。

   > **Note**
   >
   > 我們強烈建議您下載並使用最新的 PLF。表單會定期更新新資訊，包括執行個體類型和 AWS 區域 可用的執行個體類型。您可以從伺服器產品頁面找到產品的最新 PLF，方法是選取產品，然後選擇下載產品載入表單。

如需變更請求狀態的詳細資訊，請參閱 the section called “取得變更請求的狀態”。如需變更請求潛在問題的洞見，請參閱 對 上變更請求的常見錯誤進行故障診斷 AWS Marketplace。

取得變更請求的狀態

> **Important**
>
> 在 2023 年 6 月 15 日， AWS Marketplace 將停止下列程序。自助服務體驗不再需要此程序。

提交變更請求後，您可以從 的伺服器產品頁面的請求索引標籤中查看請求的狀態 AWS Marketplace 管理入口網站。狀態可以是下列任何一項：

- 審核中表示正在審核您的請求。有些請求需要 AWS Marketplace 團隊手動審核，但大多數請求會在系統中自動審核。
- 成功表示您的請求已完成。您的產品或版本已依您的要求更新。
- 必要動作表示您需要更新您的請求，以修正問題或回答有關請求的問題。選取請求以查看詳細資訊，包括任何問題。
- 失敗表示請求發生錯誤，您應該使用相同的資料建立新的變更請求。

其他資源

如需特定更新類型變更請求的詳細資訊，請參閱下列資源：

- 更新 上的 AMI 型產品資訊 AWS Marketplace
- 更新版本資訊
- 新增新版本
- 限制版本

更新 AMI 型產品可見性

身為 AWS Marketplace 賣方，您可以更新您的產品可見性，以變更哪些買方可以在其中檢視您的單一 AMI 產品 AWS Marketplace。如果可見性狀態設為公開，則所有 AWS Marketplace 買方都可看見您的產品。如果您的產品可見性設為有限，則只有您允許清單 AWS 帳戶 IDs 才會顯示您的產品。您也可以管理和更新此可查看您產品的 AWS 帳戶 IDs 允許清單。下列各節說明如何更新您的產品可見性和有限可見性允許清單。

主題

- 更新產品可見性
- 更新允許清單（預覽帳戶）

更新產品可見性

若要變更哪些買方可以在其中檢視您的產品 AWS Marketplace，您可以使用更新可見性。

更新可見性

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新可見性。

> **Note**
>
> 您可以使用此變更請求，請求將產品從有限狀態移至公有狀態。不過，變更請求必須經過 AWS Marketplace 賣方營運團隊核准程序，才能移至公有。

4. 選擇提交變更請求以提交您的請求以供檢閱。
5. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

更新允許清單 （預覽帳戶）

若要變更可在有限狀態下看到您產品的 AWS 帳戶 IDs 清單，請使用更新允許清單。

更新允許清單

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新允許清單。目前清單會隨附目前允許列出的帳戶清單。
4. 新增偏好用於可見 AWS 帳戶 IDs，並以逗號分隔 IDs。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

新增和限制 的 AMI 執行個體 AWS Marketplace

身為 AWS Marketplace 賣方，您可以管理購買者可以用於單一 Amazon Machine Imagine (AMI) 產品的執行個體。您可以為購買者可以使用的單一 AMI 產品新增執行個體。同樣地，如果您想要防止新買家使用來自特定執行個體的單一 AMI 產品，您可以限制執行個體。

如需 Amazon EC2 執行個體類型的詳細資訊，請參閱《Amazon EC2 使用者指南》中的可用執行個體類型。

下列各節說明如何新增和限制執行個體。

主題

- 新增執行個體
- 限制執行個體

新增執行個體

您可以新增新執行個體，供買方做為單一 AMI 使用。

新增執行個體

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇新增執行個體。
4. 選取執行個體架構。
5. 從可用執行個體清單中選取要新增的執行個體類型。
6. 選擇提交請求以提交您的請求以供檢閱。
7. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

> **Note**
>
> - 如果您目前的定價模型不是免費的，或使用自攜授權 (BYOL) 模型，您還必須新增價格。
> - 如果您使用新執行個體的價格建立新增執行個體，或更新定價來提高價格，則自您進行變更之日起 90 天內，您無法使用自助式新增執行個體。若要進行這些變更，請聯絡AWS Marketplace 賣方營運團隊。
> - 當您新增對新執行個體類型的支援時，已訂閱您產品的私有優惠的客戶將無法自動啟動新新增的執行個體。您必須使用您希望客戶存取的執行個體建立另一個私有優惠。接受新優惠後，客戶可以啟動新增的執行個體。只要執行個體包含在私有優惠中，未來訂閱您產品的客戶也可以啟動執行個體。如需如何建立新的私有優惠的詳細資訊，請參閱本指南在 中修訂協議 AWS Marketplace稍後的 。

限制執行個體

若要防止新買家使用 AMI 產品的執行個體，您可以限制執行個體。如有需要，您可以稍後再新增執行個體。受限制執行個體上單一 AMI 的現有使用者可以在訂閱期間繼續使用 區域中的產品。

限制執行個體

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇限制執行個體。

4. 選取您要限制的執行個體，然後選擇限制。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

> **Note**
>
> 如果核取方塊為陰影，這表示執行個體會與一到多個版本建立關聯，做為建議的執行個體類型。若要限制此類執行個體，請使用更新版本來選擇不同的建議執行個體類型。變更請求完成且您想要限制的執行個體不再是建議的執行個體類型後，您可以返回限制執行個體來限制您選擇的執行個體。

在 上管理 AMI 型產品的版本 AWS Marketplace

當您建立以 Amazon Machine Image (AMI) 為基礎的產品時 AWS Marketplace，您會包含特定版本的軟體。的 AMI 型產品的生命週期 AWS Marketplace 不會在您發佈第一個版本後結束。您應該讓您的產品與軟體的新版本保持up-to-date狀態。下列各節說明如何管理您的版本，包括更新版本資訊（例如描述和日期）、新增新版本，以及限制對先前版本的存取。

主題

- 更新版本資訊
- 新增新版本
- 限制版本

更新版本資訊

建立版本後，透過修改與該版本相關的資訊，將更新的資訊提供給買方會很有幫助。例如，如果您計劃在發行 1.1 版之後限制 1.0 版，您可以更新 1.0 版的描述，將買方導向 1.1 版，並將限制該版本的日期。您可以從 更新版本資訊 AWS Marketplace 管理入口網站。

更新版本資訊

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往目前伺服器產品頁面，在伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新版本資訊。

4. 在更新版本頁面上，選取您要更新的版本。
5. 更新您需要修改的下列任何資訊：
   - 版本備註
   - 使用說明
   - 64 位元 (x86) Amazon Machine Image (AMI) – 用量和安全群組的詳細資訊
6. 選取提交。
7. 確認請求顯示在請求索引標籤上，且狀態為審核中。

> **Note**
>
> 您無法使用此程序來更新版本標題，或與版本相關聯的 AMI。反之，請建立新的版本並限制先前的版本。

您可以隨時從伺服器產品頁面的請求索引標籤檢查請求的狀態。如需詳細資訊，請參閱取得變更請求的狀態。

新增新版本

當您變更產品、基礎映像或任何其他需要修改產品的 AMI 時，您可以新增產品版本。從 新增您產品的新版本 AWS Marketplace 管理入口網站。

> **Note**
>
> 如需為 建立 AMI 的詳細資訊 AWS Marketplace，請參閱 建置 AMIs 以搭配 使用的最佳實務 AWS Marketplace。

新增新版本

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇新增新版本。新增版本表單隨即出現，並填入最新版本的資訊。

> **Note**
>
> 您也可以選擇測試「新增版本」來掃描您的 AMI 和 AWS CloudFormation 範本（無需新增新版本）。如需詳細資訊，請參閱掃描您的 AMI 以發佈需求。

4. 在版本資訊區段中，提供下列資訊：
   - 版本標題 – 輸入有效的字串（例如 *1.1* 或 *2.0 ###*它在整個產品中必須是唯一的。
   - 版本備註 – 輸入文字來描述此版本的詳細資訊。
5. 在交付選項區段中，選取買方如何部署解決方案，可以是 AMI（獨立）、AMI 搭配 CloudFormation 或兩者。您可以使用 CloudFormation 交付選項選擇最多三個 AMI。

> **Note**
>
> 新的交付選項無法新增至現有版本。單一版本的所有交付選項都必須在相同的請求中提交。

6. 在 Amazon Machine Image (AMI) 區段中，提供下列資訊：
   - Amazon Machine Image ID – 輸入您要用於此版本的 AMI 的 AMI ID。您可以在 主控台的 AMI 清單中找到 AMIs ID。AMI 必須存在於美國東部（維吉尼亞北部）區域和您的 AWS Marketplace 賣方帳戶中。與此 AMI 相關聯的快照無法加密。
   - IAM 存取角色 ARN – 輸入 (IAM) 角色的 Amazon Resource Name AWS Identity and Access Management (ARN)， AWS Marketplace 允許 存取您的 AMI。如需如何建立 IAM 角色的說明，請參閱 提供 AMI 的 AWS Marketplace 存取權。使用 IAM ARN 的標準格式，例如：*arn#aws#iam##123456789012#role/RoleName*。ARN 必須存在於您的 AWS Marketplace 賣方帳戶中。
   - 作業系統 (OS) – 輸入 AMI 的基本作業系統。
   - 作業系統版本 – 輸入 AMI 使用的作業系統版本。
   - 作業系統使用者名稱 – 針對 Linux型 AMIs，輸入可用來登入執行個體的使用者名稱。建議使用 ec2-user。
   - 掃描連接埠 – 輸入可用於登入作業系統的連接埠號碼：LinuxAMI 的 SSH 連接埠或 Windows AMI 的 RDP 連接埠。
7. 如適用，請為 AMI 交付選項區段提供下列組態：
   - 建議的執行個體類型 – 選擇買方預設取得的執行個體類型。

- 使用說明 – 輸入使用 AMI 的說明或連結，以取得使用 AMI 的詳細資訊。例如：*########## # https://example.com/usage.htm#*
- 端點 URL – 提供買方在建立執行個體後如何存取軟體的相關資訊。輸入買方可用來存取您產品的通訊協定 (https 或 http)、相對 URL （例如 */index.html*) 和連接埠 （例如 *443*)。（主機名稱取決於 EC2 執行個體，因此您只需提供相對路徑）。
- 安全群組建議 – 輸入一或多個建議的資訊，包括通訊協定 (TCP 或 UDP)、要允許的連接埠範圍，以及 IPv4 CIDR IPs 清單 （格式為 xxx.xxx.xxx.xxx/nn，例如 *192.0.2.0/24)#*

8. 如適用，請使用 CloudFormation 交付選項為 AMI 提供下列組態設定：

   - 交付選項標題 – 這是識別此單一 AMI CloudFormation 範本交付選項的唯一名稱。
   - 簡短描述 – CloudFormation 範本的簡短描述。
   - 詳細說明 – CloudFormation 範本的詳細說明。
   - 建議的執行個體類型 – 根據預設，買方會取得的執行個體類型。
   - 使用說明 – 有關使用解決方案的說明，或有關使用解決方案的詳細資訊的連結。
   - CloudFormation 範本連結 – Amazon S3 中 CloudFormation 範本位置的 URL。對於巢狀堆疊或範本，請提供進入點或父項範本。檔案所在的 Amazon S3 儲存貯體必須可公開存取。如需詳細資訊，請參閱將 CloudFormation 範本新增至您的產品。
   - AMI 參數名稱 – 在 CloudFormation 範本中新增參數的名稱，範本中的 EC2 執行個體資源會參考其`ImageId`屬性值。 會使用 AWS Systems Manager 參數存放區參數 AWS Marketplace 取代指定的參數。當買方部署您的範本時，該參數會解析為您已發佈產品的 AWS 區域特定 AMI ID。如需詳細資訊，請參閱 AMI 詳細資訊的要求。
   - 架構圖連結 – Amazon S3 中架構圖位置的 URL。最大影像檔案的大小必須為 1560x878 或 1560x3120 像素，比例為 16：9 或 1：2。映像檔案所在的 Amazon S3 儲存貯體必須可公開存取。如需圖表需求，請參閱架構圖表。

9. 選取提交以提交請求來新增您的新版本。

   > **Note**
   >
   > 新增版本會導致掃描 AMI。如需詳細資訊，請參閱掃描 AMI 以取得發佈需求。

10. 確認請求顯示在請求索引標籤上，且狀態為審核中。如果有需要修正的錯誤，頁面會在頁面頂端的表格中顯示錯誤，而需要更新的特定欄位會以紅色顯示。

您可以隨時從伺服器產品頁面的請求索引標籤檢查請求的狀態。將檢閱新版本，如果成功，則會發佈為產品的新公有版本。如果發生問題，狀態可能是必要動作。選取請求以查看詳細資訊，包括任何問題。

如果您的請求成功，您現有的使用者會收到以下電子郵件訊息。訊息會通知他們有可用的新版本、版本版本備註的連結，並建議他們升級至最新版本。身為 AWS 帳戶 根使用者，您也會在與您的 相關聯的電子郵件帳戶中收到電子郵件訊息的副本 AWS 帳戶。

```
Greetings from AWS Marketplace,

Thank you for subscribing to <product-title>

We are writing to inform you that <seller-name> has added a new version to <product-
title> on AWS Marketplace.
As an existing customer, your subscription to the product, any running instances and
 access to previous versions
are unaffected. However, <seller-name> does recommend you to update to the latest
 version, <product-title>/<version-title>
by visiting <product-detail-page-of-new-listing>.

For additional questions or upgrade information, please contact <seller-name> directly.
 Click here <link of seller page on MP>
to visit the seller’s profile page on AWS Marketplace.

Release notes for <product-title>/<version-title>:

<release-notes>

Thank you,
The AWS Marketplace Team
https://aws.amazon.com/marketplace
AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of
 Amazon.com, Inc.
This message was produced and distributed by AWS Inc., 410 Terry Ave. North, Seattle,
 WA 98109-5210
```

限制版本

如果您想要防止買方存取您公開產品的特定版本，您可以限制該版本。

> **Note**
>
> 無論限制狀態為何，所有訂閱者都可以使用目前版本。 AWS Marketplace 準則會要求您在限制版本後繼續為現有買方提供支援 90 天。您的 AMI 會在版本受到限制後標示為已棄用。如需詳細資訊，請參閱《Amazon Elastic Compute Cloud Windows 執行個體使用者指南》中的棄用 AMI。

限制版本

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇限制版本。
4. 在限制版本頁面上，選取您要限制的版本 （ 或版本 ）。
5. 選取提交以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態為成功。

> **Note**
>
> 您無法限制產品的所有版本。如果您嘗試限制產品的最後一個剩餘公有版本，您會收到錯誤。若要完全移除產品，請參閱 the section called “移除產品”。

您可以隨時從伺服器產品頁面的請求索引標籤檢查請求的狀態。如需詳細資訊，請參閱取得變更請求的狀態。

> **Note**
>
> 限制版本最多可能需要 3 天才能完成。

如果您的請求成功，您現有的使用者會收到以下電子郵件訊息，通知他們版本限制，並建議他們使用最新的可用版本。身為 AWS 帳戶 根使用者，您也會在與您的 相關聯的電子郵件帳戶中收到電子郵件訊息的副本 AWS 帳戶。

```
Greetings from AWS Marketplace,
```

```
Thank you for subscribing to <product-title>.

We are writing to inform you that, as of <Version-Restriction-Date>, <Seller Name>
 will no longer offer version(s) "<version-title>" to new subscribers. Your use and
 subscription is unaffected for this version(s), however it is recommended that users
 upgrade to the latest version on AWS Marketplace.

For additional questions or upgrade information, please contact <seller-name> directly.
 Click here<link of seller page on MP> to visit the seller's profile page on AWS
 Marketplace.

Thank you,
The AWS Marketplace Team
https://aws.amazon.com/marketplace
AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of
 Amazon.com, Inc. This message was produced and distributed by AWS Inc., 410 Terry Ave.
 North, Seattle, WA 98109-5210
```

更新 上的 AMI 型產品資訊 AWS Marketplace

建立單一 Amazon Machine Image (AMI) 產品後，您可以變更其中與其相關聯的一些資訊 AWS Marketplace。例如，如果新版本修改了產品的描述或重點，您可以使用新資料編輯產品資訊。您也可以更新其他產品資訊，包括產品標題、SKU 描述、類別、關鍵字和其他。如需詳細資訊，請參閱下列程序。

更新產品資訊

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新產品資訊。
4. 更新您需要變更的下列任何欄位：
    - 產品標題
    - SKU
    - 簡短描述
    - 長描述
    - 產品標誌影像 URL

- 重點標示
- 產品類別
- 關鍵字
- 產品影片 URL
- 資源
- 支援資訊

> **Note**
>
> 如需標誌格式的詳細資訊，請參閱 公司和產品標誌要求。

5. 選取提交。
6. 確認請求顯示在請求索引標籤上，且狀態為審核中。您可能需要重新整理頁面，才能在清單中查看請求。

您可以隨時從伺服器產品頁面的請求索引標籤檢查請求的狀態。如需詳細資訊，請參閱取得變更請求的狀態。

依 AWS 區域 和國家/地區管理 AMI 型產品可用性

當您在 中建立產品時 AWS Marketplace，您可以選擇可用的 AWS 區域。您也可以選擇買方可以購買您產品的國家/地區。這兩個屬性類似，但不同。例如，買方可能位於美國並從中購買，但他們可能打算在歐洲（法蘭克福）區域安裝您的產品。為了讓此買方購買您的產品，您必須在您的國家/地區清單中同時包含美國，並在您的區域清單中包含歐洲（法蘭克福）區域。您可以使用下列各節，依區域和國家/地區更新您的產品可用性。

主題

- 新增 AWS 區域
- 限制 AWS 區域
- 更新對未來的支援 AWS 區域
- 依國家/地區更新可用性

新增 AWS 區域

您可以新增買方可以使用您產品的 區域。

新增區域

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇新增區域。
4. 從可用區域清單中選擇您要新增的區域。
5. 選擇提交請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

> **Note**
>
> 當您新增對新 的支援時 AWS 區域，已訂閱您產品的私有優惠的客戶將無法自動存取新增的區域。您必須使用您希望客戶存取的區域建立另一個私有優惠。接受新優惠後，客戶即可存取新增的區域。只要區域包含在私有優惠中，未來訂閱您產品的客戶也可以存取 區域。如需如何建立新的私有優惠的詳細資訊，請參閱私有優惠升級、續約和修訂。

限制 AWS 區域

若要防止新買方在特定地區使用您的產品 AWS 區域，您可以限制 區域。您可以稍後再新增區域。區域中產品的現有訂閱者只要訂閱，就可以繼續使用區域中的產品。

限制區域

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇限制區域。
4. 選取下拉式選單，以檢視目前可使用您產品的區域清單。
5. 選取您要限制的區域。
6. 您選取的區域會顯示為字符。檢閱您要限制的區域清單，然後針對您不想限制的區域輸入 X。
7. 選擇提交變更請求以提交您的請求以供檢閱。
8. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

如果您的請求成功，您現有的使用者會收到以下電子郵件訊息，通知他們要限制的區域。只要它們保持訂閱狀態，他們就可以繼續使用您的產品，但如果他們取消訂閱，就無法重新訂閱。

```
Greetings from AWS Marketplace,

This message is a notification detailing a recent change for <ProductName>.
{{{sellerName}}} has opted to restrict the <ProductType> product in <Restricted
 Region(s)> beginning <DateOfChange>.

This impacts you in the following ways:

1. As long as you're subscribed to the product, you can continue using the software
 product in the restricted Region.
2. You can't begin new instances of the software product in the restricted Region.
3. You can continue using the software product in all available AWS Regions.

Regards,
The AWS Marketplace Team
AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com (http://amazon.com/) is a
 registered
trademark of Amazon.com, Inc. This message was produced and distributed by Amazon Web
Services Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.
```

更新對未來的支援 AWS 區域

如果您希望您的產品加入新啟動的 AWS 區域，您可以使用更新未來區域支援。

更新未來的區域支援

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新未來區域支援。
4. 您可以選擇啟用未來的區域支援， AWS Marketplace 以允許 AWS 區域 代表您將產品加入新啟動的 。
5. 啟用此功能後，您可以在所有未來的區域之間進行選擇，或僅限於美國區域。
6. 選擇提交變更請求以提交您的請求以供檢閱。
7. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

依國家/地區更新可用性

如果您想要變更產品訂閱和提供的國家/地區，您可以使用更新可用性。

依國家/地區更新可用性

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新可用性。
4. 請選擇下列其中一個選項：
    1. 所有國家 - 在所有支援的國家/地區提供。
    2. 具有排除項目的所有國家/地區 – 適用於所有支援的國家/地區，但所選國家/地區除外。
    3. 自訂清單 – 提供產品的特定國家/地區清單。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

更新以 AMI 為基礎的 AWS Marketplace 產品的最終使用者授權合約 (EULA)

身為 AWS Marketplace 賣方，您可以更新最終使用者授權協議 (EULA)，該協議將規範您單一 Amazon Machine Image (AMI) 產品的使用。您的 EULA 位於 上的公有軟體清單產品清單頁面上 AWS Marketplace。您可以套用自己的 EULA 或使用標準合約 for AWS Marketplace (SCMP)。下列程序說明如何更新單一 AMI 產品的 EULA。

如需 EULA 的詳細資訊，請參閱 在 中使用標準化合約 AWS Marketplace。

更新 EULA

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 選擇伺服器產品索引標籤，在目前伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇更新最終使用者授權合約。
4. 您可以選取 AWS Marketplace (SCMP) 的標準合約，或提交您自己的自訂 EULA。對於自訂 EULA，您必須提供來自 Amazon S3 儲存貯體的自訂合約 URL。

> **Note**
>
> 必須在 Amazon S3 儲存貯體上啟用公有可存取性。

5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

在 上更新以 AMI 為基礎的產品的退款政策 AWS Marketplace

身為 AWS Marketplace 賣方，您可以為單一 Amazon Machine Image (AMI) 產品設定退款政策。如果您想要變更產品的退款政策，您可以使用更新退款政策。下列程序說明如何更新退款政策。

更新退款政策

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新退款政策。
4. 目前的退款政策詳細資訊會在文字方塊中提供，供您編輯。提交請求會覆寫目前的退款政策。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

提供 AMI 的 AWS Marketplace 存取權

當您建立包含將新的 Amazon Machine Image (AMI) 新增至 的請求時 AWS Marketplace，必須將 AMI 複製到 AWS Marketplace 系統中，然後掃描安全問題。您必須建立具有對 AMI 執行動作許可的 AWS Identity and Access Management (IAM) 角色，以及允許 擔任該角色的信任政策， AWS Marketplace 以授予對 AMI 的 AWS Marketplace 存取權。您只需要建立一次 IAM 角色。下列程序說明如何為資產 AWS Marketplace 擷取建立角色，以 AWS Marketplace 允許存取您的 AMI。

為 AWS Marketplace AMI 資產擷取建立角色

1. 登入 AWS Management Console，開啟 IAM 主控台並前往角色頁面。
2. 選取 建立角色。
3. 在建立角色頁面上，進行下列選擇：

- 選取信任實體的類型 – 選擇AWS 服務。
- 選擇使用案例 – 選擇 AWS Marketplace。
- 選取您的使用案例 – 選擇 Marketplace – AMI Assets Ingestion。
- 若要移至下一頁，請選取下一步：許可。

4. 選取AWSMarketplaceAmiIngestion政策。視需要新增許可界限，然後選取下一步：標籤以繼續。

> **Note**
>
> 您可以使用許可界限來限制您使用此角色提供的 AWS Marketplace 存取。如需詳細資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的 IAM 實體的許可界限。

5. 若要繼續，請選取下一步：檢閱。
6. 提供角色的名稱，然後選取建立角色。
7. 您應該會在頁面頂端看到「已建立角色####」，而角色應該會出現在角色清單中。

在此頁面上，當您選取剛建立的角色時，您可以看到其 ARN，格式為 arn：aws：iam：：123456789012：role/exampleRole。當您建立變更請求時，請使用 IAM 存取角色 ARN，例如將新版本新增至您的產品時。

從 移除產品 AWS Marketplace

發佈產品後，您可以從中移除產品 AWS Marketplace。這也稱為日落。若要移除產品，請識別產品並提交移除產品的請求，以及移除原因和您的聯絡電子郵件地址。如果您要使用新的產品取代目前的產品，也可以提供替代產品 ID。

在您請求產品移除後，新客戶無法訂閱產品。您必須支援任何現有客戶至少 90 天。

> **Note**
>
> 您無法從 AWS Marketplace Management Portal 刪除受限產品。入口網站會將它們保留為帳戶發佈歷史記錄的一部分。

我們處理從 移除產品的請求 AWS Marketplace ，但條件如下：

- 產品會從 AWS Marketplace 搜尋、瀏覽和其他探索工具中移除。任何訂閱按鈕或功能都會停用，而且頁面上的訊息會清楚指出產品已無法使用。產品詳細資訊頁面仍然可以使用 URL 存取，並且可以在公有搜尋引擎中編製索引。
- 必須指定移除原因（例如，支援結束、產品更新結束或替代產品）。如需已移除產品的持續支援要求，請參閱 AWS Marketplace 賣方的條款與條件。
- AWS Marketplace 透過電子郵件訊息聯絡目前的買方，通知他們產品移除、移除原因，並提供賣方聯絡資訊。
- 目前的購買者會保留軟體的存取權，直到他們取消訂閱為止。它們不會受到產品移除的任何影響。

移除使用 建立的產品 AWS Marketplace 管理入口網站

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 選擇產品索引標籤，然後選擇伺服器。
3. 在產品頁面的伺服器產品下，找到您要移除的產品。從請求變更下拉式清單中，選擇更新產品可見性。
4. 在更新產品可見性頁面上，選取限制。
5. （選用）如果有其他產品將取代您要移除的產品，請提供替換產品 ID。
6. 檢閱資訊的準確性，然後選擇提交。

答 提交產品移除請求後，會顯示下一個資訊頁面。 AWS Marketplace 賣方操作會檢閱和處理您的請求。檢視請求以檢查提交狀態。

移除您的產品後，該產品會出現在 中的目前產品清單中 AWS Marketplace 管理入口網站。在目前產品中，您可以執行的唯一動作是下載產品的試算表。您無法編輯或提交另一個日落請求。

如果您對產品移除有任何疑問，請聯絡AWS Marketplace 賣方營運團隊。

對 上變更請求的常見錯誤進行故障診斷 AWS Marketplace

當您在 上變更產品的資訊時 AWS Marketplace，您可能會遇到錯誤。本主題說明一些常見錯誤，並提供如何修正錯誤的建議。

- 掃描您的 AMI - 掃描您的 時可能會發生幾個問題AMI：

  - 您尚未授予掃描 的 AWS Marketplace 許可AMI。授予 AWS Marketplace 許可以進行存取。或者您已授予許可，但許可界限太嚴格。如需詳細資訊，請參閱提供 AMI 的 AWS Marketplace 存取權。
  - 如果掃描在您的 中發現安全問題或常見漏洞和暴露 （CVEs）AMI，請確定您使用映像中作業系統的最新修補程式。如需詳細資訊，請參閱的 AMI 型產品需求 AWS Marketplace。

  如需建置 的一般準則AMI，請參閱 建置 AMIs 以搭配 使用的最佳實務 AWS Marketplace。
- AWS Marketplace 管理入口網站 欄位 – 中的某些欄位 AWS Marketplace 管理入口網站 需要非常具體的資訊：
  - 如果您不確定欄位請求的內容，請嘗試檢查主控台中的詳細資訊。大多數欄位在欄位上方都有文字描述，並在欄位下方有格式要求。
  - 如果您嘗試提交具有一或多個無效欄位的表單，則會顯示問題清單。建議的動作可協助您修正問題。
  - 如果您被要求提供 ARN，通常可以在主控台的其他地方找到它。例如，您可以在IAM主控台的角色頁面上AMI找到您為授予 AWS Marketplace 存取權ARNIAM所建立角色的 。 https://console.aws.amazon.com/iam/home?region=us-east-1#/rolesARNs 都有類似的格式。例如，IAM角色的格式ARN為 arn：aws：iam：：123456789012：role/exampleRole。
  - 您的標誌和影片必須URL直接提供為內容的 。如需標誌格式的詳細資訊，請參閱 公司和產品標誌要求。

  如需提交產品和版本變更請求的詳細資訊，請參閱 提交您的產品以在 上發佈 AWS Marketplace。
- 產品載入表單 （PLF） 問題 – PLFs包含試算表中包含的指示。整體指示提供於指示表中。每個欄位都有如何填寫的指示 - 選取欄位以顯示指示。
- 請求進行中 – 某些請求無法並行發生。您一次只能有一個更新產品特定進行中資訊的請求。您可以在 中的伺服器產品頁面的請求索引標籤上看到所有仍在審核中的請求 AWS Marketplace 管理入口網站。如果您有未打算的待處理請求，您可以取消該請求，然後提交包含您要進行變更的新請求。
  - 當更新 （新增或限制） 版本正在進行時，您無法更新版本資訊。
  - 如果 AWS Marketplace 賣方營運團隊有待處理的請求，則您無法提交任何新的變更。
- 無法解釋的錯誤 – 如果您的提交失敗，但沒有解釋，請再試一次。伺服器載入偶爾會導致提交失敗。

如果您仍然遇到變更請求的問題，請聯絡AWS Marketplace 賣方營運團隊。

建置 AMIs 以搭配 使用的最佳實務 AWS Marketplace

本主題提供最佳實務和參考，協助您建置 Amazon Machine Image (AMIs) 以搭配 使用 AWS Marketplace。建置並提交至 AMIs AWS Marketplace 必須遵循所有 AWS Marketplace 產品政策。如需詳細資訊，請參閱下列區段。

主題

- 保護轉售權利
- 建置 AMI
- 為 準備和保護您的 AMI AWS Marketplace
- 掃描您的 AMI 以取得發佈需求
- 驗證您的軟體是否在 AMI 上執行 AWS Marketplace

保護轉售權利

對於非免費 Linux 發行版本，除了 AWS提供的 Amazon Linux、RHEL 和 SUSE 之外，您需負責保護其轉售權利的安全。您不需要保護 Windows AMIs 的轉售權利。

建置 AMI

使用下列準則來建置 AMIs：

- 確保您的 AMI 符合所有AWS Marketplace 政策。
- 在美國東部（維吉尼亞北部）區域中建立您的 AMI。
- 從 Amazon Elastic Block Store (Amazon EBS) 支援的現有、維護良好的 AMIs 建立產品，其中包含信任且信譽良好的來源提供的明確定義生命週期，例如 AWS Marketplace。
- 使用最新的作業系統、套件和軟體來建置 AMI。
- 確保您的 AMI 是以公有 Amazon EC2 AMI 為基礎，該 Amazon EC2 AMI 使用硬體虛擬機器 (HVM) 虛擬化和 64 位元架構。
- 開發一套可重複建置、更新及重新發佈 AMI 的流程。
- 在所有版本和產品中使用一致的作業系統 (OS) 使用者名稱。建議的預設使用者名稱`ec2-user`適用於 Linux 和其他類似 Unix 的系統，以及`Administrator`適用於 Windows。
- AWS Marketplace 在提交要發佈的最終 AMI 之前，請從 AMI 啟動和測試執行個體，以驗證預期的最終使用者體驗。在此執行個體上測試所有安裝方法、功能和效能。

- 檢查連接埠設定，如下所示：
  - 作為針對開放防火牆、反向代理和 SSRF 漏洞的最佳實務安全組態，IMDS 支援選項必須設定為僅限 IMDSv2。在最終建置階段註冊新的 AMI 時，可以使用下列 CLI：
    - `aws ec2 register-image --name my-image --root-device-name /dev/xvda --block-device-mappings DeviceName=/dev/xvda,Ebs={SnapshotId=snap-0123456789example} --architecture x86_64 --imds-support v2.0`

如需建立 AMI 的詳細資訊，請參閱下列資源：

- 《Amazon EC2 使用者指南》中的建立 Amazon EBS 後端 AMI Amazon EC2
- 《Amazon EC2 使用者指南》中的使用 Windows Sysprep 建立 Amazon EC2 AMI Amazon EC2
- 如何從 EBS 支援的執行個體建立 Amazon Machine Image (AMI)？
- Amazon Linux AMI
- Amazon EC2 執行個體類型和執行個體類型
- 根據預設，設定 IMDS V2 使用的 AMI

為 準備和保護您的 AMI AWS Marketplace

我們建議您遵循下列準則來建立安全的 AMIs：

- 使用 Amazon EC2 使用者指南中的共用 Linux AMIs 指導方針
- 建構您的 AMI 以部署為最低安裝，以減少攻擊面。停用或移除不必要的服務和程式。
- 盡可能使用網路流量的end-to-end加密。例如，使用 Secure Sockets Layer (SSL) 保護您和買方之間的 HTTP 工作階段。確保您的服務僅使用有效且up-to-date憑證。
- 記錄您的 AMI 產品時，提供安全群組建議給買方，以控制其執行個體的傳入流量存取。您的建議應指定下列項目：
  - 您的服務運作所需的最小連接埠集。
  - 管理存取的建議連接埠和來源 IP 地址範圍。

  這些安全群組建議可協助買方實作適當的存取控制。如需如何將新版本新增至 AMI 產品的詳細資訊，請參閱 新增新版本。
- 請考慮定期對 AWS 運算環境執行滲透測試，或考慮聘請第三方代表您執行此類測試。如需詳細資訊，包括滲透測試申請表，請參閱AWS 滲透測試。

- 請注意 Web 應用程式的前 10 個漏洞，並相應地建置您的應用程式。若要進一步了解，請參閱開放式 Web 應用程式安全專案 (OWASP) - 十大 Web 應用程式安全風險。發現新的網際網路漏洞時，請立即更新 AMI 中隨附的任何 Web 應用程式。包含此資訊的資源範例包括 SecurityFocus 和 NIST 國家漏洞資料庫。

如需安全性的詳細資訊，請參閱下列資源：

- AWS 雲端安全性
- 網際網路安全中心 (CIS)：安全基準
- 開放式 Web 應用程式安全專案 (OWASP)：安全編碼實務 - 快速參考指南
- OWASP 十大 Web 應用程式安全風險
- SANS (SysAdmin、稽核、網路和安全性） 常見弱點列舉 (CWE) 前 25 名最危險的軟體錯誤
- 安全焦點
- NIST 國家漏洞資料庫

掃描您的 AMI 以取得發佈需求

若要在 AWS Marketplace 目錄上發佈 AMI，您必須完成 AMI 掃描。AMI 掃描會檢查未修補的常見漏洞和暴露 (CVEs)，並驗證您的 AMI 是否遵循安全最佳實務。如需詳細資訊，請參閱準備和保護 的 AMI AWS Marketplace

若要執行 AMI 掃描，請選擇下列其中一個選項：

選項 1：資產功能表

此方法允許在產品建立流程之外掃描 AMIs。對於使用 SAAS Quick Launch 且需要掃描資產而不建立 AMI 產品的 SaaS 賣方也很有用。

1. 從 AWS Marketplace 管理入口網站導覽至資產功能表，然後選擇 Amazon Machine Image。
2. 若要開始掃描程序，請選擇新增 AMI。
3. 您可以返回此頁面來檢視 AMIs掃描狀態。

選項 2：請求變更功能表

此選項適用於已建立 AMI 產品的賣方。如需進一步了解，請參閱 建立 AMI 型產品。

1. 從 AWS Marketplace 管理入口網站導覽至產品選單，然後選擇伺服器。

2. 從伺服器產品中選取您的產品。這必須是 AMI 型產品。產品可以處於任何狀態，不需要處於公有發佈狀態即可進行後續步驟。
3. 導覽至請求變更選單，然後選取更新版本。
4. 選取測試「新增版本」。依照提示提交包含 AMI 詳細資訊的請求。如果請求成功，這表示 AMI 已成功通過掃描。與新增新版本選項不同，如果掃描成功，測試「新增版本」不會將新版本新增至 AMI 型產品。

> **Note**
>
> 若要了解如何授予 AMI 的 AWS Marketplace 存取權，請參閱 提供 AMI 的 AWS Marketplace 存取權。

驗證您的軟體是否在 AMI 上執行 AWS Marketplace

您可能希望軟體在執行時間驗證其是否在從 AMI 產品建立的 Amazon EC2 執行個體上執行。

若要驗證是否已從您的 AMI 產品建立 Amazon EC2 執行個體，請使用內建於 Amazon EC2 的執行個體中繼資料服務。下列步驟會引導您完成此驗證。如需使用中繼資料服務的詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的執行個體中繼資料和使用者資料。

1. 取得執行個體身分文件

   每個執行中的執行個體都有可從執行個體存取的身分文件，可提供執行個體本身的資料。下列範例顯示從執行個體使用 curl 擷取執行個體身分文件。

   IMDSv2：（建議）

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/
instance-identity/document
 {
   "accountId" : "0123456789",
   "architecture" : "x86_64",
   "availabilityZone" : "us-east-1e",
   "billingProducts" : null,
   "devpayProductCodes" : null,
   "marketplaceProductCodes" : [ "0vg0000000000000000000000" ],
```

```
   "imageId" : "ami-0123456789abcdef1",
   "instanceId" : "i-0123456789abcdef0",
   "instanceType" : "t2.medium",
   "kernelId" : null,
   "pendingTime" : "2020-02-25T20:23:14Z",
   "privateIp" : "10.0.0.2",
   "ramdiskId" : null,
   "region" : "us-east-1",
   "version" : "2017-09-30"
}
```

IMDSv1：

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document{
   "accountId" : "0123456789",
   "architecture" : "x86_64",
   "availabilityZone" : "us-east-1e",
   "billingProducts" : null,
   "devpayProductCodes" : null,
   "marketplaceProductCodes" : [ "0vg0000000000000000000000" ],
   "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. 驗證執行個體身分文件

   您可以使用簽章來驗證執行個體身分是否正確。如需此程序的詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的執行個體身分文件。

3. 驗證產品代碼

   當您最初提交 AMI 產品進行發佈時，您的產品會獲指派一個產品代碼 AWS Marketplace。您可以檢查執行個體身分文件中`marketplaceProductCodes`的欄位來驗證產品代碼，也可以直接從中繼資料服務取得：

   IMDSv2：

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
 && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/product-codes
```

如果產品代碼與您的 AMI 產品相符，則執行個體是從您的產品建立的。

您也可以驗證執行個體身分文件中的其他資訊，例如 `instanceId`和執行個體 `privateIp`。

的 AMI 產品定價 AWS Marketplace

AWS Marketplace 有多個 Amazon Machine Image (AMI) 產品的定價模型。透過賣方私有優惠，有可用於多年期和自訂持續時間合約的選項。如需多年期和自訂持續時間合約的詳細資訊，請參閱 為您的 AWS Marketplace 產品準備私有優惠和 私有優惠分期方案。下列各節提供 AMI 產品定價模型的相關資訊。

> **Note**
>
> 您必須能夠提供 W-9 稅務表單 （適用於美國實體） 或 W-8 表單 （適用於歐洲實體），如中所述註冊為 AWS Marketplace 賣方。

主題

- AMI 定價模型
- AWS 費用和軟體費用
- 使用 AWS Marketplace Metering Service 自訂 AMI 產品的計量定價
- 上的 AMI 產品的合約定價 AWS Marketplace
- 使用 將授權與 AMI 型產品建立關聯 AWS License Manager

AMI 定價模型

下表提供 AMI 產品定價模型的一般資訊。

定價方式	描述
免費	客戶可以執行 Amazon Elastic Compute Cloud (Amazon EC2) 支援的任意數量的執行個體，而不會產生額外的軟體費用。 **Note** 免費試用和年度定價無法與每月定價合併。
使用自有授權 (BYOL)	AWS Marketplace 不會向客戶收取軟體使用費，但客戶必須提供授權金鑰才能啟用產品。此金鑰是在 外部購買 AWS Marketplace。除了所有定價和帳單之外，您還會處理權利和授權強制執行。
按小時支付或按年支付	每小時 – 軟體按小時收費。每個執行個體類型的價格可以不同（但不需要），用量會四捨五入至最接近的整小時。 每小時免費試用 – 客戶僅限於僅執行一個軟體執行個體，而不會產生費用。您可以定義持續時間，介於 5 到 30 天之間。免費試用適用於執行中最昂貴的執行個體類型，而且 1 個執行個體以外的任何並行使用都會以小時費率計費。注意：這是不同於 Amazon EC2 用量 AWS 免費方案的模型，因此客戶每月可獲得 750 小時的免費用量。 每小時和每月 – 每小時和每月費用都會獨立套用。無論用量為何，每月都會收取 費用，而每小時費用只會根據每小時用量來收取。 按年計費 – 客戶可以選擇為一個執行個體類型的 Amazon EC2 執行個體預付一年的使用量。您可以為每個執行個體類型設定定價，並且可以比每小時價格節省淨價。任何超過購買年度訂閱數量的客戶用量，都會以您為該執行個體類型設定的每小時費率計費。 具有每年和自訂持續時間的每小時 – 這種類型的優惠只能透過賣方私有優惠提供。使用賣方私有優惠，您可以指定自訂合約期間，最長可達 3 年。您可以指定預付付款，或包含彈性

定價方式	描述
	的付款排程。您可以為每個執行個體類型設定定價。如果優惠中有彈性的付款排程，您也可以為優惠中包含的每個執行個體類型設定發票日期、付款金額和執行個體數量。對於具有彈性付款排程的作用中賣方私有優惠，在客戶啟動指定數量的執行個體之後，任何啟動的額外執行個體都會以賣方私有優惠中指定的每小時費率計費。如需多年期和自訂持續時間合約的詳細資訊，請參閱 為您的 AWS Marketplace 產品準備私有優惠和私有優惠分期方案。 每小時與年度（包括免費試用）– 這與具有年度選項的每小時模型相同，除了包含免費試用，允許客戶免費執行任何執行個體類型的一個執行個體，並持續您決定的固定天數。年度訂閱可隨時購買，並與免費試用訂閱結合使用。 按小時計費 – 與按小時計費的年度定價模式相同。客戶可以選擇為一個執行個體類型的 Amazon EC2 執行個體預先購買一年的用量。您可以為每個執行個體類型設定定價，並且可以比每小時價格節省淨額，但不需要節省成本。任何超過購買年度訂閱數量的客戶用量，都會以您為該執行個體類型設定的每小時費率計費。 以小時為單位的多年和自訂持續時間 – 這只能透過 提供為您的 AWS Marketplace 產品準備私有優惠。使用賣方私有優惠，您可以指定最長三年的自訂持續時間合約。您可以要求預付款項，也可以提供彈性的付款排程給客戶。您可以為合約期間的每個執行個體類型設定定價，以及啟動的其他執行個體的每小時定價。如果您提供彈性的付款排程，您也可以為優惠中包含的每個執行個體類型設定發票日期、付款金額和執行個體數量。對於具有彈性付款排程的作用中私有優惠，在啟動指定數量的執行個體之後，客戶啟動的任何其他執行個體都會以私有優惠中指定的每小時費率計費。如需多年期和自訂持續時間合約的詳細資訊，請參閱 為您的 AWS Marketplace 產品準備私有優惠和 私有優惠分期方案。

定價方式	描述
	Note 免費試用和年度定價無法與每月定價合併。
每月付費	每月 – 無論客戶執行的執行個體數量為何，都會每月固定支付軟體費用。每月費用會在註冊和取消時按比例計算。範例：當月訂閱 1 天的客戶，將支付當月 1/30 的費用。 每月每小時 – 每小時和每月費用都會獨立套用。無論用量為何，每月都會收取 費用，而每小時費用只會根據每小時用量來收取。 Note 免費試用和年度定價無法與每月定價合併。
付費用量定價	軟體會直接針對您提供的值以及您提供的四個使用類別之一收費：使用者、資料、頻寬或主機。您可以為產品定義最多 24 個維度。客戶仍每小時產生所有費用。 所有用量都會每月計算，並使用與現有 AWS Marketplace 軟體相同的機制每月計費。用量定價也稱為 AWS Marketplace Metering Service。
合約定價模型	具有合約定價的 AMI – 具有 AWS CloudFormation 堆疊的單一 AMI 產品或單一 AMI，買方需預付費用。

AWS 費用和軟體費用

Amazon Machine Image (AMI) 型產品費用分為兩個類別：

- 基礎設施定價詳細資訊 – 所有以 AMI 為基礎的產品都會產生相關聯的 AWS 基礎設施費用，具體取決於所使用的服務和基礎設施。這些費率和費用是由 定義和控制 AWS，且可能因 而異 AWS 區域。如需詳細資訊，請參閱 Amazon EC2 定價。
- 軟體定價詳細資訊 – 對於付費產品，賣方會定義使用軟體的費用。

這兩個產品費用類別會分別顯示在 AWS Marketplace 詳細資訊頁面上，以協助買方了解使用產品的潛在成本。

AMI 每小時產品的免費試用

AMI 每小時產品符合選擇性免費試用計畫的資格。在免費試用中，客戶可以訂閱產品並使用單一執行個體長達 31 天，而無需支付產品的軟體費用。仍需支付適用的 AWS 基礎設施費用。免費試用會在到期時自動轉換為付費訂閱。客戶需支付超過免費單位的額外用量費用。若要提供每小時產品免費試用，請定義試用期的持續時間，並通知AWS Marketplace 賣方營運團隊。試用期可以是 5–31 天。

當客戶訂閱免費試用產品時，會收到歡迎電子郵件訊息，其中包含免費試用期間、計算到期日，以及取消訂閱的詳細資訊。提醒電子郵件訊息會在過期日期前三天傳送。

如果您在 中提供免費試用產品 AWS Marketplace，即表示您同意退款政策中所述的特定退款政策。

其他資源

如需 AMI 產品定價的詳細資訊，請參閱下列主題：

- 使用 AWS Marketplace Metering Service 自訂 AMI 產品的計量定價
- 上的 AMI 產品的合約定價 AWS Marketplace

如需 AMI 型產品的 Bout 計費、計量和授權整合的詳細資訊，請參閱下列主題：

關於 AMI 產品定價，請參閱下列主題：

- 使用 設定 AMI 產品的自訂計量 AWS Marketplace Metering Service
- 使用 將授權與 AMI 型產品建立關聯 AWS License Manager

使用 AWS Marketplace Metering Service 自訂 AMI 產品的計量定價

使用 AWS Marketplace Metering Service，您可以修改軟體，將計量記錄傳送至端點以擷取用量。您可以選擇用量類別，並為該類別定義最多 24 個維度。這些維度每小時計量一次、彙總，並根據您定義的價格計劃收費。身為賣方，您必須決定要使用的維度。發佈 AMI 之後，您將無法變更它。下列各節提供 AWS Marketplace Metering Service 的相關資訊。

主題

- 計量服務概念
- 為您的軟體定價

- 將您的產品新增至 AWS Marketplace
- 修改您的軟體以使用計量服務
- 廠商計量標記（選用）
- 設定自訂計量
- 使用 設定 AMI 產品的自訂計量 AWS Marketplace Metering Service

計量服務概念

在使用 AWS Marketplace Metering Service 之前，請注意下列重要的服務概念：

- 用量類別 – 使用 Metering Service 定價的任何軟體產品都會根據一個用量類別進行分類，這會決定向客戶收費的適當方式。用量類別包括但不限於：
  - 使用者 – 與單一識別符相關聯的一組已定義許可。此類別適用於客戶使用者直接連線至軟體的軟體（例如，用於客戶關係管理或商業智慧報告）。
  - 主機 – 運算系統的任何伺服器、節點、執行個體、端點或其他部分。此類別適用於監控或掃描許多客戶擁有的執行個體（例如效能或安全監控）的軟體。
  - 資料 – 以 MB、GB 或 TB 為單位的儲存或資訊。此類別適用於管理預存資料或批次處理資料的軟體。
  - 頻寬 – 以 Mbps 或 Gbps 為單位測量。此類別適用於允許客戶指定要佈建頻寬量的軟體。
  - 單位 – 度量單位；請參閱下列範例。
- 用量單位 – 軟體產品的特定用量單位對應於選取的用量類別。此用量單位說明軟體將向其收費的單位。範例包括：
  - NodesHrs（對應至主機類別）
  - UserHrs（對應至使用者類別）
  - GBStored（對應至 Data 類別）
- 使用 – 透過使用計量服務費用以三種方式之一來使用的軟體產品定價：
  - 佈建 – 軟體可讓客戶設定特定數量的資源以供使用（例如，使用者數量或固定頻寬數量）。客戶每小時會支付已佈建的項目。
  - 並行 – 軟體允許任意數量的不同主機或使用者連線到軟體。客戶每小時都會根據存取軟體的主機或使用者數量付費。
  - 累積 – 軟體可讓客戶使用任何處理或存放的資料量。客戶每小時會支付彙總金額。
- 定價 – 透過使用 Metering Service 定價的軟體產品必須指定單一價格或定義最多 24 個維度，每個維度都有自己的價格。定價選項的詳細資訊包括：

  - 單一維度 – 這是最簡單的定價選項。無論大小或數量為何（例如，每位使用者每小時 0.014 美元或每部主機每小時 0.070 美元），客戶都會按每小時資源單位支付單一價格。
  - 多個維度 – 當選取的用量類別沿著多個軸變化時，此定價選項是適當的。例如，對於主機監控，可以根據主機的大小設定不同的價格。或者，對於以使用者為基礎的定價，可以根據使用者類型（例如，管理員、進階使用者和唯讀使用者）設定不同的價格。
- 計量 – 所有用量都會記錄為計量事件，每小時一次。您的軟體必須設定為將適當的維度和用量傳送至 AWS Marketplace Metering Service。
  - 配置 – 或者，您可以依您追蹤的屬性，將用量分配到配置中。這些配置會以標籤表示給買方。這些標籤可讓買方依標籤檢視其成本分割為用量。例如，如果您由使用者收費，且使用者具有「部門」屬性，則可以使用索引鍵為「部門」的標籤建立用量配置，每個值一個配置。此方法不會變更您報告的價格、維度或總用量。不過，它可讓您的客戶依適合您產品的類別來檢視其成本。

為您的軟體定價

使用 AWS Marketplace Metering Service 為您的軟體定價時，您必須先決定用量類別及其使用方式。服務支援六個不同的定價案例。您只能為您的產品選取下列其中一項：

- 佈建的使用者（每小時）
- 並行使用者（每小時）
- 佈建主機（每小時）
- 並行主機（每小時）
- 佈建頻寬（每小時）
- 累積資料（每小時）

接著，您必須決定如何為選取的用量類別定價：

- 單一價格
- 多個維度（最多 24 個）

將您的產品新增至 AWS Marketplace 說明如何提供維度和定價的客戶友好描述。

範例：具有非線性定價的佈建頻寬

假設您提供網路設備軟體。您可以選擇依佈建頻寬計費。針對您的用量類別，選取頻寬。除了透過頻寬收費，您希望隨著買方擴展而收取不同的價格。您可以在頻寬類別中定義多個維度。您可以為 25 Mbps、100 Mbps 和 1 Gbps 定義不同的價格。

範例：具有多個維度的並行主機

假設您提供軟體來監控其他 Amazon EC2 執行個體。您可以選擇依受監控的主機數量計費。針對您的用量類別，選取主機。除了透過主機收費之外，您希望收取監控大型主機的額外值費用。您可以在主機類別中使用多個維度。您可以為微型、小型、中型、大型、x 大型、2XL, 4XL和 8XL 執行個體定義不同的價格。您的軟體負責將每個特定主機映射到您定義的其中一個維度。如適用，您的軟體會負責針對用量類別的每個維度傳送個別的計量記錄。

將您的產品新增至 AWS Marketplace

若要利用 Metering Service，您必須為 建立新的產品 AWS Marketplace 才能列出。如果您的產品已在 上 AWS Marketplace，您將需要決定是否除了目前的產品之外，還會提供新的 AWS Marketplace Metering Service 產品，或者是否會將目前的產品取代為可供新使用者使用的唯一版本。如果您選擇替換，現有的產品將從 中移除， AWS Marketplace 因此不再可供新買方使用。現有客戶將繼續存取其舊產品和執行個體，但他們可以在方便的時候遷移到新產品。新產品必須測量 AWS Marketplace Metering Service 的用量，如中所述修改您的軟體以使用計量服務。

在您擁有 AMI 之後，請遵循標準程序，使用自助式工具來共用和掃描 AMI。除了使用 管理入口網站上提供的範本之外，請填寫產品載入表單並上傳以開始擷取程序。

使用下列定義來完成 AWS Marketplace Metering Service 的產品載入表單欄位。在產品載入表單上，這些欄位會標記為彈性消費定價 (FCP)，以區分它們與每小時和每月定價產品。

- 標題：如果您已在 上擁有產品， AWS Marketplace 且您使用 新增相同的產品 AWS Marketplace Metering Service，請在括號中包含 FCP 類別和維度，以區分它們（例如「產品標題（資料）」）。
- 定價模型 – 從下拉式清單中，選擇用量。
- FCP 類別 – 向客戶收取使用量定價元件付費產品的類別。從下拉式清單中，選擇使用者、主機、資料或頻寬。
- FCP 單位 – 衡量單位，針對使用量定價元件的付費產品向客戶收費。選項會根據您選取的 FCP 類別顯示在下拉式清單中。下表列出每個類別的有效單位。

類別	有效單位
使用者	UserHrs
主機	HostHrs
資料	MB、GB、TB
頻寬	Mbps、Gbps

- FCP 維度名稱 – 透過呼叫 `MeterUsage`操作傳送計量記錄時使用的名稱。它會顯示在帳單報告中。但是，因為它不是面向外部的，因此名稱不需要易於使用。名稱不可超過 15 個字元，且只能包含英數字元和底線字元。設定名稱並公開產品之後，您就無法變更它。變更名稱需要新的 AMI。
- FCP 維度描述 – 面向客戶的陳述式，描述產品的維度。描述（不可超過 70 個字元，且應易於使用。描述的範例包括：每小時的管理員和佈建的每 Mbps 頻寬。發佈產品後，您無法變更此描述。
- FCP 費率 – 此產品的每單位軟體費用。此欄位支援八個小數位數。

備註：

- 您不需要填寫每小時和每年定價欄位。
- 免費試用和年度定價不相容。
- 使用 AMI 以及叢集 AWS 和資源功能的產品無法使用 AWS Marketplace Metering Service。
- 價格、執行個體類型或 AWS 區域 變更將遵循與其他 AWS Marketplace 產品相同的程序。
- 具有 AWS Marketplace Metering Service 的產品無法轉換為其他定價模式，例如每小時、每月或自攜授權 (BYOL)。
- AWS Marketplace 建議在使用說明或文件中新增 IAM 政策資訊。
- 您總共可以包含最多 24 個 FCP 維度。建立並發佈後，您無法修改現有維度，但可以新增新維度（上限為 24)。

如果您有任何問題，請聯絡AWS Marketplace 賣方營運團隊。

修改您的軟體以使用計量服務

您需要修改軟體以記錄客戶用量、將每小時用量報告傳送至計量服務，以及處理新的故障模式。軟體的運作與定價無關，但軟體需要了解用量類別、使用方式和任何維度。

測量耗用量

您的軟體必須決定所選用量類別的數量，以及客戶已使用的維度。此值每小時會傳送給計量服務一次。在所有情況下，都假設您的軟體能夠測量、記錄和讀取資源的耗用量，以便每小時將其傳送至計量服務。

對於佈建的耗用，這通常是從軟體組態讀取為取樣值，但也可能是每小時記錄的最大設定值。對於並行耗用，這可能是週期性範例或每小時記錄的最大值。對於累積使用量，這會是每小時累積的值。

對於多個維度的定價，必須測量多個值並傳送至計量服務，每個維度一個。當您提供 AMI 時，這需要您的軟體使用已知的一組維度進行程式設計或設定。建立產品後，維度集無法變更。

對於每個定價案例，下表說明每小時測量耗用量的建議方法。

案例	如何測量
佈建的使用者	佈建使用者的目前數量（取樣）。 -或- 佈建使用者數目上限（查看該小時）。
並行使用者	目前並行使用者數量（取樣）。 -或- 並行使用者數目上限（查看該小時）。 -或- 不同使用者的總數（查看該小時）。
佈建的主機	佈建主機的目前數量（取樣）。 -或- 佈建主機的數量上限（查看該小時）。

案例	如何測量
並行主機	並行主機的目前數量（取樣）。 -或- 並行主機數量上限（查看該小時）。 -或- 不同主機的總數（查看該小時）。
佈建頻寬	目前佈建的頻寬設定（取樣）。 -或- 佈建頻寬上限（查看該小時）。
累積資料	目前 GB 的資料已儲存（取樣）。 -或- 儲存的最大 GB 資料（查看該小時）。 -或- 該小時新增或處理的資料總 GB。 -或- 該小時處理的資料總 GB。

廠商計量標記（選用）

廠商計量標記可協助獨立軟體廠商 (ISVs) 更詳細地了解買方的軟體用量，並協助他們執行成本分配。

若要標記買方的軟體用量，您需要判斷如何分配成本。首先詢問您的買方，他們想要在成本分配中看到什麼。然後，您可以將用量分割到您追蹤買方帳戶的屬性。屬性的範例包括 `Account ID`、`Cost Centers`、 `Business Unit`和您產品的其他相關中繼資料。這些屬性會以標籤的形式公開給買方。使用標籤，買方可以在帳單 AWS 主控台 (https://console.aws.amazon.com/costmanagement/：//) 中

檢視依標籤值分割為用量的成本。廠商計量標記不會變更您報告的價格、維度或總用量。它可讓您的客戶依適合您產品的類別檢視其成本。

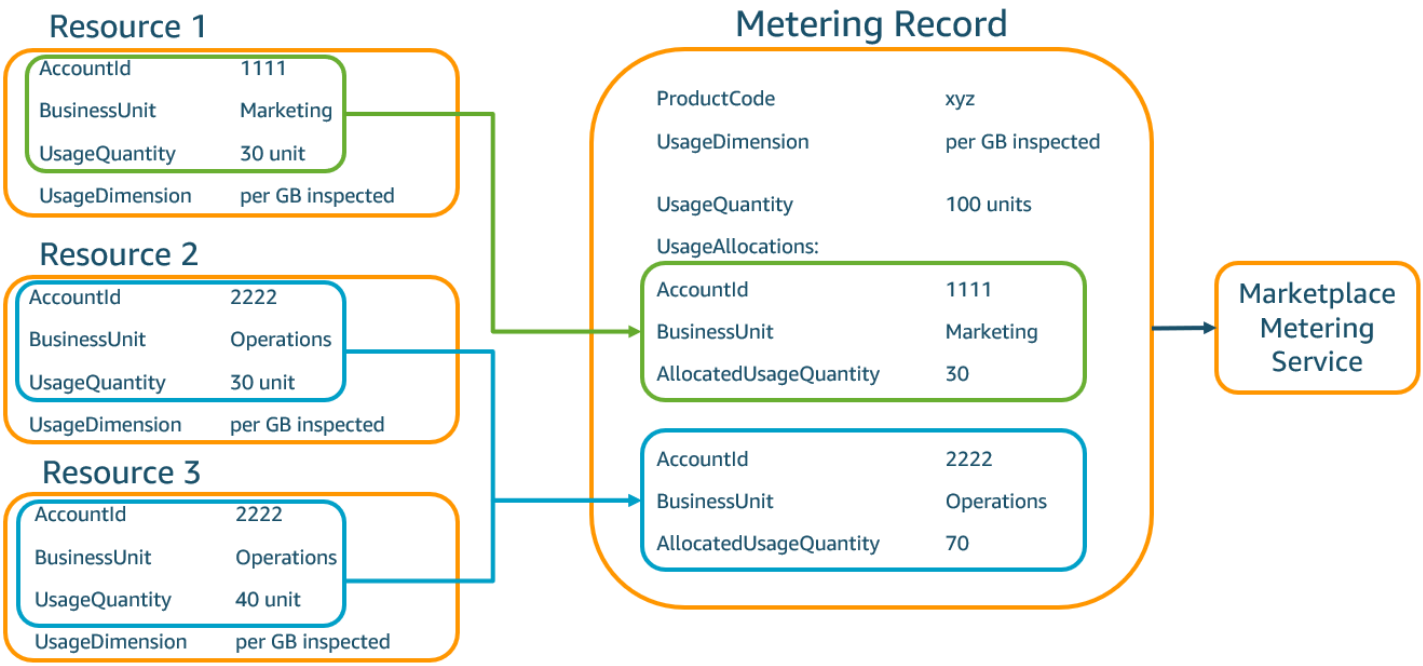
在常見的使用案例中，買方使用 訂閱您的產品 AWS 帳戶。買方也有許多與相同產品訂閱相關聯的使用者。您可以使用索引鍵為 的標籤建立用量配置Account ID，然後將用量配置給每個使用者。在此情況下，買方可以在帳單和成本管理主控台中啟用Account ID標籤，並分析個別使用者用量。

賣方體驗

賣方可以使用相同的標籤集彙總資源的計量記錄，而不是彙總所有資源的用量。例如，賣方可以建構包含不同 儲存貯體的計量記錄UsageAllocations。每個儲存貯UsageQuantity體代表一組標籤，例如 AccountId和 BusinessUnit。

在下圖中，資源 1 具有一組唯一的 AccountId 和 BusinessUnit標籤，並在計量記錄中顯示為單一項目。

資源 2 和資源 3 都有相同的AccountId標籤 2222和相同的BusinessUnit標籤 Operations。因此，它們會合併為計量記錄中的單一UsageAllocations項目。



賣方也可以將沒有標籤的資源合併為UsageAllocation具有配置用量數量的單一資源，並將其做為中的其中一個項目傳送UsageAllocations。

限制包括：

- 標籤數量 – 5

- 大小 UsageAllocations（基數）– 2，500

驗證包括：

- 標籤索引鍵和值允許的字元 – a-zA-Z0-9+ -=._：\/@
- 跨UsageAllocation清單的標籤上限 – 5
- 兩個 UsageAllocations 不能有相同的標籤（也就是標籤索引鍵和值的相同組合）。如果是這種情況，他們必須使用相同的 UsageAllocation。
- AllocatedUsageQuantity 的 總和UsageAllocation必須等於 UsageQuantity，也就是彙總用量。

買方體驗

下表顯示買方啟用 AccountId和BusinessUnit廠商標籤後買方體驗的範例。

在此範例中，買方可以在其成本用量報告中看到配置的用量。廠商計量的標籤使用字首“aws:marketplace:isv”。買方可以在帳單和成本管理、成本分配標籤、 AWS產生的成本分配標籤下啟用它們。

成本用量報告的第一列和最後一列與賣方傳送到計量服務的內容相關（如賣方體驗範例所示）。

成本用量報告（簡化）

ProductCode	買方	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	網路：每個 (GB) 已檢查	70	2222	作業
xyz	111122223333	網路：每個 (GB) 已檢查	30	3333	財務
xyz	111122223333	網路：每個 (GB) 已檢查	20	4444	IT
xyz	111122223333	網路：每個 (GB) 已檢查	20	5555	行銷部門

ProductCode	買方	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	網路：每個 (GB) 已檢查	30	1111	行銷部門

如需程式碼範例，請參閱 MeterUsage 使用用量分配標記 （選用）

設定自訂計量

如需使用 AWS Marketplace Metering Service 的詳細資訊，請參閱使用 設定 AMI 產品的自訂計量 AWS Marketplace Metering Service。

使用 設定 AMI 產品的自訂計量 AWS Marketplace Metering Service

> **Note**
>
> 對於具有自訂計量定價的 AMI 型產品，您的軟體必須使用連接至 Amazon EC2 執行個體之 Amazon Elastic Compute Cloud 的 IAM 角色臨時 AWS 憑證來呼叫 MeterUsage API。不支援使用長期存取金鑰。

AWS Marketplace Metering Service 是一項定價和計量功能，可用來依用量類別直接收取軟體費用。有五種用量類別：使用者、資料、頻寬、主機或單位。您可以使用 Metering Service 搭配 Amazon Machine Image (AMI) 型、容器型和軟體即服務 (SaaS) 型產品。下列各節提供如何使用 AWS Marketplace Metering Service 設定自訂計量的詳細資訊。

AWS Marketplace Metering Service 可啟用數個新案例。例如，如果您的軟體監控主機，您可以針對每個受監控的主機收取費用。您可以根據主機大小有不同的價格，並按每小時監控的並行主機數量收費。同樣地，如果您的軟體允許組織中的許多使用者登入，您可以按使用者數量收費。每小時會向客戶收取佈建使用者總數的費用。

如需詳細資訊，請參閱 AWS Marketplace 計量服務 API 參考。

如需整合 AWS Marketplace Metering Service API for AMI 型產品與自訂計量定價的詳細資訊，請參閱AWS Marketplace 賣方研討會自訂單位實驗室定價的列出 AMI 產品。

主題

- 要求
- 呼叫 AWS Marketplace Metering Service
- 失敗處理
- 限制
- 程式碼範例

要求

所有使用 Metering Service 的 AMI 型軟體都必須符合下列要求：

- 您的軟體必須透過 AWS Marketplace Amazon Machine Image (AMI) 從 啟動。
- 如果您在 中有現有產品 AWS Marketplace，則必須提交新的 AMI 並建立新的產品以啟用此功能。
- 所有軟體都必須使用 AWS Identity and Access Management (IAM) 角色佈建。最終客戶必須將 IAM 角色新增至使用者使用 軟體佈建的 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體。當您透過 部署軟體時，可選擇是否使用 IAM 角色 AWS Marketplace。部署 AWS Marketplace Metering Service 軟體時，這是必要的。
- 您的軟體必須能夠以某種方式判斷使用量。

呼叫 AWS Marketplace Metering Service

您的軟體必須每小時呼叫 Metering Service，並記錄該小時的耗用量值。

當軟體啟動時，它應該記錄啟動該軟體minute-of-the-hour。這稱為開始分鐘。開始分鐘的每個小時，您的軟體都必須判斷該小時的耗用值，並呼叫計量服務。如需如何取得此值的資訊，請參閱修改您的軟體以使用計量服務。

若要在啟動分鐘每小時喚醒，您的軟體必須使用下列其中一種方法：

- 軟體中的執行緒。
- 以執行個體或軟體啟動的協助程式程序。
- 應用程式啟動期間設定的 cron 任務。

> **Note**
>
> 您的軟體必須使用客戶執行個體上設定的 IAM 角色呼叫 AWS Marketplace Metering Service，並指定耗用量維度和數量。

您的軟體可以使用軟體 AWS 開發套件來呼叫 AWS Marketplace Metering Service，類似於下列範例實作：

1. 使用執行個體描述檔來建立服務用戶端。這需要為 EC2 執行個體設定的角色。SDK 會自動重新整理角色登入資料。
2. 每小時讀取您的軟體組態和狀態，以判斷該小時的耗用值。這可能包括收集value-per-dimension。
3. 使用下列參數在 SDK 用戶端上呼叫 `meterUsage`方法（針對具有 用量的每個維度另外呼叫）：
   - `timestamp` – 記錄小時的時間戳記 (UTC)。
   - `productCode` – 指派給軟體的產品程式碼。
   - `dimension` – 指派給軟體的維度（或維度）。
   - `quantity` – 小時的耗用值。
   - `allocations` –（選用）您可以跨追蹤的屬性提供用量的配置。這些配置必須加總至記錄中的總耗用量。對於買方，這些會在其帳單工具（例如 AWS 帳單與成本管理 主控台）中顯示為潛在的成本分配標籤。買方必須在其帳戶中啟用標籤，才能使用這些標籤追蹤其成本。

此外，您的軟體必須呼叫區域內 AWS Marketplace Metering Service 端點。您的產品必須設定正確的區域端點，因此 `us-east-1` 會將記錄傳送至`us-east-1`端點，並將記錄`us-west-2`傳送至`us-west-2`端點。進行區域內呼叫可為買方提供更穩定的體驗，並防止不相關的區域可用性可能影響另一個區域中執行的軟體的情況。

當您將計量記錄傳送到 服務時，您必須連線到您所在區域的 AWS Marketplace Metering Service。使用 `getCurrentRegion()` 協助程式方法來判斷 EC2 `MeteringServiceClient` 執行個體執行所在的區域，然後將此區域資訊傳遞給建構函式。如果您未在 SDK 建構函數 AWS 區域 中指定 ，則會使用預設`us-east-1`區域。如果您的應用程式嘗試對服務進行跨區域呼叫，則呼叫會遭到拒絕。如需詳細資訊，請參閱判斷應用程式的目前區域和 getCurrentRegion()。

失敗處理

您的產品必須將計量記錄傳送至公有網際網路端點 服務，以便擷取用量並進行計費。由於客戶可以修改網路設定，以防止您的計量記錄交付，因此您的產品應該透過選擇失敗模式來考慮這一點。

> **Note**
>
> 某些計量失敗可能是連線至 時的暫時性問題 AWS Marketplace Metering Service。 AWS Marketplace 強烈建議在指數退避的情況下實作重試最多 30 分鐘，以避免短期中斷或網路問題。

一般而言，軟體可能會開啟失敗（提供警告訊息但維持完整功能）或關閉失敗（停用應用程式中的所有功能，直到重新建立連線為止）。您可以選擇讓開啟、關閉或應用程式的特定項目失敗。強烈建議您避免在計量失敗不到兩小時後關閉。

作為部分開啟失敗的範例，您可以繼續允許存取軟體，但不允許買方修改軟體設定。或者，買方仍然可以存取軟體，但無法建立其他使用者。您的軟體負責定義和強制執行此失敗模式。提交 AMI 時，必須包含您軟體的失敗模式，之後無法變更。

限制

在設計和提交啟用 Metering Service 的軟體時，請記住這些限制：

- 您客戶的 IAM 角色和網際網路閘道需求 – 您的客戶必須具有網際網路閘道，而且必須使用具有特定許可的 IAM 角色來啟動您的軟體。如需詳細資訊，請參閱AWS Marketplace 計量和權利 API 許可。如果不符合這兩個條件，您的軟體就無法連線至 Metering Service。
- 無法將新的或變更用量類別新增至現有的 Metering Service 產品 – 當客戶訂閱您的軟體產品時，他們同意條款與條件。使用 Metering Service 變更產品中的用量類別需要新的產品和新的訂閱。
- 無法將維度變更為現有 Metering Service 產品 – 當客戶訂閱您的軟體產品時，他們同意條款與條件。使用 Metering Service 變更產品中的維度需要新的產品和新的訂閱。您可以將新維度新增至現有產品，上限為 24。
- 缺少免費試用和年度訂閱 – Metering Service 產品不支援啟動時的免費試用和年度訂閱。
- 多執行個體或叢集型部署考量 – 有些軟體會部署為多執行個體部署的一部分。當您設計軟體時，請考慮如何測量耗用量以及在何處發出計量記錄。

程式碼範例

以下程式碼範例可協助您整合 AMI 產品與發佈和維護產品所需的 AWS Marketplace APIs。

`MeterUsage` 使用用量分配標記（選用）

下列程式碼範例適用於使用定價模型的 AMI 產品。Python 範例會傳送具有適當用量分配標籤的計量記錄給 AWS Marketplace ，以向您的客戶收取pay-as-you-go。

```
# 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
)
```

如需 的詳細資訊MeterUsage，請參閱《計量服務 API 參考》中的 MeterUsage。 AWS Marketplace

回應範例

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

上的 AMI 產品的合約定價 AWS Marketplace

以 Amazon Machine Image (AMI) 為基礎的產品的合約定價表示買方需為單一 AMI 產品或具有 AWS CloudFormation 堆疊的單一 AMI 支付預付費用。對於具有合約定價的 AMI 型產品， 會根據您與客戶之間的合約，預先或根據您定義的付款排程向您的客戶 AWS Marketplace 收費。之後，他們有權使用這些資源。本主題提供有關合約定價的詳細資訊。

若要設定定價，請選擇您提供給客戶的一或多個合約持續時間。您可以為每個合約期間輸入不同的價格。您的選項為 1 個月、12 個月、24 個月和 36 個月的持續時間。對於私有優惠，您可以指定以月為單位的自訂持續時間 （最多 60 個月）。

> **Note**
>
> 免費試用不適用於具有合約定價的 AMI 產品。

選擇最能描述您產品定價的類別。定價類別會顯示在 AWS Marketplace 網站上的客戶。您可以選擇頻寬 (GB/s、MB/s)、資料 (GB、MB、TB)、主機、請求、層或使用者。如果沒有任何預先定義的類別符合您的需求，您可以選擇更通用的單位類別。

優惠最多允許新增 24 個維度。每個維度都需要下列資料：

- 合約類別 – 對於沒有以消耗為基礎的定價的合約產品，您可以選擇最接近合約中維度類別的類別，或者如果沒有值類似於合約中維度的單位，請選擇單位
- 合約單位 – 根據選取的類別，為最符合您維度的單位選擇其中一個可用值。
- 合約維度 允許多次購買 – 此欄位用於指示優惠是分層定價優惠還是非分層優惠：

  分層優惠 – 允許買方僅訂閱優惠中其中一個可用的維度。分層優惠中的維度沒有數量的概念。簽署具有特定維度的合約基本上表示買方已選擇該維度所指示的特定功能。

  非分層優惠 – 允許客戶在合約中採購多個維度，並允許他們採購每個維度的多個單位。

  為此欄位設定 true 值表示優惠是非分層優惠。為此欄位設定 false 值表示優惠是分層優惠。

使用產品載入表單 (PLF) 為您的 AMI 產品建立合約時，您必須為您的定價維度定義下列欄位：

- 合約 DimensionX API 名稱 – 應該出現在買方 AWS License Manager 帳戶中產生的授權中的名稱。此名稱也會做為 `Checkoutlicense` API 呼叫`Entitlement`中 `Name` 的值。
- 合約維DimensionX 顯示名稱 – 面向客戶之維度的名稱，會顯示在 AWS Marketplace 網站的產品詳細資訊和採購頁面上。建立易於使用的名稱。名稱的長度上限為 24 個字元。清單為公有之後，`Name`的值就無法變更。
- Contracts DimensionX 描述 – 面向客戶的維度描述，提供產品維度的其他資訊，例如特定維度提供的功能。描述的長度上限為 70 個字元。
- 合約 DimensionX 數量 – 用於計算產品協議修訂時的比例。對於所有合約優惠，此欄位的這個值應該設定為 1。不應對其進行編輯。
- 合約 DimensionX 1-Month費率 – 針對此維度收取一個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援三個小數位數。
- 合約 DimensionX 12-Month費率 – 對維度收取 12 個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援三個小數位數。
- 合約 DimensionX 24-Month費率 - 針對維度收取 24 個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援三個小數位數。
- 合約 DimensionX 36-Month費率 - 針對維度收取 36 個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援三個小數位數。

範例：資料儲存應用程式

	1 個月價格	12 個月價格	24 個月價格	36 個月價格
未加密的資料 (GB)	1.50 USD/GB	16.00 USD/GB	30.00 USD/GB	60.00 USD/GB
加密資料 (GB)	$1.55/GB	16.60 USD/GB	$31.20/GB	61.20 USD/GB

範例：日誌監控產品

	1 個月價格	12 個月價格	24 個月價格	36 個月價格
基本（監控 10 個主機、監控 5 個容器）	100 美元	1000 美元	2000 美元	4000 美元

	1 個月價格	12 個月價格	24 個月價格	36 個月價格
標準（監控 20 個主機、監控 10 個容器）	$200	2000 美元	4000 美元	8000 美元
Pro（監控 40 個主機、監控 20 個容器）	400 美元	4000 美元	8000 美元	16，000 美元
每小時監控的其他主機	10 美元	100 美元	$200	400 美元
每小時監控的其他容器	10 美元	100 美元	$200	400 美元

> **Note**
>
> 價格的持續時間如下：1 個月、12 個月、24 個月或 36 個月。您可以選擇為您的產品提供一或多個選項。每個維度的持續時間必須相同。
>
> Example
>
> 例如，如果您有 `ReadOnlyUsers`和 `AdminUsers`維度，如果您提供 ReadOnlyUsers 的年價，您`AdminUsers`也必須提供 的年價。

自動續約

當客戶 AWS Marketplace 使用 AMI 合約購買您的產品時，他們可以同意自動續約合約條款。客戶每個月或 1、2 或 3 年繼續支付權利。

客戶可以隨時修改續約設定。如需詳細資訊，請參閱《 AWS Marketplace 買方指南》中的修改現有合約。

使用 將授權與 AMI 型產品建立關聯 AWS License Manager

對於具有合約定價的 Amazon Machine Image (AMI) 型產品，您可以使用 AWS License Manager 將授權與產品建立關聯。 AWS License Manager 是一種授權管理工具，可讓您的應用程式追蹤和更新客戶

購買的授權（也稱為權利）。整合完成後，您可以在其上發佈您的產品清單 AWS Marketplace。下列各節提供使用 AWS License Manager 將授權與以 AMI 為基礎的產品建立關聯的詳細資訊。

如需詳細資訊 AWS License Manager，請參閱 AWS License Manager 使用者指南和 AWS CLI 命令參考的 AWS License Manager一節。

如需 AWS License Manager 整合 AMI 型產品與合約定價的詳細資訊，請參閱AWS Marketplace 賣方研討會預付付款實驗室定價的列出 AMI 產品。

> **Note**
>
> - 客戶無法在合約到期期間之後啟動 AMI 的新執行個體。不過，在合約期間，他們可以啟動任意數量的執行個體。這些授權不會鎖定節點或繫結至特定執行個體。
> - 私有優惠建立 – 賣方可以使用 中的私有優惠建立工具，為產品產生私有優惠 AWS Marketplace 管理入口網站。
> - 報告 – 您可以在 的報告區段中設定 Amazon S3 儲存貯體，以設定資料饋送 AWS Marketplace 管理入口網站。如需詳細資訊，請參閱 中的賣方報告、資料饋送和儀表板 AWS Marketplace。

主題

- 授權模型
- 整合工作流程
- License Manager 整合先決條件
- 將 AMI 型產品與 整合 AWS License Manager
- 授權續約和升級

授權模型

AWS Marketplace 與 整合 AWS License Manager 支援兩種授權模型：

- 可設定的授權模型
- 分層授權模型

可設定的授權模型

在買方取得授權之後，可設定的授權模型（也稱為可量化授權模型）可讓買方獲得特定數量的資源。

您可以設定定價維度和每個單價。然後，買方可以選擇他們想要購買的資源數量。

Example 定價維度和每個單價的

您可以設定定價維度（例如資料備份）和每單位價格（例如 $30/單位）

買方可以選擇購買 5、10 或 20 個單位。

您的產品會追蹤和測量用量，以測量消耗的資源數量。

使用組態模型時，會以下列兩種方式之一計算權利：

- 下調授權
- 浮動授權

下調授權

授權是從使用時允許的授權數量集區中提取。該權利會永久簽出，且無法傳回至授權集區。

Example 處理有限數量的資料

使用者有權處理 500 GB 的資料。隨著他們繼續處理資料，數量會從 500 GB 的集區中提取，直到所有 500 GB 授權都用盡為止。

對於提取授權，您可以使用 `CheckoutLicense` API 操作來查看已耗用的授權單位。

Example 的備份到 Amazon S3 的單位/年

您有一個儲存產品，允許備份到 Amazon Simple Storage Service 最多 1024 個單位的資料一年。您可以使用多個 Amazon EC2 執行個體啟動您的應用程式。您的應用程式具有追蹤和彙總資料的機制。您的軟體會在每次備份時以產品 ID 呼叫 `CheckoutLicense` API 操作，或以固定間隔更新耗用的數量。

在此範例中，您的軟體會呼叫 `CheckoutLicense` 來檢查 10 個單位的資料。當總容量達到客戶購買的備份限制時，API 呼叫會失敗。

請求

```
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=l0, Unit=Count" \
--client-token "AKIAIOSFODNN7EXAMPLE"
```

回應

```
{
  "CheckoutType": "PERPETUAL",
  "EntitlementsAllowed": [
    {
      "Name": "DataConsumption",
      "Count": 10,
      "Units": "Count",
      "Value": "Enabled"
    }
},
  "Expiration":    "2021-04-22Tl9:02: 36",
  "IssuedAt": "2021-04-22Tl8:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken":  "AKIAIOSFODNN7EXAMPLE"
}
```

浮動授權

授權在使用後會傳回至允許授權數量的集區。

Example 來自固定上限的使用者數量

使用者有權在應用程式上同時存取 500 名使用者。當使用者登入和登出時，會繪製使用者並傳回 500 名使用者的集區。不過，應用程式無法從集區中提取超過 500 個使用者，因為 500 個同時使用者是固定的上限。

對於浮動授權，您可以使用 `CheckInLicense` API 操作將授權單位傳回權利集區。

Example 一年的並行使用者數量

您的產品會根據並行使用者數量定價。客戶為 10 個使用者購買授權一年。客戶提供 AWS Identity and Access Management (IAM) 許可來啟動軟體。當使用者登入時，您的應用程式會呼叫 `CheckoutLicense` API 操作，將數量減少 1。當使用者登出時，應用程式會呼叫 `CheckInLicense` API 操作，將該授權傳回至集區。如果您不呼叫 `CheckInLicense`，授權單位會在 1 小時後自動簽入。

> **Note**
>
> 在下列請求中，`key-fingerprint` 不是預留位置值，而是將發佈所有授權的指紋實際值。

請求

```
linux-machine ~]$ aws license-manager checkout-license\
--product-sku "2205b290-19e6-4c76-9eea-377d6bf7la47" \
--checkout-type "PROVISIONAL" \
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements "Name=ReadOnlyUSers, Value=l0, Unit=Count" \
--client-token "AKIAIOSFODNN7EXAMPLE"
```

回應

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "ReadOnlyUsers",
      "Count": 10,
      "Units": "Count",
      "Value": "Enabled"
    }
},
  "Expiration": "2021-04-22Tl9:02: 36",
  "IssuedAt": "2021-04-22Tl8:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

分層授權模型

在買方取得授權之後，分層授權模型會授予買方特定層級或層級的應用程式功能。

您可以為您的產品建立方案，例如 Basic、Intermediate 和 Premium。然後，買方會選取其中一個預先定義的層。

應用程式不需要追蹤或計量應用程式的用量。

使用分層授權模型時，不會計算權利，而是表示客戶購買的服務層級。

如果您想要同時提供綁定功能，建議您使用分層授權模型。

Example 基本、中級和高級方案

客戶可以為軟體的三個可能方案之一簽署合約：基本、中級或高級。每個方案都有自己的定價。您的軟體可以透過叫用 CheckoutLicense API 操作並在請求中指定所有可能的層來識別客戶已註冊的層。

請求的回應包含與客戶已取得的方案對應的權利。根據此資訊，軟體可以佈建適當的客戶體驗。

請求

```
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"
```

回應

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "IntermediateTier",
      "Units": "None"
    }
},
  "Expiration": "2021-04-22Tl9:02:36",
  "IssuedAt": "2021-04-22Tl8:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

整合工作流程

下列步驟顯示將您的 AMI 產品與 整合的工作流程 AWS License Manager：

1. 賣方建立具有 AWS License Manager 整合的產品。
2. 賣方列出產品 AWS Marketplace。
3. 買方在 上找到 AWS Marketplace 並購買產品。

4. 授權會傳送給其 中的買方 AWS 帳戶。
5. 買方透過啟動 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、Amazon Elastic Container Service (Amazon ECS) 任務或 Amazon Elastic Kubernetes Service (Amazon EKS) Pod 軟體來使用軟體。客戶使用 IAM 角色部署 。
6. 軟體會讀取買方 AWS License Manager 帳戶中的授權、探索購買的權利，並相應地佈建功能。

> **Note**
>
> License Manager 不會執行任何追蹤或更新；這由賣方的應用程式完成。

License Manager 整合先決條件

在發佈產品之前，您必須執行下列動作：

1. 在 中建立新的 AMI 產品 AWS Marketplace 管理入口網站，並記下其產品代碼。
2. 在產品載入表單 (PLF) 中填寫必要的價格資訊，並將其傳回給我們進行處理。
3. 針對執行應用程式的 任務或 Pod 使用 IAM 角色，並具有呼叫 `CheckoutLicense`、`ExtendLicenseConsumption`和 所需的 IAM 許可`CheckInLicense`。

   必要的 IAM 許可詳述於下列 IAM 政策。

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"VisualEditor0",
         "Effect":"Allow",
         "Action":[
            "license-manager:CheckoutLicense",
            "license-manager:GetLicense",
            "license-manager:CheckInLicense",
            "license-manager:ExtendLicenseConsumption",
            "license-manager:ListReceivedLicenses"
         ],
         "Resource":"*"
      }
```

```
    ]
}
```

4. 使用您定義的所有定價維度的記錄對 `RegisterUsage` API 操作進行測試呼叫。

將 AMI 型產品與 整合 AWS License Manager

您可以使用 AWS License Manager API 將您的 AMI 型產品與 License Manager 整合。使用 AMI 型產品啟動 AWS Marketplace Amazon EC2 執行個體。

> **Note**
>
> 在執行下列程序the section called “License Manager 整合先決條件”之前，請確定您已完成 。

將您的 AMI 型產品與 License Manager 整合

1. 完成the section called “在 License Manager 中建立測試授權”中的程序。您必須在 License Manager 中建立測試授權，以測試整合。
2. 使用您在步驟 1 中取得的授權 Amazon Resource Name (ARN) 執行 GetLicense API 操作。請注意`GetLicense`回應的 `KeyFingerprint` 屬性值，以供日後使用。
3. 在您的應用程式中下載並包含最新的公有 AWS SDK。
4. 若要確認買方有權為您的應用程式使用授權，請執行 CheckoutLicense API 操作。使用您在步驟 1 中取得之測試授權的權限詳細資訊和金鑰指紋。

   如果找不到授權的權限，或超過權利上限計數，`CheckoutLicense`則 API 操作會傳回 `NoEntitlementsAllowedException`。如果權利有效或可供使用，`CheckoutLicense`操作會傳回成功回應，其中包含請求的權限及其值。
5. （僅適用於浮動權利）使用`CheckoutLicense`回應中`LicenseConsumptionToken`收到的 執行 CheckinLicense API 操作。此動作會將先前簽出的權限釋出回可用權限的集區。
6. 成功驗證 License Manager 與您在步驟 1 中建立的測試授權整合後，請將程式碼中的金鑰指紋更新為 `aws:294406891311:AWS/Marketplace:issuer-fingerprint`。現在，您已準備好使用 發行的授權 AWS Marketplace。

遵循將應用程式建置至 AMI 產品的發行程序，然後 AWS Marketplace 依照產品發佈程序提交產品至 。

在 License Manager 中建立測試授權

您可以使用 AWS Command Line Interface (AWS CLI) 第 2 版在其中建立測試授權 AWS License Manager。此測試授權僅用於驗證和測試 AWS License Manager 整合。測試完成後，您可以刪除測試授權。實際授權是由 AWS Marketplace 使用不同的金鑰指紋產生。

AWS Marketplace 支援兩種類型的 權利 AWS License Manager。不過，一個產品只能啟用一種類型。建立授權時，包括測試授權，您必須指定下列其中一種權利類型：

分層權利 – 分層授權模型讓客戶有權使用某些應用程式功能。客戶無法定義想要購買的單位數量。不過，他們可以選取單一預先定義的套件或層。客戶稍後可以修改合約以訂閱另一個方案。

可設定的權利 – 當客戶取得授權時，可設定的授權模型會將權利授予特定數量的資源。客戶會選擇要在訂閱程序期間購買的單位數量，並根據單價計費。客戶也可以訂閱多個維度。

在 `CheckoutLicense` API 操作中使用的必要參數如下：

- `CheckoutType` – 有效值為 `Perpetual`或 `Provisional`：
  - `Perpetual` – 用於從集區中耗盡已簽出的權限數量時。範例：買方有權處理 500 GB 的資料。隨著他們繼續處理資料，數量會從 500 GB 的集區中消耗。取得已購買授權的狀態，了解授權是否過期或即將過期，以傳送通知給客戶。
  - `Provisional` – 用於浮動授權權利，其中權利會從集區中簽出，並在使用後傳回。範例：使用者有權在應用程式中同時存取 500 名使用者。當使用者登入和登出時，會繪製使用者並傳回 500 名使用者的集區。如需浮動授權權利的詳細資訊，請參閱 License Manager 中的賣方發行 AWS 的授權。
- `ClientToken` – 唯一且區分大小寫的識別符，以確保發生確切的結果，無論嘗試多少次都相同。我們建議您針對每個請求使用隨機通用唯一識別碼 (UUID)。
- `Entitlements` – 要簽出的權利清單。
  - 對於分層權利，請提供 `Name`和 `Unit` 屬性，如下所示：

```
{

"Name": "<Entitlement_Name>",

"Unit": "None"

}
```

  - 對於可設定的權限，請提供 `Name`、 `Unit`和 `Value`屬性，如下所示：

```
{

"Name": "<Entitlement_Name>",

"Unit": "<Entitlement_Unit>",

"Value": <Desired_Count>{

}
```

- KeyFingerprint – 使用此金鑰指紋來驗證授權是由 發出 AWS Marketplace。所發行授權的金鑰指紋 AWS Marketplace 如下所示：

  aws:294406891311:AWS/Marketplace:issuer-fingerprint

- Product SKU – 具有與 產品相關聯之全域唯一識別符 (GUID) 格式 AWS Marketplace 的產品 ID。

Example 可設定權利的

以下是使用 CheckoutLicense API 操作來檢查名為 之可設定權利的請求範例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 分層權利的

以下是使用 CheckoutLicense API 操作來檢查名為 之功能權利的請求範例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"
```

為您的 AMI 型產品建立測試授權

1. 在已安裝 AWS CLI v2 的本機環境中，執行下列指令碼。指令碼會建立測試授權並設定適當的產品詳細資訊。

> **Note**
>
> 使用 AWS 帳戶 與您要部署和測試軟體的測試 AWS 帳戶 不同的 。無法在相同的 中建立、授予和簽出授權 AWS 帳戶。

```
#!/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. 使用 AWS License Manager 主控台授予授權。如需詳細資訊，請參閱《 License Manager 使用者指南》中的分發權利。

3. 登入做為買方帳戶的 AWS 帳戶 ，您將在其中部署和測試軟體。這必須與建立和授予授權 AWS 帳戶 的 AWS 帳戶 不同。
4. 前往 AWS License Manager 主控台以接受並啟用授予的授權。如需詳細資訊，請參閱《 License Manager 使用者指南》中的管理您授予的授權。
5. 在您的環境中執行下列命令。

```
# 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)
```

先前的命令使用 PROVISIONAL做為 CheckoutType 參數的值。如果權利使用提取授權，請使用 PERPETUAL做為 值。

License Manager API 呼叫

若要管理存放在客戶 License Manager 帳戶中的授權，您的軟體可以使用下列 API 呼叫：

- GetLicense – 取得已購買授權的狀態，了解授權是否已過期或即將過期，以傳送通知給客戶。
- CheckoutLicense – 探索使用者已購買的授權。當使用者已取用一些授權數量時，您也可以使用它來更新授權數量。使用 CheckoutLicense，您可以繼續檢查客戶使用的授權數量。當客戶耗盡所有授權時，此呼叫會傳回錯誤。如需執行 的建議節奏資訊CheckoutLicense，請參閱 the section called “授權續約和升級”。
- ExtendLicenseConsumption – 在浮動維度的情況下，當軟體簽出授權時，會在 60 分鐘後自動將授權傳回至集區。如果您想要延長授權保持簽出的時間，您的軟體可以呼叫 ExtendLicenseConsumption 將授權再延長 60 分鐘。
- CheckInLicense – 如果是浮動維度，當您想要將授權傳回權利集區時，請使用 CheckInLicense。
- ListReceivedLicenses – 列出買方購買的授權。

授權續約和升級

客戶可以在 上續約或升級其授權 AWS Marketplace 管理入口網站。進行其他購買後， AWS Marketplace 會產生反映新權利的授權新版本。您的軟體會使用相同的 API 呼叫讀取新權利。在 License Manager 整合方面，您不需要採取任何不同的動作來處理續約和升級。

由於授權續約、升級、取消等，我們建議您的產品在使用產品時定期執行 `CheckoutLicense` API 呼叫。透過定期使用 `CheckoutLicense` API 操作，產品可以偵測權利的變更，例如升級和過期。

建議您每 15 分鐘執行一次 `CheckoutLicense` API 呼叫。

在 上接收 AMI 產品的 Amazon SNS 通知 AWS Marketplace

若要接收您產品的客戶訂閱變更通知，您可以訂閱 Amazon Simple Notification Service (Amazon SNS) 主題，以在產品建立期間 AWS Marketplace 提供給您。例如，您可以知道客戶何時接受私有優惠。 Amazon SNS 主題： aws-mp-subscription-notification是適用於 AMI 產品的 Amazon SNS 主題。本主題會在買方訂閱或取消訂閱產品時通知您。此通知適用於每小時定價模型，包括每小時和每小時與每年。如需詳細資訊，請參閱下列區段。

> **Note**
>
> 在產品建立過程中，會為您的產品建立 Amazon SNS 主題。若要訂閱通知，您需要 Amazon SNS 主題的 Amazon Resource Name (ARN)（ 例如 `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`)。ARN 不適用於伺服器產品的賣方入口網站。請聯絡 AWS Marketplace 操作團隊以請求 ARN。

主題

- Amazon SNS 主題： aws-mp-subscription-notification
- 訂閱 Amazon SQS 佇列至 Amazon SNS 主題

Amazon SNS 主題： aws-mp-subscription-notification

`subscribe-success` 和 `subscribe-fail`動作`aws-mp-subscription-notification`主題中的每個訊息都具有下列格式。

```
{
```

```
    "action": "action-name",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123"
}
```

<action-name> 會根據通知而有所不同。可能的動作包括：

- subscribe-success
- subscribe-fail
- unsubscribe-pending
- unsubscribe-success

只有在動作為 subscribe-success或 時， offer-identifier才會包含在通知中subscribe-fail。當動作為 unsubscribe-pending或 時，它不會包含在通知中unsubscribe-success。對於 2024 年 1 月之前建立的優惠，此識別符僅包含在私有優惠的通知中。對於 2024 年 1 月及更新版本建立的優惠，此識別符會包含在所有優惠的通知中，包括私有優惠和公開優惠。

如需優惠類型的資訊，請參閱 DescribeEntity API 的回應或協議續約儀表板中的協議優惠可見性。

> **Note**
>
> 對於 DescribeEntity API，如果您 AWS 帳戶 在以該優惠規則為目標的面向帳戶中找到 ，則這是私有優惠。如果 AWS 帳戶 帳戶中沒有以該優惠規則為目標的面向為目標的 ，則這是公開優惠。

訂閱 Amazon SQS 佇列至 Amazon SNS 主題

我們建議訂閱 Amazon SQS 佇列至提供的 SNS 主題。如需建立 SQS 佇列和訂閱佇列至主題的詳細說明，請參閱《Amazon Simple Notification Service 開發人員指南》中的訂閱 Amazon SQS 佇列至 Amazon SNS 主題。

> **Note**
>
> 您只能從 AWS 帳戶 用來銷售產品的 訂閱 AWS Marketplace SNS 主題。不過，您可以將訊息轉送到不同的 帳戶。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的將 Amazon SNS 訊息傳送至不同帳戶中的 Amazon SQS 佇列。

輪詢 SQS 佇列以取得通知

將 SQS 佇列訂閱 SNS 主題後，訊息會儲存在 SQS 中。您必須定義持續輪詢佇列、尋找訊息並相應地處理它們的服務。

的 AMI 產品檢查清單 AWS Marketplace

提交 Amazon Machine Image (AMI) 產品之前 AWS，請使用此檢查清單來驗證您的提交。遵循這些要求有助於確保有效率的發佈程序。

產品用量

- AMI 可立即生產。
- 產品用量不受時間或任何其他測量的限制。
- AMI 與一鍵式履行體驗相容。
- 使用產品所需的一切都在 軟體中，包括用戶端應用程式。需要外部相依性的產品，例如軟體套件或用戶端應用程式，請遵循包含適當揭露產品使用政策的 。
- AMI 符合強制性 的 AMI 型產品需求 AWS Marketplace。
- 使用產品不需要額外的授權。
- 買方不需要提供個人身分識別資訊 （ 例如，他們的電子郵件地址） 即可使用產品。

AMI 準備

- 您的產品名稱和描述必須符合您提供的 AMI 產品的描述欄位。
- 使用硬體虛擬機器 (HVM) 虛擬化和 64 位元架構。
- 產品不包含任何已知的漏洞、惡意軟體或病毒。
- 客戶可以透過網路存取執行個體，並使用管理存取。
- AMI 上沒有硬式編碼秘密。硬式編碼秘密的範例包括系統使用者和服務的預設密碼、私有金鑰和登入資料。
- AMI 上沒有硬式編碼的 SSH 授權公有金鑰。
- 測試「新增版本」驗證測試成功完成，沒有問題。

Windows AMI

- 對於 Windows Server 2012 和更新版本的作業系統，您的 AMI 使用最新版本的 EC2Launch v2。
- 如果您使用的是 EC2Launch v2，請完成下列操作：
  - 在 Amazon EC2Launch 設定中，選擇設定管理員帳戶下的隨機，以在執行時間產生管理員密碼。
  - 在 Amazon EC2Launch 設定中，選取重新啟用並在 Sysprep 之後啟動 SSM 服務。
  - 將 UserData 新增至 EC2 v2 任務組態。
- 對於 Windows Server 2012 和更新版本，請避免使用 EC2Config。如果需要 EC2Config，請確定您使用的是最新版本。
- 如果您使用 EC2Config，請在 AMI 的設定檔案中啟用下列參數：
  - `Ec2SetPassword`
  - `Ec2WindowsActivate`
  - `Ec2HandleUserData`
- 確保不存在訪客帳戶或遠端桌面使用者。

Linux AMI

- 禁止以超級使用者身分遠端登入。
- 禁止以密碼為基礎的遠端存取。

產品載入表單或產品索引標籤

- 所有必要欄位皆已完成。
- 所有值都在指定的字元限制內。
- 所有 URLs載入時都不會發生錯誤。
- 產品影像至少為 110 像素寬，比例介於 1：1 和 2：1 之間。
- 定價是針對所有已啟用的執行個體類型（適用於每小時、每小時每月定價和每小時年度定價模型）指定。
- 指定每月定價（適用於以小時為基礎的每月和每月定價模型）。

的 AMI 型產品需求 AWS Marketplace

AWS Marketplace 針對所有 Amazon Machine Image (AMI) 產品和產品維護下列政策。本節中的政策旨在為客戶提供安全、可靠且可靠的運算平台。

提交時，會檢閱所有產品及其相關中繼資料，以確保它們符合或超過目前的 AWS Marketplace 政策。這些政策會定期更新，以符合不斷發展的安全指導方針。 AWS Marketplace 會持續掃描產品，以確認現有的清單持續符合這些要求的任何變更。如果產品不合規， AWS Marketplace 將聯絡賣方更新其產品以符合新標準。在某些情況下，在問題解決之前，新訂閱用戶可能會暫時無法使用產品。此程序有助於為所有使用者維護 AWS Marketplace 平台的安全性和可信度。

在提交您的產品之前，強烈建議您在 中使用測試「新增版本」功能 AWS Marketplace 管理入口網站 ，以確保符合目前的政策。

主題

- AMI 產品賣方政策
- 安全政策
- 架構政策
- AMI 產品使用說明
- AMI 產品版本政策
- 客戶資訊政策
- 產品使用政策

AMI 產品賣方政策

所有 AMIs都必須遵守下列賣方政策：

- 根據預設， AWS Marketplace 賣方上限為 75 個公有 AMI 產品清單。所有超過限制的賣方都必須定期進行效能審查，而且可能需要限制效能不佳的清單。 AWS Marketplace 可以自行決定授予和撤銷提高此限制。

安全政策

一般政策

所有 AMIs都必須遵守下列政策：

- AMIs 必須通過 AWS Marketplace AMI 掃描工具執行的所有安全檢查，未顯示已知的漏洞或惡意軟體。
- AMIs必須使用目前支援的作業系統和軟體。不允許已達到生命週期結束的作業系統和軟體。
- 禁止執行個體服務的密碼型身分驗證。即使使用者在啟動時產生、重設或定義密碼，這也適用。不允許空密碼和空白密碼。

  例外狀況：

  - 在 Windows 執行個體EC2Config/EC2Launch上由 產生的管理員密碼。
  - 在沒有其他身分驗證方法的情況下，對主機服務（例如 Web 應用程式）的非管理存取權。如果使用高強度密碼，則必須為每個執行個體隨機產生密碼，由服務管理員用於初始身分驗證一次，並在第一次登入後立即變更。
- AMI 不得包含硬式編碼秘密，例如系統使用者和服務密碼（包括雜湊密碼）、私有金鑰或登入資料。
- AMIs不得請求 AWS 登入資料來存取 AWS 服務。如果您的產品需要存取 AWS 服務，則應為執行個體指派最低權限 AWS Identity and Access Management (IAM) 角色。使用者可以手動或使用 AWS CloudFormation 範本建立角色。使用 CloudFormation 交付方法的產品啟用單一 AMI 啟動時，使用說明必須包含建立最低權限 IAM 角色的明確指引。如需詳細資訊，請參閱使用 AWS CloudFormation 交付 AMI 型產品。
- 賣方不得存取客戶執行的執行個體。如果需要此類存取以進行支援或其他用途，可以指示客戶明確啟用。

SSH（安全殼層）存取政策

除了一般政策之外，提供 SSH（安全殼層）存取權AMIs 必須符合下列安全政策：

- AMIs不得允許使用 SSH 進行密碼型身分驗證。為了確保這一點，請在您的 `sshd_config` 檔案中，將 `PasswordAuthentication`設定為 `no`。
- AMIs停用超級使用者帳戶的密碼型遠端登入。如需詳細資訊，請參閱停用根使用者的密碼型遠端登入。
- AMIs 不得包含 SSH 存取的授權公有金鑰。
- AMIs上的 SSH 必須可供 AWS Marketplace 內部審核程序存取。
  - SSH 服務必須在為 AMI 掃描指定的 TCP 連接埠上接聽。如需詳細資訊，請參閱新增版本。
  - SSH 必須可從子網路`10.0.0.0/16`和執行個體啟動時由 Amazon Elastic Compute Cloud (Amazon EC2) 指派的 IP `10.2.0.0/16` 地址存取。

以 Linux 和其他類似 Unix 作業系統為基礎的 AMIs 政策

除了一般政策之外，以 Linux 和其他類似 Unix 的作業系統為基礎的 AMIs 必須遵循下列安全政策：

- AMIs 必須允許使用者取得完全特權存取（例如，允許sudo存取）。

Windows 型 AMIs的政策

除了一般政策之外，Windows 型 AMIs還必須符合下列安全政策：

- AMIs不得包含訪客帳戶。
- 只有管理員帳戶才能獲得執行個體的遠端桌面存取權。
- Windows AMIs 必須在 EC2Launch （或適用於 Windows 2016 及更舊版本的 EC2Config) 中啟用這些選項來產生管理員密碼：
  - `Ec2SetPassword`
  - `Ec2WindowsActivate`
  - `Ec2HandleUserData`
- AMIs可用於自動審核。至少必須實作下列其中一項要求：
  - （建議選項）SSM 代理程式已安裝，並具有管理許可和傳出網路存取權。
  - Windows Remote Management (WinRM) 服務已啟用、接聽 TCP 連接埠 5985，並且可從子網路`10.0.0.0/16`和執行個體啟動時由 Amazon Elastic Compute Cloud (Amazon EC2) 指派`10.2.0.0/16`的 IP 地址存取。
  - Microsoft Server Message Block (SMB) Protocol 和 Common Internet File System (CIFS) Protocol 服務已啟用、接聽 TCP 連接埠 139 和 445，並且可從子網路`10.0.0.0/16`和執行個體啟動時由 Amazon Elastic Compute Cloud (Amazon EC2) 指派`10.2.0.0/16`的 IP 地址存取。

架構政策

所有 AMIs都必須遵守下列架構政策：

- 的來源 AMIs AWS Marketplace 必須在美國東部（維吉尼亞北部）區域提供。
- AMIs必須使用 HVM 虛擬化。
- AMIs必須使用 x86-64 或 64 位元 ARM 架構。
- AMIs 必須是 Amazon Elastic Block Store (Amazon EBS) 支援的 AMIs。我們不支援 Amazon Simple Storage Service 支援的 AMIs。

- AMIs不得使用加密的 EBS 快照。
- AMIs不得使用加密的檔案系統。
- 必須建置 AMIs，才能在所有 中執行， AWS 區域 且與區域無關。不允許針對不同區域以不同方式建置的 AMIs。

AMI 產品使用說明

為 AMI 產品建立使用說明時，請遵循 中的步驟和指引the section called “AMI 和容器產品使用說明”。

AMI 產品版本政策

AWS Marketplace 使用 S-AMI、AMI 搭配 CloudFormation 範本和容器產品，自動化 AWS 客戶和賣方的版本管理體驗。透過自動版本封存，賣方限制超過兩年的任何產品版本都會自動封存。封存版本不再可供 AWS Marketplace 新客戶從 啟動，但現有使用者可以透過啟動範本和 Amazon EC2 Auto Scaling 群組指定 AMI ID，繼續使用封存版本。過去 13 個月內未用於啟動新執行個體的任何封存版本都會遭到刪除。刪除封存版本後，就無法再為新的或現有的使用者啟動。

客戶資訊政策

所有 AMIs都必須遵守下列客戶資訊政策：

- 除非 BYOL （自帶授權）要求，否則軟體不得在客戶不知情且明確同意的情況下收集或匯出客戶資料。收集或匯出客戶資料的應用程式必須遵循下列準則：
  - 客戶資料的收集必須是自助式、自動化且安全的。買方不需要等待賣方核准部署軟體。
  - 客戶資料的收集必須符合您與 的協議 AWS，包括但不限於 AWS Marketplace 條款與條件、AWS 服務條款、AWS 隱私權聲明和AWS 客戶協議。
  - 不得收集付款資訊。

產品使用政策

所有 AMIs都必須遵守下列產品使用政策：

- 產品不得依時間、使用者數量或其他限制來限制對產品或產品功能的存取。不支援 Beta 版和發行前版本產品或唯一目的是提供試用或評估功能的產品。支援商業軟體的開發人員、社群和 BYOL 版本，前提是也提供同等的付費版本 AWS Marketplace。
- 所有 AMIs都必須與從網站啟動體驗或透過 AMI 型交付相容 AWS CloudFormation。對於從網站啟動，AMI 無法在執行個體建立時要求客戶或使用者資料才能正常運作。

- AMIs 及其軟體必須以自助式方式部署，且不得需要額外的付款方式或成本。需要部署外部相依性的應用程式必須遵循下列準則：
  - 要求必須在清單的描述或使用說明中公開。例如，此產品需要網際網路連線才能正確部署。下列套件會在部署時下載：<list of package>。
  - 賣方負責使用並確保所有外部相依性的可用性和安全性。
  - 如果外部相依性不再可用， AWS Marketplace 也必須從 中移除產品。
  - 外部相依性不得需要額外的付款方式或成本。
- 需要與不受買方直接控制之外部資源持續連線的 AMIs，例如外部 APIs 或由賣方或第三方 AWS 服務 管理的 AMI，必須遵循下列準則：
  - 要求必須在清單的描述或使用說明中公開。例如，此產品需要持續的網際網路連線。需要下列持續的外部服務才能正常運作：<list of resources>。
  - 賣方負責使用並確保所有外部資源的可用性和安全性。
  - 如果外部資源不再可用， AWS Marketplace 也必須從 移除產品。
  - 外部資源不得需要額外的付款方式或成本，且連線的設定必須自動化。
- 產品軟體和中繼資料不得包含將使用者重新導向至 中無法使用的其他雲端平台、其他產品或追加銷售服務的語言 AWS Marketplace。
- 如果您的產品是另一個產品或其他 ISV 產品的附加元件，您的產品描述必須指出它擴展了另一個產品的功能，如果沒有它，您的產品具有非常有限的公用程式。例如，此產品擴展了 <product name> 的功能，如果沒有它，此產品的公用程式非常有限。請注意，<product name> 可能需要自己的授權才能使用此清單的完整功能。

中的 EC2 Image Builder 元件型產品 AWS Marketplace

身為 AWS Marketplace 賣方，您可以使用 Image Builder 元件將產品交付給買方。Image Builder 元件是 Image Builder 服務中用來建立和維護自訂 EC2 Image Builder 元件的基本建置區塊。元件是可重複使用的模組化組態片段，可定義要在映像建立過程中執行的特定任務集。

元件會定義在建立映像之前自訂執行個體 （ 建置元件 ），或從建立的映像 （ 測試元件 ） 啟動的執行個體所需的步驟順序。映像建置器元件可以包含三個項目：

- 元件文件，宣告式、YAML 或 JSON 檔案，指定元件用於建置映像時要執行的步驟和動作。
- 提供元件組態選項的元件屬性。它們有助於定義元件的特性、行為和相容性。
- 軟體套件，必須安裝的相依性，元件才能正常運作。

使用 AWS Marketplace 目錄 API AWS Marketplace 支援在 上發佈映像建置器元件。如需使用目錄 API 發佈元件的說明，請參閱AWS Marketplace 《 API 參考》中的使用 EC2 Image Builder 元件產品。

在 中建立 EC2 Image Builder 產品的變更請求 AWS Marketplace

若要變更 中的產品或版本 AWS Marketplace，您可以透過 提交變更請求 AWS Marketplace 管理入口網站。變更請求會新增至佇列，根據請求的類型，可能需要幾分鐘到幾天的時間才能解決。您可以在 中查看請求的狀態 AWS Marketplace 管理入口網站。本主題提供可用於為單一 產品建立變更請求的程序 AWS Marketplace，包括使用自助式體驗的選項。

您可以為下列情況建立變更請求：

- 您已儲存進行中的步驟，但尚未完成整個程序，同時使用自助式體驗來建立單一產品清單。若要完成其餘步驟，請建立變更請求。
- 您想要修改處於有限或公有狀態之產品的產品資訊。若要更新資訊，您可以建立變更請求。如需您可以為 EC2 Image Builder 產品請求的變更類型詳細資訊，請參閱 建立變更請求。

> **Note**
>
> 除了 之外 AWS Marketplace 管理入口網站，您也可以使用AWS Marketplace 目錄 API 建立變更請求。

主題

- 使用自助式建立變更請求
- 建立變更請求
- 取得變更請求的狀態
- 其他資源

使用自助式建立變更請求

若要修改版本或產品資訊，您可以在 中建立變更請求 AWS Marketplace 管理入口網站。變更請求是您用來變更產品的自助式清單的建置區塊。每次選取儲存並退出步驟，或選取提交以進行任何更新時，您都會提出變更請求。您可以在 AWS Marketplace 管理入口網站 請求索引標籤上找到您的請求。

使用自助式建立變更請求

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，並登入您的賣方帳戶，然後前往伺服器產品頁面。
2. 在伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中選擇選項。
4. 提出變更請求後，系統會等待系統處理您的請求，反映在審核中。當請求完成時，將導致成功或失敗。

    - 提交請求後，它會開始處理這些狀態：檢閱中、準備變更和套用變更。
    - 成功表示已處理請求的變更，且變更會在系統中反映。
    - 失敗表示請求發生錯誤，因此不會處理變更。如果狀態為失敗，您可以選取請求以尋找錯誤代碼，提供如何更正錯誤的建議。此時，您可以對錯誤進行故障診斷，並建立新的變更請求。若要讓程序更快，您可以選擇複製到新請求，以複製失敗請求的詳細資訊。然後，您可以進行調整並重新提交請求。

建立變更請求

> **Important**
>
> 在 2023 年 6 月 15 日， AWS Marketplace 將停止下列程序。2023 年 6 月 15 日之後，請使用 the section called “使用自助式建立變更請求” 程序。

若要修改版本或產品資訊，您可以在 中建立變更請求 AWS Marketplace 管理入口網站。

建立變更請求

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，並登入您的賣方帳戶，然後前往伺服器產品頁面。
2. 在伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中選擇選項。

   對於大多數變更請求，請在使用者介面中填寫表單並提交。

如需變更請求狀態的詳細資訊，請參閱 the section called “取得變更請求的狀態”。如需變更請求潛在問題的洞見，請參閱 對 上變更請求的常見錯誤進行故障診斷 AWS Marketplace。

取得變更請求的狀態

> **Important**
>
> 在 2023 年 6 月 15 日， AWS Marketplace 將停止下列程序。自助服務體驗不再需要此程序。

提交變更請求後，您可以從 的伺服器產品頁面的請求索引標籤中查看請求的狀態 AWS Marketplace 管理入口網站。狀態可以是下列任何一項：

- 審核中表示正在審核您的請求。有些請求需要 AWS Marketplace 團隊手動審核，但大多數請求會在系統中自動審核。
- 成功表示您的請求已完成。您的產品或版本已依照您的要求更新。
- 必要動作表示您需要更新請求以修正問題或回答有關請求的問題。選取請求以查看詳細資訊，包括任何問題。
- 失敗表示請求發生錯誤，您應該使用相同的資料建立新的變更請求。

其他資源

如需特定更新類型變更請求的詳細資訊，請參閱下列資源：

- 更新 上的 EC2 Image Builder 產品資訊 AWS Marketplace

更新 EC2 Image Builder 產品可見性

身為 AWS Marketplace 賣方，您可以更新您的產品可見性，以變更哪些買方可以檢視您的產品 AWS Marketplace。如果可見性狀態設為公開，所有 AWS Marketplace 買方都可看見您的產品。如果您的產品可見性設為有限，則只有您允許清單 AWS 帳戶 IDs 才會顯示您的產品。您也可以管理和更新 AWS 帳戶 IDs 此可查看您產品的 ID 允許清單。下列各節說明如何更新產品可見性和有限可見性允許清單。

主題

- 更新產品可見性
- 更新允許清單（預覽帳戶）

更新產品可見性

若要變更哪些買方可以在其中檢視您的產品 AWS Marketplace，您可以使用更新可見性。

更新可見性

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇更新可見性。

> **Note**
>
> 您可以使用此變更請求，請求將產品從有限狀態移至公有狀態。不過，變更請求必須經過 AWS Marketplace 賣方營運團隊核准程序，才能移至公有。

4. 選擇提交變更請求以提交您的請求以供檢閱。
5. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

更新允許清單（預覽帳戶）

若要變更可在有限狀態下看到您產品的 AWS 帳戶 IDs 清單，請使用更新允許清單。

更新允許清單

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇更新允許清單。目前清單隨目前允許列出的帳戶清單提供。
4. 新增偏好用於可見 AWS 帳戶 IDs，並以逗號分隔 IDs。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

新增和限制 的 Amazon EC2 執行個體 AWS Marketplace

身為 AWS Marketplace 賣方，您可以管理買方可以使用的執行個體。您可以為買方可以使用的產品新增執行個體。同樣地，如果您想要防止新買方從特定執行個體使用您的產品，您可以限制執行個體。

如需 Amazon EC2 執行個體類型的詳細資訊，請參閱《Amazon EC2 使用者指南》中的可用執行個體類型。

下列各節說明如何新增和限制執行個體。

主題

- 新增執行個體
- 限制執行個體

新增執行個體

您可以新增新執行個體，供買方做為 EC2 Image Builder 元件使用。

新增執行個體

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇新增執行個體。
4. 選取執行個體架構。
5. 從可用執行個體清單中選取要新增的執行個體類型。

6. 選擇提交請求以提交您的請求以供檢閱。
7. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

> **Note**
>
> - 如果您目前的定價模式不是免費的，或使用自帶授權 (BYOL) 模型，您也必須新增價格。
> - 如果您使用新執行個體的價格或更新定價來建立新增執行個體以增加價格，則自您進行變更之日起 90 天內，您無法使用自助式新增執行個體。若要進行這些變更，請聯絡AWS Marketplace 賣方營運團隊。
> - 當您新增對新執行個體類型的支援時，已訂閱您產品的私有優惠的客戶將無法自動啟動新增的執行個體。您必須使用您希望客戶存取的執行個體建立另一個私有優惠。接受新優惠後，客戶可以啟動新增的執行個體。未來訂閱您產品的客戶也可以啟動執行個體，只要執行個體包含在私有優惠中。如需如何建立新的私有優惠的詳細資訊，請參閱本指南在 中修訂協議 AWS Marketplace稍後的 。

限制執行個體

若要防止新買方使用執行個體，您可以限制執行個體。如有需要，您可以稍後再新增執行個體。受限執行個體上 EC2 Image Builder 元件的現有使用者，可以在訂閱期間繼續使用 區域中的產品。

限制執行個體

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇限制執行個體。
4. 選取您要限制的執行個體，然後選擇限制。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

> **Note**
>
> 如果核取方塊已著色，這表示執行個體會與 1 到多個版本建立關聯，做為建議的執行個體類型。若要限制此類執行個體，請使用更新版本來選擇不同的建議執行個體類型。變更

請求完成且您想要限制的執行個體不再是建議的執行個體類型後，您可以返回限制執行個體來限制您選擇的執行個體。

在 上管理 EC2 Image Builder 產品的版本 AWS Marketplace

無法從 UI 完成 EC2 Image Builder 元件的版本管理。如需詳細資訊，請參閱下列目錄 API 區段：

- 新增新版本
- 更新現有版本的相關資訊
- 限制版本

更新 上的 EC2 Image Builder 產品資訊 AWS Marketplace

建立單一 EC2 Image Builder 元件之後，您可以變更其中與其相關聯的一些資訊 AWS Marketplace。例如，如果新版本修改產品的描述或重點，您可以使用新資料編輯產品資訊。您也可以更新其他產品資訊，包括產品標題、SKU 描述、類別、關鍵字等。如需詳細資訊，請參閱下列程序。

更新產品資訊

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇更新產品資訊。
4. 更新您需要變更的下列任何欄位：
    - 產品標題
    - SKU
    - 簡短描述
    - 長描述
    - 產品標誌影像 URL
    - 重點標示
    - 產品類別
    - 關鍵字
    - 產品影片 URL

- 資源
- 支援資訊

> **Note**
>
> 如需標誌格式的詳細資訊，請參閱 公司和產品標誌要求。

5. 選取提交。
6. 確認請求出現在請求索引標籤上，且狀態為審核中。您可能需要重新整理頁面，才能在清單中查看請求。

依 AWS 區域 和國家/地區管理 EC2 Image Builder 產品可用性

當您在 中建立產品時 AWS Marketplace，您可以選擇可用的 AWS 區域。您也可以選擇買方可以購買您產品的國家/地區。這兩個屬性類似，但不同。例如，買方可能位於美國並從中購買，但他們可能打算在歐洲（法蘭克福）區域安裝您的產品。為了讓此買方購買您的產品，您必須在您的國家/地區清單中同時包含美國，並在您的區域清單中包含歐洲（法蘭克福）區域。您可以使用下列各節，依區域和國家/地區更新您的產品可用性。

主題

- 新增 AWS 區域
- 限制 AWS 區域
- 更新對未來的支援 AWS 區域
- 依國家/地區更新可用性

新增 AWS 區域

您可以新增買方可以使用您產品的 區域。

新增區域

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇新增區域。

4. 從可用區域清單中選擇您要新增的區域。
5. 選擇提交請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

> **Note**
>
> 當您新增對新 的支援時 AWS 區域，已訂閱您產品的私有優惠的客戶將無法自動存取新增的區域。您必須使用您希望客戶存取的區域建立另一個私有優惠。接受新優惠之後，客戶即可存取新增的區域。只要區域包含在私有優惠中，未來訂閱您產品的客戶也可以存取 區域。如需如何建立新的私有優惠的詳細資訊，請參閱私有優惠升級、續約和修訂。

限制 AWS 區域

若要防止新買方在特定地區使用您的產品 AWS 區域，您可以限制 區域。您可以稍後再新增區域。區域中產品的現有訂閱者只要訂閱，就可以繼續使用區域中的產品。

限制區域

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇限制區域。
4. 選取下拉式選單，以檢視目前可使用您產品的區域清單。
5. 選取您要限制的區域。
6. 您選取的區域會顯示為字符。檢閱您要限制的區域清單，並針對您不想限制的區域輸入 X。
7. 選擇提交變更請求以提交您的請求以供檢閱。
8. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

如果您的請求成功，您現有的使用者會收到以下電子郵件訊息，通知他們要限制的區域。只要他們仍然訂閱，他們就可以繼續使用您的產品，但如果他們取消訂閱，就無法重新訂閱。

```
Greetings from AWS Marketplace,

This message is a notification detailing a recent change for <ProductName>.
```

```
{{{sellerName}}} has opted to restrict the <ProductType> product in <Restricted
 Region(s)> beginning <DateOfChange>.

This impacts you in the following ways:

1. As long as you're subscribed to the product, you can continue using the software
 product in the restricted Region.
2. You can't begin new instances of the software product in the restricted Region.
3. You can continue using the software product in all available AWS Regions.

Regards,
The AWS Marketplace Team
AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com (http://amazon.com/) is a
 registered
trademark of Amazon.com, Inc. This message was produced and distributed by Amazon Web
Services Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.
```

更新對未來的支援 AWS 區域

如果您希望您的產品加入新啟動的 AWS 區域，您可以使用更新未來區域支援。

更新未來的區域支援

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，然後在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新未來區域支援。
4. 您可以選擇啟用未來的區域支援， AWS Marketplace 以允許 AWS 區域 代表您將產品加入新啟動的 。
5. 啟用此功能後，您可以在所有未來的區域之間進行選擇，或僅限於美國區域。
6. 選擇提交變更請求以提交您的請求以供檢閱。
7. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

依國家/地區更新可用性

如果您想要變更產品訂閱和提供的國家/地區，您可以使用更新可用性。

依國家/地區更新可用性

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新可用性。
4. 請選擇下列其中一個選項：
    1. 所有國家 - 在所有支援的國家/地區提供。
    2. 具有排除項目的所有國家/地區 – 適用於所有支援的國家/地區，但所選國家/地區除外。
    3. 自訂清單 – 提供產品的特定國家/地區清單。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

更新 EC2 Image Builder AWS Marketplace 產品的最終使用者授權合約 (EULA)

身為 AWS Marketplace 賣方，您可以更新最終使用者授權合約 (EULA)，該合約將控管 EC2 Image Builder 元件的使用。您的 EULA 位於公開軟體清單的產品清單頁面上 AWS Marketplace。您可以套用自己的 EULA，或使用 標準合約 AWS Marketplace (SCMP)。下列程序說明如何更新 EC2 Image Builder 元件的 EULA。

如需 EULA 的詳細資訊，請參閱 在 中使用標準化合約 AWS Marketplace。

更新 EULA

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 選擇伺服器產品索引標籤，在目前伺服器產品索引標籤上，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇更新最終使用者授權合約。
4. 您可以選取 AWS Marketplace (SCMP) 的標準合約，或提交您自己的自訂 EULA。對於自訂 EULA，您必須從 Amazon S3 儲存貯體提供自訂合約的 URL。

> **Note**
>
> 必須在 Amazon S3 儲存貯體上啟用公有可存取性。

5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

在 上更新 EC2 Image Builder 產品的退款政策 AWS Marketplace

身為 AWS Marketplace 賣方，您可以設定 EC2 Image Builder 元件的退款政策。如果您想要變更產品的退款政策，您可以使用更新退款政策。下列程序說明如何更新退款政策。

更新退款政策

1. AWS Marketplace 管理入口網站 在 開啟 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新退款政策。
4. 文字方塊中提供目前的退款政策詳細資訊，供您編輯。提交請求會覆寫目前的退款政策。
5. 選擇提交變更請求以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

從 移除產品 AWS Marketplace

發佈產品後，您可以從中移除產品 AWS Marketplace。這也稱為日落。若要移除產品，請識別產品並提交移除產品的請求，以及移除原因和您的聯絡電子郵件地址。如果您要將目前的產品取代為新的產品，也可以提供替代產品 ID。

在您請求產品移除後，新客戶無法訂閱產品。您必須支援任何現有客戶至少 90 天。

> **Note**
>
> 您無法從 AWS Marketplace 管理入口網站刪除受限產品。入口網站會將它們保留為帳戶發佈歷史記錄的一部分。

我們處理從 移除產品的請求 AWS Marketplace ，條件如下：

- 產品會從 AWS Marketplace 搜尋、瀏覽和其他探索工具中移除。任何訂閱按鈕或功能都會停用，且頁面上的訊息會清楚指出產品已無法使用。產品詳細資訊頁面仍然可以使用 URL 存取，並且可以在公有搜尋引擎中編製索引。
- 必須指定移除原因（例如，終止支援、產品更新結束或替換產品）。如需繼續支援已移除產品的需求，請參閱 AWS Marketplace 賣方的條款與條件。
- AWS Marketplace 透過電子郵件訊息聯絡目前買方，通知他們產品移除、移除原因，並提供賣方聯絡資訊。
- 目前的買方會保留軟體的存取權，直到他們取消訂閱為止。它們不會受到產品移除的任何影響。

移除使用 建立的產品 AWS Marketplace 管理入口網站

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 選擇產品索引標籤，然後選擇伺服器。
3. 在產品頁面的伺服器產品下，找到您要移除的產品。從請求變更下拉式清單中，選擇更新產品可見性。
4. 在更新產品可見性頁面上，選取限制。
5. （選用）如果有另一個產品將取代您要移除的產品，請提供替換產品 ID。
6. 檢閱資訊的準確性，然後選擇提交。

答 提交產品移除請求後，會顯示下一個資訊頁面。 AWS Marketplace 賣方操作會檢閱和處理您的請求。檢視請求，檢查提交的狀態。

移除您的產品後，產品會出現在 中的目前產品清單中 AWS Marketplace 管理入口網站。在目前產品中，您可以執行的唯一動作是下載產品的試算表。您無法編輯或提交另一個日落請求。

如果您對產品移除有任何疑問，請聯絡AWS Marketplace 賣方營運團隊。

上的容器型產品 AWS Marketplace

AWS Marketplace 支援使用 Docker 容器的軟體產品。容器產品由交付選項組成，這些選項是一組容器映像和部署範本。您至少為產品提交一個交付選項，最多四個。對於每個交付選項，您提供一組容器映像、使用說明和部署範本的連結，供客戶啟動該交付選項。本主題提供 上容器型產品的相關資訊 AWS Marketplace。

AWS Marketplace 購買者會在發佈的產品詳細資訊頁面上查看可用的交付選項，這些選項可供他們使用。在他們訂閱產品並選擇偏好的交付選項之後，買方會看到有關啟動和使用產品的資訊和說明。對於容器映像交付選項，買方會看到可用部署範本和容器映像 URLs的連結。他們也會收到如何提取個別容器映像的指示。對於 Helm Chart 交付選項，買方將看到使用 Helm 啟動的step-by-step說明。

如需購買體驗的逐步解說，請參閱此影片：在 Amazon ECS 叢集上部署 AWS Marketplace 容器 (3：34)。

您可以在任何環境中的任何 Kubernetes 叢集 AWS Marketplace 上，從 尋找、訂閱和部署第三方 Kubernetes 應用程式。您可以在 Amazon Elastic Container Service (Amazon ECS) AWS Fargate、Amazon Elastic Kubernetes Service (Amazon EKS) 和內部部署上使用 Amazon EKS Anywhere (EKS Anywhere) 部署第三方 Kubernetes 應用程式。您也可以在內部部署或 Amazon Elastic Compute Cloud (Amazon EC2) 中的自我管理 Kubernetes 叢集上部署這些叢集。

您可以在任何 Docker 相容執行時間執行免費和自攜授權模型 (BYOL) 容器產品。

主題

- 取得說明
- 容器產品入門
- 的容器型產品需求 AWS Marketplace
- 的容器產品定價 AWS Marketplace
- 容器產品計費、計量和授權整合
- 容器產品的 Amazon SNS 通知

取得說明

如需容器產品的協助，請聯絡您的 業務開發合作夥伴 AWS Marketplace 或 AWS Marketplace 賣方營運團隊。

容器產品入門

身為 AWS Marketplace 賣方，您可以建立容器型軟體產品。容器產品由一組容器映像和部署範本組成的交付選項組成。下列主題說明如何開始使用容器產品。

- 產品生命週期
- 先決條件
- 步驟 1：為您的容器產品建立產品 ID 和產品代碼
- 步驟 2：建立初始清單
- 步驟 3：新增您產品的初始版本
- 步驟 4：（僅適用於付費產品）整合計量或合約定價
- 後續步驟
- 容器產品掃描安全問題

產品生命週期

當您在 中建立產品時 AWS Marketplace，它最初會以有限的可見性發佈，以便允許清單上的帳戶可以查看它，包括建立產品的帳戶。當您準備好時，您可以將其發佈至 AWS Marketplace 目錄，以允許買方訂閱和購買您的產品。

在伺服器產品頁面上，您可以檢視您的產品清單。視其所在階段而定，產品將具有下列其中一種狀態。

- 預備：您仍在為其新增資訊的不完整產品。在第一次儲存並退出自助式體驗時，成功的變更請求會建立未發佈的產品，其中包含您提交之已完成步驟中的資訊。在此狀態中，您可以繼續將資訊新增至產品，或透過變更請求變更已提交的詳細資訊。
- 有限 – 產品提交至系統後即完成，並在系統中傳遞所有驗證。然後，產品會釋出為有限狀態。此時，產品有一個詳細資訊頁面，只有您的帳戶以及允許列入清單的對象才能存取。您可以透過詳細資訊頁面測試您的產品。如需詳細資訊或協助，請聯絡AWS Marketplace 賣方營運團隊。
- 公開 – 當您準備好發佈產品，以便買方可以檢視和訂閱產品時，您可以使用更新可見性變更請求。此請求會啟動工作流程，讓 AWS Marketplace 賣方營運團隊根據 AWS 政策檢閱和稽核您的產品。核准產品並處理變更請求後，產品的狀態會從有限移至公開。如需 AWS 準則的資訊，請參閱 的容器型產品需求 AWS Marketplace。
- 受限 – 如果您想要停止新使用者訂閱您的產品，您可以使用更新可見性變更請求來限制產品。受限狀態表示現有的允許清單使用者可以繼續使用產品。不過，公開或新使用者將無法再看到該產品。

先決條件

開始之前，您必須完成下列先決條件：

1. 存取和使用 AWS Marketplace 管理入口網站。這是您用來註冊為賣方和管理您銷售產品的工具 AWS Marketplace。如需詳細資訊，請參閱AWS Marketplace 管理入口網站。
2. 註冊為賣方，並提交您的稅務和銀行資訊。如需詳細資訊，請參閱註冊為 AWS Marketplace 賣方。
3. 在 Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS) 或 中建立至少一個容器 AWS Fargate。請確定您有關聯映像的連結。
4. 規劃您將如何建立和整合容器產品 AWS Marketplace。

   我們建議您在公開發佈產品之前預先規劃定價、權利和計量策略。

   - 如需容器型產品需求的相關資訊，請參閱 的容器型產品需求 AWS Marketplace。
   - 如需設定產品定價的資訊，請參閱 的容器產品定價 AWS Marketplace。
   - 如需付費容器型產品的自訂計量資訊，請參閱 使用 的每小時和自訂計量 AWS Marketplace Metering Service。

概觀：建立容器產品

建立容器產品需要下列步驟：

1. 步驟 1：為您的容器產品建立產品 ID 和產品代碼
2. 步驟 2：建立初始清單
3. 步驟 3：新增您產品的初始版本
4. 步驟 4：（僅適用於付費產品）整合計量或合約定價
5. 更新產品可見性

如需產品生命週期的資訊，請參閱 產品生命週期。

步驟 1：為您的容器產品建立產品 ID 和產品代碼

若要開始使用容器產品，您必須在其中建立產品 ID 和產品代碼記錄 AWS Marketplace。產品 ID 用於追蹤您產品的整個生命週期。

使用下列程序在 中建立新的容器產品 AWS Marketplace 管理入口網站，並產生產品 ID。

> **Note**
>
> 此程序也會為與您產品配對的容器建立公有金鑰。

建立容器產品 ID

1. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
2. 從選單列中選取產品，然後選擇伺服器。
3. 選擇建立伺服器產品，然後選擇容器。
4. 產生容器產品 ID 和程式碼。

> **Note**
>
> （選用）您可以為產品加上標籤型授權的標籤。如需詳細資訊，請參閱標記您的 AWS 資源。

5. 選擇繼續以繼續建立您的產品。

步驟 2：建立初始清單

產生產品 ID、產品代碼和公有金鑰後，您將使用精靈建立初始清單。

1. 提供產品清單的產品資訊。
2. 判斷您產品的定價模式。

> **Note**
>
> 如需詳細資訊，請參閱容器產品定價。

> **Note**
>
> 對於付費產品，您的產品將從 0.01 USD 定價開始，允許您和 AWS Marketplace 賣方營運團隊測試產品，而不會產生高成本。您會在公開時提供實際價格。

3. 提供其他優惠資訊，包括退款政策、EULA 和優惠可用性。

4. 為您的容器產品新增初始儲存庫。
5. 在最後一個步驟選擇提交，將產品移至有限可見性。

> **Note**
>
> 您的容器產品最初是使用預留位置版本建立的。當產品可見性有限時，您將新增最終版本。

步驟 3：新增您產品的初始版本

您的產品在其生命週期內可能有多個版本。每個版本都有一組特定於該版本的容器映像。若要新增產品的初始版本，請參閱 在 上新增新版本的容器產品 AWS Marketplace。

步驟 4：（僅適用於付費產品）整合計量或合約定價

對於具有用量定價的容器型產品，您可以使用 AWS Marketplace Metering Service來檢查是否有權使用您的產品和計量用量來計費。您必須測量您在設定定價資訊時建立的定價模型。如需詳細資訊，請參閱使用 的每小時和自訂計量 AWS Marketplace Metering Service。

合約定價

對於具有合約定價的容器型產品，您可以使用 AWS License Manager 將授權與產品建立關聯。

如需 與 整合的詳細資訊 AWS License Manager，請參閱 使用 的容器產品的合約定價 AWS License Manager。

步驟 5：更新產品可見性

當您在 中建立產品時 AWS Marketplace，它最初會以有限的可見性發佈，以便允許清單上的帳戶可以查看它，包括建立產品的帳戶。您可以更新產品可見性，以允許買方訂閱和購買您的產品。或者，您可以更新要新增的產品允許清單 AWS 帳戶。本主題說明如何管理哪些買方可以檢視您的產品 AWS Marketplace。

如需產品可見性和生命週期的詳細資訊，請參閱 產品生命週期。

主題

- 更新產品可見性
- 更新 IDs的 AWS 帳戶 允許清單

更新產品可見性

更新可見性

1. 在 開啟 AWS Marketplace 管理入口網站 https://aws.amazon.com/marketplace/management/tour/，然後登入您的賣方帳戶。
2. 前往伺服器產品頁面，在目前伺服器產品索引標籤上，選取您要修改的容器型產品。
3. 從請求變更下拉式清單中，選擇更新可見性。

> **Note**
>
> 您可以使用此變更請求，請求將產品從有限狀態移至公有狀態。不過，變更請求必須經過 AWS Marketplace 賣方營運團隊核准程序，才能移至公有。

4. 選擇提交以提交您的請求以供檢閱。
5. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

更新 IDs的 AWS 帳戶 允許清單

您可以變更可在有限狀態下檢視產品的 AWS 帳戶 IDs 清單。允許列出的帳戶會在產品詳細資訊頁面上顯示產品版本的有限徽章。

1. 在 AWS Marketplace 管理入口網站 https：//https://aws.amazon.com/marketplace/management/tour/ 開啟 並登入您的賣方帳戶。
2. 在伺服器產品頁面中，選取要修改的容器產品。
3. 從請求變更下拉式清單中，選取更新允許清單。隨即顯示允許列出的目前帳戶清單。
4. 在允許清單 AWS 帳戶欄位中，輸入 AWS 帳戶 IDs並使用逗號分隔。
5. 選擇提交以提交您的請求以供檢閱。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會更新為成功或失敗。

後續步驟

建立容器產品之後，您可以使用下列主題中的資訊來設定和管理它：

- 更新 上容器產品的產品資訊 AWS Marketplace
- 在 上新增新版本的容器產品 AWS Marketplace

- 在 上管理容器產品定價 AWS Marketplace
- 依 中的國家/地區更新容器產品可用性 AWS Marketplace
- 在 上更新容器產品的最終使用者授權合約 (EULA) AWS Marketplace
- 在 上測試並發行容器產品 AWS Marketplace

容器產品掃描安全問題

當您建立變更請求以將新版本新增至容器產品時，我們會掃描該新版本中包含的容器映像，並檢查是否有安全漏洞。為此，我們在映像上執行layer-by-layer靜態掃描。如果我們找到具有可遠端利用風險向量的關鍵漏洞，我們會為您提供找到的問題清單。我們強烈建議您使用容器映像掃描器執行自己的安全分析Twistlock，例如 Clair、Aqua Security、 或 Trend Micro ，以避免擷取和發佈程序延遲。

您選擇的建置容器映像的基礎映像可能會對最終映像的安全性設定檔產生重大影響。如果您選擇已具有已知關鍵漏洞的基本映像，則會因為基底層而標記這些映像，即使您的應用程式軟體層是乾淨的。我們建議您在建置映像並將其提交至 之前，先驗證您是否從沒有漏洞的基本容器開始 AWS Marketplace。

更新 上容器產品的產品資訊 AWS Marketplace

建立產品 ID 並設定定價後，您可以編輯產品資訊，包括客戶會在 中看到哪些容器產品 AWS Marketplace。例如，如果新版本修改了產品的描述或重點，您可以使用新資料編輯產品資訊。您也可以更新其他產品資訊，包括產品標題、SKU描述、類別、關鍵字和其他。下列程序概述為您的產品建立產品詳細資訊。

建立或更新容器產品的產品詳細資訊

1. 登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中選取伺服器。
3. 在伺服器產品索引標籤上，選取您要修改的產品。
4. 從請求變更下拉式清單中，選擇更新產品資訊 。
5. 更新您想要變更的下列任何欄位：
    - 產品標題
    - SKU
    - 簡短描述
    - 詳細描述

- 產品標誌映像 URL
- 重點標示
- 產品類別
- 關鍵字
- 產品影片 URL
- 資源
- 支援資訊

> **Note**
>
> 映像URLs必須位於可公開存取的 Amazon S3 儲存貯體中。如需標誌格式的詳細資訊，請參閱 公司和產品標誌要求。

6. 選擇提交。
7. 確認請求顯示在請求索引標籤上，且狀態為審核中。您可能需要重新整理頁面，才能在清單中查看請求。

您可以隨時從伺服器產品頁面的請求索引標籤檢查請求的狀態。

在 上新增新版本的容器產品 AWS Marketplace

身為 AWS Marketplace 賣方，您可以新增容器產品的新版本、管理版本，以及更新版本資訊。您的產品在其生命週期內可能有多個版本。每個版本都有一組專屬於該版本的容器映像。下列主題說明如何管理容器產品的產品版本。

> **Note**
>
> 您必須先建立產品 ID 和產品定價，才能將版本新增至您的產品。如需這些步驟的詳細資訊，請參閱 步驟 1：為您的容器產品建立產品 ID 和產品代碼。

主題

- 步驟 1：新增儲存庫
- 步驟 2：將容器映像和成品上傳至儲存庫
- 步驟 3：將新版本新增至您的容器產品

- 步驟 4：更新版本資訊
- 限制 Amazon EKS 附加元件的版本

步驟 1：新增儲存庫

您產品的容器映像和其他成品會存放在 的儲存庫中 AWS Marketplace。一般而言，您可以為每個所需的成品建立一個儲存庫，但儲存庫可以存放多個版本的成品（具有不同的標籤）。

> **Note**
>
> 產品部署中的所有映像都必須使用儲存庫中的 AWS Marketplace 映像。

下列程序說明如何在 中新增任何所需的儲存庫 AWS Marketplace。

新增儲存庫

1. 登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中選取伺服器。
3. 在伺服器產品索引標籤上，選取您要修改的產品，然後從請求變更下拉式清單中選擇新增儲存庫。
4. 輸入您要建立之儲存庫的名稱。如果您想要建立多個新儲存庫：
   - 為每個額外的儲存庫選擇新增儲存庫。
   - 提供唯一的名稱。您選擇的唯一名稱必須跨賣方帳戶中的所有產品。

   > **Note**
   >
   > 儲存庫將具有此結構：`<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>`。當您將項目新增至儲存庫時（在下列程序中），它們會取得標籤並具有此結構：`<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>:<tag>`。`repositoryID` 是 的內部 ID AWS Marketplace。`sellerName` 是根據您為賣方帳戶建立的名稱。您可以在此步驟`respositoryName`中定義 。當您將成品上傳到儲存庫時，`tag`會設定 。
5. 選取提交。

> **Note**
>
> 每個產品最多可以有 50 個儲存庫。

系統會建立新的請求，並顯示在請求索引標籤上。完成後，您可以在幾分鐘內開始將容器映像和其他成品新增至您建立的儲存庫。

步驟 2：將容器映像和成品上傳至儲存庫

將容器映像和成品上傳至儲存庫

1. 登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中，選擇伺服器。
3. 在伺服器產品索引標籤上，選取您要修改的產品。
4. 從請求變更下拉式清單中選擇新增儲存庫。
5. 選擇檢視現有的儲存庫。
6. 選取您要上傳的儲存庫。
7. 選取檢視推送命令以開啟指示清單，包括可用來將Docker容器映像和Helm圖表推送至該儲存庫的命令。

   如需如何將容器映像和其他成品推送至儲存庫的一般資訊，請參閱《Amazon Elastic Container Registry 使用者指南》中的推送映像。

   > **Note**
   >
   > 您可以在呼叫 `docker pull`或 時使用下列 Amazon Elastic Container Registry (Amazon ECR) API 操作`docker push`：
   >
   > - `DescribeImages` – 用來檢閱儲存庫中映像的相關中繼資料。
   > - `GetAuthorizationToken` – 使用 在將成品上傳到儲存庫之前進行驗證，然後使用 `docker pull`或 `docker push`命令。
   > - `ListImages` – 使用 來檢視您推送的映像清單。

8. 使用列出的命令，將任何所需的成品從本機儲存庫推送至您產品的 AWS Marketplace 儲存庫。

> **Note**
>
> **tag** 您在push命令中提供的 用於區分您上傳到儲存庫的成品版本。使用對成品所屬版本有意義的標籤。

9. 針對版本中所需的每個容器映像或成品重複此步驟。

> **Note**
>
> 您的版本在每個交付選項中最多可包含 50 個容器映像或成品。如需交付選項的詳細資訊，請參閱下列程序。

上傳成品之後，您就可以建立產品的版本。

> **Note**
>
> 您的容器映像會自動掃描，以查看是否符合 的容器型產品需求 AWS Marketplace。如需詳細資訊，請參閱 容器產品掃描安全問題。

新增交付選項

每個版本的容器產品都需要一個交付選項。交付選項指定買方可用的部署選項。根據以下其中一個交付選項，您需要將適當的成品上傳到儲存庫。

- 對於容器映像交付選項，請將產品安裝所需的所有容器映像上傳至 AWS Marketplace 主控台中建立的 Amazon Elastic Container Registry (Amazon ECR) 儲存庫。
- 對於Helm圖表交付選項，請將Helm圖表和容器映像上傳到 AWS Marketplace 主控台中建立的 Amazon ECR 儲存庫。
- 對於 Amazon EKS 主控台附加元件交付選項，請將Helm圖表和容器映像上傳到主控台中建立的 Amazon ECR 儲存庫 AWS Marketplace 。

步驟 3：將新版本新增至您的容器產品

> **Note**
>
> 如果您在將新版本新增至容器時收到任何錯誤，請參閱 AWS Marketplace Catalog API 參考中的新增新版本非同步錯誤表格。

將新版本新增至您的容器產品

1. 登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中選擇伺服器。
3. 在伺服器產品索引標籤上，選取您要新增版本的產品。然後從請求變更下拉式清單中選擇新增版本。
4. 在新增版本頁面上，輸入您版本的版本標題和版本備註。
5. 輸入版本詳細資訊後，下一個步驟是新增交付選項。交付選項是一組指示和資訊，買方可用來從您的產品版本啟動軟體。交付選項稱為買方履行選項。

   > **Note**
   >
   > 您的產品可以支援具有不同容器映像的多個平台（例如 Kubernetes和 Ubuntu 部署）。您可以為每個客戶設定產品的方式建立一個交付選項，每個產品版本最多四個交付選項。

   a. 如果產品在其他版本中已有交付選項，您可以使用現有選項做為範本，將交付選項新增至新版本。在交付選項中，從清單中選擇您要新增的交付選項。您可以使用下列步驟中的指示來編輯選項。

   b. 若要新增交付選項，請選擇新增交付選項。新增選項後，請依照下列步驟中的指示進行設定。
6. 選擇交付選項的交付方法。交付方法決定買方如何啟動您的軟體。

   - 對於容器映像交付選項，提供在主控台中建立的 Amazon Elastic Container Registry (Amazon ECR) 儲存庫中容器映像的 AWS Marketplace 路徑。買方使用容器映像路徑，透過直接將映像提取至其環境來啟動軟體。
   - 對於Helm圖表交付選項，提供在 AWS Marketplace 主控台中建立的 Amazon ECR 儲存庫中的Helm圖表路徑。買方會在其部署環境中安裝Helm圖表以啟動軟體。

- 對於 Amazon EKS 主控台附加元件交付選項，請在主控台中建立的 Amazon ECR 儲存庫中提供Helm AWS Marketplace 圖表路徑。買方使用 Amazon EKS 主控台或原生 Amazon EKS 附加元件 APIs 安裝容器，以啟動軟體。如需詳細資訊，請參閱來自 Amazon EKS 的可用 Amazon EKS 附加元件。

a. 若要新增容器映像交付選項，請執行下列步驟：

   i. 在容器映像中，將 Amazon ECR URL 新增至包含產品版本軟體的容器映像。

   ii. 在交付選項標題和部署選項描述中，輸入此交付選項的標題和描述。

   iii. 在使用說明中，輸入詳細資訊，以協助您的買方在啟動軟體後使用您的軟體。

   iv. 在支援的 服務中，選取買方可以在其中啟動軟體的環境。

   v. 在部署範本中，新增買方可用來啟動軟體的資源。為每個範本輸入資源的標題和 URL。

b. 若要新增Helm圖表交付選項，請執行下列步驟：

   i. 在Helm圖表中，將 Amazon ECR URL 新增至購買者將在其部署環境中安裝的Helm圖表，以啟動您的軟體。

   ii. 在容器映像中，將 Amazon ECR URL 新增至包含產品版本軟體的容器映像。

   iii. 在交付選項標題和部署選項描述中，輸入此交付選項的標題和描述。

   iv. 在使用說明中，輸入詳細資訊，以協助您的買方在啟動軟體後使用您的軟體。

   v. 在支援的 服務中，選取買方可以在其中啟動軟體的環境。

   vi. 選用 - 在Helm發行名稱中，輸入要安裝Helm圖表的Kubernetes命名空間名稱。

   vii. 選用 - 在Helm安裝命名空間中，輸入 `helm install`命令將使用的Helm發行版本名稱。

   viii. 選用 - 在Kubernetes服務帳戶名稱中，輸入將用於連線至 AWS Identity and Access Management (IAM) Kubernetes的服務帳戶名稱。Kubernetes 服務帳戶會呼叫 AWS 授權或計量等服務。

   ix. 選擇在此產品版本上啟用 QuickLaunch。快速啟動是 中的功能 AWS Marketplace。買方可以使用 Quick Launch 快速建立 Amazon EKS 叢集，並使用 啟動您的軟體 AWS CloudFormation。如需詳細資訊，請參閱 中的快速啟動 AWS Marketplace。

   x. 在覆寫參數中，輸入將在啟動軟體的 CLI Helm 命令中使用的參數。這些參數可讓買方覆寫提供的預設值。如果您已啟用快速啟動，請同時輸入 CloudFormation 表單的參數名稱和描述。使用 AWS Marketplace 管理主控台時有 15 個參數的限制，但使用 時沒有限制 AWS Marketplace Catalog API。如需詳細資訊，請參閱將新版本新增至容器型產品。

> **Note**
>
> 某些覆寫參數是必要的。Amazon EKS Anywhere `DefaultValue` 產品需要具備 授權秘密的覆寫參數`"${AWSMP_LICENSE_SECRET}"`。對於付費產品，您必須使用 `DefaultValue`的 為服務帳戶組態提供一個覆寫參數`"${AWSMP_SERVICE_ACCOUNT}"`。

xi. 選擇隱藏密碼和秘密，以在主控台、命令列工具和 APIs 中遮罩敏感資訊。如需詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的參數NoEcho文件。

c. 若要新增 Amazon EKS 主控台附加元件交付選項，請確定成品符合 Amazon EKS 附加元件產品的需求，然後執行下列步驟：

> **Note**
>
> 每個版本僅支援一個 Amazon EKS 附加元件交付選項。在 Amazon EKS 主控台上發佈您正在使用的目前版本之前，您無法新增版本。

i. 在Helm圖表中，將 Amazon ECR URL 新增至購買者將在其部署環境中安裝的Helm圖表，以啟動您的軟體。

ii. 在容器映像中，將 Amazon ECR URL 新增至包含產品版本軟體的容器映像。確定已列出Helm圖表中的所有影像。

iii. 在交付選項標題和部署選項描述中，輸入此交付選項的標題和描述。

iv. 在可見性中，保持選取 Limited 的預設值。

v. 在附加元件名稱中，輸入此附加元件的唯一名稱。您在 Amazon EKS 主控台中顯示時，您輸入的附加元件名稱會附加賣方的名稱。

vi. 在附加元件版本中，輸入安裝或升級此附加元件時可見的附加元件版本。遵循格式 `major.minor.patch`。

vii. 在附加元件類型中，從下拉式清單中選取附加元件的類別。

viii. 在Kubernetes版本中，選取您的附加元件將支援的所有Kubernetes版本。

ix. 在架構中，選取您的附加元件支援的平台架構。選項為 AMD64 和 ARM64。我們建議您支援這兩種架構，以最大化相容性。如果您的附加元件不支援 ARM64 裝置，您必須先指定新增支援的計劃日期，才能將產品發佈至所有商業廣告 AWS 區域。

x. 在命名空間中，輸入要安裝附加元件的唯一Kubernetes命名空間。安裝第三方附加元件時不支援 `kube-system`、 `default`和 `kube-public`命名空間。

xi. 在環境覆寫參數中，您可以從 Amazon EKS 附加元件架構中選取最多 2 個環境參數。您可以將參數名稱從 values.yaml 映射到這些環境變數，即 `${AWS_REGION}`和 `${AWS_EKS_CLUSTER_NAME}`。

7. 若要新增其他交付選項，請選擇新增交付選項，然後重複先前步驟中的指示進行設定。
8. 選擇提交。

步驟 4：更新版本資訊

建立版本之後，透過修改與該版本相關聯的資訊，將更新的資訊提供給買方會很有幫助。例如，如果您計劃在發行 1.1 版之後限制 1.0 版，您可以更新 1.0 版的描述，將買方導向 1.1 版。提供 1.0 版將受到限制的日期。您可以從 更新版本資訊 AWS Marketplace 管理入口網站。

更新版本資訊

1. 登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中選取伺服器。
3. 在伺服器產品索引標籤上，選取您要修改的產品。
4. 從請求變更下拉式清單中，選擇更新版本資訊。
5. 在更新版本頁面上，選取您要更新的版本。
6. 更新所選版本。可供更新的欄位取決於產品版本或交付選項的狀態。

   a. 對於所有版本，您可以更新版本備註。

   b. 對於尚未公開提供的版本，您可以更新版本標題。

   c. 對於尚未受到限制的交付選項，您可以更新下列欄位：

      - Description
      - 使用說明
      - 支援的服務

   d. 對於尚未公開提供的版本中的交付選項，您可以更新下列欄位：

      - 交付選項標題
      - Helm 圖表 （僅適用於Helm圖表交付選項）
      - 容器映像

- 部署資源
- AddOn名稱
- AddOn 版本
- AddOn 類型
- Helm 圖表 URI
- CompatibleKubernetesVersions
- SupportedArchitectures
- 命名空間
- EnvironmentOverrideParameters

e. 對於公開提供的版本中的交付選項，您可以更新 SupportedArchitectures。

7. 選擇提交。
8. 確認請求顯示在請求索引標籤上，且狀態為審核中。

您可以隨時從伺服器產品頁面的請求索引標籤檢查請求的狀態。

限制 Amazon EKS 附加元件的版本

若要限制以附加元件發佈的容器產品版本，請使用 底部的聯絡我們表單聯絡 AWS Marketplace 營運團隊AWS Marketplace 管理入口網站。

在 上管理容器產品定價 AWS Marketplace

身為 AWS Marketplace 賣方，您可以管理和更新容器產品定價。對於容器產品，您可以列出免費產品、自攜授權模型（BYOL）產品和 Amazon Elastic Container Service（Amazon ECS）、Amazon Elastic Kubernetes Service（Amazon EKS）和 的付費產品 AWS Fargate。每個產品只能設定一個價格。您可以新增或更新定價維度。根據您的定價模型，您可以新增合約和用量維度。您也可以更新定價條款或退款政策。如需詳細資訊，請參閱下列區段。

如需 容器產品定價模型的詳細資訊 AWS Marketplace，請參閱 的容器產品定價 AWS Marketplace。

主題

- 新增定價維度
- 更新維度資訊
- 更新定價條件

- 更新產品的退款政策

新增定價維度

您可以將定價維度新增至您用於產品帳單的定價模型。如需定價模型的詳細資訊，請參閱 容器定價模型。

> **Note**
>
> 在 上無法使用為具有消耗定價的合約新增定價維度 （ 例如，額外用量的 pay-as-you-go定價 ）AWS Marketplace 管理入口網站。
> 您無法在合約、用量和使用量定價的合約之間變更定價模型。如需協助，請聯絡AWS Marketplace 賣方營運團隊。

1. 在 https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace 管理入口網站 開啟 ，並登入您的賣方帳戶。
2. 在伺服器產品頁面中，選取要修改的容器產品。
3. 從請求變更下拉式清單中，選取更新定價維度 ，然後選取新增定價維度 。
4. 根據定價模型，您可以提供API識別碼 、顯示名稱 和描述 的資訊，以新增合約維度或用量維度。
5. 選擇下一個 ，然後輸入您的合約維度定價。
6. 選擇提交以提交您的審核請求。
7. 確認請求索引標籤顯示請求狀態為審核中 。當請求完成時，狀態將更新為成功或失敗 。

更新維度資訊

您可以變更產品的維度資訊。如需定價模型的詳細資訊，請參閱 容器定價模型。

1. 在 https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace 管理入口網站 開啟 ，並登入您的賣方帳戶。
2. 在伺服器產品頁面中，選取要修改的容器產品。
3. 從請求變更下拉式清單中，選取更新定價維度 ，然後選取更新維度資訊 。
4. 您可以透過提供要更新之維度的顯示名稱和描述的資訊來新增維度資訊。
5. 選擇提交以提交您的審核請求。

6. 確認請求索引標籤顯示請求狀態為審核中 。當請求完成時，狀態將更新為成功或失敗 。

更新定價條件

您可以變更產品的定價條件。如果您變更產品提供的國家/地區，可能需要更新定價條件。

1. 在 https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace 管理入口網站 開啟 ，並登入您的賣方帳戶。
2. 在伺服器產品頁面中，選取要修改的容器產品。
3. 從請求變更清單中，選取更新公開優惠 ，然後選取更新定價條件 。
4. 目前的定價會預先填入，供您編輯。您可以刪除目前的定價並新增您的新價格。我們建議您在提交請求以供審核之前，先審核您請求的價格。
5. 選擇提交以提交您的審核請求。
6. 確認請求索引標籤顯示請求狀態為審核中 。當請求完成時，狀態將更新為成功或失敗 。

> **Note**
>
> 如果您使用更新定價條件 提高維度的價格，您將無法更新定價 90 天。價格調升會鎖定從您啟動變更當日起 90 天內的價格。此價格鎖定僅在價格上漲時生效，而不是價格降低時生效。

更新產品的退款政策

您可以更新產品的退款政策。退款政策的更新對所有使用者生效。如需詳細資訊，請參閱中的產品退款 AWS Marketplace。

1. 在 https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace 管理入口網站 開啟 ，並登入您的賣方帳戶。
2. 在伺服器產品頁面中，選取要修改的容器產品。
3. 從請求變更下拉式清單中，選取更新公開優惠 ，然後選取更新退款政策 。
4. 目前的退款政策已預先填入，供您編輯。您可以刪除目前的退款政策，並新增新的退款政策。我們建議您在提交請求以供檢閱之前，先檢閱您請求的退款政策。提交請求會覆寫目前的退款政策。
5. 選擇提交以提交您的審核請求。
6. 確認請求索引標籤顯示請求狀態為審核中 。當請求完成時，狀態將更新為成功或失敗 。

依 中的國家/地區更新容器產品可用性 AWS Marketplace

身為 AWS Marketplace 賣方，您可以變更容器產品可以提供和訂閱的國家/地區。如需詳細資訊，請參閱國家/地區。下列程序說明如何依國家/地區更新容器產品可用性。

1. 在 https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace 管理入口網站 開啟 ，並登入您的賣方帳戶。
2. 在伺服器產品頁面中，選取要修改的容器產品。
3. 從請求變更下拉式清單中，選取更新公開優惠 ，然後選取依國家/地區更新可用性。
4. 選取以下其中一個選項：
   - 所有國家 - 在所有支援的國家/地區提供。
   - 具有排除項目的所有國家 - 在所有支援的國家/地區提供，但所選國家/地區除外。
   - 僅限允許列出的國家 - 僅供買方在您指定的國家/地區購買。
5. 選擇提交以提交您的審核請求。
6. 確認請求索引標籤顯示請求狀態為審核中 。當請求完成時，狀態將更新為成功或失敗 。

在 上更新容器產品的最終使用者授權合約 (EULA) AWS Marketplace

身為 AWS Marketplace 賣方，您可以更新最終使用者授權合約 (EULA)，以使用 的標準合約 AWS Marketplace或自訂 EULA。針對訂閱您產品和產品續約的新使用者，EULA 的更新會生效。若要更新您的 EULA，請使用下列程序。

1. 在 https：//https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace 管理入口網站 開啟 並登入您的賣方帳戶。
2. 在伺服器產品頁面中，選取要修改的容器產品。
3. 從請求變更下拉式清單中，選取更新公開優惠，然後選取更新 EULA。
4. 選取 的標準合約 AWS Marketplace或提交您的自訂 EULA。對於自訂 EULA，您必須提供來自 Amazon S3 儲存貯體的合約。
5. 選擇提交以提交您的審核請求。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會更新為成功或失敗。

在 上測試並發行容器產品 AWS Marketplace

將新版本的容器產品新增至 後 AWS Marketplace，您可以測試產品並將其發佈給公有。本主題概述測試和公開發佈產品所需的特定步驟和程序。

主題

- 容器映像和 Helm 圖表交付選項
- Amazon EKS 附加元件交付選項

容器映像和 Helm 圖表交付選項

本節提供有關釋出容器映像和 的指引 Helm 圖表。

您的新版本請求已建立，並應在幾分鐘內完成。您可以從伺服器產品頁面的請求索引標籤追蹤請求。如果您在測試或釋出附加元件時收到任何錯誤，請參閱AWS Marketplace Catalog API 參考 中的新增新版本的非同步錯誤資料表。

> **Note**
>
> 如果您的產品目前設為有限可用性，則只有產品可用的買方才能存取產品版本。如果您的產品目前設為公有可用性，所有 AWS Marketplace 買家都可以存取產品版本。

如果這是您的第一個版本集，您的產品現在已準備好發佈。

Amazon EKS 附加元件交付選項

本節提供測試和發行 Amazon EKS附加元件的指引。

測試您的附加元件

- 提交附加元件後， 會 AWS Marketplace 處理您的請求，並在有限狀態下發佈附加元件，供您在Amazon EKS附加元件目錄中進行驗證。您可以從 中的伺服器產品頁面的請求索引標籤追蹤請求 AWS Marketplace 管理入口網站。擷取時間從 5 到 10 個工作日不等，具體取決於我們正在處理的請求量。

  當您的請求處於審核中狀態時， AWS 團隊仍在從 將附加元件發佈 AWS Marketplace 至 Amazon EKS附加元件目錄。附加元件發佈至有限狀態後，請求狀態變更為成功。您可以在此之後開始測試您的附加元件。

- 附加元件可用後，您可以在亞太區域（首爾）區域中找到該附加元件以供測試之用。 AWS Marketplace 仰賴您的專業知識來驗證軟體的功能。若要測試您的附加元件，您必須在允許附加元件的賣方帳戶中的亞太區域（首爾）區域建立 Amazon EKS叢集。若要測試您的附加元件，請遵循這些詳細說明 。請務必在每個 上進行測試 Kubernetes 軟體支援的版本。
- 如果您要提供付費產品，請為下列內部 建立私有優惠 AWS 帳戶。這些帳戶有助於將您的軟體整合到所有商業 中的 Amazon EKS主控台 AWS 區域。

```
288092140294, 288092140294, 408202761791
```

- 在 AWS Marketplace 核准並移動您的附加元件版本至公有版本之前，請讓測試叢集保持作用中狀態。

> **Note**
>
> AWS Marketplace 不會承擔在 Amazon EKS叢集上測試容器產品時產生的 AWS 基礎設施成本。您可以在驗證測試結果時，遵循正確的大小調整機制，將節點調低到最低的操作成本。

將附加元件發佈至公有

透過 Amazon EKS叢集驗證軟體做為附加元件後，您可以使用 AWS Marketplace 管理入口網站或 將 Amazon EKS附加元件版本發佈至公有的請求 AWS Marketplace Catalog API。

如需詳細資訊，請參閱 AWS Marketplace Catalog API 參考 中的更新 Amazon EKS附加元件的可見性。

您可以從 中的伺服器產品頁面的請求索引標籤追蹤請求 AWS Marketplace 管理入口網站。擷取時間會有所不同。

的容器型產品需求 AWS Marketplace

AWS Marketplace 會維護所有容器型產品和方案的下列需求 AWS Marketplace。這些要求有助於為我們的客戶提升安全、可靠且可靠的目錄。我們也鼓勵賣方審查適用的其他控制和協定的實作，以滿足其特定產品的需求。

提交時，會檢閱所有產品及其相關中繼資料，以確保它們符合或超過目前的 AWS Marketplace 政策。這些政策會定期更新，以符合不斷發展的安全準則。 AWS Marketplace 會持續掃描產品，以確認現有的清單持續符合這些要求的任何變更。如果產品不合規， AWS Marketplace 將聯絡賣方更新其產品以

符合新標準。在某些情況下，在問題解決之前，新訂閱用戶可能會暫時無法使用產品。此程序有助於為所有使用者維護 AWS Marketplace 平台的安全性和可信度。

主題

- 安全政策
- 客戶資訊要求
- 產品使用需求
- 架構需求
- 容器產品使用說明
- Amazon EKS 附加元件產品的需求

安全政策

所有容器型產品都必須遵守下列安全要求：

- 容器映像不得包含任何已知的漏洞、惡意軟體或End-of-Life(EoL) 軟體套件和作業系統。
- 容器不得請求 AWS 登入資料來存取 AWS 服務。當您的產品需要存取 AWS 服務時，您必須使用下列其中一項：
  - 服務帳戶的 IAM 角色，適用於 Amazon Elastic Kubernetes Service (Amazon EKS) 工作負載。
  - 任務的 IAM 角色，適用於 Amazon Elastic Container Service (Amazon ECS) 工作負載。
- 容器型產品必須只需要最低權限才能執行。如需詳細資訊，請參閱 Amazon Elastic Container Service 中的安全性和 Amazon EKS 中的安全性。
- 根據預設，容器映像應設定為使用非根權限執行。
- 容器不得包含任何硬式編碼秘密，例如系統使用者和服務的密碼（即使是雜湊）、私有金鑰、登入資料等。
- 容器內執行的任何服務中的身分驗證都不得使用密碼型身分驗證，即使使用者在啟動時產生、重設或定義密碼。也不允許使用空密碼和空白密碼。
- 容器映像不得包含具有不支援架構的層（例如，傳入認證架構中繼資料）。

客戶資訊要求

所有容器型產品都必須遵守下列客戶資訊要求：

- 除非 BYOL （自帶授權） 要求，否則軟體不得在客戶不知情且明確同意的情況下收集或匯出客戶資料。收集或匯出客戶資料的應用程式必須遵循下列準則：
  - 客戶資料的收集必須是自助式、自動化且安全的。買方不需要等待賣方核准部署軟體。
  - 客戶資料的收集必須符合您與 的協議 AWS，包括但不限於 AWS Marketplace 條款與條件、AWS 服務條款、AWS 隱私權聲明和AWS 客戶協議。
  - 不得收集付款資訊。

產品使用需求

所有容器型產品都必須遵守下列產品使用需求：

- 賣方只能列出功能完整的產品。不允許用於試用或評估目的的 Beta 版或發行前產品。如果賣方 AWS Marketplace 在提供免費版本的 90 天內提供對等的付費版本，則支援商業軟體版本的開發人員、社群和 BYOL 版本。
- 所有容器型產品的使用說明必須包含部署容器型產品的所有步驟。使用說明必須提供指向 上對應容器映像的命令和部署資源 AWS Marketplace。
- 容器型產品必須包含訂閱者使用軟體所需的所有容器映像。此外，容器型產品不得要求使用者使用來自外部的任何映像來啟動產品 AWS Marketplace （例如，來自第三方儲存庫的容器映像）。
- 容器及其軟體必須以自助式方式部署，且不得需要額外的付款方式或成本。需要部署外部相依性的應用程式必須遵循下列準則：
  - 要求必須在清單的描述或使用說明中公開。例如，此產品需要網際網路連線才能正確部署。下列套件會在部署時下載：<list of package>。
  - 賣方負責使用並確保所有外部相依性的可用性和安全性。
  - 如果外部相依性不再可用， AWS Marketplace 也必須從 中移除產品。
  - 外部相依性不得需要額外的付款方式或成本。
- 需要持續連線到不受買方直接控制的外部資源的容器，例如外部 APIs 或由賣方或第三方 AWS 服務管理的容器，必須遵循下列準則：
  - 要求必須在清單的描述或使用說明中公開。例如，此產品需要持續的網際網路連線。需要下列持續的外部服務才能正常運作：<list of resources>。
  - 賣方負責使用並確保所有外部資源的可用性和安全性。
  - 如果外部資源不再可用， AWS Marketplace 也必須從 移除產品。
  - 外部資源不得需要額外的付款方式或成本，且連線的設定必須自動化。

- 產品軟體和中繼資料不得包含將使用者重新導向至其他雲端平台、其他產品或不可使用的追加銷售服務的語言 AWS Marketplace。
- 如果您的產品是另一個產品或其他 ISV 產品的附加元件，您的產品描述必須指出它擴展了另一個產品的功能，如果沒有它，您的產品具有非常有限的公用程式。例如，此產品擴展了 <product name> 的功能，如果沒有它，此產品的公用程式非常有限。請注意，<product name> 可能需要自己的授權才能使用此清單的完整功能。

架構需求

所有容器型產品都必須遵守下列架構要求：

- 的來源容器映像 AWS Marketplace 必須推送至 擁有的 Amazon Elastic Container Registry (Amazon ECR) 儲存庫 AWS Marketplace。您可以在伺服器產品下的 AWS Marketplace 管理入口網站 中，為每個容器產品清單建立這些儲存庫。
- 容器映像必須以 Linux 為基礎。
- 付費容器型產品必須能夠部署在 Amazon ECS、Amazon EKS 或 上AWS Fargate。
- 具有合約定價和 整合的付費容器型產品 AWS License Manager 應該部署在 Amazon EKS、Amazon ECS AWS Fargate、Amazon EKS Anywhere、Amazon ECS Anywhere、Red Hat OpenShift Service on AWS (ROSA)、內部部署或 Amazon Elastic Compute Cloud 上。

容器產品使用說明

建立容器產品的使用說明時，請遵循 中的步驟和指引the section called “AMI 和容器產品使用說明”。

Amazon EKS 附加元件產品的需求

Amazon EKS 附加元件是一種軟體，可為Kubernetes應用程式提供操作功能，但並非特定於應用程式。例如，Amazon EKS 附加元件包含可觀測性代理程式或Kubernetes驅動程式，允許叢集與基礎 AWS 資源互動，以進行聯網、運算和儲存。

身為容器產品的賣方，您可以選擇多種部署選項，包括 Amazon EKS。您可以將產品版本做為 AWS Marketplace 附加元件發佈至 Amazon EKS 附加元件目錄。您的附加元件會顯示在 Amazon EKS 主控台中，位於 AWS 和其他廠商維護的附加元件旁。您的買方可以將軟體部署為附加元件，就像部署其他附加元件一樣輕鬆。

如需詳細資訊，請參閱《Amazon EKS 使用者指南》中的 Amazon EKS 附加元件。

將容器產品準備為 AWS Marketplace 附加元件

若要將容器產品發佈為 AWS Marketplace 附加元件，必須符合下列要求：

- 您的容器產品必須在 中發佈 AWS Marketplace。
- 您的容器產品必須建置與 AMD64 和 ARM64 架構相容。
- 您的容器產品不得使用自帶授權 (BYOL) 定價模型。

> **Note**
>
> Amazon EKS 附加元件交付不支援 BYOL。

- 您必須遵循所有容器型產品需求，包括將所有容器映像和Helm圖表推送至 AWS Marketplace 受管 Amazon ECR 儲存庫。此需求包含開放原始碼映像，例如 nginx。映像和圖表無法託管在其他外部儲存庫中，包括但不限於 Amazon ECR Public Gallery、 Docker Hub和 Quay。
- Helm 圖表 - 準備並封裝您的軟體做為Helm圖表。Amazon EKS 附加元件架構會將Helm圖表轉換為 Kubernetes 資訊清單。Amazon EKS 系統不支援某些Helm功能。下列清單說明將軟體加入為 Amazon EKS 附加元件之前必須符合的要求。在此清單中，所有Helm命令都使用 3.8.1 Helm版：
  - 支援所有`Capabilities`物件，但 除外`.APIVersions`。 `.APIVersions`不支援non-built-in的自訂 Kubernetes API。 APIs
  - 僅支援 `Release.Name`和 `Release.Namespace` 物件。
  - Helm 不支援勾點和 `lookup`函數。
  - 所有相依圖表都必須位於主Helm圖表中 （ 以儲存庫路徑檔案：//... 指定 ）。
  - Helm 圖表必須成功通過 Helm Lint 和 Helm 範本，沒有錯誤。命令如下所示：
    - Helm Lint – `helm lint` *`helm-chart`*

      常見問題包括父圖表中繼資料中未宣告的圖表。例如 `chart metadata is missing these dependencies: chart-base Error: 1 chart(s) linted, 1 chart(s) failed`
    - Helm 範本 – `helm template` *`chart-name chart-location`* `–set k8version=`*`Kubernetes-version`* `–kube-version` *`Kubernetes-version`* `–namespace` *`addon-namespace`* `–include-crds –no-hooks –f` *`any-overriden-values`*

      使用 –f旗標傳遞任何覆寫的組態。
- 將所有容器二進位檔存放在 AWS Marketplace Amazon ECR 儲存庫中。若要建立資訊清單，請使用先前顯示的Helm範本命令。搜尋資訊清單以取得任何外部影像參考，例如 busybox或 gcr

影像。使用請求下拉式清單中的新增儲存庫選項，將所有容器映像以及相依性上傳至建立的 AWS Marketplace Amazon ECR 儲存庫。

- 自訂組態 – 您可以在部署期間新增自訂變數。如需如何識別最終使用者體驗、命名軟體`aws_mp_configuration_schema.json`，以及使用Helm圖表封裝成包裝函式的詳細資訊，請參閱 Amazon EKS 附加元件：進階組態。

  根據「$schema」關鍵字， `$schema` 必須是指向有效`application/schema+json`資源的 URI。

  此檔案不得接受任何敏感資訊，例如密碼、授權金鑰和憑證。

  若要處理秘密和憑證安裝，您可以為最終使用者提供安裝後或pre-Add-on元件安裝步驟。產品不應依賴任何外部授權。產品應該根據 AWS Marketplace 權利運作。

  如需 限制的詳細資訊`aws_mp_configuration_schema.json`，請參閱 附加元件供應商的附加元件組態需求和最佳實務。
- 識別並建立要部署軟體的命名空間 – 在產品的第一個版本中，您必須新增範本化命名空間，以識別要部署軟體的命名空間。
- 自訂資源定義 (CRDs) – Amazon EKS 附加元件架構不支援根據使用相同附加元件套用的 CRDs CRDs 和自訂資源宣告。如果您的附加元件具有自訂資源並依賴 CRDs，您可以：
  - 將兩個附加元件：分割 CRD 定義發佈至個別的附加元件 （ 個別的 Helm Chart)，並將實際的自訂資源安裝發佈至個別的附加元件。
  - 發佈具有其他手動指示的單一附加元件：發佈單一附加元件，在叢集上安裝 CRDs。提供使用說明以及 kubernetes 資訊清單檔案，以供最終使用者設定依賴這些 CRDs自訂資源。
- 如果適用，**`serviceAccount`**請建立 - 如果軟體是 上的付費軟體 AWS Marketplace 或必須與其他軟體連線 AWS 服務，請確定Helm圖表`serviceAccount`預設會建立。如果`serviceAccount`建立是由`values.yaml`檔案中的參數處理，請將參數值設定為 `true`。例如 `serviceAccount.create = true`。這是必要的，因為客戶可能會選擇從已具有必要許可的基礎節點執行個體繼承許可來安裝附加元件。如果 Helm Chart 未建立 `serviceAccount`，則許可無法繫結至 `serviceAccount`。
- 可追蹤的部署或協助程式集 – 確保您的 Helm Chart 具有協助程式集或部署。Amazon EKS 附加元件架構會使用它們來追蹤 Amazon EKS 資源的部署。如果沒有可追蹤的部署或協助程式集，您的附加元件將面臨部署錯誤。如果您的附加元件沒有部署或協助程式集，例如，如果您的附加元件部署了大量自訂資源或無法追蹤的 Kubernetes 任務，請新增虛擬部署或協助程式集物件。
- 支援 AMD 和 ARM 架構 – 許多 Amazon EKS 客戶現在使用 ARM64 來使用 AWS Graviton 執行個體。第三方軟體必須支援這兩種架構。

- 與來自 的授權或計量 APIs 整合 AWS Marketplace - AWS Marketplace 支援多個帳單模型。如需詳細資訊，請參閱容器產品計費、計量和授權整合。如果您想要透過 PAYG 機制銷售您的產品，請參閱 使用 AWS Marketplace Metering Service 設定容器產品的自訂計量。如果您想要透過預付或合約模型銷售產品，請參閱 使用 的容器產品的合約定價 AWS License Manager。
- 上傳軟體和所有成品和相依性 – Helm Chart 必須獨立，且不得需要來自外部來源的相依性，例如 GitHub。如果軟體需要外部相依性，則必須將相依性推送至相同 AWS Marketplace 清單下的 AWS Marketplace 私有 Amazon ECR 儲存庫。
- 在您的網站上提供部署指示 – 我們要求您託管部署指南，讓客戶了解如何透過 create-addon 命令部署您的軟體。
- 附加元件許可/IAM 角色 – 如果您從 發佈的附加元件 AWS Marketplace 需要存取 AWS 服務，您的軟體應該有一個 Kubernetes 服務帳戶，並標註 IAM 政策來存取 AWS 服務。您可以從兩個選項中選擇服務帳戶對 AWS 服務提出 API 請求：
  - 透過 IRSA 的登入資料：此選項可讓您的軟體從 Identity and Access Management (IAM) Role Service (IRSA) 取得擔任登入資料。如需詳細資訊，請參閱服務帳戶的 IAM 角色。
  - Amazon EKS Pod 身分：此選項可讓您的軟體使用 Amazon EKS Pod 的 Pod 身分向 AWS 服務提出 API 請求。如需詳細資訊，請參閱了解 EKS Pod Identity 如何授予 Pod 對 AWS 服務的存取權

  您的附加元件必須在與目前自訂組態結構描述 () 相同的目錄中，在 Helm Chart `aws_mp_addon_parameters.json` 的最上層具有名為 的額外組態檔案`aws_mp_configuration_schema.json`。目前，此檔案只會處理與 Pod 身分相容的許可。檔案格式如下：

```
{
  "permissions": {
      "isPodIdentityCompatible" : true,
      "permissionsList": [
       {
        "serviceAccount" : "String",
        "managedPolicies" : ["Policy Arn"],
       }
     ]
    }
  }
```

  檔案名稱： **`aws_mp_addon_parameters.json`**

> **Note**
>
> `aws_mp_addon_parameters.json` 檔案會在 Amazon EKS 主控台的附加元件組態設定頁面中啟用附加元件存取區段

欄位名稱	Type	備註	範例值
isPodIdentityCompatible	Boolean	目前僅支援 `true`。欄位顯示下列 permissionsList 清單中描述的許可是否與 Pod-identity 相符	TRUE
serviceAccount	字串	附加元件用來存取許可的服務帳戶名稱	`kpow`
managedPolicies	List<String>	可能由 EKS 附加元件擔任，用於此服務帳戶的政策 Arn 清單	`["arn:aws:iam::aws:policy/ReadOnlyAccess"]`

> **Note**
>
> 來自 Pay-as-you-go(PAYG) 附加元件產品 AWS Marketplace 無法使用 Amazon EKS Pod 身分，且必須使用服務帳戶的 IAM 角色 (IRSA) 進行存取控制。

- 版本更新 – Amazon EKS 會在上游發行數週後發行新的 Kubernetes 版本。隨著新的 Amazon EKS 叢集版本正式推出，廠商有 45 天的時間來認證或更新其軟體，使其與新的 Amazon EKS 叢集版本版本相容。如果您目前的附加元件版本支援新的 Kubernetes 版本，請驗證並認證相同的版本，以便我們更新版本相容性矩陣。如果需要新的附加元件版本以支援新的 Kubernetes 版本版本，請提交新版本以供加入。
- 合作夥伴的軟體必須屬於下列其中一種類型，或是可增強 Kubernetes 或 Amazon EKS 的操作軟體：Gitops | 監控 | 記錄 | cert-management | 政策管理 | 成本管理 | Autoscaling | 儲存 | kubernetes-

management | service-mesh | etcd-backup | ingress-service-type | load-balancer | local-registry | networking | Security | 備份 | 輸入控制器 | 可觀測性

- 軟體不能是容器網路界面 (CNI)。
- 軟體必須透過 銷售， AWS Marketplace 並與付費產品的授權和計量 APIs整合。不接受 BYOL 產品。

附加元件供應商的附加元件組態需求和最佳實務

Amazon EKS 需要來自附加元件供應商的 Helm JSON 結構描述字串組態。需要必要組態或允許選用組態的附加元件必須包含提交 Helm Chart `aws_mp_configuration_schema.json` 的檔案 AWS Marketplace。Amazon EKS 將使用此結構描述來驗證客戶的組態輸入，並拒絕具有不符合結構描述之輸入值的 API 呼叫。附加元件組態通常分為兩個類別：

- 一般 Kubernetes 屬性的組態，例如標籤、公差、 nodeSelector 等。
- 附加元件特定的組態，例如授權金鑰、功能啟用、URLs等。

本節著重於與一般 Kubernetes 屬性相關的第一個類別。

Amazon EKS 建議遵循有關 Amazon EKS 附加元件組態的最佳實務。

- 結構描述要求
- 允許組態的常見參數
- 不允許用於組態的常見參數

結構描述要求

定義 json 結構描述時，請確定您使用 Amazon EKS 附加元件支援的 jsonschema 版本。

支援的結構描述清單：

- 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

使用任何其他 json 結構描述版本與 Amazon EKS 附加元件不相容，將導致附加元件無法釋出，直到修正此問題為止。

範例 Helm 結構描述檔案

```
{
"$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

組態參數必須是 camelCase，如果未遵循此格式，則會遭到拒絕。

描述為必要項目

一律包含結構描述屬性的有意義的描述。此描述將用於在每個組態參數的 Amazon EKS 主控台中轉譯標籤名稱。

RBAC 定義

附加元件提供者需要使用最低權限原則來定義和提供成功安裝附加元件所需的 RBAC 許可。如果 RBAC 許可需要變更較新版本的附加元件或任何修正以解決 CVE，則附加元件提供者將需要將此變更通知 Amazon EKS 團隊。每個 Kubernetes 資源的必要許可應限於物件的資源名稱。

```
apiGroups: ["apps"]
resources: ["daemonsets"]
resourceNames: ["ebs-csi-node"]
verbs: ["create", "delete", "get", "list", "patch", "update", "watch"]
```

秘密管理

本節僅適用於需要客戶設定機密資訊的附加元件，例如應用程式金鑰、API 金鑰、密碼等。由於安全性影響，Amazon EKS APIs 目前不支援以純文字形式傳入秘密資訊。不過，客戶可以使用組態來傳入包含附加元件所需金鑰的 Kubernetes Secret 名稱。客戶需要建立 Kubernetes Secret 物件，其中包含與先決條件步驟具有相同命名空間的金鑰，然後在建立附加元件時使用組態 Blob 傳入 Secret 的名稱。我們建議附加元件提供者為結構描述屬性命名，以便客戶不會意外地將其誤認為實際金鑰。例如：appSecretName、connectionSecretName 等。

總而言之，附加元件提供者可以利用結構描述，讓客戶能夠傳入秘密的名稱，但不能傳遞實際上會保留秘密本身的金鑰。

範例組態值

您可以在結構描述中包含組態範例，以協助客戶設定附加元件。下列範例來自 AWS Distro for OpenTelemetry 附加元件的結構描述。

```
"examples": [
      {
        "admissionWebhooks": {
          "namespaceSelector": {},
          "objectSelector": {}
        },
        "affinity": {},
        "collector": {
          "amp": {
            "enabled": true,
            "remoteWriteEndpoint": "https://aps-workspaces.us-west-2.amazonaws.com/
workspaces/ws-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/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": []
    }
  ]
```

允許組態的常見參數

以下是面向客戶的 Helm 結構描述檔案中的建議參數。

參數	描述	應該有預設值嗎？
additionalLabels	將 Kubernetes 標籤新增至附加元件管理的所有 Kubernetes 物件。	否
additionalAnnotations	將 Kubernetes 註釋新增至附加元件管理的所有 Kubernetes 物件。	否
podLabels	將 Kubernetes 標籤新增至 附加元件管理的 Pod。	否
podAnnotations	將 Kubernetes 註釋新增至附加元件管理的 Pod。	否
logLevel	附加元件管理之元件的日誌層級。	是
nodeSelector	建議最簡單的節點選取限制條件。您可以將 nodeSelector 欄位新增至 Pod 規格，並指定您希望目標節點擁有的節點標籤。	可能，例如僅限 Linux 節點
容差	公差會套用至 Pod。容錯允許排程器使用相符的污點來排程 Pod。容錯允許排程，但不保證排程。	或許， 協助程式集更常見

參數	描述	應該有預設值嗎？
親和性	親和性功能包含兩種類型的親和性：節點親和性函數，例如 nodeSelector 欄位，但更具表達性，並可讓您指定軟性規則，Pod 間親和性/反親和性可讓您限制 Pod 與其他 Pod 上的標籤。	也許可以
topologySpreadConstraints	您可以使用拓撲分散限制來控制 Pod 如何在故障網域之間分散到叢集，例如區域、區域、節點和其他使用者定義的拓撲網域。這有助於實現高可用性和高效的資源使用率。	也許可以
資源請求/限制	指定每個容器需要多少 cpu/記憶體。強烈建議設定請求。限制是選用的。	是
複本	附加元件管理的 Pod 複本數量。不適用於協助程式集。	是

> **Note**
>
> 對於工作負載排程組態參數，您可能需要在必要時將結構描述中的頂層元件分開。例如，Amazon EBS CSI 驅動程式包含兩個主要元件：控制器和節點代理程式 - 客戶需要每個元件的不同節點選擇器/容錯。

> **Note**
>
> JSON 結構描述中定義的預設值僅用於使用者文件，不會取代`values.yaml`檔案中具有正確預設值的需求。如果使用預設屬性，請確保 中的預設值`values.yaml`符合結構描述中的預設值，且每當對 Helm Chart 進行變更時，兩個成品 (`values.schema.json` 和 `values.yaml`) 都會保持同步。

```
"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"
            ]
          }
```

不允許用於組態的常見參數

各種附加元件（例如 Elastic Load Balancing 控制器）可能需要叢集中繼資料參數，例如 `clusterName` `region` `vpcId`、`accountId`、、 和其他參數。Amazon EKS 服務已知的任何類似參數都會由 Amazon EKS 附加元件自動注入，而不會承擔使用者將 指定為組態選項的責任。這些參數包括：

- AWS 區域
- Amazon EKS 叢集名稱
- 叢集的 VPC ID
- 容器登錄檔，特別是用於 build-prod 帳戶，供聯網附加元件使用
- DNS 叢集 IP，特別是核心附加元件
- Amazon EKS 叢集 API 端點
- 叢集上已啟用 IPv4
- 叢集上已啟用 IPv6
- 叢集上啟用 IPv6 的字首委派

附加元件提供者需要確保您已為此類適用參數定義範本。上述每個參數都會有 Amazon EKS 定義的預先定義`parameterType`屬性。發行中繼資料將指定 範本中 參數`parameterType`的名稱/路徑之間的映射。如此一來，Amazon EKS 即可動態傳入這些值，而無需客戶透過組態指定這些值，也提供附加元件提供者彈性來定義自己的範本名稱/路徑。Amazon EKS 動態注入所需的上述參數應該從結構描述檔案中排除。

來自發行中繼資料的範例映射

```
"defaultConfiguration": [
       {
            "key": "image.containerRegistry",
            "parameterType": "CONTAINER_REGISTRY"
       }
]
```

以下是不建議在面向客戶的 Helm 結構描述檔案中設定的參數。這些參數應具有不可修改的預設值，或完全不包含在附加元件範本中。

參數	描述	應該有預設值嗎？
image	將部署在 Kubernetes 叢集上的容器映像。	否，透過附加元件定義進行管理
imagePullSecrets	設定 Pod 以使用秘密從私有登錄檔提取。	N/A
livenessProbe	Kubelet 程序使用活體探測來知道何時重新啟動容器。例如，活體探測可能會捕捉應用程式正在執行但無法進行進度的死結。重新啟動處於此類狀態的容器有助於讓應用程式在發生錯誤時更可用。	是
readinessProbe	請務必為容器準備準備探查。如此一來，在資料平面上執行的 Kubelet 程序就會知道容器何時準備好為流量提供服務。當 Pod 的所有容器都就緒時，就會將其視為就緒。此訊號的一個用途是控制哪些 Pod 用作 服務的後端。當 Pod 尚未就緒時，會從服務負載平衡器中移除。	是
startupProbe	kubelet 使用啟動探查來了解容器應用程式何時啟動。如果已設定此類探查，則會停用活體和準備度檢查，直到成功為止，確保這些探查不會干擾應用程式啟動。這可用於在啟動緩慢的容器上採用活體檢查，避免它們在啟動和執行之前被 kubelet 刪除。	選用
podDisruptionBudget	定義 Pod 中斷預算 (PDB)，以確保在自願中斷期間持續執行的 PODS 數	是，如果預設為兩個以上的複本

參數	描述	應該有預設值嗎？
	量下限。PDB 會限制複寫應用程式因自願中斷而同時關閉的 Pod 數量。例如，以規定人數為基礎的應用程式想要確保執行的複本數目永遠不會低於規定人數所需的數目。Web 前端可能想要確保提供負載的複本數目永遠不會低於總數的特定百分比。	
serviceAccount （名稱）	服務帳戶 Pod 的名稱將在 下執行。	是
serviceAccount （註釋）	套用至服務帳戶的註釋。通常用於服務帳戶的 IAM 角色功能	否，IAM 服務帳戶角色 ARN 是在頂層 Amazon EKS 附加元件 API 中設定。此規則的例外是，如果您的附加元件具有多個部署/控制器 （例如 Flux)，並且需要單獨的 IRSA 角色 ARNs。
priorityClassName	優先順序表示 Pod 相對於其他 Pod 的重要性。如果無法排程 Pod，排程器會嘗試先佔 （移出） 優先順序較低的 Pod，以便排程待定的 Pod。	是。大多數附加元件對叢集功能至關重要，且預設應設定優先順序類別。
podSecurityContext	安全內容會定義 Pod 或容器的權限和存取控制設定。通常用於設定 fsGroup - 這是 v1.19 和較低叢集中 IRSA 所需的。	不太可能，指定的 Amazon EKS 不再支援 Kubernetes 1.19 版

參數	描述	應該有預設值嗎？
securityContext	安全內容會定義 Pod 或容器的權限和存取控制設定。	是
updateStrategy	指定用來將舊 Pod 取代為新 Pod 的策略。	是
nameOverride	覆寫 Pod 的名稱。	否
podSecurityPolicy	對參數強制執行限制。	否 - PSPs已棄用
extraVolumeMounts/extraVolumes	用於非 Amazon EKS 叢集中的 IRSA。	否

的容器產品定價 AWS Marketplace

在 上 AWS Marketplace，您可以列出免費產品、自有授權模型 (BYOL) 產品，以及 Amazon Elastic Container Service (Amazon ECS)、Amazon Elastic Kubernetes Service (Amazon EKS) 和 的付費產品 AWS Fargate。每個產品只能設定一個價格。本主題概述容器產品的可用定價模型。

> **Note**
>
> 您可以使用 AWS Marketplace 計量服務來強制執行付費產品的權利和計量用量。對於每個任務或每個 Pod 定價，用量會自動計量 AWS。

您為容器產品設定的價格適用於所有 AWS 區域。每當您降低容器產品的價格時，就會立即為您的買方實作新的價格。對於價格上漲，現有買方會在變更影響其帳單前 90 天收到通知。新買方需支付新金額的費用。

> **Note**
>
> 對於新訂閱者，價格變更會立即生效。對於現有的訂閱者，價格變更會在從傳送價格變更通知當日開始的 90 天期間之後的下個月第一天生效。例如，假設您在 3 月 16 日傳送價格變更通

知。6 月 16 日大約是 3 月 16 日之後的 90 天。由於價格變更發生在 90 天期間之後當月的第一天，因此變更的生效日期是 7 月 1 日。

主題

- 容器定價模型
- 容器產品的合約定價

容器定價模型

AWS Marketplace 有多個容器產品的定價模型。

下表提供有關容器型產品的定價模型的一般資訊。

容器產品的定價模型

定價方式	描述
使用自有授權 (BYOL)	BYOL 是透過您與買方維持 AWS Marketplace 的外部帳單關係在 外部進行管理。
每月	每月固定價格 每月固定價格，為使用者提供下個月無限的產品用量。 範例：您將產品的價格設定為每月 99 美元。您的產品包含三個使用 Amazon ECS 任務定義部署的不同容器映像。 購買者訂閱您的產品後，會立即向他們收取 99 美元的費用，該費用每月重複一次，直到他們取消訂閱為止。買方也會獲得產品的無限用量。買方也會針對任務執行所在的任何基礎設施單獨付費。訂閱後，他們可以存取您的容器映像。他們可以在任何組態中，從 Amazon ECS 或 Amazon EKS 上的這些映像啟動和執行任意數量的容器。 如果買方在一個月中取消其訂閱，則會失去 AWS Marketplace 存放容器映像的 Amazon ECR 儲存庫存取權。買方可能已提取並存放原始映像。不過，他們無法再提取您透過 提供的新容器映像版本 AWS Marketplace。買方會獲得最後一個月未使用部分的退款。您是根據買方的使用量減去約定 AWS Marketplace 費用來支付。

定價方式	描述
自訂指標定價維度	根據您定義的維度 （例如使用者、節點、儲存庫或 GB) 自訂計量價格，每個產品最多 24 個維度。 範例：使用者收取的產品費用。您有管理員使用者和一般使用者，而您定義定價為管理員使用者 2 美元，一般使用者 1 美元。您可以在列出產品時將它們設定為不同的維度。您每天會向登入的使用者收費，並測量每天的用量。
每個任務或每個 Pod 每小時價格	Amazon ECS 任務或 Amazon EKS Pod 每個 Amazon ECS 任務或每個 Amazon EKS Pod 定價，我們以每小時的價格設定來測量到第二個。 範例：您的產品包含三種不同的容器映像：控制器節點、工作者節點和分析節點。由於您的產品沒有控制器節點就無法運作或有用，因此您決定 是您要收取用量費用的映像。您設定每小時 6 美元的價格。 您可以修改 容器映像中的軟體，讓控制器節點與 AWS Marketplace Metering Service `RegisterUsage` API 操作整合。這可確保只有具有作用中訂閱的買方才能啟動和執行該容器映像，並根據其執行時間來衡量其用量。 每個執行中的 Amazon EKS 控制器 Pod 每小時都會向買方收取 6 美元的使用費。如果買方啟動五個包含控制器節點容器的 Amazon EKS 控制器 Pod，則會收取每小時 30 美元的費用 （每個 Pod 6 美元）。買方也會分別支付 Pod 執行的任何基礎設施的費用。 對於每小時定價，計費為每秒，最少 1 分鐘。如果客戶執行此控制器容器 20 分鐘 30 秒，則會向 收費`20 x ($6/60) + 30 x ($6/60/60) = $2 + $0.05 = $2.05`。您是根據買方的使用量減去約定 AWS Marketplace 費用來支付。

定價方式	描述
具有長期合約的每小時/用量	長期合約，以較低的價格預付或定期分期付款。長期合約可以新增至具有自訂計量定價的現有產品，或每個任務和每個 Pod 定價。買方在取用超過長期合約中購買的價格時，會支付計量價格。 範例：對於計量定價模型，您可以為買方新增長期合約價格，以獲得預付的折扣。假設您通常會對使用的某些單位收取 1 美元的費用。使用每小時 1 個單位的買方每年將支付 8760 美元 (`365 days x 24 hours x $1 per hour`)。您可以啟用合約，讓買方以該價格的一半 (4380 美元），對那些 365 天每小時使用 1 個單位。在這種情況下，買方承諾預付一年合約的費用，價格從每單位 1 美元降至每單位 0.5 美元。您也可以讓買方購買其中多個合約。如果計量的數量顯示買方在一小時內耗用 10 個單位，而且他們有兩個合約，則 2 個單位將包含在 2 個合約中。這 8 個額外單位將以每小時 1 美元的一般費率計費，該小時總計 8 美元。 對於每個任務或每個 Pod 範例，您也可以為買方新增長期合約價格，以取得預付的折扣。如果您通常為每個 Pod 收取 6 美元，您可以設定 365 天的長期合約持續時間，價格為 13，140 美元 (`365 days x 24 hours x $3 per pod per hour`)。然後，在這 365 天期間，一份合約將授予客戶每小時 1 個 Pod。客戶可以選擇購買多個合約。例如，客戶可以購買兩個合約，讓他們每小時 2 個 Pod。如果客戶每小時執行的 Pod 超過授權的合約，則多餘的 Pod 將按您正常的每小時價格計費。 在這兩種情況下，購買長期合約的買方都會預先計費，可以是一次性付款或定期排程的未來付款。買方也會依計量費率，支付超出合約的任何額外用量的費用。
容器合約定價	具有合約定價的容器 – 以容器為基礎的產品，買方會預付費用。

容器產品的合約定價

對於具有合約定價的容器型產品， 會根據您與客戶之間的合約，預先或根據您定義的付款排程向您的客戶 AWS Marketplace 收費。之後，他們有權使用這些資源。

若要設定定價，請選擇您提供給客戶的一或多個合約持續時間。您可以為每個合約期間輸入不同的價格。您的選項為 1 個月、12 個月、24 個月和 36 個月的持續時間。對於私有優惠，您可以指定以月為單位的自訂持續時間 （最多 60 個月）。

選擇最能描述您產品定價的類別。定價類別會顯示在 AWS Marketplace 網站上的客戶。您可以選擇頻寬 (GB/s、MB/s)、資料 (GB、MB、TB)、主機、請求、層或使用者。如果沒有預先定義的類別符合您的需求，您可以選擇更通用的單位類別。

優惠最多可新增 24 個維度。每個維度都需要下列資料：

- 合約類別 – 如果產品支援根據合約定價的消耗計量，則合約類別用於測量或計量您的產品。對於沒有以消耗為基礎的定價的合約產品，您可以選擇最接近合約中維度類別的類別。如果沒有與合約中維度的單位類似的值，請選擇 `Units`。
- 合約單位 – 如果產品支援以消耗為基礎的計量，則合約單位會與計量類別搭配使用。根據選取的類別，為最符合您維度的單位選擇其中一個可用值。
- 合約維度 允許多次購買 – 此欄位用於指示優惠是分層定價優惠還是允許購買多個維度的非分層優惠。

  分層優惠 – 允許買方僅訂閱優惠中其中一個可用的維度。分層優惠中的維度沒有數量的概念。簽署具有特定維度的合約基本上表示買方已選擇該維度所指示的特定功能。

  非分層優惠 – 允許客戶在合約中採購多個維度，並允許他們採購每個維度的多個單位。

  為此欄位設定 true 值表示優惠是非分層優惠。為此欄位設定 false 值表示優惠是分層優惠。

使用產品載入表單 (PLF) 建立容器產品的合約時，您必須為您的定價維度定義下列欄位：

- Contracts DimensionX API Name – 應該出現在買方 AWS License Manager 帳戶中產生的授權中的名稱。此名稱也會做為 `Checkoutlicense` API 呼叫`Entitlement`中 `Name` 的值。
- Contracts DimensionX 顯示名稱 – 面向客戶的名稱，將顯示在 AWS Marketplace 網站的產品詳細資訊和採購頁面上。建立易於使用的名稱 名稱的長度上限為 24 個字元。清單為公有之後， `Name` 的值就無法變更。
- Contracts DimensionX 描述 – 面向客戶的維度描述，提供有關產品維度的其他資訊，例如特定維度提供的功能。描述的長度上限為 70 個字元。
- Contracts DimensionX 數量 – 用於計算產品協議修訂時的按比例分配。對於所有合約優惠，此欄位的這個值應該設定為 1。不應對其進行編輯。
- Contracts DimensionX 1-Month費率 – 針對此維度收取 1 個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援八個小數位數。
- Contracts DimensionX 12-Month費率 – 針對維度收取 12 個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援八個小數位數。

- Contracts DimensionX 24-Month費率 – 針對維度收取 24 個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援八個小數位數。
- Contracts DimensionX 36-Month費率 – 針對維度收取 36 個月權利的合約費率。對於非分層優惠，此費率會針對所採購維度的每個單位收費。此欄位支援八個小數位數。

範例：資料儲存應用程式

	1 個月價格	12 個月價格	24 個月價格	P36-month價格
未加密的資料 (GB)	$1.50/GB	$16.00/GB	$30.00/GB	$60.00/GB
加密資料 (GB)	$1.55/GB	$16.60/GB	$31.20/GB	$61.20/GB

範例：日誌監控產品

	1 個月價格	12 個月價格	24 個月價格	36 個月價格
基本（監控 10 個主機，監控 5 個容器）	100 美元	1000 美元	2000 美元	4000 美元
標準（監控 20 個主機，監控 10 個容器）	$200	2000 美元	4000 美元	8000 美元
Pro（監控 40 個主機，監控 20 個容器）	400 美元	4000 美元	8000 美元	16，000 美元
每小時監控的其他主機	10 美元	100 美元	$200	400 美元
每小時監控的其他容器	10 美元	100 美元	$200	400 美元

> **Note**
>
> 價格可能持續下列期間：1 個月、12 個月、24 個月或 36 個月。您可以選擇為您的產品提供一或多個選項。每個維度的持續時間必須相同。
>
> Example
>
> 例如，在您有 `ReadOnlyUsers`和 `AdminUsers`維度的情況下，如果您提供 ReadOnlyUsers 的年價，您`AdminUsers`也必須提供 的年價。

自動續約

當客戶 AWS Marketplace 使用容器合約購買您的產品時，他們可以同意自動續約合約條款。客戶每個月或 1、2 或 3 年繼續支付 權限。

客戶可以隨時修改續約設定。如需詳細資訊，請參閱 AWS Marketplace 買方指南中的修改現有合約。

當容器合約結束時

容器合約產品有合約到期。當合約結束時，會發生下列事件：

1. 您的容器產品會收到`entitlement-updated`通知，指出買方的權限已變更，而 AWS Marketplace Entitlement Service 會傳回空的回應。
2. 您有一小時的時間來衡量客戶的任何剩餘用量。之後，您就無法再傳送此客戶的計量記錄。

容器產品計費、計量和授權整合

AWS Marketplace 與其他 整合 AWS 服務 ，為您的容器產品提供計量和以合約為基礎的定價。對於使用量定價的容器型產品，您可以使用 AWS Marketplace Metering Service來檢查是否有權使用您的產品和計量用量來計費。對於具有合約定價的容器型產品，您可以使用 AWS License Manager 將授權與產品建立關聯。下列各節提供有關使用 AWS Marketplace Metering Service 的每小時和自訂計量，以及 的合約定價的詳細資訊 AWS License Manager。

主題

- 使用 的每小時和自訂計量 AWS Marketplace Metering Service
- 使用 的合約定價 AWS License Manager
- 使用 AWS Marketplace Metering Service 設定每小時計量

- 使用 AWS Marketplace Metering Service 設定容器產品的自訂計量
- 使用 的容器產品的合約定價 AWS License Manager

使用 的每小時和自訂計量 AWS Marketplace Metering Service

若要同時檢查使用您產品的權利和計量帳單用量，請使用 AWS Marketplace 計量服務。如果您想要定義自己的定價單位和計量，讓我們使用 進行計費，請使用 MeterUsage API 操作整合 。如果您想要根據使用的任務或 Pod 數量定價產品，並自動讓 AWS 計量該用量，請使用 RegisterUsage API 操作進行整合。對於這兩種定價類型，您可以新增長期合約價格，而無需變更與 整合的方式 AWS Marketplace Metering Service。

當您在 中建立新的容器產品時 AWS Marketplace 管理入口網站，我們會提供一組產品識別符 （產品代碼和公有金鑰），用於整合您的產品與 AWS Marketplace Metering Service。

Entitlement

與 整合 AWS Marketplace Metering Service 可讓您驗證執行付費軟體的客戶是否已訂閱您的產品 AWS Marketplace，以防止您在容器啟動時遭到未經授權的使用。若要驗證權利，請根據您的定價模型，使用 MeterUsage 或 RegisterUsage API 操作。對於每小時和固定每月定價模型，請使用 `RegisterUsage` API 操作。對於自訂計量定價模型，請使用 `MeterUsage` API 操作。

如果買方無權使用您的產品，這些 API 操作會傳回`CustomerNotEntitledException`例外狀況。

> **Note**
>
> 如果買方在執行產品時取消訂閱，他們有權繼續執行產品。不過，它們無法為您的產品啟動其他容器。

整合準則

當您建立和發佈容器產品，並使用 `MeterUsage`或 `RegisterUsage` API 操作進行權利和計量時，請記住下列準則：

- 請勿在軟體或 Docker 容器映像中設定 AWS 登入資料。當容器映像在 Amazon ECS 任務或 Amazon EKS Pod 中執行時，會在執行時間自動取得買方的 AWS 登入資料。
- 若要從 Amazon EKS 呼叫 `MeterUsage`或 `RegisterUsage` API 操作，您必須使用支援的 AWS SDK。若要測試`MeterUsage`或`RegisterUsage`整合 Amazon EKS，您必須執行執行 Kubernetes 1.13.x 或更新版本的 Amazon EKS 叢集。 AWS Identity and Access Management (IAM) 角色需要

Kubernetes 1.13 才能支援 Pod。執行中的 Pod 需要 IAM 角色，才能取得在 Amazon EKS 上叫用這些動作所需的 AWS 登入資料。

- 您可以進行本機開發，但您會收到`PlatformNotSupportedException`例外狀況。當您在容器服務 (Amazon ECS、Amazon EKS 和 Fargate) 上啟動 AWS 容器時，不會發生此例外狀況。

支援的 AWS 區域

如需所有 AWS Marketplace 支援的清單 AWS 區域，請參閱全球基礎設施網站上的區域表。

取得用於計量 AWS 區域 的

將用於計量的容器與 `MeterUsage`或 `RegisterUsage` API 操作整合時，請勿將 AWS SDK 設定為使用特定 AWS 區域。區域必須在執行時間動態取得。

Example

例如，客戶啟動 Amazon ECS 任務或 Amazon EKS Pod。`RegisterUsage` API 操作會在與啟動 Amazon ECS 任務或 Amazon EKS Pod 的區域不同的區域中呼叫。因此，`RegisterUsage`API 操作會擲回`InvalidRegionException`錯誤。

AWS SDK 語言不會`AWS_REGION`以一致的方式判斷 。如果您的 SDK 未自動挑選 `AWS_REGION`，則需要手動撰寫軟體來判斷 `AWS_Region`。例如，當環境變數或其他組態不存在時， 適用於 Java 的 AWS SDK 會自動使用 Amazon EC2 執行個體中繼資料 （特別是 `ec2InstanceMetadata`) 來取得區域。在此執行個體中，只有在環境`AWS_REGION`變數不存在`ec2InstanceMetadata`時才呼叫 。

如需有關如何在 AWS 區域 執行時間動態取得 的資訊，請參閱適用於您的程式設計語言的 AWS SDK 開發人員指南。

防止計量修改

介紹買方修改或覆寫對 的呼叫，`RegisterUsage`或`MeterUsage`可能導致不良帳單和付款問題的方法。我們強烈建議您整合計量和權利邏輯。

設計您的產品以防止計量修改時，請記住下列事項：

- 如果買方可以插入包含 `CMD`或 `ENTRYPOINT`指示的新映像層，請直接將 `RegisterUsage`或 整合`MeterUsage`到買方透過容器映像執行的軟體中。否則，買方可能會覆寫透過`RegisterUsage`基本映像`CMD`或從`ENTRYPOINT`基本映像`MeterUsage`執行的呼叫。
- 我們建議您管理軟體用來輸入 `RegisterUsage` 或以買方無法修改`MeterUsage`的方式使用 AWS Marketplace 的產品代碼。不過，如果您的產品以客戶可以覆寫的方式管理產品代碼，例如 AWS

CloudFormation、Helm Chart 或 Kubernetes 資訊清單，您必須維護信任 AWS Marketplace 的產品代碼清單。這是為了確保軟體做為輸入傳遞至 `RegisterUsage`或 的產品程式碼`MeterUsage`有效。

- 如果您的任何受信任產品代碼是免費產品，請確保它們無法取代付費產品代碼。

使用 的合約定價 AWS License Manager

對於具有合約定價的容器型產品，您可以使用 AWS License Manager 將授權與產品建立關聯。

AWS License Manager 是一種授權管理工具，可讓您的應用程式追蹤和更新客戶購買的授權 （ 也稱為權利 ）。本節提供如何將您的 產品與 整合的相關資訊 AWS License Manager。整合完成後，您可以在其上發佈您的產品清單 AWS Marketplace。

如需 的詳細資訊 AWS License Manager，請參閱 AWS License Manager 使用者指南和 AWS CLI 命令參考的 AWS License Manager一節。

> **Note**
>
> - 客戶無法在合約到期期間之後啟動容器的新執行個體。不過，在合約期間，他們可以啟動任意數量的執行個體。這些授權不會繫結至特定節點或執行個體。只要在任何節點上任何容器上執行的任何軟體具有指派 AWS 的登入資料，就可以簽出授權。
> - 私有優惠建立 – 賣方可以使用 中的私有優惠建立工具，為產品產生私有優惠 AWS Marketplace 管理入口網站。
> - 報告 – 您可以在 的報告區段中設定 Amazon S3 儲存貯體，以設定資料摘要 AWS Marketplace 管理入口網站。如需詳細資訊，請參閱中的賣方報告、資料饋送和儀表板 AWS Marketplace。

整合工作流程

下列步驟顯示將容器產品與 整合的工作流程 AWS License Manager：

1. 賣方建立具有 AWS License Manager 整合的產品。
2. 賣方列出產品 AWS Marketplace。
3. 買方在 上找到 AWS Marketplace 並購買產品。
4. 授權會傳送給其 中的買方 AWS 帳戶。

5. 買方透過啟動 Amazon EC2 執行個體、Amazon ECS 任務或 Amazon EKS Pod 軟體來使用軟體。客戶使用 IAM 角色部署 。
6. 軟體會讀取買方 AWS License Manager 帳戶中的授權、探索購買的權利，並相應地佈建功能。

> **Note**
>
> License Manager 不會執行任何追蹤或更新；這由賣方的應用程式完成。

使用 AWS Marketplace Metering Service 設定每小時計量

> **Note**
>
> 對於 Amazon EKS 部署，您的軟體必須使用服務帳戶 (IRSA) 的 IAM 角色來簽署 API 操作的 `RegisterUsage` API 呼叫。不支援使用 EKS Pod Identity、節點角色或長期存取金鑰。
> 對於 Amazon ECS 部署，您的軟體必須使用 Amazon ECS 任務 IAM 角色來簽署 API 操作的 `RegisterUsage` API 呼叫。不支援使用節點角色或長期存取金鑰。

如果您的容器產品使用每小時每個任務或每個 Pod 定價，而不是自訂計量定價維度，則不需要定義自訂計量維度。您可以使用 AWS Marketplace Metering Service 搭配 中的容器產品進行每小時計量 AWS Marketplace。下列各節說明如何使用 AWS Marketplace Metering Service 設定每小時計量。

`RegisterUsage` API 操作會測量每個 Amazon Elastic Container Service (Amazon ECS) 任務或每個 Amazon Elastic Kubernetes Service (Amazon EKS) Pod 每小時使用的軟體，使用量按比例分配到第二個。最少 1 分鐘的用量適用於短期運作的任務或 Pod。軟體使用的連續計量由 自動處理 AWS Marketplace Metering Control Plane。您的軟體不需要執行任何計量特定動作，除了呼叫`RegisterUsage`一次 開始計量軟體使用。

`RegisterUsage` 必須在啟動容器時立即呼叫 。如果您未在容器啟動的前 6 小時內註冊容器，AWS Marketplace Metering Service 不會提供前幾個月的任何計量保證。不過，計量會繼續執行當月，直到容器結束為止。

無論客戶的訂閱狀態為何， AWS Marketplace Metering Control Plane都會繼續向客戶收取執行 Amazon ECS 任務和 Amazon EKS Pod 的費用。這樣就不需要軟體在任務或 Pod 初次成功啟動後執行權利檢查。

如需整合 AWS Marketplace Metering Service API 與容器產品與每小時定價的詳細資訊，請參閱與AWS Marketplace 賣方研討會的每小時計量實驗室整合。

主題

- 每小時計量先決條件
- 測試 的整合 RegisterUsage
- 的錯誤處理 RegisterUsage
- 使用 整合您的容器產品與 AWS Marketplace Metering Service 適用於 Java 的 AWS SDK

每小時計量先決條件

在發佈產品之前，您必須執行下列動作：

1. 在 中建立新的容器產品 AWS Marketplace 管理入口網站，並記下其產品代碼。

   如需詳細資訊，請參閱概觀：建立容器產品。

2. 使用 AWS Identity and Access Management (IAM) 角色執行應用程式的任務或 Pod，並具有呼叫 所需的 IAM 許可`RegisterUsage`。IAM 受管政策`AWSMarketplaceMeteringRegisterUsage`具有這些許可。如需政策的詳細資訊，請參閱《AWS 受管政策參考》中的 AWSMarketplaceMeteringFullAccess。

3. （選用）如果您想要查看記錄，建議您在任務或 Pod 定義中啟用 AWS CloudTrail 記錄。

4. 使用您定義的所有定價維度的記錄來測試呼叫 `RegisterUsage` API 操作。

測試 的整合 **`RegisterUsage`**

使用 `RegisterUsage` API 操作來測試整合，然後再將映像提交至 AWS Marketplace 進行發佈。

在 Amazon ECS 或 Amazon EKS 上執行您的產品，`RegisterUsage`從容器映像呼叫 。使用您用來列出產品的 AWS 帳戶 AWS Marketplace。您的計量整合必須動態設定 AWS 區域，而不是硬式編碼。不過，在測試時，請啟動至少一項 Amazon ECS 任務或 Amazon EKS Pod，其中包含美國東部（維吉尼亞北部）區域中的付費容器。透過這樣做， AWS Marketplace 操作團隊可以驗證您在該區域中使用日誌的工作。

> **Note**
>
> 如果您的產品同時支援 Amazon ECS 和 Amazon EKS，您只需要在 Amazon EKS 中啟動 ，讓我們驗證您的整合。

在產品發佈所有必要中繼資料和定價資訊之前，您無法完整測試整合。如果請求， AWS Marketplace 目錄操作團隊可以驗證接收您的計量記錄。

的錯誤處理 **RegisterUsage**

如果您的容器映像與 整合， AWS Marketplace Metering Service 並收到容器啟動`ThrottlingException`時 以外的例外狀況，您應該終止容器，以防止未經授權的使用。

以外的例外`ThrottlingException`狀況只會在初始呼叫 `RegisterUsage` API 操作時擲回。`CustomerNotSubscribedException` 即使客戶在任務或 Pod 仍在執行時取消訂閱，來自相同 Amazon ECS 任務或 Amazon EKS Pod 的後續呼叫也不會擲回。這些客戶在取消訂閱並追蹤其用量後，仍需支付執行中容器的費用。

下表說明 `RegisterUsage` API 操作可能擲回的錯誤。每種 AWS SDK 程式設計語言都有一組錯誤處理準則，您可以參閱這些準則以取得其他資訊。

錯誤	Description
`InternalServiceErrorException`	`RegisterUsage` 無法使用。
`CustomerNotEntitledException`	客戶沒有有效的產品訂閱。
`InvalidProductCodeException`	作為請求的一部分傳入`ProductCode` 的值不存在。
`InvalidPublicKeyException`	在請求中傳入`PublicKeyVersion` 的值不存在。
`PlatformNotSupportedException`	AWS Marketplace 不支援從基礎平台計量用量。僅 AWS Fargate 支援 Amazon ECS、Amazon EKS 和 。
`ThrottlingException`	對 的呼叫`RegisterUsage` 會受到調節。
`InvalidRegionException`	`RegisterUsage` 必須在啟動 AWS 區域 Amazon ECS 任務或 Amazon EKS Pod 的相同 中呼叫 。這可防止容器在呼叫 時選擇區域 （例如 `withRegion(“us-east-1”)`)`RegisterUsage` 。

使用 整合您的容器產品與 AWS Marketplace Metering Service 適用於 Java 的 AWS SDK

您可以使用 適用於 Java 的 AWS SDK 與 AWS Marketplace Metering Service 整合。軟體使用的持續計量由 自動處理 AWS Marketplace Metering Control Plane。除了呼叫`RegisterUsage`一次 開始計量軟體使用之外，您的軟體不需要執行任何計量特定動作。本主題提供使用 適用於 Java 的 AWS SDK 與 AWS Marketplace Metering Service `RegisterUsage`動作整合的範例實作。

`RegisterUsage` 必須在啟動容器時立即呼叫 。如果您在容器啟動的前 6 小時內未註冊容器，AWS Marketplace Metering Service 不會提供前幾個月的任何計量保證。不過，計量將在當月繼續，直到容器結束為止。

如需完整的原始程式碼，請參閱 RegisterUsage Java 範例。無論 AWS SDK 語言為何，這些步驟都適用。

AWS Marketplace Metering Service 整合的範例步驟

1. 登入 AWS Marketplace 管理入口網站。
2. 從資產選擇容器以開始建立新的容器產品。建立產品會產生產品的產品代碼，以與您的容器映像整合。如需設定 IAM 許可的詳細資訊，請參閱 the section called “AWS Marketplace 計量和權利 API 許可”。
3. 下載公有 AWS Java 開發套件。

   > **Important**
   >
   > 若要從 Amazon EKS 呼叫計量 APIs，您必須使用支援的 AWS SDK，並在執行 Kubernetes 1.13 或更新版本的 Amazon EKS 叢集上執行。

4. （選用）如果您要與 `RegisterUsage`動作整合，而且想要執行數位簽章驗證，則需要在應用程式 classpath 中設定 BouncyCastle 簽章驗證程式庫。

   如果您想要使用 JSON Web Token (JWT)，您還必須在應用程式 classpath 中包含 JWT Java 程式庫。使用 JWT 提供更簡單的簽章驗證方法，但並非必要，您可以改為使用獨立的 BouncyCastle。無論您使用 JWT 或 BouncyCastle，都需要使用 Maven 等建置系統，在應用程式類別路徑中包含 BouncyCastle 或 JWT 的傳輸相依性。

   ```
   // 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. RegisterUsage 從您產品方案中的每個付費容器映像呼叫 。 ProductCode和 PublicKeyVersion是必要的參數，而所有其他輸入都是選用的。以下是 的範例承載RegisterUsage。

```
{
    "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**
>
> 連接到 AWS Marketplace Metering Service 時，可能會出現暫時性問題。 AWS Marketplace 強烈建議在指數退避的情況下，實作重試最多 30 分鐘，以避免短期中斷或網路問題。

6. RegisterUsage 使用 SHA-256 產生 RSA-PSS 數位簽章，可用於驗證請求真實性。簽章包含下列欄位：ProductCode、 PublicKeyVersion和 Nonce。若要驗證數位簽章，您必須從請求中保留這些欄位。下列程式碼是對RegisterUsage呼叫的範例回應。

```
{
"Signature": "<<JWT Token>>"
}

// Where the JWT Token is composed of 3 dot-separated,
// base-64 URL Encoded sections.
```

```
// e.g. eyJhbGcVCJ9.eyJzdWIMzkwMjJ9.rrO9Qw0SXRWTe

// 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
"rrO9Q4FEi3gweH3X4lrt2okf5zwIatUUwERlw016wTy_21Nv8S..."
```

7. 重建新版本的容器映像，其中包含RegisterUsage呼叫、標記容器，並將其推送至與 Amazon ECS 或 Amazon EKS 相容的任何容器登錄檔，例如 Amazon ECR 或 Amazon ECR Public。如果您使用的是 Amazon ECR，請確定啟動 Amazon ECS 任務或 Amazon EKS Pod 的帳戶具有 Amazon ECR 儲存庫的許可。否則，啟動會失敗。
8. 建立 IAM 角色，授予容器呼叫 的許可RegisterUsage，如下列程式碼所定義。您必須在 Amazon ECS 任務或 Amazon EKS Pod 定義的任務角色參數中提供此 IAM 角色。

   JSON

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

9. 建立 Amazon ECS 任務或 Amazon EKS Pod 定義，參考與 整合的容器， AWS Marketplace 並參考您在步驟 7 中建立的 IAM 角色。如果您想要查看 AWS CloudTrail 記錄，您應該在任務定義中啟用記錄。
10. 建立 Amazon ECS 或 Amazon EKS 叢集來執行您的任務或 Pod。如需建立 Amazon ECS 叢集的詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的建立叢集。如需建立 Amazon EKS 叢集 （使用 Kubernetes 1.1.3.x 版或更新版本） 的詳細資訊，請參閱建立 Amazon EKS 叢集。
11. 在 us-east-1 中設定 Amazon ECS 或 Amazon EKS 叢集，並啟動您建立的 Amazon ECS 任務定義或 Amazon EKS Pod AWS 區域。只有在此測試程序期間，在產品上線之前，您必須使用此區域。
12. 當您從 取得有效的回應時`RegisterUsage`，您可以開始建立您的容器產品。如有疑問，請聯絡AWS Marketplace 賣方營運團隊。

RegisterUsage Java 範例

下列範例使用 適用於 Java 的 AWS SDK 和 AWS Marketplace Metering Service 來呼叫`RegisterUsage`操作。簽章驗證是選用的，但如果您想要執行簽章驗證，則必須包含必要的數位簽章驗證程式庫。此範例僅供說明之用。

```
import com.amazonaws.auth.PEM;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMetering;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMeteringClientBuilder;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageRequest;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageResult;
import com.amazonaws.util.json.Jackson;
import com.fasterxml.jackson.databind.JsonNode;
import com.nimbusds.jose.JWSObject;
import com.nimbusds.jose.JWSVerifier;
import com.nimbusds.jose.crypto.RSASSAVerifier;
import java.io.ByteArrayInputStream;
import java.nio.charset.StandardCharsets;
import java.security.PublicKey;
import java.security.Security;
import java.security.Signature;
import java.security.interfaces.RSAPublicKey;
import java.util.Base64;
import java.util.Optional;
import java.util.UUID;
import org.bouncycastle.jce.provider.BouncyCastleProvider;
```

```
/**
 * Class for making calls out to &MKT; Metering Service.
 */
class RegisterUsage {

    private static final String PRODUCT_CODE = ".......";

    private final AWSMarketplaceMetering registerUsageClient;
    private final SignatureVerifier signatureVerifier;
    private final int publicKeyVersion;

    public RegisterUsage(final SignatureVerifier signatureVerifier) {
        this.signatureVerifier = signatureVerifier;
        this.publicKeyVersion = PublicKeyProvider.PUBLIC_KEY_VERSION;
        this.registerUsageClient =
 AWSMarketplaceMeteringClientBuilder.standard().build();
    }

    /**
     * Shows how to call RegisterUsage client and verify digital signature.
     */
    public void callRegisterUsage() {
        RegisterUsageRequest request = new RegisterUsageRequest()
                .withProductCode(PRODUCT_CODE)
                .withPublicKeyVersion(publicKeyVersion)
                .withNonce(UUID.randomUUID().toString());

        // Execute call to RegisterUsage (only need to call once at container startup)
        RegisterUsageResult result = this.registerUsageClient.registerUsage(request);

        // Verify Digital Signature w/o JWT
        boolean isSignatureValid = this.signatureVerifier.verify(request, result);
        if (!isSignatureValid) {
            throw new RuntimeException("Revoke entitlement, digital signature
 invalid.");
        }
    }
}

/**
 * Signature verification class with both a JWT-library based verification
 * and a non-library based implementation.
 */
class SignatureVerifier {
```

```
    private static BouncyCastleProvider BC = new BouncyCastleProvider();

    private static final String SIGNATURE_ALGORITHM = "SHA256withRSA/PSS";

    private final PublicKey publicKey;

    public SignatureVerifier(PublicKeyProvider publicKeyProvider) {
        this.publicKey = publicKeyProvider.getPublicKey().orElse(null);
        Security.addProvider(BC);
    }

    /**
     * Example signature verification using the NimbusJOSEJWT library to verify the JWT
Token.
     *
     * @param request RegisterUsage Request.
     * @param result  RegisterUsage Result.
     * @return true if the token matches.
     */
    public boolean verifyUsingNimbusJOSEJWT(final RegisterUsageRequest request, final
RegisterUsageResult result) {
        if (!getPublicKey().isPresent()) {
            return false;
        }

        try {
            JWSVerifier verifier = new RSASSAVerifier((RSAPublicKey)
getPublicKey().get());
            JWSObject jwsObject = JWSObject.parse(result.getSignature());
            return jwsObject.verify(verifier) &&
validatePayload(jwsObject.getPayload().toString(), request, result);
        } catch (Exception e) {
            // log error
            return false;
        }
    }

    /**
     * Example signature verification without any JWT library support.
     *
     * @param request RegisterUsage Request.
     * @param result  RegisterUsage Result.
     * @return true if the token matches.
     */
```

```
    public boolean verify(final RegisterUsageRequest request, final RegisterUsageResult
result) {
        if (!getPublicKey().isPresent()) {
            return false;
        }
        try {
            String[] jwtParts = result.getSignature().split("\\.");
            String header = jwtParts[0];
            String payload = jwtParts[1];
            String payloadSignature = jwtParts[2];

            Signature signature = Signature.getInstance(SIGNATURE_ALGORITHM, BC);
            signature.initVerify(getPublicKey().get());
            signature.update(String.format("%s.%s", header,
payload).getBytes(StandardCharsets.UTF_8));
            boolean verified = signature.verify(Base64.getUrlDecoder()
                    .decode(payloadSignature.getBytes(StandardCharsets.UTF_8)));

            String decodedPayload = new String(Base64.getUrlDecoder().decode(payload));
            return verified && validatePayload(decodedPayload, request, result);
        } catch (Exception e) {
            // log error
            return false;
        }
    }

    /**
     * Validate each value in the returned payload matches values originally
     * supplied in the request to RegisterUsage. TimeToLiveInMillis and
     * PublicKeyExpirationTimestamp will have the values in the payload compared
     * to values in the signature
     */
    private boolean validatePayload(final String payload, final RegisterUsageRequest
request,
                                    final RegisterUsageResult result) {
        try {
            JsonNode payloadJson = Jackson.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"
                    + "UWAY9iR3fy4arWNA1KoS8kVw33cJibXr8bvwUAUparCwlvdbH6dvEOfou0/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();
        }
    }
```

```
}
```

使用 AWS Marketplace Metering Service 設定容器產品的自訂計量

> **Note**
>
> 對於 Amazon EKS 部署，您的軟體必須使用服務帳戶 (IRSA) 的 IAM 角色來簽署 API 操作的 `MeterUsage` API 呼叫。不支援使用 EKS Pod Identity、節點角色或長期存取金鑰。
> 對於 Amazon ECS 部署，您的軟體必須使用 Amazon ECS 任務 IAM 角色來簽署 API 操作的 `MeterUsage` API 呼叫。不支援使用節點角色或長期存取金鑰。

AWS Marketplace 容器產品可以對每個產品最多 24 個不同的定價維度進行自訂計量。每個維度都可以有與其相關聯的長期合約價格。若要啟用自訂計量，請將您的容器產品與 AWS Marketplace Metering Service 整合。您可以使用 `MeterUsage` API 操作，為該用量定義自己的定價單位和自訂計量到 AWS 進行計費。下列各節說明如何設定容器產品的自訂計量。

價格維度在兩個位置定義，一次是在 AWS Marketplace 管理入口網站（賣方入口網站）中建立產品時，另一次在軟體中定義以執行`MeterUsage`操作。此雙重因素方法可確保後續優惠在公開前如預期般運作。

若要設定自訂計量，您需要選擇用量類別、單位類型和定價維度：

- 用量類別 – 用量類別可協助買方了解您的產品是什麼，以及如何使用它。
- 單位類型 – 單位類型定義計費的度量單位。例如，以 GBps 或 MBps 測量的頻寬、主機數量或以 MB、GB 或 TB 測量的資料。
- 定價維度 – 定價維度代表您為 設定每單位價格的功能或服務（例如，使用者、掃描、vCPUs 或部署的代理程式）。定價維度為公有。不過，您仍然可以為公開產品定義私有和自攜授權 (BYOL) 優惠。請勿在計量記錄中傳送定價。您可以測量單位數量，並使用它以及您建立產品以計算買方帳單時定義的價格。

  如果您的產品定價不符合任何預先定義的類別或單位類型，您可以選擇一般單位類別。然後，使用維度描述來描述什麼是單位。

或者，您可以根據您追蹤的屬性，將用量分配到配置。配置會以標籤表示給買方。這些標籤可讓買方檢視依標籤值分割為用量的成本。例如，如果您由使用者收費，且使用者具有「部門」屬性，則可以使用索引鍵為「部門」的標籤建立用量配置，每個值一個配置。這不會變更您報告的價格、維度或總用量，但可讓您的客戶依適合您產品的類別檢視其成本。

建議您每小時傳送計量記錄。不過，您也可以彙總每日或每月期間的用量。如果您遇到中斷，您可以彙總買方軟體的使用，並在下列時間計量中傳送。您無法每小時傳送超過一筆記錄。

如需將容器產品的 AWS Marketplace Metering Service API 與自訂計量定價整合的詳細資訊，請參閱與賣方研討會的自訂計量實驗室整合。 AWS Marketplace

> **Important**
>
> 免費試用和預付權利會每小時追蹤一次。因此，在 中分別傳送這些記錄可能會導致買方過度收費。

主題

- 自訂計量先決條件
- 測試MeterUsage整合
- 的錯誤處理 MeterUsage
- 廠商計量標記 （選用）
- 程式碼範例
- 使用自訂計量與 AWS Marketplace Metering Service 和 整合您的容器產品 適用於 Java 的 AWS SDK

自訂計量先決條件

在發佈產品之前，您必須執行下列動作：

1. 在 中建立新的容器產品 AWS Marketplace 管理入口網站，並記下其產品代碼。
2. 針對執行應用程式的 任務或 Pod，使用 AWS Identity and Access Management (IAM) 角色，並具有呼叫 所需的 IAM 許可`MeterUsage`。IAM 受管政策`AWSMarketplaceMeteringRegisterUsage`具有這些許可。如需政策的詳細資訊，請參閱《AWS 受管政策參考》中的 AWSMarketplaceMeteringFullAccess。
3. （選用）如果您想要查看 AWS CloudTrail 記錄，建議您在任務或 Pod 定義中啟用記錄。
4. 使用您定義的所有定價維度的記錄對 `MeterUsage` API 操作進行測試呼叫。

測試`MeterUsage`整合

使用 `MeterUsage`操作來測試整合，然後再將映像提交至 AWS Marketplace 進行發佈。

在 Amazon Elastic Container Service (Amazon ECS) 或 Amazon Elastic Kubernetes Service (Amazon EKS) 上執行您的產品，並使用 AWS 帳戶 列出產品，以MeterUsage從容器映像呼叫 AWS Marketplace。您的計量整合必須動態設定 AWS 區域，而不是硬式編碼。不過，在測試時，請至少啟動一個 Amazon ECS 任務或 Amazon EKS Pod，其中包含您在美國東部（維吉尼亞北部）區域的付費容器，以便 AWS Marketplace 操作團隊可以驗證您在該區域中使用日誌的工作。

> **Note**
>
> - 如果您的產品同時支援 Amazon ECS 和 Amazon EKS，您只需要在 Amazon EKS 中啟動 ，我們就能驗證您的整合。
> - 在將產品啟動到公有環境之前，以及在新增維度之後，測試每個維度。如果您未針對與容器產品相關聯的每個維度傳送計量記錄，則會導致請求失敗的錯誤。

在產品發佈所有必要的中繼資料和定價資訊之前，您無法完整測試整合。如果請求， AWS Marketplace 目錄操作團隊可以驗證是否收到您的計量記錄。

的錯誤處理 `MeterUsage`

如果您的容器映像與 MeterUsage操作整合，並收到容器啟動ThrottlingException時 以外的例外狀況，您應該終止容器，以防止未經授權的使用。

以外的例外僅在對 的初始呼叫中ThrottlingException擲回MeterUsage。CustomerNotSubscribedException 即使客戶在任務或 Pod 仍在執行時取消訂閱，來自相同 Amazon ECS 任務或 Amazon EKS Pod 的後續呼叫也不會擲回。這些客戶在取消訂閱並追蹤其用量後，仍需支付執行中容器的費用。

如需 常見錯誤的詳細說明，請參閱 AWS Marketplace Metering Service API 參考中的 MeterUsageMeterUsage。每種 AWS SDK 程式設計語言都有一組錯誤處理準則，如需其他資訊，請參閱這些準則。

廠商計量標記（選用）

廠商計量標記可協助獨立軟體廠商 (ISVs) 為買方提供更精細的軟體用量洞察，並協助他們執行成本分配。

標記買方軟體用量的方法有很多。其中一種方法是先詢問您的買方，他們想要在成本分配中看到什麼。然後，您可以將用量分割到您追蹤買方帳戶的屬性。屬性的範例包括 AccountId、Cost Centers、

Business Unit和您產品的其他相關中繼資料。這些屬性會以標籤的形式公開給買方。使用標籤，買方可以在帳單 AWS 主控台 (https://console.aws.amazon.com/costmanagement/：//) 中檢視依標籤值分割為用量的成本。廠商計量標記不會變更您報告的價格、維度或總用量。它可讓您的客戶依適合您產品的類別檢視其成本。

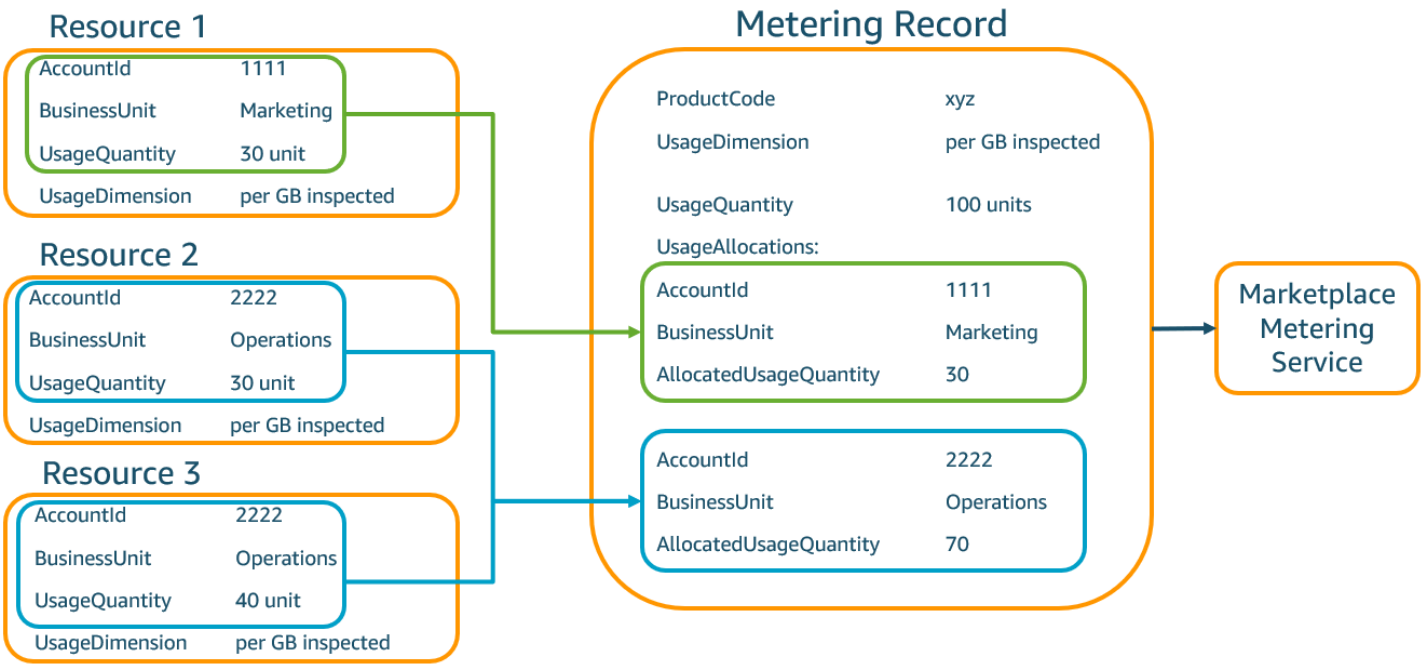
在常見的使用案例中，買方使用 訂閱您的產品 AWS 帳戶。買方也有許多與相同產品訂閱相關聯的使用者。您可以使用索引鍵為 的標籤建立用量配置AccountId，然後將用量配置給每個使用者。在此情況下，買方可以在帳單和成本管理主控台中啟用AccountId標籤，並分析個別使用者用量。

賣方體驗

賣方可以使用相同的標籤集彙總資源的計量記錄，而不是彙總所有資源的用量。例如，賣方可以建構包含不同 儲存貯體的計量記錄UsageAllocations。每個儲存貯UsageQuantity體代表一組標籤，例如 AccountId和 BusinessUnit。

在下圖中，資源 1 具有一組唯一的 AccountId 和 BusinessUnit標籤，並在計量記錄中顯示為單一項目。

資源 2 和資源 3 都有相同的AccountId標籤 2222和相同的BusinessUnit標籤 Operations。因此，它們會合併為計量記錄中的單一UsageAllocations項目。



賣方也可以將沒有標籤的資源合併為UsageAllocation具有配置用量數量的單一資源，並將其做為中的其中一個項目傳送UsageAllocations。

限制包括：

- 標籤數量 – 5
- 大小 UsageAllocations（基數）– 2，500

驗證包括：

- 標籤索引鍵和值允許的字元 – a-zA-Z0-9+ -=._：\/@
- 跨UsageAllocation清單的標籤上限 – 5
- 兩個UsageAllocations不能有相同的標籤（也就是標籤索引鍵和值的相同組合）。如果是這種情況，他們必須使用相同的 UsageAllocation。
- AllocatedUsageQuantity 的 總和UsageAllocation必須等於 UsageQuantity，即彙總用量。

買方體驗

下表顯示買方啟用 AccountId和BusinessUnit廠商標籤後買方體驗的範例。

在此範例中，買方可以在成本用量報告中看到配置的用量。廠商計量的標籤使用字首“aws:marketplace:isv”。買方可以在帳單和成本管理、成本分配標籤、 AWS產生的成本分配標籤下啟用它們。

成本用量報告的第一列和最後一列與賣方傳送到計量服務的內容相關（如賣方體驗範例所示）。

成本用量報告（簡化）

ProductCode	買方	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	網路：每個 (GB) 已檢查	70	2222	作業
xyz	111122223333	網路：每個 (GB) 已檢查	30	3333	財務
xyz	111122223333	網路：每個 (GB) 已檢查	20	4444	IT

ProductCode	買方	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	網路：每個 (GB) 已檢查	20	5555	行銷部門
xyz	111122223333	網路：每個 (GB) 已檢查	30	1111	行銷部門

如需程式碼範例，請參閱「MeterUsage 具有用量配置標記的程式碼範例 （選用）」。

程式碼範例

以下程式碼範例可協助您將容器產品與發佈和維護產品所需的 AWS Marketplace APIs 整合。

MeterUsage 具有用量配置標記的程式碼範例 （選用）

下列程式碼範例與具有耗用定價模型的容器產品相關。Python 範例會將含有適當用量分配標籤的計量記錄傳送至 AWS Marketplace ，以向您的客戶收取pay-as-you-go費用。

```
# 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
)
```

如需 的詳細資訊`MeterUsage`，請參閱 AWS Marketplace Metering Service API 參考中的 MeterUsage。

回應範例

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

使用自訂計量與 AWS Marketplace Metering Service 和 整合您的容器產品 適用於 Java 的 AWS SDK

AWS Marketplace 容器產品可以對每個產品最多 24 個不同的定價維度進行自訂計量。若要啟用自訂計量，您可以將容器產品與 AWS Marketplace Metering Service 整合。您可以使用 `MeterUsage` API 操作，為該用量定義自己的定價單位和自訂計量到 AWS 進行計費。下列範例概述使用 適用於 Java 的 AWS SDK 與 AWS Marketplace Metering Service `MeterUsage`操作整合的實作。

如需完整詳細資訊，請參閱 `MeterUsage Java 範例`。無論語言為何，都適用下列許多步驟。

範例： AWS Marketplace Metering Service 整合

1. 登入 AWS Marketplace 管理入口網站。
2. 從資產中，選擇容器以開始建立新的容器產品。建立產品會產生產品的產品代碼，以與您的容器映像整合。如需設定 AWS Identity and Access Management (IAM) 許可的詳細資訊，請參閱 the section called “AWS Marketplace 計量和權利 API 許可”。
3. 下載公有 AWS Java 開發套件。

> **Important**
>
> 若要從 Amazon Elastic Kubernetes Service (Amazon EKS) 呼叫計量 API 操作，您必須使用支援的 AWS SDK，並在執行 Kubernetes 1.13 或更新版本的 Amazon EKS 叢集上執行。

4. 每個維度用量每小時從任務或 Pod 呼叫一次 `MeterUsage`操作。API 操作接受 `Dimension`、`Resource`和 唯一組合的一個計量記錄`Hour`。資源可以是 Amazon Elastic Container Service (Amazon ECS) 任務或 Amazon EKS Pod。

```
{
    "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**
>
> 連接到 時，可能會出現暫時性問題 AWS Marketplace Metering Service。 AWS Marketplace 強烈建議在指數退避的情況下，實作重試長達 30 分鐘，以避免短期中斷或網路問題。

5. 重建容器映像的新版本，其中包含`MeterUsage`呼叫、標記容器，並將其推送至與 Amazon ECS 或 Amazon EKS 相容的任何 Docker 登錄檔，例如 Amazon Elastic Container Registry (Amazon ECR)。如果您使用的是 Amazon ECR，請確定啟動 Amazon ECS 任務或 Amazon EKS Pod 的帳戶具有 Amazon ECR 儲存庫的許可。否則，操作會失敗。

6. 建立 IAM 角色，授予容器呼叫 的許可MeterUsage，如下列程式碼範例所定義。您必須在 Amazon ECS 任務或 Amazon EKS Pod 定義的任務角色參數中提供此 AWS Identity and Access Management (IAM) 角色。

   JSON

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

7. 建立 Amazon ECS 任務或 Amazon EKS Pod 定義，參考與 整合的容器， AWS Marketplace 並參考您在步驟 6 中建立的 IAM 角色。如果您想要查看記錄，請在任務定義中啟用 AWS CloudTrail 記錄。
8. 建立 Amazon ECS 或 Amazon EKS 叢集來執行您的任務或 Pod。如需建立 Amazon ECS 叢集的詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的建立叢集。如需建立 Amazon EKS 叢集 （使用 Kubernetes 1.1.3.x 版或更新版本） 的詳細資訊，請參閱建立 Amazon EKS 叢集。
9. 設定 Amazon ECS 或 Amazon EKS 叢集，並在 us-east-1 AWS Region 中啟動您在步驟 8 中建立的 Amazon ECS 任務定義或 Amazon EKS Pod。只有在此測試程序期間，在產品上線之前，您必須使用此區域。
10. 當您從 取得MeterUsage針對產品發佈的每個維度的有效回應時，您可以開始建立容器產品。如有疑問，請聯絡AWS Marketplace 賣方營運團隊。

MeterUsage Java 範例

下列程式碼範例使用 適用於 Java 的 AWS SDK 和 AWS Marketplace Metering Service 來呼叫 MeterUsage操作。

下列程式碼範例在沒有 的情況下呼叫 MeterUsage操作UsageAllocations。

```
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);
    }
}
```

下列程式碼範例使用 呼叫 MeterUsage操作UsageAllocations。

```
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();
    }
```

使用 的容器產品的合約定價 AWS License Manager

對於具有合約定價的容器型產品，您可以使用 AWS License Manager 將授權與產品建立關聯。 AWS License Manager 是一種授權管理工具，可讓您的應用程式追蹤和更新客戶購買的授權 （ 也稱為權利 ）。本節提供如何將產品與 整合的相關資訊 AWS License Manager。整合完成後，您可以在其上發佈您的產品清單 AWS Marketplace。

如果您要將 License Manager 與適用於 Amazon EKS Anywhere、Amazon ECS Anywhere、Amazon Elastic Compute Cloud (Amazon EC2) 或內部部署基礎設施 AWS Marketplace 的 for Containers Anywhere 產品整合，請遵循 中的指示將 AWS Marketplace for Containers Anywhere 與 License Manager 整合。

如需詳細資訊 AWS License Manager，請參閱 AWS License Manager 使用者指南和 AWS CLI 命令參考的 AWS License Manager一節。

如需將 AWS License Manager 與容器產品與合約定價整合的詳細資訊，請參閱 與賣方研討會的前期付款實驗室整合。 AWS Marketplace

主題

- 授權模型

- AWS License Manager 整合先決條件
- 將容器產品與 License Manager 整合
- License Manager API 操作
- 授權續約和升級
- 將 AWS Marketplace for Containers Anywhere 與 License Manager 整合

授權模型

AWS Marketplace 與 整合 AWS License Manager 支援兩種授權模型：

- 可設定的授權模型
- 分層授權模型

可設定的授權模型

在買方取得授權之後，可設定的授權模型 （ 也稱為可量化授權模型 ） 可讓買方獲得特定數量的資源。

您可以設定定價維度和每個單價。然後，買方可以選擇他們想要購買的資源數量。

Example 定價維度和每個單價的

您可以設定定價維度 （ 例如資料備份 ） 和每單位價格 （ 例如 $30/單位 ）。

買方可以選擇購買 5、10 或 20 個單位。

您的產品會追蹤和測量用量，以測量消耗的資源數量。

使用組態模型時，會以下列兩種方式之一計算權利：

- Drawdown 授權
- 浮動授權

下調授權

授權是從使用時允許的授權數量集區中提取。該權利會永久簽出，且無法傳回至授權集區。

Example 處理有限數量的資料

使用者有權處理 500 GB 的資料。隨著他們繼續處理資料，數量會從 500 GB 的集區中提取，直到所有 500 GB 授權都用盡為止。

對於提取授權，您可以使用 CheckoutLicense API 操作來查看使用的授權單位 （權限）。

Example 的備份到 Amazon S3 的單位/年

您有一個儲存產品，允許備份到 Amazon Simple Storage Service 最多 1，024 個單位的資料一年。您可以使用多個 Amazon EC2 執行個體啟動您的應用程式。您的應用程式具有追蹤和彙總資料的機制。您的軟體會在每次備份時以產品 ID 呼叫 CheckoutLicense API 操作，或以固定間隔更新耗用的數量。

在此範例中，您的軟體會呼叫 CheckoutLicense API 操作來檢查 10 個單位的資料。當總容量達到客戶購買的備份限制時，API 呼叫會失敗。

請求

```
linux-machine ~]$ aws license-manager checkout-license\
--product-sku "2205b290-19e6-4c76-9eea-377d6bf7la47" \
--checkout-type "PERPETUAL" \
--key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements "Name=DataConsumption, Value=l0, Unit=Count" \
--client-token "AKIAIOSFODNN7EXAMPLE"
```

回應

```
{"CheckoutType": "PERPETUAL",
"EntitlementsAllowed": [{
"Name": "IntermediateTier",
"Units": "None"
}],
"Expiration": "2021-04-22Tl9:02:36",
"IssuedAt": "2021-04-22Tl8:02:36",
"LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
"LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

浮動授權

授權在使用後會傳回至允許授權數量的集區。

對於浮動授權，應用程式會在使用資源時，使用 CheckoutLicense API 操作從權利集區簽出權利。CheckoutLicense API 操作的回應包含授權使用字符，這是結帳的唯一識別符。授權使用字符可用來對簽出的權限執行其他動作，例如將它們簽回授權或延長簽出。

若要將權利檢查回集區，請在不再使用資源時使用 CheckInLicense API 操作。

```
aws license-manager check-in-license --license-consumption-token
 "f1603b3c1f574b7284db84..."
```

如果無法簽入權利（如果應用程式當機），權利會在 60 分鐘後自動簽入集區。如果資源使用時間超過 60 分鐘，最佳實務是使用 ExtendLicenseConsumption API 操作，在資源使用期間，保持從集區中檢查權利。

```
aws license-manager extend-license-consumption --license-consumption-token
 "f1603b3c1f574b7284..."
```

Example 來自固定上限的使用者數量

使用者有權在應用程式上同時存取 500 名使用者。當使用者登入和登出時，會繪製使用者並傳回 500 名使用者的集區。不過，應用程式無法從集區中提取超過 500 個使用者，因為 500 個同時使用者是固定的上限。

對於浮動權利，您可以使用 CheckInLicense API 操作將授權單位傳回至權利集區。

Example 一年的並行使用者數量

您的產品會根據並行使用者數量定價。客戶為 10 個使用者購買授權一年。客戶提供 AWS Identity and Access Management (IAM) 許可來啟動軟體。當使用者登入時，您的應用程式會呼叫 CheckoutLicense API 操作，將數量減少 1。當使用者登出時，應用程式會呼叫 CheckInLicense API 操作，將該授權傳回至集區。如果您不呼叫 CheckInLicense，授權單位會在 1 小時後自動簽入。

> **Note**
>
> 在下列請求中， key-fingerprint 不是預留位置值，而是將發佈所有授權的指紋實際值。

請求

```
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"
```

回應

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "ReadOnlyUsers",
      "Count": 10,
      "Units": "Count",
      "Value": "Enabled"
    }
},
  "Expiration": "2021-04-22Tl9:02: 36",
  "IssuedAt": "2021-04-22Tl8:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

分層授權模型

在買方取得授權之後，分層授權模型會授予買方特定層級或層級的應用程式功能。

您可以為您的產品建立方案，例如 Basic、Intermediate 和 Premium。然後，買方會選取其中一個預先定義的層。

應用程式不需要追蹤或計量應用程式的用量。

使用分層授權模型時，不會計算權利，而是表示客戶購買的服務層級。

如果您想要同時提供綁定功能，最好使用 層。

Example 基本、中級和高級方案

客戶可以為軟體的三個可能方案之一簽署合約：基本、中級或高級。每個方案都有自己的定價。您的軟體可以透過叫用 CheckoutLicense API 操作並在請求中指定所有可能的層來識別客戶已註冊的層。

請求的回應包含與客戶已取得的方案對應的權利。根據此資訊，軟體可以佈建適當的客戶體驗。

請求

```
linux-machine  ~]$ aws  license-manager   checkout-license\
--product-sku  "2205b290-19e6-4c76-9eea-377d6bf7la47"  \
```

```
--checkout-type  "PROVISIONAL"  \
--key-fingerprint  "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements  "Name=BasicTier,  Unit=None"   "Name=IntermediateTier,  Unit=None" \
 "Name=PremiumTier, Unit=None"
```

回應

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "IntermediateTier",
      "Units": "None"
    }
},
  "Expiration": "2021-04-22Tl9:02:36",
  "IssuedAt": "2021-04-22Tl8:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

AWS License Manager 整合先決條件

在發佈產品之前，您必須執行下列動作：

1. 在 中建立新的容器產品 AWS Marketplace 管理入口網站，並記下其產品代碼。

   如需詳細資訊，請參閱概觀：建立容器產品。

2. 針對執行應用程式的 任務或 Pod 使用 IAM 角色，並具有呼叫 `CheckoutLicense`、`ExtendLicenseConsumption`和 `CheckInLicense` API 操作所需的 IAM 許可。

   必要的 IAM 許可詳述於下列 IAM 政策。

   JSON

   ```
   {
      "Version":"2012-10-17",
      "Statement":[
         {
            "Sid":"VisualEditor0",
            "Effect":"Allow",
   ```

```
            "Action":[
               "license-manager:CheckoutLicense",
               "license-manager:GetLicense",
               "license-manager:CheckInLicense",
               "license-manager:ExtendLicenseConsumption",
               "license-manager:ListReceivedLicenses"
            ],
            "Resource":"*"
         }
      ]
}
```

3. 使用您定義的所有定價維度的記錄對 `RegisterUsage` API 操作進行測試呼叫。

將容器產品與 License Manager 整合

將您的容器型產品與 License Manager 整合

1. 設定 IAM 許可以呼叫 License Manager。如需詳細資訊，請參閱AWS License Manager 整合先決條件。
2. 下載 AWS SDK。

> **Note**
>
> 請勿在 software. AWS credentials 中設定 AWS 登入資料。當容器在 Amazon EC2 執行個體、Amazon ECS 任務或 Amazon EKS Pod 中執行時，系統會在執行時間自動取得買方登入資料。

3. 將授權檢查新增至您的產品。

   您的產品可以在應執行授權檢查的位置呼叫 `CheckoutLicense` API 操作。若要檢查授權，您的產品必須知道：

   1. 授權的信任發行者 (AWS Marketplace)
   2. 應用程式的產品 SKU （ 產品 ID)
   3. 檢查此應用程式的權利

   API 呼叫會根據您設定的定價授權類型而有所不同。
4. 發佈您的產品清單 AWS Marketplace。

License Manager API 操作

若要管理存放在客戶 License Manager 帳戶中的授權，您的軟體可以使用下列 API 操作：

- `GetLicense` – 軟體可查詢的 API。它會擷取已購買授權的狀態（即已過期或即將過期），並傳送狀態通知給客戶。
- `CheckoutLicense` – 探索使用者已購買的授權。當使用者已取用一些授權數量時，您也可以使用 `CheckoutLicense` API 操作來更新授權數量。使用 `CheckoutLicense`，您可以繼續檢查客戶使用的授權數量。當客戶耗盡所有授權時，此呼叫會傳回錯誤。如需執行 的建議節奏資訊`CheckoutLicense`，請參閱 the section called “授權續約和升級”。
- `ExtendLicenseConsumption` – 在浮動維度的情況下，當軟體簽出授權時，授權會在 60 分鐘後自動返回集區。如果您想要延長授權保持簽出的時間，請使用 `ExtendLicenseConsumption` API 操作將授權再延長 60 分鐘。
- `CheckInLicense` – 如果是浮動維度，當您想要將授權傳回權利集區時，請使用 `CheckInLicense` API 操作。
- `ListReceivedLicenses` API – 列出買方購買的授權。

授權續約和升級

客戶可以在 上續約或升級其授權 AWS Marketplace 管理入口網站。進行額外的購買後， AWS Marketplace 會產生反映新權利的授權新版本。您的軟體會使用相同的 API 操作讀取新的權利。在 License Manager 整合方面，您不需要採取任何不同的動作來處理續約和升級。

由於授權續約、升級、取消等，我們建議您的產品在使用產品時定期呼叫 `CheckoutLicense` API 操作。透過定期使用 `CheckoutLicense` API 操作，產品可以偵測權利的變更，例如升級和過期。

我們建議您每 15 分鐘執行一次 `CheckoutLicense` API 呼叫。

將 AWS Marketplace for Containers Anywhere 與 License Manager 整合

身為 AWS Marketplace 賣方，您可以 AWS License Manager 整合適用於 Amazon EKS Anywhere、Amazon ECS Anywhere、Amazon EC2 或內部部署基礎設施 AWS Marketplace 的 for Containers Anywhere 產品。以下各節提供此整合的說明。

如需 License Manager 與 整合的一般資訊 AWS Marketplace，包括可用的授權模型，請參閱 使用 的容器產品的合約定價 AWS License Manager。如需 的詳細資訊 AWS License Manager，請參閱 AWS License Manager 使用者指南和 AWS CLI 命令參考的 AWS License Manager一節。

主題

- 將 AWS Marketplace for Containers Anywhere 產品與 License Manager 整合
- 在本機測試 License Manager 整合
- 在 Amazon EKS 上測試 License Manager 整合
- 使用 License Manager 浮動授權權利
- 與現場部署的 License Manager 整合的最佳實務
- LicenseManagerCredentialsProvider - Java 實作
- LicenseManagerCredentialsProvider - Golang實作

將 AWS Marketplace for Containers Anywhere 產品與 License Manager 整合

使用以下指示將您的 AWS Marketplace for Containers Anywhere 產品與 整合 AWS License Manager。

將您的 AWS Marketplace for Containers Anywhere 產品與 License Manager 整合

1. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
2. 執行下列步驟，為您的容器產品建立產品 ID。在後續步驟中，您將在容器映像中使用此 ID 進行授權檢查。

   a. 在選單列中，展開資產，然後選擇容器。

   b. 輸入產品面向客戶的名稱，然後選擇建立。您可以稍後變更此名稱。

   c. 記下產品 ID。您將在建立或更新產品定價詳細資訊時使用它。

   > **Tip**
   >
   > 如果您遺失產品 ID，您可以從資產功能表中 AWS Marketplace 管理入口網站 選擇容器，在 中找到它。Containers 頁面會顯示您的產品清單及其相關聯的產品 IDs。

3. 下載最新的公有 AWS SDK，然後將其安裝在您的容器應用程式中。您可以在在 AWS 上建置的工具中找到您偏好 AWS SDK 的安裝指示。

   > **Note**
   >
   > 若要從 Amazon EKS Anywhere 或 未提供的 Kubernetes 叢集呼叫 License Manager API 操作 AWS，您必須使用支援的 AWS SDK。若要檢視支援的 AWS SDKs 清單，請參閱使用支援的 AWS SDK。

4. 使用自訂登入資料提供者建立 AWS License Manager 用戶端，以便為部署在 上的容器應用程式 AWS 以及內部部署提供登入資料。如需自訂登入資料提供者的完整原始碼LicenseCredentialProvider，請參閱下列各節：

   - LicenseManagerCredentialsProvider - Java 實作
   - LicenseManagerCredentialsProvider - Golang實作

   LicenseCredentialsProvider 透過新增 來擴展 AWS SDK 的預設登入資料提供者鏈，以供內部部署使用LicenseManagerTokenCredentialsProvider。這透過在內部部署環境中使用 License Manager OIDC 發行的身分字符來提供憑證。您必須在應用程式 classpath LicenseCredentialsProvider中包含 的原始碼。

   > **Note**
   >
   > 延伸 DefaultCredentialsProvider可讓相同的容器應用程式在內部部署環境中執行 AWS 和 時取得登入資料。如果容器應用程式已使用自訂登入資料提供者鏈而非預設值，也可以透過將 LicenseManagerTokenCredentialsProvider新增至自訂鏈來延伸。

   下列程式碼片段是使用 Java 建立 AWS License Manager 用戶端的範例。

   ```
   LicenseManagerClientBuilder clientBuilder =
    LicenseManagerClient.builder().credentialsProvider(LicenseCredentialsProvider.create());
   ```

5. 使用產品方案中每個付費容器映像的 aws license-manager checkout-license命令來呼叫 CheckoutLicense API 操作。這會檢查買方是否有權為您的應用程式使用授權。如果買方有權使用應用程式， 會CheckoutLicense成功並傳回請求的權限及其值。如果買方無權使用應用程式， 會CheckoutLicense擲回例外狀況。

   呼叫 CheckoutLicense API 操作時需要下列參數：

   - CheckoutType – 有效值為 PROVISIONAL或 PERPETUAL：
     - 當已從集區中耗盡已簽出的權利數量PERPETUAL時，請使用 。

       範例：買方有權處理 500 GB 的資料。隨著他們繼續處理資料，數量會從 500 GB 的集區中消耗。
     - PROVISIONAL 用於浮動授權權利，其中權利會從集區中簽出，並在使用後傳回。

範例：使用者有權在應用程式上同時存取 500 名使用者。當使用者登入或登出時，系統會將使用者繪製或傳回至 500 名使用者的集區。若要進一步了解浮動授權權利，請參閱 使用 License Manager 浮動授權權利。

- `ClientToken` – 唯一且區分大小寫的識別符。我們建議針對每個唯一請求使用隨機 UUID。
- `Entitlements` – 要簽出的權限清單。
  - 針對功能權利，提供 `Name`和 `Unit` 屬性，如下所示。

```
{
  "Name": "<Entitlement_Name>",
  "Unit": "None"
}
```

  - 針對計數權利，提供 `Name`、 `Unit`和 `Count` 屬性，如下所示。

```
{
  "Name": "<Entitlement_Name>",
  "Unit": "<Entitlement_Unit>",
  "Value": <Desired_Count>
}
```

- `KeyFingerprint` – 所發行授權的金鑰指紋 AWS Marketplace 為 `aws:294406891311:AWS/Marketplace:issuer-fingerprint`。使用此金鑰指紋可確保授權是由 發出 AWS Marketplace ，而不是由不可靠的實體發出。
- `ProductSKU` – 上一個步驟 AWS Marketplace 管理入口網站 中在 上產生的產品 ID。

下列程式碼片段是使用 的 `CheckoutLicense` API 操作呼叫的範例 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**
>
> 若要檢查授權，容器應用程式需要傳出網路存取權才能使用 License Manager。部署在內部部署的應用程式可能會遇到不可靠或緩慢的傳出網路存取。呼叫 License Manager 時，這些應用程式應包含足夠的重試次數。如需詳細資訊，請參閱與現場部署的 License Manager 整合的最佳實務。

6. 定期呼叫 `CheckoutLicense` API 操作，以識別因續約、升級或取消 而導致客戶授權的任何變更 AWS Marketplace。節奏取決於應用程式。我們建議您每天檢查一次授權，以自動收取變更，無需任何買方介入。

   部署於內部部署的應用程式可能具有不可靠的傳出網路存取權，以定期檢查授權。在這種情況下，應用程式應使用快取的授權以獲得足夠的彈性。如需詳細資訊，請參閱與現場部署的 License Manager 整合的最佳實務。

7. 將`CheckoutLicense`呼叫與容器應用程式整合後，請建置具有變更的 Docker 容器映像新版本。

8. 更新應用程式的 Helm Chart 以接受 Kubernetes 秘密作為選用輸入，其中包含使用 License Manager APIs組態。組態秘密將包含 License Manager 發行的身分字符和 角色，該 AWS Identity and Access Management 角色將由先前描述的自訂登入資料提供者使用，以取得在現場部署容器應用程式時呼叫 License Manager APIs AWS 登入資料。此外，將 新增 AWS 區域 為預設值為 的輸入`us-east-1`。

   現場部署容器應用程式的買方可以透過容器產品的 AWS Marketplace 買方體驗來建立 Kubernetes 秘密。提供 Kubernetes 秘密名稱做為`helm install`命令的輸入。組態秘密的設定格式如下。

   ```
   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. 針對與 整合的容器映像，更新 Helm Chart 中的應用程式部署範本 AWS License Manager ，以包含下列項目：

- Pod 的服務帳戶 – 在 Amazon EKS 上的 Helm 部署需要 服務帳戶。它用於透過在容器映像上設定服務帳戶的 IAM 角色來取得呼叫 License Manager API 操作的許可。如需服務帳戶的 IAM 角色詳細資訊，請參閱服務帳戶的 IAM 角色。
- 內部部署的授權存取 – 需要授權組態秘密，才能提供登入資料和適當的許可，以在內部部署環境中呼叫 Helm 部署的 License Manager API 操作。買方會從 AWS Marketplace 買方體驗產生授權秘密並提供給 Helm。

下列程式碼片段是具有服務帳戶、授權組態和映像提取秘密的範例部署規格。

```
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.awsmp.licenseConfigSecretName }}
          //Mount the license volume to the container image
          volumeMounts:
            - name: awsmp-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: awsmp-product-license
          secret:
            secretName: {{ .Values.aws.licenseConfigSecretName }}
            optional: true
{{ - end }}
```

> **Note**
>
> 授權組態秘密是選用的。買方只會將 值用於內部部署。對於 AWS 部署，部署規格必須包含 License Manager 整合映像的服務帳戶。

10. 透過執行以下章節中的步驟，在本機和 Amazon EKS 上測試 License Manager 整合：
    a. 在本機測試 License Manager 整合
    b. 在 Amazon EKS 上測試 License Manager 整合
11. 在現場部署 AWS 和現場部署成功驗證 License Manager 整合之後，您可以依照中的步驟建立容器產品清單概觀：建立容器產品。

在本機測試 License Manager 整合

您可以使用 minikube 或任何其他設定來測試本機任何 Kubernetes 叢集上的 License Manager 整合。請確定 Kubernetes 叢集具有傳出網際網路存取權，以呼叫 License Manager API 操作。

在本機測試 License Manager 整合

1. 在具有所需權利的測試賣方帳戶中建立測試授權。若要設定測試授權，請參閱 AWS License Manager API 參考中的 CreateLicense。或者，使用以下指令碼建立測試授權，然後建立授權給測試買方帳戶以取用授權。下列指令碼使用測試賣方帳戶登入資料。

```
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. 使用先前定義的秘密格式，使用授權字符和 IAM 角色建立 Kubernetes 秘密。使用 License Manager `CreateToken` API 操作來產生授權字符。然後，使用 IAM `CreateRole` API 操作建立具有許可和信任政策的 IAM 角色。請參閱下列指令碼中的範例。下列指令碼使用測試買方帳戶登入資料。

```
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"c
```

3. 設定在外部託管的任何 Kubernetes 叢集 AWS。使用它來測試容器應用程式是否可以從 以外的環境連線至 AWS License Manager API， AWS 並且自訂登入資料提供者已與應用程式充分整合。

4. 將先前產生的授權字符和 IAM 角色部署到本機 Kubernetes 叢集。

```
kubectl create secret generic "awsmp-license-access-config" \
--from-literal=license_token=${TOKEN} \
```

```
--from-literal=iam_role=${ROLE_ARN}
```

5. 透過 Helm 使用秘密名稱做為輸入來部署應用程式，並確認應用程式可以呼叫 License Manager API 操作來執行權利檢查。如需 Helm 和部署規格變更，請參閱 中的步驟 9[將 AWS Marketplace for Containers Anywhere 產品與 License Manager 整合](#)。

在 Amazon EKS 上測試 License Manager 整合

您也可以在 Amazon EKS 上測試 License Manager 整合。測試以確保應用程式可以在沒有授權組態秘密的情況下呼叫 License Manager API 操作。此外，請確定服務帳戶可用來設定服務帳戶的 IAM 角色 (IRSA)，並提供應用程式相關登入資料。

在 Amazon EKS 上測試 License Manager 整合

1. 在具有所需權利的測試賣方帳戶中建立測試授權。請參閱 CreateLicense API 參考以設定您的測試授權，或使用下列指令碼建立一個授權，並建立授權給測試買方帳戶以取用授權。下列指令碼使用測試賣方帳戶登入資料。

```
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. 建立所需組態的測試 Amazon EKS 叢集，或執行下列命令以使用預設組態。

```
aws ec2 create-key-pair --region us-west-2 --key-name eks-key-pair
```

```
eksctl create cluster \
--name awsmp-eks-test-example \
--region us-west-2 \
--with-oidc \
--ssh-access \
--ssh-public-key eks-key-pair
```

3. 為現有叢集建立服務帳戶，並將其與 IAM 角色建立關聯。下列命令會使用 建立 IAM 角色`AWSLicenseManagerConsumptionPolicy`。然後，命令會將它連接到 Amazon EKS 叢集`test_sa`的服務帳戶，其中應部署 License Manager 整合映像。因此，服務帳戶可以取得適當的登入資料來呼叫 License Manager API 操作。

```
eksctl create iamserviceaccount \
--name test_sa \
--namespace test_namespace \
```

```
--cluster awsmp-eks-test-example \
--attach-policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLicenseManagerConsumptionPolicy" \
--approve \
--override-existing-serviceaccounts
```

4. 在 IAM 角色與上一個命令相關聯的服務帳戶中，透過 Helm 部署應用程式。確認應用程式可以呼叫 License Manager API 操作來執行權利檢查。

使用 License Manager 浮動授權權利

使用浮動授權時，當使用者登入應用程式時，會從可用授權集區中提取授權。當使用者登出時，授權會加回可用授權的集區。

對於浮動授權，應用程式會使用 CheckoutLicense API 操作，在使用資源時從權利集區簽出權利。CheckoutLicense API 操作的回應包含授權使用字符，這是結帳的唯一識別符。授權使用字符可以對已簽出的權限執行其他動作，例如將它們簽回授權集區或延長簽出。

當資源不再使用時，應用程式會使用 CheckInLicense API 操作將權利檢查回集區。

```
aws license-manager check-in-license \
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

如果將授權檢查回集區失敗，例如，如果應用程式在操作期間當機，則會在 60 分鐘後自動將權利檢查回集區。因此，如果資源的使用時間超過 60 分鐘，最佳實務是將權利保留在集區之外。若要這樣做，只要資源正在使用，請使用 ExtendLicenseConsumption API 操作。

```
aws license-manager extend-license-consumption \
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

與現場部署的 License Manager 整合的最佳實務

內部部署環境中的容器應用程式部署可能會遇到不可靠的傳出網路存取。使用下列最佳實務來新增彈性，以避免因網際網路連線不佳造成潛在問題而導致買方的服務中斷：

- 充分重試 – 暫時性網路問題可能會讓您的應用程式無法連線到 AWS License Manager。實作重試最多 30 分鐘，並呈指數退避。這有助於避免短期中斷或網路問題。
- 避免硬性限制 – 部署在連線叢集中的應用程式可以定期檢查授權，以識別因升級或續約而產生的任何變更。透過不可靠的傳出存取，應用程式可能無法識別這些變更。盡可能避免由於無法透過

License Manager 檢查授權，導致買方的服務中斷。當授權過期且無法檢查授權是否有效時，應用程式可以回到免費試用或開放原始碼體驗。

- 通知客戶 – 使用快取的授權時，授權的任何變更（包括續約或升級）都不會自動反映在執行中的工作負載上。通知您的客戶（他們必須暫時允許傳出存取應用程式，以便應用程式可以更新其快取的授權。例如，透過應用程式本身或文件通知客戶。同樣地，當回復到一組較低的功能時，請通知客戶其權利已耗盡或授權已過期。然後，他們可以選擇升級或續約。

LicenseManagerCredentialsProvider - Java 實作

LicenseCredentialsProvider 透過新增 來擴展 AWS SDK 的預設登入資料提供者鏈，以供內部部署使用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 在內部部署環境中使用 License Manager OIDC 發行的身分字符來提供登入資料。您必須在應用程式 classpath LicenseCredentialsProvider中包含 的原始碼。

```
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.getStringV
            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實作

LicenseCredentialsProvider

LicenseCredentialsProvider 透過新增 來擴展 AWS SDK 的預設登入資料提供者鏈，以供內部部署使用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 在內部部署環境中使用 License Manager OIDC 發行的身分字符來提供登入資料。您必須在應用程式 classpath LicenseCredentialsProvider中包含 的原始碼。

```
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
 }
}
```

容器產品的 Amazon SNS 通知

若要接收通知，您可以訂閱產品建立期間 AWS Marketplace 提供給您的 Amazon Simple Notification Service (Amazon SNS) 主題。這些主題會針對您產品的客戶訂閱提供變更通知。例如，您可以使用這些通知來了解客戶何時接受私有優惠。

> **Note**
>
> 在產品建立過程中，會為您的產品建立 Amazon SNS 主題。若要訂閱通知，您需要 Amazon SNS 主題的 Amazon Resource Name (ARN)（例如 `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`)。ARN 不適用於伺服器產品的賣方入口網站。請聯絡 AWS Marketplace 操作團隊以請求 ARN。

下列 Amazon SNS 主題適用於容器產品：

- Amazon SNS 主題： `aws-mp-subscription-notification` – 本主題會在買方訂閱或取消訂閱產品時通知您。這適用於每小時定價模型，包括長期每小時和每小時。

Amazon SNS 主題： **`aws-mp-subscription-notification`**

`aws-mp-subscription-notification` 主題中的每個訊息都有下列格式。

```
{
    "action": "<action-name>",
```

```
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123"
}
```

<action-name> 會根據通知而有所不同。可能的動作包括：

- subscribe-success
- subscribe-fail
- unsubscribe-pending
- unsubscribe-success

只有在動作為 subscribe-success或 時， offer-identifier才會包含在通知中subscribe-fail。當動作為 unsubscribe-pending或 時，它不會包含在通知中unsubscribe-success。對於 2024 年 1 月之前建立的優惠，此識別符僅包含在私有優惠的通知中。對於 2024 年 1 月及更新版本建立的優惠，此識別符包含在所有優惠的通知中，包括私有優惠和公開優惠。

如需優惠類型的資訊，請參閱 DescribeEntity API 的回應或協議續約儀表板中的協議優惠可見性。

> **Note**
>
> 對於 DescribeEntity API，如果您 AWS 帳戶 在以該優惠規則為目標的面向帳戶中找到 ，則這是私有優惠。如果 AWS 帳戶 帳戶中沒有以該優惠規則為目標的面向為目標的 ，則這是公開優惠。

訂閱 Amazon SQS 佇列至 Amazon SNS 主題

我們建議訂閱 Amazon SQS 佇列至提供的 SNS 主題。如需建立 SQS 佇列和訂閱佇列至主題的詳細說明，請參閱《Amazon Simple Notification Service 開發人員指南》中的訂閱 Amazon SQS 佇列至 Amazon SNS 主題。

> **Note**
>
> 您只能從 AWS 帳戶 用來銷售產品的 訂閱 AWS Marketplace SNS 主題。不過，您可以將訊息轉送到不同的 帳戶。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的將 Amazon SNS 訊息傳送至不同帳戶中的 Amazon SQS 佇列。

輪詢 SQS 佇列以取得通知

將 SQS 佇列訂閱 SNS 主題後，訊息會儲存在 SQS 中。您必須定義持續輪詢佇列、尋找訊息並相應地處理它們的服務。

中的機器學習產品 AWS Marketplace

身為 AWS Marketplace 賣方，您可以建立買方可以部署的機器學習 (ML) 演算法和模型 AWS。本主題提供 中所列 Amazon SageMaker AI 產品類型的相關資訊 AWS Marketplace。

中列出兩種類型的 SageMaker AI 產品 AWS Marketplace：

模型套件

預先訓練的模型，用於進行不需要買方進一步訓練的預測。

演算法

需要買方在進行預測之前提供訓練資料的模型。包含訓練演算法。

這些產品可透過 Amazon SageMaker AI 主控台或 提供給買方 AWS Marketplace。買方可以檢閱產品描述、文件、客戶評論、定價和支援資訊。當他們訂閱模型套件產品或演算法產品時，它會新增到 SageMaker AI 主控台上的產品清單中。買方也可以使用 AWS SDKs、 AWS Command Line Interface (AWS CLI) 或 SageMaker AI 主控台來建立完全受管的 REST 推論端點，或在資料批次上執行推論。

如需使用 Amazon SageMaker AI 建立機器學習產品的支援，請聯絡AWS Marketplace 賣方營運團隊。

了解機器學習產品

AWS Marketplace 支援兩種使用 Amazon SageMaker AI 的機器學習產品類型。模型套件產品和演算法產品這兩種類型都會產生可部署的推論模型，以進行預測。

SageMaker AI 模型套件

Amazon SageMaker AI 模型套件產品包含預先訓練的模型。預先訓練的模型可以部署在 SageMaker AI 中，以即時或批次進行推論或預測。如果有，此產品包含訓練過的推論元件，其中包含模型成品。身為賣方，您可以使用 SageMaker AI 訓練模型，或自攜模型。

SageMaker AI 演算法

買方可以使用 SageMaker AI 演算法產品來執行完整的機器學習工作負載。演算法產品有兩個邏輯元件：訓練和推論。在 SageMaker AI 中，買方會使用自己的資料集，以您的訓練元件建立訓練任務。當

訓練元件中的演算法完成時，會產生機器學習模型的模型成品。SageMaker AI 會將模型成品儲存在買方的 Amazon Simple Storage Service (Amazon S3) 儲存貯體中。在 SageMaker AI 中，買方接著可以部署您的推論元件以及這些產生的模型成品，以即時或批次執行推論（或預測）。

部署推論模型

無論是從模型套件還是演算法建立推論模型，有兩種方法可以部署它們：

- 端點 – 此方法使用 SageMaker AI 部署模型並建立 API 端點。買方可以使用此端點做為後端服務的一部分，為其應用程式提供支援。將資料傳送至端點時，SageMaker AI 會將資料傳遞至模型容器，並在 API 回應中傳回結果。端點和容器會持續執行，直到買方停止為止。

  > **Note**
  >
  > 在 中 AWS Marketplace，端點方法稱為即時推論，在 SageMaker AI 文件中，它稱為託管服務。如需詳細資訊，請參閱在 Amazon SageMaker AI 中部署模型。

- 批次轉換任務 – 在此方法中，買方將資料集存放在 Amazon S3 中以進行推論。當批次轉換任務開始時，SageMaker AI 部署模型、將資料從 S3 儲存貯體傳遞至模型的容器，然後將結果傳回至 Amazon S3 儲存貯體。當任務完成時，SageMaker AI 會停止任務。如需詳細資訊，請參閱使用批次轉換。

  > **Note**
  >
  > 這兩種方法對模型都是透明的，因為 SageMaker AI 會將資料傳遞給模型，並將結果傳回給買方。

機器學習產品生命週期

中的機器學習產品 AWS Marketplace 包含一或多個軟體版本和相關聯的中繼資料。產品組態包含基本屬性，例如名稱、描述、使用說明、定價、分類和搜尋關鍵字。

機器學習產品建立程序

若要在 中列出機器學習產品 AWS Marketplace，您必須完成下列操作：

1. the section called “在 SageMaker AI 中準備您的產品”
2. the section called “在 中列出您的產品 AWS Marketplace”

建立機器學習產品後，您可以編輯和管理產品。如需詳細資訊，請參閱the section called “管理您的產品”。

機器學習產品狀態

新產品一開始的可見性有限，只有允許列出的帳戶和產品建立者才能存取。在測試和驗證之後，您可以發佈您的產品，讓所有買方都能在 AWS Marketplace 目錄中使用它。中的產品 AWS Marketplace 可以有下列狀態值：

Status	定義
Staging	此狀態表示您仍在新增資訊的不完整產品。在您第一次儲存並結束自助式體驗後， 會 AWS Marketplace 建立未發佈的產品，其中包含已完成步驟中的資訊。在此狀態中，您可以繼續新增資訊或修改提交的詳細資訊。
Limited	A product attains this status after it's submitted to AWS Marketplace and passes all validation checks. At this point, the product has a detail page accessible only to your account and allowlisted entities. You can conduct product testing through this detail page.
Public	When you're prepared to make your product visible to buyers for subscription, update the product visibility in the console. Once processed, the product transitions from Limited to Public status. For information about AWS guidelines, see the section called “需求和最佳實務”.
Restricted	若要防止新使用者訂閱您的產品，您可以透過更新可見性設定來限制它。受限狀態允許現有的允許清單使用者繼續使用產品，但新使用者將無法再看到它。

如需詳細資訊或支援，請聯絡AWS Marketplace 賣方營運團隊。

的機器學習產品定價 AWS Marketplace

您可以從 Amazon SageMaker AI 產品的數個可用定價模型中進行選擇 AWS Marketplace。訂閱您產品的買方會在自己的 SageMaker AI 中執行它 AWS 帳戶。買方的價格是其 中執行之資源的基礎設施成本 AWS 帳戶 以及您設定之產品定價的組合。下列各節提供有關 中 SageMaker AI 產品的定價模型資訊 AWS Marketplace

主題

- 基礎設施定價
- 軟體定價

基礎設施定價

買方在使用產品時，需負責 SageMaker AI 的所有基礎設施成本。這些成本由 設定 AWS ，並可在 Amazon SageMaker AI 定價頁面上取得。

軟體定價

您可以判斷使用您產品的買方 AWS Marketplace 收取的軟體價格。當您將機器學習產品新增至 時，您可以設定定價和條款 AWS Marketplace。

每個執行個體類型的所有基礎設施和軟體價格都會在買方訂閱 AWS Marketplace 之前，在 的產品清單頁面上呈現給買方。

主題

- 免費定價
- 每小時定價
- 推論定價
- 免費試用

免費定價

您可以選擇免費提供您的產品。在此情況下，買方只會支付基礎設施成本。

每小時定價

您可以在 SageMaker AI 中執行之軟體的每個執行個體，提供每小時價格的產品。您可以針對軟體執行所在的每個執行個體類型，收取不同的每小時價格。當買方執行您的軟體時， 會 AWS Marketplace 追蹤用量，然後相應地向買方收取費用。用量會按比例分配到分鐘。

對於模型套件產品，買方可以透過兩種不同的方式執行您的軟體。他們可以持續託管端點，以執行即時推論或在資料集上執行批次轉換任務。您可以為買方執行軟體的兩種方式設定不同的定價。

對於演算法產品，除了判斷執行推論的價格之外，如前所述，您也可以判斷訓練任務的每小時價格。您可以針對訓練映像支援的每個執行個體類型，收取不同的每小時價格。

推論定價

當買方透過託管端點以持續執行即時推論來執行您的軟體時，您可以選擇為每個推論設定價格。

> **Note**
>
> 批次轉換程序一律使用每小時定價。演算法產品的訓練任務也一律使用每小時定價。您可以獨立於推論定價和彼此設定這些價格。

根據預設，透過推論定價， 會針對端點的每個調用向您的買方 AWS Marketplace 收取費用。不過，在某些情況下，您的軟體會在單一調用中處理一批推論 （ 也稱為迷你批次)。對於端點部署，您可以指出 AWS Marketplace 應該向買方收取該單一調用費用的自訂推論數量。若要這樣做，請在呼叫的 HTTP 回應標頭中包含自訂計量標頭，如下列範例所示。此範例顯示呼叫，向買方收取三個推論的費用。

```
X-Amzn-Inference-Metering: {"Dimension": "inference.count", "ConsumedUnits": 3}
```

> **Note**
>
> 對於推論定價， AWS Marketplace 只會針對 HTTP 回應碼為 的請求向買方收取費用2XX。

免費試用

或者，您可以為您的產品建立免費試用，並定義免費試用的天數。免費試用可以是 5–31 天。在免費試用期間，買方可以根據需要執行您的軟體，而且不會收取您的軟體費用。買方在免費試用期間需支付基礎設施費用。試用期結束後，會向您的一般軟體價格以及基礎設施成本收費。

當買方訂閱具有免費試用的產品時，會收到歡迎電子郵件訊息。訊息包含免費試用期限、計算過期日期，以及取消訂閱的詳細資訊。提醒電子郵件訊息會在過期日期前三天傳送。

如果您在 中提供產品的免費試用 AWS Marketplace，即表示您同意免費試用的特定退款政策。

> **Note**
>
> 如需機器學習的私有優惠相關資訊，請參閱私有優惠。

中的機器學習產品的服務限制和配額 AWS Marketplace

本節說明 中機器學習 (ML) 產品的限制和配額 AWS Marketplace。

主題

- 網路隔離
- 映像大小
- 儲存體大小
- 執行個體大小
- 推論的承載大小
- 推論的處理時間
- Service Quotas
- 非同步推論
- 無伺服器推論
- 受管 Spot 訓練
- Docker 映像和 AWS 帳戶
- 從內建演算法或 發佈模型套件 AWS Marketplace
- 支援 AWS 區域 發佈

網路隔離

基於安全考量，當買方訂閱您的容器化產品時，Docker 容器會在沒有網路存取權的隔離環境中執行。當您建立容器時，請不要依賴透過網際網路進行傳出呼叫，因為它們會失敗。呼叫 AWS 服務 也會失敗。

映像大小

您的 Docker 映像大小由 Amazon Elastic Container Registry (Amazon ECR) 服務配額管理。Docker 映像大小會影響訓練任務、批次轉換任務和端點建立期間的啟動時間。為了獲得更好的效能，請維持最佳的 Docker 影像大小。

儲存體大小

當您建立端點時，Amazon SageMaker AI 會將 Amazon Elastic Block Store (Amazon EBS) 儲存磁碟區連接至託管端點的每個 ML 運算執行個體。（端點也稱為即時推論或 Amazon SageMaker AI 託管服務。）儲存磁碟區的大小取決於執行個體類型。如需詳細資訊，請參閱《Amazon SageMaker AI 開發人員指南》中的主機執行個體儲存磁碟區。

如需批次轉換，請參閱《Amazon SageMaker AI 開發人員指南》中的批次轉換中的儲存。

執行個體大小

SageMaker AI 提供多種執行個體類型，這些執行個體類型已針對不同的 ML 使用案例進行最佳化。執行個體類型由 CPU、GPU、記憶體和聯網容量的不同組合組成。執行個體類型可讓您靈活地選擇適當的資源組合，以建置、訓練和部署 ML 模型。如需詳細資訊，請參閱 Amazon SageMaker AI ML 執行個體類型。

推論的承載大小

對於端點，將每次呼叫的輸入資料大小上限限制為 6 MB。無法調整此值。

對於批次轉換，每次調用的輸入資料大小上限為 100 MB。此值無法調整。

推論的處理時間

對於端點，每次調用的處理時間上限為 60 秒。此值無法調整。

對於批次轉換，每次調用的處理時間上限為 60 分鐘。此值無法調整。

Service Quotas

如需訓練和推論相關配額的詳細資訊，請參閱Amazon SageMaker AI Service Quotas。

非同步推論

在 中發佈的模型套件和演算法 AWS Marketplace 無法部署到為 Amazon SageMaker AI 非同步推論設定的端點。針對非同步推論設定的端點需要模型具有網路連線能力。所有 AWS Marketplace 模型都以網路隔離運作。如需詳細資訊，請參閱無網路存取。

無伺服器推論

在 中發佈的模型套件和演算法 AWS Marketplace 無法部署到為 Amazon SageMaker AI Serverless Inference 設定的端點。針對無伺服器推論設定的端點需要模型具有網路連線能力。所有 AWS Marketplace 模型都以網路隔離運作。如需詳細資訊，請參閱無網路存取。

受管 Spot 訓練

對於來自 的所有演算法 AWS Marketplace，即使實作了受管 Spot 訓練的檢查點， 的值`MaxWaitTimeInSeconds`也會設為 3，600 秒 (60 分鐘）。無法調整此值。

Docker 映像和 AWS 帳戶

若要發佈，映像必須存放在賣方 擁有 AWS 帳戶 的 Amazon ECR 儲存庫中。無法發佈存放在另一個儲存庫中的映像 AWS 帳戶。

從內建演算法或 發佈模型套件 AWS Marketplace

使用 Amazon SageMaker AI 內建演算法或 AWS Marketplace 訂閱演算法從訓練任務建立的模型套件無法發佈。

您仍然可以使用訓練任務中的模型成品，但發佈模型套件需要您自己的推論影像。

支援 AWS 區域 發佈

AWS Marketplace 支援從 AWS 區域 下列位置發佈模型套件和演算法資源：

- Amazon SageMaker AI 支援的 區域
- 預設選擇加入的可用區域 （例如 describe-regions 傳回 `"OptInStatus": "opt-in-not-required"`）

發佈模型套件或演算法產品所需的所有資產都必須存放在您選擇發佈的相同區域中。這包含下列項目：

- 在 Amazon SageMaker AI 中建立的模型套件和演算法資源
- 上傳至 Amazon ECR 儲存庫的推論和訓練映像
- 存放在 Amazon Simple Storage Service 中並在模型部署期間動態載入模型套件資源的模型成品（如果有的話）
- Amazon S3 中存放的推論和訓練驗證測試資料

您可以在 SageMaker AI 支援的任何區域中開發和訓練您的產品。但是，您必須先將所有資產複製到 AWS Marketplace 支援從中發佈的區域中並重新建立資源，才能發佈。

Amazon SageMaker AI 的安全和智慧財產權

Amazon SageMaker AI 會保護您的智慧財產權和買方資料，以取得從中取得的模型和演算法 AWS Marketplace。下列各節提供 SageMaker AI 保護智慧財產權和客戶資料安全性方式的詳細資訊。

主題

- 保護智慧財產權
- 無網路存取
- 客戶資料的安全性

保護智慧財產權

當您建立產品時，程式碼會封裝在 Docker 容器映像中。如需詳細資訊，請參閱本指南在 SageMaker AI 中準備您的產品稍後的 。當您上傳容器映像時，映像和成品會在傳輸中和靜態時加密。在發佈之前，也會掃描映像是否有漏洞。

為了協助保護您的智慧財產權，SageMaker AI 僅允許買方透過 AWS 服務 端點存取您的產品。買方無法直接存取或提取容器映像或模型成品，也無法存取基礎基礎設施。

無網路存取

與買方建立的 SageMaker AI 模型和演算法不同，當買方從中啟動您的產品時 AWS Marketplace，模型和演算法會在沒有網路存取的情況下部署。SageMaker AI 會在無法存取網路或 AWS 服務 端點的環境中部署映像。例如，容器映像無法對網際網路、VPC 端點或任何其他 上的服務進行傳出 API 呼叫 AWS 服務。

客戶資料的安全性

您的產品在買方的 SageMaker AI 中執行 AWS 帳戶。因此，當買方使用您的產品執行資料推論時，身為賣方的您無法存取其資料。

對於演算法產品，模型成品會在每次訓練任務後由訓練影像輸出。模型成品會存放在買方的帳戶中。當買方使用推論映像部署模型時，會使用訓練任務中的模型成品。若要保護模型成品中可能包含的任何智慧財產權，請在輸出前對其進行加密。

> **Important**
>
> 此安全模型可防止您的程式碼在執行時間存取網際網路。因此，您的程式碼無法使用來自網際網路的資源或程式庫，因此請在 Docker 容器映像中封裝您的相依性。如果您選擇加密訓練任務輸出的成品，這尤其重要。無法在執行時間透過網際網路存取加密和解密成品的金鑰。它們必須與映像一起封裝。

如需詳細資訊，請參閱 Amazon SageMaker AI 中的安全性。

中的機器學習報告 AWS Marketplace

AWS Marketplace 為您的 Amazon SageMaker AI 產品產生報告，其中包含買方、財務、用量和稅金的資料。所有報告都可在 報告頁面上 AWS Marketplace 管理入口網站 的 中取得。如需詳細資訊，請參閱賣方報告。下列各節提供機器學習產品報告的摘要資訊。

主題

- 每日業務報告
- 每月營收報告
- 支出報告
- 其他報告和分析

每日業務報告

每日業務報告提供執行個體類型、使用時數、軟體費用收入，以及每個買方和產品的其他詳細資訊。買方可以透過其 AWS 帳戶 ID 來識別。如需詳細資訊，請參閱每日業務報告。

每月營收報告

每月收入報告為您提供每月收入，已向您的買方收取使用您的軟體的費用。如需詳細資訊，請參閱每月計費營收報告。

支出報告

每月支出報告提供在軟體費用的結算期間代您收取的所有資金明細。報告中反映的總和解金額應與存入您銀行帳戶的金額相符。如需詳細資訊，請參閱 支出報告。

其他報告和分析

如需其他可用的報告，請參閱賣方報告。

您也可以使用 中的賣方交付資料摘要 AWS Marketplace 提供的 建立自訂報告 AWS Marketplace。

在 SageMaker AI 中準備您的產品

您必須先在 Amazon SageMaker AI 中準備產品 AWS Marketplace，才能在 中發佈產品。中列出兩種類型的 SageMaker AI 產品 AWS Marketplace：模型套件和演算法。如需詳細資訊，請參閱中的機器學習產品 AWS Marketplace。本主題概述準備產品所需的三個步驟：

1. 將您的程式碼封裝到 中的機器學習產品映像中 AWS Marketplace – 若要準備模型套件或演算法產品，您必須為產品建立 Docker 容器映像。
2. 將映像上傳至 Amazon Elastic Container Registry – 在容器映像中封裝程式碼並在本機測試它們之後，上傳映像並掃描它們是否有已知的漏洞。修正任何漏洞後再繼續。
3. 建立 Amazon SageMaker AI 資源 – 成功掃描映像後，您可以使用它們在 SageMaker AI 中建立模型套件或演算法資源。

將您的程式碼封裝到 中的機器學習產品映像中 AWS Marketplace

中的機器學習產品 AWS Marketplace 使用 Amazon SageMaker AI 來建立和執行您提供給買方的機器學習邏輯。SageMaker AI 會執行包含邏輯的 Docker 容器映像。SageMaker AI 會在安全且可擴展的基礎設施中執行這些容器。如需詳細資訊，請參閱Amazon SageMaker AI 的安全和智慧財產權。下列各節提供如何將程式碼封裝至 SageMaker AI 的 Docker 容器映像的相關資訊。

主題

- 我要建立哪種類型的容器映像？
- 建立模型套件映像
- 建立演算法映像

我要建立哪種類型的容器映像？

容器映像的兩種類型是推論映像和訓練映像。

若要建立模型套件產品，您只需要推論映像。如需詳細說明，請參閱 建立模型套件映像。

若要建立演算法產品，您需要訓練和推論映像。如需詳細說明，請參閱 建立演算法映像。

若要將程式碼正確封裝到容器映像中，容器必須遵循 SageMaker AI 檔案結構。容器必須公開正確的端點，以確保服務可以往返您的容器傳遞資料。下列各節說明此程序的詳細資訊。

> **Important**
>
> 基於安全考量，當買方訂閱您的容器化產品時，Docker 容器會在沒有網際網路連線的隔離環境中執行。當您建立容器時，請不要依賴透過網際網路的外撥通話，因為它們會失敗。呼叫 AWS 服務 也會失敗。如需詳細資訊，請參閱 Amazon SageMaker AI 的安全和智慧財產權 一節。

或者，建立推論和訓練映像時，請使用可用深度學習容器映像中的容器做為起點。映像已正確封裝不同的機器學習架構。

建立模型套件映像

Amazon SageMaker AI 模型套件是預先訓練的模型，可進行預測，不需要買方進行任何進一步訓練。您可以在 SageMaker AI 中建立模型套件，並在其中發佈機器學習產品 AWS Marketplace。以下各節說明如何建立 的模型套件 AWS Marketplace。這包括在本機建立容器映像，以及建置和測試映像。

主題

- 概觀
- 建立模型套件的推論映像

概觀

模型套件包含下列元件：

- 儲存在Amazon Elastic Container Registry (Amazon ECR) 中的推論映像
- （選用）模型成品，分別存放在 Amazon S3 中

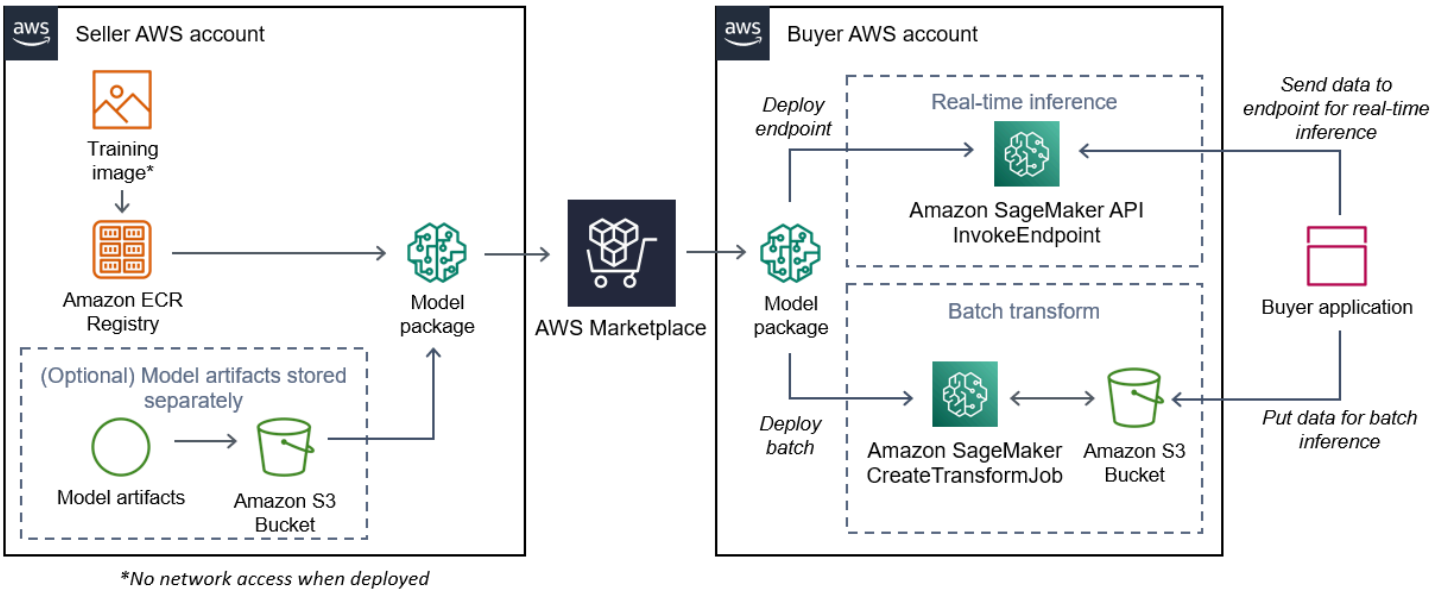
> **Note**
>
> 模型成品是模型用來進行預測的檔案，通常是您訓練程序的結果。成品可以是模型所需的任何檔案類型，但必須使用.tar.gz 壓縮。對於模型套件，它們可以綁定在您的推論映像中，或分別存放在 Amazon SageMaker AI 中。Amazon S3 中存放的模型成品會在執行時間載入推論容

器。發佈模型套件時，這些成品會發佈並存放在買方無法直接存取的 AWS Marketplace 擁有 Amazon S3 儲存貯體中。

Tip

如果您的推論模型使用 Gluon、Keras、MXNet、PyTorch、TensorFlow、TensorFlow-Lite 或 ONNX 等深度學習架構建置，請考慮使用 Amazon SageMaker AI Neo。Neo 可以自動最佳化部署到特定雲端執行個體類型的推論模型，例如 `ml.c4`、 `ml.p2`和其他。如需詳細資訊，請參閱《Amazon SageMaker AI 開發人員指南》中的使用 Neo 最佳化模型效能。

下圖顯示發佈和使用模型套件產品的工作流程。



為 建立 SageMaker AI 模型套件的工作流程 AWS Marketplace 包括以下步驟：

1. 賣方會建立推論映像 （部署時無法存取網路），並將其推送至 Amazon ECR 登錄檔。

   模型成品可以綁定在推論映像中，或分別存放在 S3 中。
2. 然後，賣方在 Amazon SageMaker AI 中建立模型套件資源，並在其中發佈 ML 產品 AWS Marketplace。
3. 買方訂閱 ML 產品並部署模型。

> **Note**
>
> 模型可以部署為即時推論的端點，也可以部署為批次任務，以同時取得整個資料集的預測。如需詳細資訊，請參閱部署推論模型。

4. SageMaker AI 會執行推論映像。推論映像中未綁定的任何賣方提供的模型成品都會在執行時間動態載入。
5. SageMaker AI 使用容器的 HTTP 端點將買方的推論資料傳遞至容器，並傳回預測結果。

建立模型套件的推論映像

本節提供逐步解說，將您的推論程式碼封裝到模型套件產品的推論映像中。程序包含下列步驟：

步驟

- 步驟 1：建立容器映像
- 步驟 2：在本機建置和測試映像

推論映像是包含推論邏輯的 Docker 映像。執行時間的容器公開 HTTP 端點，以允許 SageMaker AI 將資料傳遞至容器，以及從容器傳遞資料。

> **Note**
>
> 以下是推論映像的封裝程式碼範例。如需詳細資訊，請參閱在 GitHub 上使用 Docker 容器搭配 SageMaker AI 和 AWS Marketplace SageMaker AI 範例。
> 下列範例使用 Web 服務 Flask 來簡化操作，且不被視為可立即生產。

步驟 1：建立容器映像

若要讓推論映像與 SageMaker AI 相容，Docker 映像必須公開 HTTP 端點。當您的容器執行時，SageMaker AI 會將買方輸入傳遞至容器的 HTTP 端點以進行推論。推論結果會在 HTTP 回應的內文中傳回。

下列逐步解說使用 Linux Ubuntu 分佈在開發環境中使用 Docker CLI。

- 建立 Web 伺服器指令碼
- 建立容器執行的指令碼

- 建立 Dockerfile
- 封裝或上傳模型成品

建立 Web 伺服器指令碼

此範例使用名為 Flask 的 Python 伺服器，但您可以使用適用於架構的任何 Web 伺服器。

> **Note**
>
> 此處使用 Flask 以簡化操作。它不被視為生產就緒的 Web 伺服器。

在 SageMaker AI 使用的 TCP 連接埠 8080 上建立 Flask Web 伺服器指令碼，提供兩個 HTTP 端點。以下是兩個預期的端點：

- `/ping` – SageMaker AI 會向此端點發出 HTTP GET 請求，以檢查您的容器是否已準備就緒。當您的容器準備就緒時，它會在此端點使用 HTTP 200 回應碼來回應 HTTP GET 請求。
- `/invocations` – SageMaker AI 會對此端點發出 HTTP POST 請求以進行推論。推論的輸入資料會在請求內文中傳送。使用者指定的內容類型會在 HTTP 標頭中傳遞。回應的內文是推論輸出。如需逾時的詳細資訊，請參閱 建立機器學習產品的需求和最佳實務。

`./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)
```

在上一個範例中，沒有實際推論邏輯。針對您的實際推論映像，將推論邏輯新增至 Web 應用程式，以便處理輸入並傳回實際預測。

您的推論映像必須包含所有必要的相依性，因為它將無法存取網際網路，也無法呼叫任何 AWS 服務。

> **Note**
>
> 即時和批次推論都會呼叫相同的程式碼

建立容器執行的指令碼

建立名為 SageMaker AI `serve` 在執行 Docker 容器映像時執行的指令碼。下列指令碼會啟動 HTTP Web 伺服器。

`./serve`

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker
flask run --host 0.0.0.0 --port 8080
```

建立 **`Dockerfile`**

在建置內容`Dockerfile`中建立 。此範例使用 Ubuntu 18.04，但您可以從任何適用於架構的基本映像開始。

./Dockerfile

```
FROM ubuntu:18.04

# Specify encoding
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8

# Install python-pip
RUN apt-get update \
&& apt-get install -y python3.6 python3-pip \
&& ln -s /usr/bin/python3.6 /usr/bin/python \
&& ln -s /usr/bin/pip3 /usr/bin/pip;

# Install flask server
RUN pip install -U Flask;

# Add a web server script to the image
# Set an environment to tell flask the script to run
COPY /web_app_serve.py /web_app_serve.py
ENV FLASK_APP=/web_app_serve.py

# Add a script that Amazon SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

會將先前建立的兩個指令碼Dockerfile新增至映像。serve 指令碼的目錄會新增至 PATH，以便在容器執行時執行。

封裝或上傳模型成品

從訓練模型到推論映像的兩種模型成品方式如下：

- 使用推論映像靜態封裝。
- 在執行時間動態載入。因為它是動態載入的，因此您可以使用相同的映像來封裝不同的機器學習模型。

如果您想要使用推論映像封裝模型成品，請在 中包含成品Dockerfile。

如果您想要動態載入模型成品，請將這些成品分別存放在 Amazon S3 的壓縮檔案 (.tar.gz) 中。建立模型套件時，請指定壓縮檔案的位置，SageMaker AI 會在執行容器`/opt/ml/model/`時擷取內容並將其複製到容器目錄。發佈模型套件時，這些成品會發佈並存放在買方無法直接存取的擁有 Amazon S3 儲存貯體中 AWS Marketplace 。

步驟 2：在本機建置和測試映像

在建置內容中，現在存在下列檔案：

- `./Dockerfile`
- `./web_app_serve.py`
- `./serve`
- 您的推論邏輯和 （選用） 相依性

接下來建置、執行和測試容器映像。

建置映像

在建置內容中執行 Docker 命令，以建置和標記映像。此範例使用標籤 `my-inference-image`。

```
sudo docker build --tag my-inference-image ./
```

執行此 Docker 命令來建置映像後，您應該會看到輸出，因為 Docker 會根據 中的每行建置映像`Dockerfile`。完成後，您應該會看到類似以下內容的內容。

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```

在本機執行

建置完成後，您可以在本機測試映像。

```
sudo docker run \
  --rm \
  --publish 8080:8080/tcp \
  --detach \
  --name my-inference-container \
  my-inference-image \
```

```
  serve
```

以下是 命令的詳細資訊：

- `--rm` – 停止後自動移除容器。
- `--publish 8080:8080/tcp` – 公開連接埠 8080 以模擬 SageMaker AI 傳送 HTTP 請求的連接埠。
- `--detach` – 在背景執行容器。
- `--name my-inference-container` – 為此執行中的容器命名。
- `my-inference-image` – 執行建置的映像。
- `serve` – 執行與執行容器時 SageMaker AI 執行的相同指令碼。

執行此命令後，Docker 會從您建置的推論映像建立容器，並在背景執行。容器會執行`serve`指令碼，這會啟動您的 Web 伺服器以供測試之用。

測試 ping HTTP 端點

當 SageMaker AI 執行您的容器時，它會定期 Ping 端點。當端點傳回狀態碼為 200 的 HTTP 回應時，它會向 SageMaker AI 發出訊號，表示容器已準備好進行推論。您可以執行下列命令來測試此項目，該命令會測試端點並包含回應標頭。

```
curl --include http://127.0.0.1:8080/ping
```

輸出範例如下所示。

```
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
```

測試推論 HTTP 端點

當容器將 200 狀態碼傳回 ping 表示已準備就緒時，SageMaker AI 會透過`POST`請求將推論資料傳遞至`/invocations` HTTP 端點。執行下列命令來測試推論點。

```
curl \
```

```
  --request POST \
  --data "hello world" \
  http://127.0.0.1:8080/invocations
```

輸出範例如下所示。

```
{"prediction": "a", "text": "hello world"}
```

當這兩個 HTTP 端點運作時，推論映像現在與 SageMaker AI 相容。

> **Note**
>
> 模型套件產品的模型有兩種部署方式：即時和批次。在這兩種部署中，SageMaker AI 在執行 Docker 容器時使用相同的 HTTP 端點。

若要停止容器，請執行下列命令。

```
sudo docker container stop my-inference-container
```

當您的推論映像準備就緒並經過測試時，您可以繼續 將映像上傳至 Amazon Elastic Container Registry。

建立演算法映像

Amazon SageMaker AI 演算法要求買方在進行預測之前，先攜帶自己的資料進行訓練。身為 AWS Marketplace 賣方，您可以使用 SageMaker AI 來建立機器學習 (ML) 演算法和模型，供買方部署 AWS。下列各節說明如何建立演算法映像 AWS Marketplace。這包括建立 Docker 訓練映像來訓練您的演算法，以及包含推論邏輯的推論映像。發佈演算法產品時，需要訓練和推論影像。

主題

- 概觀
- 建立演算法的訓練映像
- 建立演算法的推論映像

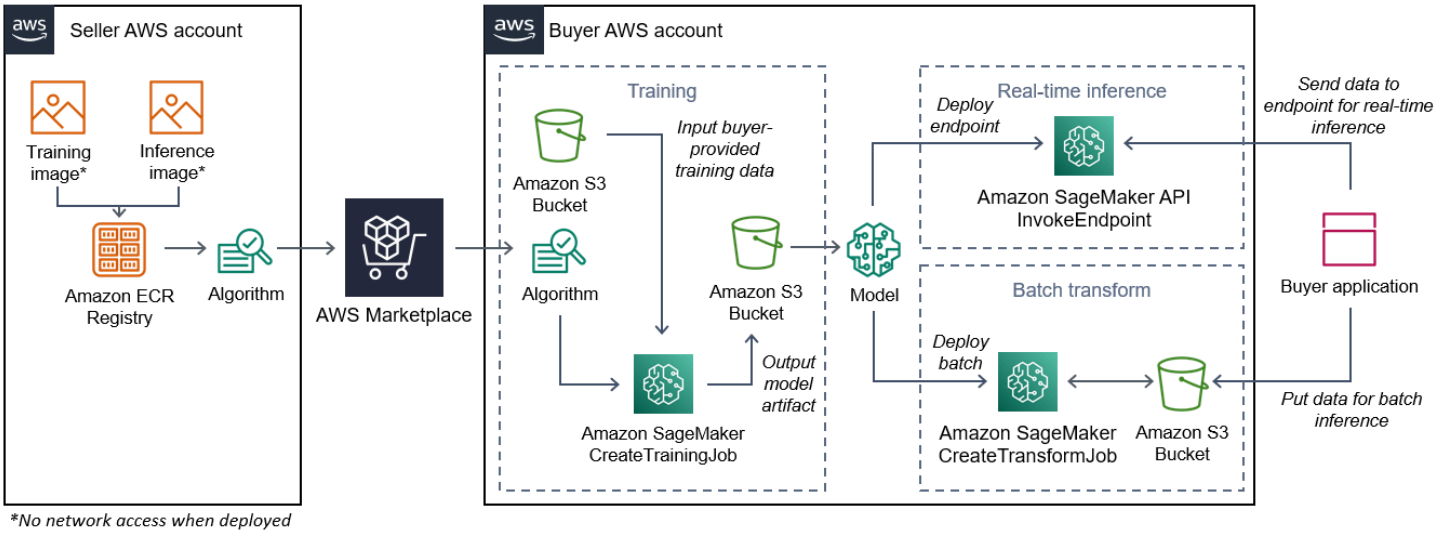
概觀

演算法包含下列元件：

- 存放在 Amazon ECR 中的訓練映像
- 儲存在 Amazon Elastic Container Registry (Amazon ECR) 中的推論映像

> **Note**
>
> 對於演算法產品，訓練容器會產生模型成品，這些成品會在模型部署時載入推論容器。

下圖顯示發佈和使用演算法產品的工作流程。



為 建立 SageMaker AI 演算法的工作流程 AWS Marketplace 包括以下步驟：

1. 賣方會建立訓練映像和推論映像（部署時無法存取網路），並將其上傳至 Amazon ECR 登錄檔。
2. 然後，賣方會在 Amazon SageMaker AI 中建立演算法資源，並在 上發佈 ML 產品 AWS Marketplace。
3. 買方訂閱 ML 產品。
4. 買方會使用相容的資料集和適當的超參數值來建立訓練任務。SageMaker AI 會執行訓練映像，並將訓練資料和超參數載入訓練容器。當訓練任務完成時，位於 中的模型成品`/opt/ml/model/`會壓縮並複製到買方的 Amazon S3 儲存貯體。
5. 買方會建立模型套件，其中包含 Amazon S3 中所存放訓練的模型成品，並部署模型。
6. SageMaker AI 會執行推論映像、擷取壓縮的模型成品，並將檔案載入推論容器目錄路徑，`/opt/ml/model/`供推論的程式碼使用。
7. 無論模型部署為端點或批次轉換任務，SageMaker AI 都會代表買方將資料傳遞至容器的 HTTP 端點，並將預測結果傳回。

> **Note**
>
> 如需詳細資訊，請參閱訓練模型。

建立演算法的訓練映像

本節提供將訓練程式碼封裝至訓練映像的逐步解說。建立演算法產品需要訓練影像。

訓練映像是包含訓練演算法的 Docker 映像。容器會遵守特定的檔案結構，以允許 SageMaker AI 將資料複製到您的容器或從容器複製資料。

發佈演算法產品時需要訓練和推論影像。建立訓練影像後，您必須建立推論影像。這兩個影像可以合併為一個影像或保留為個別影像。是否合併影像或將其分開由您決定。一般而言，推論比訓練更簡單，您可能想要不同的映像來協助推論效能。

> **Note**
>
> 以下是訓練映像的封裝程式碼範例。如需詳細資訊，請參閱使用您自己的演算法和模型搭配 AWS Marketplace和 GitHub 上的 AWS Marketplace SageMaker AI 範例。

步驟

- 步驟 1：建立容器映像
- 步驟 2：在本機建置和測試映像

步驟 1：建立容器映像

為了使訓練映像與 Amazon SageMaker AI 相容，它必須遵守特定檔案結構，以允許 SageMaker AI 將訓練資料和組態輸入複製到容器中的特定路徑。當訓練完成時，產生的模型成品會存放在 SageMaker AI 複製來源容器中的特定目錄路徑中。

以下使用安裝在 Linux Ubuntu 發行版本開發環境中的 Docker CLI。

- 準備您的程式以讀取組態輸入
- 準備您的程式以讀取資料輸入
- 準備您的程式以撰寫訓練輸出
- 建立容器執行的指令碼

- 建立 Dockerfile

準備您的程式以讀取組態輸入

如果您的訓練計畫需要任何買方提供的組態輸入，則下列是在執行時複製到容器內的組態輸入。如有必要，您的程式必須從這些特定檔案路徑中讀取。

- `/opt/ml/input/config` 是 目錄，其中包含控制程式執行方式的資訊。
  - `hyperparameters.json` 是超參數名稱和值的 JSON 格式字典。這些值是字串，因此您可能需要轉換它們。
  - `resourceConfig.json` 是 JSON 格式的檔案，描述用於分散式訓練的網路配置。如果您的訓練映像不支援分散式訓練，您可以忽略此檔案。

> **Note**
>
> 如需組態輸入的詳細資訊，請參閱 Amazon SageMaker AI 如何提供訓練資訊。

準備您的程式以讀取資料輸入

訓練資料可以下列兩種模式之一傳送到容器。您在容器中執行的訓練程式會在這兩種模式中之一摘要訓練資料。

檔案模式

- `/opt/ml/input/data/<channel_name>/` 包含該頻道的輸入資料。這些頻道是根據對 `CreateTrainingJob` 操作的呼叫建立，但頻道通常必須符合演算法預期的內容。每個頻道的檔案會從 Amazon S3 複製到此目錄，保留 Amazon S3 金鑰結構所指示的樹狀結構。

管道模式

- `/opt/ml/input/data/<channel_name>_<epoch_number>` 是指定 epoch 的管道。Epoch 從零開始，每次讀取時都會增加一個。您可以執行的 epoch 數目沒有限制，但您必須先關閉每個管道，才能讀取下一個 epoch。

準備您的程式以撰寫訓練輸出

訓練的輸出會寫入下列容器目錄：

- /opt/ml/model/ 是您撰寫模型或訓練演算法產生的模型成品的目錄。您的模型可以是您想要的任何格式。它可以是單一檔案或整個目錄樹狀目錄。SageMaker AI 會將此目錄中的任何檔案封裝為壓縮檔案 (.tar.gz)。此檔案可在 DescribeTrainingJob API 操作傳回的 Amazon S3 位置使用。
- /opt/ml/output/ 是一個目錄，其中演算法可以撰寫描述任務失敗原因failure的檔案。此檔案的內容會在DescribeTrainingJob結果的 FailureReason 欄位中傳回。對於成功的任務，沒有理由寫入此檔案，因為其被忽略。

建立容器執行的指令碼

建立 SageMaker AI 在執行 Docker 容器映像時執行的 train Shell 指令碼。當訓練完成且模型成品寫入其個別目錄時，請結束指令碼。

./train

```
#!/bin/bash

# Run your training program here
#
#
#
#
```

建立 **Dockerfile**

在建置內容Dockerfile中建立 。此範例使用 Ubuntu 18.04 作為基礎映像，但您可以從適用於架構的任何基礎映像開始。

./Dockerfile

```
FROM ubuntu:18.04

# Add training dependencies and programs
#
#
#
#
#
# Add a script that SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /train /opt/program/train
```

```
RUN chmod 755 /opt/program/train
ENV PATH=/opt/program:${PATH}
```

會將先前建立的train指令碼Dockerfile新增至映像。指令碼的目錄會新增至 PATH，以便在容器執行時執行。

在上一個範例中，沒有實際的訓練邏輯。針對您的實際訓練影像，將訓練相依性新增至 Dockerfile，並新增邏輯以讀取訓練輸入，以訓練和產生模型成品。

您的訓練映像必須包含所有必要的相依性，因為它將無法存取網際網路。

如需詳細資訊，請參閱使用您自己的演算法和模型搭配 AWS Marketplace和 GitHub 上的 AWS Marketplace SageMaker AI 範例。

步驟 2：在本機建置和測試映像

在建置內容中，現在存在下列檔案：

- ./Dockerfile
- ./train
- 您的訓練相依性和邏輯

接下來，您可以建置、執行和測試此容器映像。

建置映像

在建置內容中執行 Docker 命令，以建置和標記映像。此範例使用標籤 my-training-image。

```
sudo docker build --tag my-training-image ./
```

執行此 Docker 命令來建置映像後，您應該會看到輸出，因為 Docker 會根據 中的每行建置映像Dockerfile。完成後，您應該會看到類似以下內容的內容。

```
Successfully built abcdef123456
Successfully tagged my-training-image:latest
```

在本機執行

完成後，請在本機測試映像，如下列範例所示。

```
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
```

以下是命令詳細資訊：

- `--rm` – 停止後自動移除容器。
- `--volume '<path_to_input>:/opt/ml/input:ro'` – 讓容器可使用測試輸入目錄做為唯讀。
- `--volume '<path_to_model>:/opt/ml/model'` – 訓練測試完成時，繫結掛載模型成品存放在主機機器的路徑。
- `--volume '<path_to_output>:/opt/ml/output'` – 繫結掛載路徑，其中`failure`檔案中的失敗原因寫入主機機器。
- `--name my-training-container` – 為這個執行中的容器命名。
- `my-training-image` – 執行建置的映像。
- `train` – 執行相同的指令碼 SageMaker AI 會在執行容器時執行。

執行此命令後，Docker 會從您建置的訓練映像建立容器並執行它。容器會執行`train`指令碼，以啟動您的訓練計畫。

訓練程式完成且容器結束之後，請檢查輸出模型成品是否正確。此外，請檢查日誌輸出以確認它們未產生您不希望的日誌，同時確保提供有關訓練任務的足夠資訊。

如此即完成為演算法產品封裝訓練程式碼。由於演算法產品也包含推論映像，請繼續下一個區段：建立演算法的推論映像。

建立演算法的推論映像

本節提供逐步解說，將您的推論程式碼封裝到演算法產品的推論映像中。

推論映像是包含推論邏輯的 Docker 映像。執行時間的容器公開 HTTP 端點，以允許 SageMaker AI 將資料傳遞至容器，以及從容器傳遞資料。

發佈演算法產品時，需要訓練和推論映像。如果您尚未這樣做，請參閱上一節有關 的章節建立演算法的訓練映像。這兩個影像可以合併為一個影像，或保留為個別影像。是否合併或分隔影像由您決定。一般而言，推論比訓練更簡單，您可能希望個別的影像有助於推論效能。

> **Note**
>
> 以下是推論映像的封裝程式碼範例。如需詳細資訊，請參閱使用您自己的演算法和模型搭配 AWS Marketplace和 GitHub 上的 AWS Marketplace SageMaker AI 範例。
> 下列範例使用 Web 服務 Flask 來簡化操作，且不被視為可立即生產。

步驟

- 步驟 1：建立推論映像
- 步驟 2：在本機建置和測試映像

步驟 1：建立推論映像

若要讓推論映像與 SageMaker AI 相容，Docker 映像必須公開 HTTP 端點。當您的容器正在執行時，SageMaker AI 會將買方提供的推論輸入傳遞到容器的 HTTP 端點。推論的結果會在 HTTP 回應的內文中傳回。

以下使用安裝在 Linux Ubuntu 發行版本開發環境中的 Docker CLI。

- 建立 Web 伺服器指令碼
- 建立容器執行的指令碼
- 建立 `Dockerfile`
- 準備您的程式以動態載入模型成品

建立 Web 伺服器指令碼

此範例使用名為 Flask 的 Python 伺服器，但您可以使用適用於架構的任何 Web 伺服器。

> **Note**
>
> 此處使用 Flask 以簡化操作。它不被視為生產就緒的 Web 伺服器。

在 SageMaker AI 使用的 TCP 連接埠 8080 上建立 Flask Web 伺服器指令碼，提供兩個 HTTP 端點。以下是兩個預期的端點：

- `/ping` – SageMaker AI 會向此端點發出 HTTP GET 請求，以檢查您的容器是否已準備就緒。當您的容器準備就緒時，它會在此端點使用 HTTP 200 回應碼來回應 HTTP GET 請求。

- /invocations – SageMaker AI 會對此端點發出 HTTP POST 請求以進行推論。推論的輸入資料會在請求內文中傳送。使用者指定的內容類型會在 HTTP 標頭中傳遞。回應的內文是推論輸出。

./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)
```

在上一個範例中，沒有實際推論邏輯。針對您的實際推論映像，將推論邏輯新增至 Web 應用程式，以便處理輸入並傳回預測。

您的推論映像必須包含所有必要的相依性，因為它將無法存取網際網路。

建立容器執行的指令碼

建立名為 SageMaker AI `serve` 在執行 Docker 容器映像時執行的指令碼。在此指令碼中，啟動 HTTP Web 伺服器。

`./serve`

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker AI
flask run --host 0.0.0.0 --port 8080
```

建立 **`Dockerfile`**

在建置內容`Dockerfile`中建立 。此範例使用 Ubuntu 18.04，但您可以從任何適用於架構的基本映像開始。

`./Dockerfile`

```
FROM ubuntu:18.04

# Specify encoding
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8

# Install python-pip
RUN apt-get update \
&& apt-get install -y python3.6 python3-pip \
&& ln -s /usr/bin/python3.6 /usr/bin/python \
&& ln -s /usr/bin/pip3 /usr/bin/pip;

# Install flask server
RUN pip install -U Flask;

# Add a web server script to the image
# Set an environment to tell flask the script to run
COPY /web_app_serve.py /web_app_serve.py
ENV FLASK_APP=/web_app_serve.py

# Add a script that Amazon SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
```

```
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

會將先前建立的兩個指令碼Dockerfile新增至映像。serve 指令碼的目錄會新增至 PATH，以便在容器執行時執行。

準備您的程式以動態載入模型成品

對於演算法產品，買方會使用自己的資料集搭配您的訓練映像來產生唯一的模型成品。當訓練程序完成時，您的訓練容器會將模型成品輸出至容器目錄 /opt/ml/model/。SageMaker AI 會將該目錄中的內容壓縮為 .tar.gz 檔案，並將其存放在 Amazon S3 中的買方 AWS 帳戶 中。

當模型部署時，SageMaker AI 會執行您的推論映像，從存放在 Amazon S3 中買方帳戶中的 .tar.gz 檔案擷取模型成品，並將其載入目錄中的推論容器中/opt/ml/model/。在執行時間，您的推論容器程式碼會使用模型資料。

> **Note**
>
> 若要保護模型成品檔案中可能包含的任何智慧財產權，您可以選擇先加密檔案再輸出。如需詳細資訊，請參閱Amazon SageMaker AI 的安全和智慧財產權。

步驟 2：在本機建置和測試映像

在建置內容中，現在存在下列檔案：

- ./Dockerfile
- ./web_app_serve.py
- ./serve

接下來，您可以建置、執行和測試此容器映像。

建置映像

執行 Docker 命令來建置和標記映像。此範例使用標籤 my-inference-image。

```
sudo docker build --tag my-inference-image ./
```

執行此 Docker 命令來建置映像後，您應該會看到輸出，因為 Docker 會根據 中的每行建置映像Dockerfile。完成後，您應該會看到類似以下內容的內容。

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```

在本機執行

建置完成後，您可以在本機測試映像。

```
sudo docker run \
  --rm \
  --publish 8080:8080/tcp \
  --volume '<path_to_model>:/opt/ml/model:ro' \
  --detach \
  --name my-inference-container \
  my-inference-image \
  serve
```

以下是命令詳細資訊：

- `--rm` – 停止後自動移除容器。
- `--publish 8080:8080/tcp` – 公開連接埠 8080 以模擬連接埠 SageMaker AI 將 HTTP 請求傳送至其中。
- `--volume '<path_to_model>:/opt/ml/model:ro'` – 繫結將測試模型成品存放在主機機器的路徑掛載為唯讀，以便可供容器中的推論程式碼使用。
- `--detach` – 在背景執行容器。
- `--name my-inference-container` – 為此執行中的容器命名。
- `my-inference-image` – 執行建置的映像。
- `serve` – 執行相同的指令碼 SageMaker AI 會在執行容器時執行。

執行此命令後，Docker 會從推論映像建立容器，並在背景執行。容器會執行`serve`指令碼，這會啟動您的 Web 伺服器以進行測試。

測試 ping HTTP 端點

當 SageMaker AI 執行您的容器時，它會定期 Ping 端點。當端點傳回狀態碼為 200 的 HTTP 回應時，它會向 SageMaker AI 發出訊號，表示容器已準備好進行推論。

執行下列命令來測試端點並包含回應標頭。

```
curl --include http://127.0.0.1:8080/ping
```

範例輸出如下列範例所示。

```
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
```

測試推論 HTTP 端點

當容器傳回 200 狀態碼表示已準備好時，SageMaker AI 會透過POST請求將推論資料傳遞至 `/invocations` HTTP 端點。

執行下列命令來測試推論端點。

```
curl \
  --request POST \
  --data "hello world" \
  http://127.0.0.1:8080/invocations
```

範例輸出如下列範例所示。

```
{"prediction": "a", "text": "hello world"}
```

當這兩個 HTTP 端點運作時，推論映像現在與 SageMaker AI 相容。

> **Note**
>
> 演算法產品的模型有兩種部署方式：即時和批次。對於這兩種部署，SageMaker AI 在執行 Docker 容器時使用相同的 HTTP 端點。

若要停止容器，請執行下列命令。

```
sudo docker container stop my-inference-container
```

在演算法產品的訓練和推論映像都準備好並進行測試後，請繼續 將映像上傳至 Amazon Elastic Container Registry。

將映像上傳至 Amazon Elastic Container Registry

建立推論和訓練映像之後，您可以將它們上傳至 Amazon Elastic Container Registry。Amazon ECR 是全受管的 Docker 登錄檔。Amazon SageMaker AI 從 Amazon ECR 提取映像，以建立模型套件以進行推論，或建立用於訓練任務的演算法。 AWS Marketplace 也會從 Amazon ECR 擷取這些映像以發佈模型套件和演算法產品。本主題提供將推論和訓練映像上傳至 Amazon ECR 的逐步解說

主題

- 我必須上傳哪些影像？
- 需要哪些 IAM 許可？
- 將您的 Docker 用戶端登入 AWS
- 建立儲存庫並上傳映像
- 掃描上傳的映像

我必須上傳哪些影像？

如果您要發佈模型套件，請僅上傳推論映像。如果您要發佈演算法，請上傳推論映像和訓練映像。如果推論和訓練映像已合併，則只能上傳合併映像一次。

需要哪些 IAM 許可？

下列步驟假設本機電腦具有賣方中 AWS Identity and Access Management (IAM) 角色或使用者的正確 AWS 登入資料 AWS 帳戶。角色或使用者必須同時為 AWS Marketplace 和 Amazon ECR 制定正確的政策。例如，您可以使用下列 AWS 受管政策：

- AWSMarketplaceSellerProductsFullAccess – 用於存取 AWS Marketplace
- AmazonEC2ContainerRegistryFullAccess – 用於存取 Amazon ECR

> **Note**
>
> 這些連結會帶您前往 AWS 受管政策參考。

將您的 Docker 用戶端登入 AWS

為您要從中發佈 AWS 區域 的 設定變數 （ 請參閱 支援 AWS 區域 發佈)。在此範例中，請使用美國東部 （ 俄亥俄 ） 區域。

```
region=us-east-2
```

執行下列命令，以您的 AWS 帳戶 ID 設定變數。此範例假設 current AWS Command Line Interface (AWS CLI) 憑證屬於賣方的 AWS 帳戶。

```
account=$(aws sts get-caller-identity --query Account --output text)
```

若要使用您區域的 AWS 帳戶 Amazon ECR Docker 登錄檔來驗證 Docker CLI 用戶端，請執行下列命令。

```
aws ecr get-login-password \
--region ${region} \
| sudo docker login \
--username AWS \
--password-stdin \
${account}.dkr.ecr.${region}.amazonaws.com
```

建立儲存庫並上傳映像

為上傳影像的標籤設定變數，並為上傳影像儲存庫的名稱設定另一個變數。

```
image=my-inference-image
repo=my-inference-image
```

> **Note**
>
> 在本指南先前建置推論和訓練影像的章節中，它們分別標記為 my-inference-image 和 my-training-image。在此範例中，建立推論映像，並將其上傳至同名的儲存庫。

執行下列命令，在 Amazon ECR 中建立映像儲存庫。

```
aws ecr --region ${region} create-repository --repository-name "${repo}"
```

Amazon ECR 儲存庫位置的完整名稱由下列部分組成： `<account-id>.dkr.ecr.<region>.amazonaws.com/<image-repository-name>`

若要將映像推送至儲存庫，您必須使用儲存庫位置的完整名稱來標記映像。

為映像儲存庫位置的完整名稱以及latest標籤設定變數。

```
fullname="${account}.dkr.ecr.${region}.amazonaws.com/${repo}:latest"
```

使用全名標記映像。

```
sudo docker tag ${image} ${fullname}
```

最後，將推論映像推送至 Amazon ECR 中的儲存庫。

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

上傳完成後，映像會出現在您發佈所在區域的 Amazon ECR 主控台儲存庫清單中。在上一個範例中，映像被推送到美國東部（俄亥俄）區域的儲存庫。

掃描上傳的映像

在 Amazon ECR 主控台中，選擇您要從中發佈 AWS 區域 的 ，然後開啟映像上傳到的儲存庫。選取您上傳的映像，然後開始掃描以檢查已知漏洞。 會在發佈前 AWS Marketplace 檢查 Amazon SageMaker AI 資源中所用容器映像的 Amazon ECR 掃描結果。您必須先修正具有嚴重或高嚴重性漏洞的容器映像，才能建立產品。

成功掃描映像之後，即可用來建立模型套件或演算法資源。

若您認為您的產品在掃描中發生錯誤，且結果為誤報，請聯絡AWS Marketplace 賣方營運團隊提供錯誤的相關資訊。

後續步驟

- 請參閱 中的大小限制 建立機器學習產品的需求和最佳實務
- 繼續前往 建立 Amazon SageMaker AI 資源

建立 Amazon SageMaker AI 資源

若要發佈模型套件或演算法產品，您必須在 Amazon SageMaker AI 中建立個別的模型套件資源或演算法資源。當您為 AWS Marketplace 產品建立資源時，必須透過驗證步驟進行認證。驗證步驟需要您提供資料來測試模型套件或演算法資源，才能發佈。下列各節說明如何建立 SageMaker AI 資源，可以是模型套件資源或演算法資源。這包括設定驗證規格，告知 SageMaker AI 如何執行驗證。

> **Note**
>
> 如果您尚未為產品建立映像並將其上傳到 Amazon Elastic Container Registry (Amazon ECR)，請參閱 將您的程式碼封裝到 中的機器學習產品映像中 AWS Marketplace和 將映像上傳至 Amazon Elastic Container Registry 以取得如何執行此操作的資訊。

主題

- 建立模型套件
- 建立您的演算法

建立模型套件

以下是為 建立模型套件的要求 AWS Marketplace：

- 儲存在Amazon ECR 中的推論映像
- （選用）模型成品，分別存放在 Amazon S3 中
- 用於推論的測試資料，存放在 Amazon Simple Storage Service 中

> **Note**
>
> 以下是關於建立模型套件產品。如需 SageMaker AI 中模型套件的詳細資訊，請參閱建立模型套件資源。

建立模型套件資源

下列程序會逐步引導您建立模型套件資源。

步驟 1：建立模型套件資源

1. 開啟 Amazon SageMaker AI 主控台。
2. 查看頁面右上角 AWS ，確保您位於要從中發佈的區域。如需發佈，請參閱 支援 AWS 區域 發佈一節。您在先前步驟中上傳至 Amazon ECR 的推論映像必須位於相同的區域。
3. 在左側導覽功能表中，選擇模型套件。
4. 選擇 Create model package (建立模型套件)。

建立套件之後，您需要設定推論套件的規格。

步驟 2：設定推論規格

1. 為您的模型套件提供名稱（例如 *my-model-package*）。
2. 針對推論影像的位置，輸入上傳至 Amazon ECR 的推論影像的 URI。您可以透過在 Amazon ECR 主控台中放置映像來擷取 URI。
3. 如果您的訓練模型成品與推論影像中的邏輯綁定在一起，請將模型資料成品的位置保留空白。否則，請指定模型成品壓縮檔案 (.tar.gz) 的完整 Amazon S3 位置。
4. 使用下拉式清單，為即時推論（也稱為端點) 和批次轉換任務選擇支援的推論映像執行個體類型。
5. 選擇下一步。

在建立和發佈模型套件之前，需要驗證以確保其如預期般運作。這需要您使用測試資料來執行批次轉換任務，以進行您提供的推論。驗證規格會告知 SageMaker AI 如何執行驗證。

步驟 3：設定驗證規格

1. 將 中的發佈此模型套件 AWS Marketplace設定為是。如果您將此設定為否，則無法稍後發佈此模型套件。選擇是可驗證 的模型套件 AWS Marketplace ，且需要驗證步驟。
2. 如果這是第一次完成此程序，請選擇為 IAM 角色建立新角色。 Amazon SageMaker AI 會在部署模型套件時使用此角色。這包括動作，例如從 Amazon ECR 提取影像，以及從 Amazon S3 提取成品。檢閱設定，然後選擇建立角色。在此建立角色會將 AmazonSageMakerFullAccess IAM 政策所述的許可授予您建立的角色。
3. 在驗證設定檔中編輯 JSON。如需允許值的詳細資訊，請參閱 TransformJobDefinition。

    1. `TransformInput.DataSource.S3Uri`：將 設定為存放推論測試資料的位置。
    2. `TransformInput.ContentType`：指定您的測試資料內容類型（例如 `application/json`、`image/png` 、 `text/plain`或任何其他值）。SageMaker AI 不會驗證實際的輸入資料。此值會傳遞至 `Content-type`標頭值中的容器 HTTP 端點。
    3. `TransformInput.CompressionType`：`None`如果 Amazon S3 中的推論測試資料未壓縮，請將 設定為 。
    4. `TransformInput.SplitType`：設定為 `None`以將 Amazon S3 中的每個物件整體傳遞以進行推論。
    5. `TransformOutput.S3OutputPath`：將 設定為存放推論輸出的位置。
    6. `TransformOutput.AssembleWith`：將 設定為 `None`，將每個推論輸出為 Amazon S3 中的個別物件。

4. 選擇 Create model package (建立模型套件)。

SageMaker AI 從 Amazon ECR 提取推論映像、將任何成品複製到推論容器，並使用測試資料執行批次轉換任務以進行推論。驗證成功後，狀態會變更為已完成。

> **Note**
>
> 驗證步驟不會使用您的測試資料評估模型的準確性。驗證步驟會檢查容器是否如預期執行和回應。

您已完成建立模型產品資源。繼續進行在 中列出您的產品 AWS Marketplace。

建立您的演算法

以下是為 建立演算法的要求 AWS Marketplace：

- 儲存在 Amazon ECR 中的推論映像
- 存放在 Amazon ECR 的訓練映像
- 您的訓練測試資料，存放在 Amazon S3 中
- 您的推論測試資料，存放在 Amazon S3 中

> **Note**
>
> 下列逐步解說會建立演算法產品。如需詳細資訊，請參閱建立演算法資源。

建立演算法資源

下列程序會逐步引導您在演算法套件中建立資源。

步驟 1：建立演算法資源

1. 開啟 Amazon SageMaker AI 主控台。
2. 查看頁面右上角 AWS ，確保您位於要從中發佈的區域（請參閱 支援 AWS 區域 發佈)。您在先前步驟中上傳至 Amazon ECR 的訓練和推論映像必須位於相同的區域。
3. 在左側導覽功能表中，選擇演算法。

4. 選擇建立演算法。

建立演算法套件之後，您必須設定模型訓練和調校的規格。

步驟 2：設定訓練和調校規格

1. 輸入演算法的名稱（例如 *my-algorithm*)。
2. 對於訓練映像，請貼上已上傳至 Amazon ECR 的訓練映像的完整 URI 位置。您可以透過在 Amazon ECR 主控台中放置映像來擷取 URI。
3. 使用下拉式方塊，選擇訓練映像支援的訓練執行個體類型。
4. 在頻道規格區段下，為您的演算法支援的每個輸入資料集新增頻道，最多 20 個頻道的輸入來源。如需詳細資訊，請參閱輸入資料組態。
5. 選擇下一步。
6. 如果您的演算法支援超參數和超參數調校，您必須指定調校參數。
7. 選擇下一步。

> **Note**
>
> 我們強烈建議您的演算法支援超參數調校，並讓適當的參數可調校。這可讓資料科學家調校模型，以取得最佳結果。

設定調校參數之後，如果有，您必須設定推論映像的規格。

步驟 3：設定推論影像規格

1. 針對推論影像的位置，貼上上傳至 Amazon ECR 的推論影像的 URI。您可以在 Amazon ECR 主控台中放置映像來擷取 URI。
2. 使用下拉式清單，為您的即時推論（也稱為端點) 和批次轉換任務的推論映像選擇支援的執行個體類型。
3. 選擇下一步。

在建立和發佈您的演算法之前，需要驗證以確保其如預期般運作。這需要您使用訓練的測試資料來執行訓練任務，並使用測試資料來執行批次轉換任務，以進行您提供的推論。驗證規格會告知 SageMaker AI 如何執行驗證。

步驟 4：設定驗證規格

1. 將 中的發佈此演算法 AWS Marketplace設定為是。如果您將此設定為否，您稍後就無法發佈此演算法。選擇是可驗證您的演算法， AWS Marketplace 且需要驗證規格。
2. 如果這是您第一次為 建立機器學習套件 AWS Marketplace，請選擇為 IAM 角色建立新角色。Amazon SageMaker AI 會在訓練您的演算法和部署後續模型套件時使用此角色。這包括從 Amazon ECR 提取影像、在 Amazon S3 中儲存成品，以及從 Amazon S3 複製訓練資料等動作。檢閱設定，然後選擇建立角色。在此建立角色會將 AmazonSageMakerFullAccess IAM 政策所述的許可授予您建立的角色。
3. 在訓練任務定義的驗證設定檔中編輯 JSON 檔案。如需允許值的詳細資訊，請參閱 TrainingJobDefinition。
    1. `InputDataConfig`：在此 JSON 陣列中，為您在訓練規格步驟中指定的每個頻道新增頻道物件。針對每個頻道，指定訓練的測試資料存放位置。
    2. `OutputDataConfig`：訓練完成後，訓練容器目錄路徑中的模型成品`/opt/ml/model/`會壓縮並複製到 Amazon S3。指定存放壓縮檔案 (.tar.gz) 的 Amazon S3 位置。
4. 在轉換任務定義的驗證設定檔中編輯 JSON 檔案。如需允許值的詳細資訊，請參閱 TransformJobDefinition。
    1. `TransformInput.DataSource.S3Uri`：將 設定為存放推論測試資料的位置。
    2. `TransformInput.ContentType`：指定您的測試資料內容類型。例如，`application/json`、`image/png`、 `text/plain`或任何其他值。Amazon SageMaker AI 不會驗證實際輸入資料。此值會傳遞至 `Content-type`標頭值中的容器 HTTP 端點。
    3. `TransformInput.CompressionType`：`None`如果 Amazon S3 中的推論測試資料未壓縮，請將 設定為 。
    4. `TransformInput.SplitType`：選擇您希望 S3 分割中物件的方式。例如， 會將 Amazon S3 中的每個物件整體`None`傳遞以進行推論。如需詳細資訊，請參閱《Amazon SageMaker AI API 參考》中的 SplitType。
    5. `TransformOutput.S3OutputPath`：將 設定為存放推論輸出的位置。
    6. `TransformOutput.AssembleWith`：將 設定為 `None`，將每個推論輸出為 Amazon S3 中的個別物件。
5. 選擇建立演算法套件。

SageMaker AI 從 Amazon ECR 提取訓練映像，使用您的資料執行測試訓練任務，並將模型成品存放在 Amazon S3 中。然後，它會從 Amazon ECR 提取推論映像、將成品從 Amazon S3 複製到推論容器中，並使用測試資料執行批次轉換任務以進行推論。驗證成功後，狀態會變更為已完成。

> **Note**
>
> 驗證步驟不會使用您的測試資料評估訓練或模型的準確性。驗證步驟會檢查容器是否如預期執行和回應。
>
> 驗證步驟只會驗證批次處理。您可以自行驗證即時處理是否適用於您的產品。

您已完成建立演算法產品資源。繼續進行在 中列出您的產品 AWS Marketplace。

在 中列出您的產品 AWS Marketplace

將程式碼封裝成模型套件映像或演算法映像、上傳映像並建立 Amazon SageMaker AI 資源後，您可以在 中發佈機器學習產品 AWS Marketplace。以下各節會逐步引導您完成發佈程序，其中包括建立您的產品清單、測試您的產品，以及簽署以進行發佈。發佈產品後，您可以請求變更以更新您的清單。如需詳細資訊，請參閱the section called “管理您的產品”。

主題

- 先決條件
- 步驟 1：建立新的清單
- 步驟 2：提供產品資訊
- 步驟 3：新增初始產品版本
- 步驟 4：設定定價模型
- 步驟 5：設定退款政策
- 步驟 6：設定 EULA
- 步驟 7：設定允許清單

先決條件

在 中發佈模型套件或演算法之前 AWS Marketplace，您必須具有下列項目：

- 已註冊為 AWS Marketplace 賣方 AWS 帳戶 的 。您可以在 中執行此操作AWS Marketplace 管理入口網站。

- 中設定頁面下的已完成賣方設定檔 AWS Marketplace 管理入口網站。
- 若要發佈付費產品，您必須完成稅務面試和銀行表單。這對於發佈免費產品並非必要。如需詳細資訊，請參閱賣方註冊程序。
- 您必須具有存取 AWS Marketplace 管理入口網站 和 Amazon SageMaker AI 的許可。如需詳細資訊，請參閱所需的 許可。

所需的 許可

若要發佈 Amazon SageMaker AI 產品，您必須指定與服務 AWS Marketplace 委託人具有信任關係的有效 IAM 角色 ARN。此外，您以 身分登入的 IAM 使用者或角色需要必要的許可。

設定登入許可

- 將下列許可新增至 IAM 角色：

  1. sagemaker：DescribeModelPackage — 用於列出模型套件
  2. sagemaker：DescribeAlgorithm — 用於列出演算法

     JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeModelPackage",
                "sagemaker:DescribeAlgorithm"
            ],
            "Resource": "*"
       }
    ]
}
```

設定 IAM 角色 AddVersion/Create 產品

1. 依照步驟建立具有自訂信任政策的角色。如需詳細資訊，請參閱使用自訂信任政策建立 IAM 角色（主控台）。
2. 針對自訂信任政策陳述式輸入下列內容：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": "assets.marketplace.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

3. 輸入下列許可政策：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeModelPackage",
                "sagemaker:DescribeAlgorithm"
            ],
            "Resource": "*"
       }
    ]
}
```

4. 請求時提供角色 ARN。角色應該遵循格式：`arn:aws:iam::<account-id>:role/<role-name>`。

如需所需 AWS Marketplace 許可或管理賣方帳戶的許可，請參閱AWS Marketplace 賣方的政策和許可。

必要的資產

在建立機器學習產品清單之前，請確定您擁有下列必要資產：

- Amazon Resource Name (ARN) — 在您從中發佈的 中提供模型套件或演算法資源 AWS 區域 的 ARN （請參閱 支援 AWS 區域 發佈)。
  - 模型套件的 ARN 具有此表單： `arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>`

    若要尋找您的模型套件 ARN，請參閱我的市集模型套件。
  - 演算法的 ARN 具有以下形式： `arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>`

    若要尋找您的演算法資源 ARN，請參閱我的演算法。
- the section called “用量資訊的需求” — 提供輸入、輸出和程式碼範例的詳細資訊。
- the section called “輸入和輸出的需求” — 提供檔案或文字。
- the section called “Jupyter 筆記本的需求” — 示範完整的產品使用方式。

步驟 1：建立新的清單

若要開始使用機器學習產品，您可以透過設定產品名稱、新增組織的選用資源標籤，以及產生產品 ID 來啟動列出程序。產品 ID 用於追蹤您產品的整個生命週期。

> **Note**
>
> 建立清單之前，請確定您擁有 中指定的必要資源建立機器學習產品的需求和最佳實務。

1. 登入您的賣方 AWS 帳戶 並前往 AWS Marketplace 管理入口網站。
2. 在頂端選單中，前往產品，然後選擇機器學習。
3. 選擇建立機器學習產品。
4. 在產品名稱下，輸入將向買方顯示在產品清單頁面頂端和搜尋結果中的唯一產品名稱。
5. （選用）在標籤下，輸入您要與產品建立關聯的任何標籤。如需詳細資訊，請參閱標記 AWS 資源。
6. 在產品 ID 和程式碼下，選擇產生產品 ID 和程式碼。
7. 選擇繼續精靈。您將開始在精靈中新增詳細產品資訊的程序。

步驟 2：提供產品資訊

在 中列出您的機器學習產品時 AWS Marketplace，提供完整且準確的產品資訊至關重要。使用精靈中的提供產品資訊步驟來擷取您產品的基本詳細資訊，例如產品類別和支援資訊。

1. 輸入產品的相關資訊。
2. 選擇下一步以移至精靈中的下一個步驟。

步驟 3：新增初始產品版本

此頁面會引導您新增產品的初始版本。您的產品在其整個生命週期中可能有多個版本，每個版本都由唯一的 SageMaker AI ARN 識別。

1. 在 Amazon Resource Name (ARNs下：

   a. 輸入模型或演算法 Amazon SageMaker AI ARN。

      - 範例模型套件 ARN： `arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>`

        若要尋找您的模型套件 ARN，請參閱我的市集模型套件。
      - 演算法 ARN 範例： `arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>`

        若要尋找您的演算法資源 ARN，請參閱我的演算法。

   b. 輸入 IAM 存取角色 ARN。

      IAM ARN 範例： `arn:aws:iam::<account-id>:role/<role-name>`
2. 在版本資訊下，輸入版本名稱和版本備註。
3. 在模型輸入詳細資訊下，輸入模型輸入的摘要，並提供即時和批次任務輸入的範例輸入資料。或者，您可以提供任何輸入限制。
4. （選用）在輸入參數下，提供產品支援的每個輸入參數的詳細資訊。您可以提供參數名稱、描述、限制條件，並指定 參數為必要或選用。您最多可以提供 24 個輸入參數。
5. （選用）在自訂屬性下，提供產品支援的任何自訂調用參數。對於每個屬性，您可以提供名稱、描述、限制條件，並指定屬性為必要或選用。
6. 在模型輸出詳細資訊下，輸入模型輸出的摘要，並提供即時和批次任務輸出的範例輸出資料。或者，您可以提供任何輸出限制。

7. （選用）在輸出參數下，提供產品支援的每個輸出參數的詳細資訊。您可以提供參數名稱、描述、限制條件，並指定 參數為必要或選用。您最多可以提供 24 個輸出參數。
8. 在使用說明下，提供有效使用模型的明確指示，例如最佳實務、如何處理常見的邊緣案例或效能最佳化建議。
9. 在 Git 儲存庫和筆記本連結下，提供範例筆記本和 Git 儲存庫的連結。範例筆記本應包含如何叫用您的模型。您的 Git 儲存庫應包含筆記本、資料檔案和其他開發人員工具。
10. 在建議的執行個體類型下，為您的產品選取建議的執行個體類型。

    對於模型套件，您將為批次轉換和即時推論選取建議的執行個體類型。

    對於演算法套件，您將為訓練任務選取建議的執行個體類型。

    > **Note**
    >
    > 可供選取的執行個體類型僅限於模型或演算法套件支援的執行個體類型。當您最初在 Amazon SageMaker AI 中建立資源時，會決定這些支援的執行個體類型。這可確保您的產品僅與可有效執行機器學習解決方案的硬體組態相關聯。

11. 選擇下一步以移至精靈中的下一個步驟。

步驟 4：設定定價模型

設定產品的定價模式時，您可以免費提供產品，或實作以用量為基礎的定價。發佈產品後，無法變更您的定價模型。

1. 選擇定價模型。批次轉換和演算法訓練產品只能免費或按小時計費。
   - 如果您選擇免費提供產品，請選擇下一步並繼續精靈。
   - 如果您選擇用量定價，請繼續這些步驟。
2. 如果您選擇根據用量收費，您可以輸入用量成本。您可以選擇輸入適用於所有執行個體類型的價格，或輸入每個執行個體類型的價格，以取得更精細的定價。
3. 選取是，如果您想要提供產品的免費試用，請提供免費試用。
4. 選擇下一步以移至精靈中的下一個步驟。

步驟 5：設定退款政策

雖然您不需要提供退款，但您必須向 提交官方退款政策 AWS Marketplace。

1. 輸入退款政策。
2. 選擇下一步以移至精靈中的下一個步驟。

步驟 6：設定 EULA

在此步驟中，您將選擇法律協議，以控管客戶如何使用您的產品。您可以選取 AWS標準合約條款，或上傳自己的自訂最終使用者授權合約 (EULA)。

1. 選取標準合約或提供自訂最終使用者授權合約。
2. 選擇下一步以移至精靈中的下一個步驟。

步驟 7：設定允許清單

在提交您的產品之前，您需要指定哪些 AWS 帳戶 可以存取它。此選用步驟可控制產品的初始可見性，限制存取您自己的 帳戶 AWS 帳戶 ，以及您新增至允許清單的任何特別授權。

1. 輸入您要存取產品的 AWS 帳戶 IDs。
2. 選擇提交以提交您的產品。

   您的產品將具有有限可見性狀態，而且只有 AWS 帳戶 建立產品和其他允許清單的 才能看見 AWS 帳戶。

   如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

您可以在產品清單的可見性有限時檢視和測試產品清單。當您準備好變更產品的可見性時，請參閱 the section called “”。

管理您的機器學習產品

在 AWS Marketplace 管理入口網站中，選擇請求變更以修改產品或版本 AWS Marketplace。當您提交變更時，系統會處理它們。處理時間會因分鐘到天而異，取決於修改類型。您可以在 AWS Marketplace 管理入口網站中監控變更的狀態。

您可以透過自助式服務 AWS Marketplace 在 中修改 ML 產品：

- 更新產品資訊
- 更新產品可見性

- 更新允許清單
- 管理產品版本
- 更新產品定價
- 更新您的退款政策
- 更新您的 EULA
- 移除產品

> **Note**
>
> 除了透過 AWS Marketplace 管理入口網站進行變更之外，您也可以使用 進行變更AWS Marketplace Catalog API。

更新產品資訊

建立機器學習 (ML) 產品後，您可以在 中修改特定產品資訊 AWS Marketplace，例如描述、重點、標題、SKU、類別和關鍵字。

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的目標產品。
3. 選擇請求變更，然後選取更新產品資訊。
4. 視需要更新欄位。

   > **Note**
   >
   > 如需標誌規格，請參閱 the section called “公司和產品標誌要求”。

5. 選擇提交。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

更新產品可見性

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。

3. 選擇請求變更，選取更新產品可見性，然後選取公開或限制。
4. 檢閱您的變更，然後選擇提交。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

更新允許清單

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇請求變更，然後選取更新允許清單。
4. 修改您需要變更的資訊，然後選擇提交。如需詳細資訊，請參閱the section called “步驟 7：設定允許清單”。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

管理產品版本

身為賣方，您可以透過更新現有版本資訊、新增新版本或移除不再支援的版本， AWS Marketplace 在中管理您的產品版本。每個版本都有唯一的 SageMaker AI ARN 和相關資訊，供買方用來評估和部署您的產品。

> **Note**
>
> 新增版本之前，請先建立產品 ID 並建立定價。如需詳細資訊，請參閱the section called “步驟 1：建立新的清單”。

更新版本資訊

建立版本後，您可以修改其相關資訊，例如版本備註、使用說明和執行個體建議。

> **Note**
>
> 版本名稱和 ARNs 無法修改。這些變更需要建立新的版本。

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇請求變更，然後選取更新版本資訊。
4. 選取您要更新的版本。
5. 選擇編輯版本。
6. 修改必要欄位，然後選擇下一步。
7. 輸入您的定價資訊，然後選擇提交。如需詳細資訊，請參閱the section called “步驟 4：設定定價模型”。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

新增新版本

您可以新增產品的新版本，以引進功能、更新或改進，同時保持對舊版的存取。

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇版本，然後選擇新增版本。
4. 依照中的步驟輸入新版本的資訊the section called “步驟 3：新增初始產品版本”。
5. 輸入您的定價資訊，然後選擇提交。如需詳細資訊，請參閱the section called “步驟 4：設定定價模型”。

當您成功新增新版本時，買方會收到電子郵件通知，告知有新版本可用。

限制版本

當版本過期或您想要停止其可用性時，您可以限制買方存取該版本，同時維持對其他版本的存取。

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇版本，然後選擇限制版本。

> **Note**
>
> 您必須至少有一個可用的版本。

4. 選擇提交。

當您成功限制版本時，買方會收到該版本受到限制的電子郵件通知。

更新產品定價

您可以修改機器學習產品的費率和免費試用期 AWS Marketplace，但定價模型本身無法變更。請注意，對於付費模型，價格上漲會在 90 天的通知期間之後，於下個月的第一天生效。無法在此通知期間進行其他價格變更。

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇請求變更，選取更新公開優惠，然後選取編輯優惠資訊。
4. 修改您需要變更的資訊，然後選擇提交。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

更新您的退款政策

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇請求變更，選取更新公開優惠，然後選取更新退款政策。
4. 修改您需要變更的資訊，然後選擇提交。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

更新您的 EULA

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。

2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇請求變更，選取更新公開優惠，然後選取更新 EULA。
4. 修改您需要變更的資訊，然後選擇提交。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

移除產品

您可以從中移除（隨機）已發佈的產品 AWS Marketplace。一旦移除，新客戶就無法訂閱，但您必須支援現有客戶至少 90 天。

以下是從中移除產品的條件 AWS Marketplace：

- 產品將從 AWS Marketplace 搜尋和探索工具中移除。
- 訂閱功能將會停用。
- 產品詳細資訊頁面仍可透過直接 URL 存取。
- 目前的訂閱者會保留存取權，直到取消訂閱為止。
- AWS Marketplace 通知目前的買方有關移除。

若要移除您的機器學習產品：

1. 在 AWS Marketplace 管理入口網站中登入您的賣方帳戶。
2. 前往機器學習產品頁面，然後選取您的產品。
3. 選擇請求變更，選取更新產品可見性，然後選取限制。
4. （選用）輸入替代產品 ID。
5. 檢閱變更，然後選擇提交。

您可以從機器學習產品頁面的請求索引標籤監控您的請求。如需狀態的詳細資訊，請參閱 the section called “產品狀態”。

移除後，產品會出現在目前產品清單中，您只能下載產品試算表。如果您對移除產品有任何疑問，請聯絡AWS Marketplace 賣方營運團隊。

為機器學習產品建立私有優惠

您可以為機器學習產品直接交涉並提供私有優惠給客戶。如需私有優惠的詳細資訊，請參閱 準備您的私有優惠。

事前準備：

- 您必須擁有其中的付費清單 AWS Marketplace。
- 您必須擁有 AWS Marketplace Management Portal (AMMP) 的存取權。

若要為機器學習產品建立私有優惠：

1. 登入 AWS Marketplace 管理入口網站。
2. 選擇優惠，然後選擇建立私有優惠
3. 在建立私有優惠頁面上，選取要為其建立私有優惠的產品。您只能為可用的產品建立優惠。
4. 在優惠詳細資訊頁面上：
   a. 輸入優惠名稱和描述。
   b. 選取續約選項。
   c. 設定優惠過期日期。優惠會在設定日期 UTC 的 23：59：59 過期。
5. 選擇 Next (下一步) 兩次。
6. 在設定優惠定價和持續時間頁面上，指定：
   - 定價選項

     (如需詳細資訊，請參閱 the section called “機器學習產品定價”。)
   - 合約持續時間
   - 優惠貨幣
   - 產品維度

   > **Note**
   > 如需分期付款計劃的詳細資訊，請參閱 私有優惠分期方案。
7. 選擇下一步。
8. 在新增買方頁面上，輸入買方 AWS 帳戶 IDs。然後選擇下一步。

> **Important**
>
> 若要讓連結帳戶受益於私有優惠：
>
> - 包含付款人 AWS 帳戶 ID。
> - 付款人帳戶必須先接受私有優惠的每小時條款。
> - 付款人帳戶接受之後，連結的帳戶就可以接受私有優惠。

9. 在設定法律條款和優惠文件頁面上，新增任何自訂條款，然後選擇下一步。

> **Note**
>
> 您最多可以新增五個檔案（法律條款、工作說明、物料清單、定價表或附錄）。系統會將這些合併為一份文件。

10. 在檢閱和建立頁面上，驗證優惠詳細資訊，然後選擇建立優惠。
11. 優惠顯示在管理私有優惠頁面上後，開啟動作選單，選擇複製優惠 URL，然後透過電子郵件傳送給買方。

> **Note**
>
> 優惠可能需要一些時間才能發佈。您可以在管理私有優惠頁面上編輯優惠，直到買方接受為止。

建立機器學習產品的需求和最佳實務

您的買方務必能輕鬆測試模型套件和演算法產品。下列各節說明 ML 產品的最佳實務。如需需求和建議的完整摘要，請參閱 ML 產品清單的需求和建議摘要。

> **Note**
>
> 如果您的已發佈產品不符合這些要求， AWS Marketplace 代表可能會與您聯絡，協助您滿足這些要求。

主題

- ML 產品的一般最佳實務
- 用量資訊的需求
- 輸入和輸出的需求
- Jupyter 筆記本的需求
- ML 產品清單的需求和建議摘要

ML 產品的一般最佳實務

提供機器學習產品的下列資訊：

- 如需產品說明，請包含下列項目：
  - 模型的功能
  - 目標客戶是誰
  - 最重要的使用案例是什麼
  - 模型的訓練方式或已使用的資料量
  - 什麼是效能指標，以及使用的驗證資料
  - 如果是醫療，無論您的模型是否用於診斷用途
- 根據預設，機器學習產品會設定為具有公有可見性。不過，您可以建立可見性有限的產品。如需詳細資訊，請參閱the section called “步驟 7：設定允許清單”。
- （選用）針對付費產品，提供 14–30 天的免費試用，讓客戶試用您的產品。如需詳細資訊，請參閱的機器學習產品定價 AWS Marketplace。

用量資訊的需求

描述您產品的預期輸入和輸出（使用範例）的明確用量資訊對於推動正面買方體驗至關重要。

對於您新增至產品清單的每個新版本資源，您必須提供用量資訊。

若要編輯特定版本的現有用量資訊，請參閱 the section called “更新版本資訊”。

輸入和輸出的需求

使用範例清楚說明支援的輸入參數和傳回的輸出參數，對於協助您的買方了解和使用您的產品至關重要。此了解可協助您的買方對輸入資料執行任何必要的轉換，以取得最佳推論結果。

將 Amazon SageMaker AI 資源新增至產品清單時，系統會提示您輸入下列項目。

推論輸入和輸出

對於推論輸入，提供產品對即時端點和批次轉換任務預期之輸入資料的描述。包含用於任何必要預先處理資料的程式碼片段。如果適用，請包含限制。提供 GitHub 上託管的輸入範例。

針對推論輸出，提供產品針對即時端點和批次轉換任務傳回的輸出資料描述。如果適用，請包含限制。提供 GitHub 上託管的輸出範例。

針對範例，請提供可搭配您產品的輸入檔案。如果您的模型執行多類別分類，請為每個類別提供至少一個範例輸入檔案。

訓練輸入

在訓練模型的資訊區段中，提供輸入資料格式和程式碼片段，以進行任何必要的資料預先處理。如果適用，請包含值和限制的描述。提供 GitHub 上託管的輸入範例。

說明買方可以提供的選用和強制性功能，並指定是否支援PIPE輸入模式。如果支援分散式訓練（使用超過 1 個 CPU/GPU 執行個體進行訓練），請指定此選項。針對調校，請列出建議的超參數。

Jupyter 筆記本的需求

將 SageMaker AI 資源新增至產品清單時，請提供 GitHub 上託管的範例 Jupyter 筆記本的連結，示範完整的工作流程，而不會要求買方上傳或尋找任何資料。

使用 AWS SDK for Python (Boto)。精心開發的範例筆記本可讓買方更輕鬆地嘗試和使用您的清單。

對於模型套件產品，您的範例筆記本示範準備輸入資料、建立用於即時推論的端點，以及批次轉換任務的效能。如需詳細資訊，請參閱 GitHub 上的模型套件清單和範例筆記本。如需範例筆記本，請參閱 auto_management。筆記本適用於所有 AWS 區域，無需輸入任何參數，也不需要買方尋找範例資料。

> **Note**
>
> 未顯示多個可能輸入和資料預先處理步驟的未開發範例 Jupyter 筆記本，可能會讓買方難以完全了解產品的價值主張。

對於演算法產品，範例筆記本示範完整的訓練、調校、建立模型、建立用於即時推論的端點，以及批次轉換任務的效能。如需詳細資訊，請參閱 GitHub 上的演算法清單和範例筆記本。如需範例筆記本，請參閱 GitHub 上的 amazon_demo_product 和 automl。這些範例筆記本適用於所有區域，無需輸入任何參數，也不需要買方尋找範例資料。

> **Note**
>
> 缺少範例訓練資料可能會使您的買方無法成功執行 Jupyter 筆記本。未開發的範例筆記本可能會讓您的買方無法使用您的產品並阻礙採用。

ML 產品清單的需求和建議摘要

下表提供機器學習產品清單頁面的需求和建議摘要。

詳細資訊	對於模型套件清單	對於演算法清單
Product descriptions		
Explain in detail what the product does for supported content types (for example, “detects X in images").	Required	Required
Provide compelling and differentiating information about the product (avoid adjectives like "best" or unsubstantiated claims).	Recommended	Recommended
List most important use case(s) for this product.	Required	Required
Describe the data (source and size) it was trained on and list any known limitations.	Required	Not applicable
Describe the core framework that the model was built on.	Recommended	Recommended
Summarize model performan ce metric on validation data (for example, "XX.YY percent	Required	Not applicable

詳細資訊	對於模型套件清單	對於演算法清單
accuracy benchmarked using the Z dataset").		
Summarize model latency and/or throughput metrics on recommended instance type.	Required	Not applicable
Describe the algorithm category. For example, “This decision forest regressio n algorithm is based on an ensemble of tree-structured classifiers that are built using the general technique of bootstrap aggregation and a random choice of features.”	Not applicable	Required
Usage information		
For inference, provide a description of the expected input format for both the real-time endpoint and batch transform job. Include limitatio ns, if applicable. See 輸入和輸出的需求.	Required	Required
For inference, provide input samples for both the real-time endpoint and batch transform job. Samples must be hosted on GitHub. See 輸入和輸出的需求.	Required	Required

詳細資訊	對於模型套件清單	對於演算法清單
For inference, provide the name and description of each input parameter. Provide details about the its limitations and specify if it is required or optional.	Recommended	Recommended
For inference, provide details about the output data your product returns for both the real-time endpoint and batch transform job. Include any limitations, if applicable. See 輸入和輸出的需求.	Required	Required
For inference, provide output samples for both the real-time endpoint and batch transform job. Samples must be hosted on GitHub. See 輸入和輸出的需求.	Required	Required
For inference, provide an example of using an endpoint or batch transform job. Include a code example using the AWS Command Line Interface (AWS CLI) commands or using an AWS SDK.	Required	Required
For inference, provide the name and description of each output parameter. Specify if it is always returned.	Recommended	Recommended

詳細資訊	對於模型套件清單	對於演算法清單
For training, provide details about necessary information to train the model such as minimum rows of data required. See 輸入和輸出的需求.	Not applicable	Required
For training, provide input samples hosted on GitHub. See 輸入和輸出的需求.	Not applicable	Required
For training, provide an example of performing training jobs. Describe the supported hyperparameters, their ranges, and their overall impact. Specify if the algorithm supports hyperparameter tuning, distributed training, or GPU instances. Include code example such as AWS CLI commands or using an AWS SDK, for example.	Not applicable	Required
Provide a Jupyter notebook hosted on GitHub demonstrating complete use of your product. See Jupyter 筆記本的需求.	Required	Required
Provide technical information related to the usage of the product, including user manuals and sample data.	Recommended	Recommended

針對機器學習產品的問題進行故障診斷

本節提供說明，說明您在發佈機器學習產品的過程中可能遇到的一些常見錯誤。如果您的問題未列出，請聯絡AWS Marketplace 賣方營運團隊。

一般：當我在 中新增模型套件或演算法的 Amazon Resource Name (ARN) 時，收到 400 錯誤 AWS Marketplace 管理入口網站

常見原因

在 SageMaker AI 中建立機器學習產品時，您未選擇在其中發佈產品 AWS Marketplace。

Resolution

如果您使用 Amazon SageMaker AI 主控台來建立資源，則必須在發佈此模型套件 AWS Marketplace的程序最後一頁選擇是，或在其中發佈此演算法 AWS Marketplace時選擇是。您無法選擇否，也無法稍後發佈。選取是不會發佈模型套件或演算法。不過，它會在建立模型套件或演算法資源時驗證，這是在 中使用的必要項目 AWS Marketplace。

如果您使用 AWS SDK 來建立模型套件或建立演算法，請確定 參數`CertifyForMarketplace`設定為 `true`。

重新建立已驗證和已驗證的模型套件或演算法資源後，請在 中新增新的 ARN AWS Marketplace 管理入口網站。

一般：當我在 中新增模型套件或演算法的 ARN 時，收到 404 錯誤 AWS Marketplace 管理入口網站

常見原因

發生此錯誤的原因有幾個：

- ARN 可能無效。
- 模型套件或演算法資源未在 AWS 帳戶 與賣方帳戶相同的 中建立。
- 您用於發佈的使用者或角色沒有存取模型套件或演算法資源的正確 IAM 許可。

Resolution

1. 檢查 ARN 以確保其為正確的 ARN，且為預期的格式：

對於模型套件，ARNs看起來應該類似於 `arn:aws:sagemaker:us-east-2:000123456789:model-package/my-model-package-name`。

對於演算法，ARNs 看起來應該類似於 `arn:aws:sagemaker:us-east-2:000123456789:algorithm/my-algorithm`。

2. 確定要發佈的所有資源和資產都位於您要發佈的賣方帳戶中。
3. 確保您的使用者或角色具有下列許可：

   對於模型套件，必須允許模型套件資源`sagemaker:DescribeModelPackage`上的 動作。

   對於演算法，必須允許演算法資源`sagemaker:DescribeAlgorithm`上的 動作。

Amazon SageMaker AI：當我建立模型套件或演算法資源時，收到「用戶端錯誤：登錄檔存取遭拒」失敗訊息

常見原因

當用來建立模型套件或演算法的映像存放在屬於另一個模型套件的 Amazon ECRrepository 中時，可能會發生此錯誤 AWS 帳戶。模型套件或演算法驗證不支援跨帳戶映像。

Resolution

將映像複製到您用來發佈之 擁有 AWS 帳戶 的 Amazon ECR 儲存庫。然後，繼續使用新的映像位置建立資源。

Amazon SageMaker AI：我收到「未開始」和「用戶端錯誤：無掃描排程...」 當我建立模型套件或演算法資源時失敗訊息

常見原因

當 SageMaker AI 無法開始掃描存放在 Amazon ECR 中的 Docker 容器映像時，可能會發生此錯誤。

Resolution

如果發生這種情況，請開啟 Amazon ECR 主控台，尋找映像上傳到的儲存庫，選擇映像，然後選擇掃描。

中的 SaaS 型產品 AWS Marketplace

透過軟體即服務 (SaaS) 產品，您可以部署託管在基礎設施上的 AWS 軟體，並授予 AWS Marketplace 買方存取您 AWS 環境中的軟體。您負責管理軟體中的客戶存取、帳戶建立、資源佈建和帳戶管理。建立初始 SaaS 產品後，您可以提交變更請求，並在其中設定其屬性 AWS Marketplace，包括您的產品描述、依國家/地區的可用性、定價等。您也可以設定 SaaS 訂閱和合約整合。下列主題為您提供開始使用所需的資訊。

如需整合 APIs與 Amazon API Gateway 的相關資訊，請參閱《Amazon API Gateway 開發人員指南》中的透過 銷售 API Gateway APIs AWS Marketplace。 Amazon API Gateway

如需 SaaS 產品的協助，請聯絡AWS Marketplace 賣方營運團隊。

主題

- 上的 SaaS 產品入門 AWS Marketplace
- 規劃您的 SaaS 產品
- 的 SaaS 產品指導方針 AWS Marketplace
- 中的 SaaS 產品定價 AWS Marketplace
- 在 中建立 SaaS 免費試用優惠 AWS Marketplace
- 透過 將客戶加入您的 SaaS 產品 AWS Marketplace
- SaaS 產品的 Amazon SNS 通知
- 存取 AWS Marketplace 計量和權利服務 APIs
- 上的 SaaS 產品報告 AWS Marketplace
- SaaS 產品整合的程式碼範例
- 使用 透過 Amazon VPC 交付您的產品 AWS PrivateLink

上的 SaaS 產品入門 AWS Marketplace

身為 AWS Marketplace 賣方，您可以將軟體新增為服務 (SaaS) 產品 AWS Marketplace。這包括根據 SaaS 產品的帳單模型，整合您的 SaaS 產品與適當的 AWS Marketplace API 操作。本主題提供建立和設定 SaaS 產品的程序概觀，從您開始使用所需的必要先決條件開始。

先決條件

開始之前，您必須完成下列先決條件：

1. 存取和使用 AWS Marketplace 管理入口網站。這是您用來註冊為賣方和管理您銷售產品的工具 AWS Marketplace。如需詳細資訊，請參閱AWS Marketplace 管理入口網站。
2. 註冊為賣方，並提交您的稅務和銀行資訊。如需詳細資訊，請參閱註冊為 AWS Marketplace 賣方。
3. 規劃您將如何建立和整合 SaaS 產品 AWS Marketplace。如需詳細資訊，請參閱規劃您的 SaaS 產品。

後續步驟

完成必要的先決條件後，您可以建立和設定 SaaS 產品。

主題

- 中的 SaaS 產品生命週期 AWS Marketplace
- 在 中建立 SaaS 產品 AWS Marketplace
- 在 上建立初始 SaaS 產品頁面 AWS Marketplace
- 在 中設定 SaaS 產品設定 AWS Marketplace
- 將您的 SaaS 訂閱產品與 整合 AWS Marketplace
- 將您的 SaaS 合約產品與 整合 AWS Marketplace
- 將您的 SaaS 合約與pay-as-you-go產品與 整合 AWS Marketplace
- 部署無伺服器 SaaS 整合

中的 SaaS 產品生命週期 AWS Marketplace

當您在 中建立軟體即服務 (SaaS) 產品時 AWS Marketplace，最初發佈時可見性有限，因此只有您的帳戶可以存取它。當您準備好時，您可以將其發佈至 AWS Marketplace 目錄，以允許買方訂閱和購買您的產品。下列主題提供有關 SaaS 產品生命週期的資訊。如需建立 SaaS 產品的詳細資訊，請參閱在 中建立 SaaS 產品 AWS Marketplace、

在 SaaS 產品頁面上，您可以檢視您的產品清單。根據產品生命週期中的階段，產品將具有下列其中一種狀態：

- 預備：您仍在為其新增資訊的不完整產品。在第一次儲存並退出自助式體驗時，成功的變更請求會建立未發佈的產品，其中包含您提交之已完成步驟中的資訊。在此狀態中，您可以繼續新增資訊至產品，或透過變更請求變更已提交的詳細資訊。

- 有限 – 產品提交至系統後即完成，並通過系統中的所有驗證。然後，產品會釋出為有限狀態。此時，產品有一個詳細資訊頁面，只有您的帳戶和您已允許列出的任何人才能存取。您可以透過詳細資訊頁面來測試您的產品。如需詳細資訊或協助，請聯絡AWS Marketplace 賣方營運團隊。
- 公開 – 當您準備好發佈產品，以便買方可以檢視和訂閱產品時，您可以使用更新可見性變更請求。此請求會啟動工作流程，讓 AWS Marketplace 賣方營運團隊根據 AWS 政策來檢閱和稽核您的產品。核准產品並處理變更請求後，產品的狀態會從限制為公開。如需 AWS 指導方針的相關資訊，請參閱 SaaS 產品指導方針。
- 受限 – 如果您想要停止新使用者訂閱您的產品，您可以使用更新可見性變更請求來限制產品。受限狀態表示現有使用者可以繼續使用產品。不過，大眾將無法再看到該產品，也無法供新使用者使用。

您可以在預備狀態、有限狀態和公開狀態更新您的產品。如需詳細資訊，請參閱更新產品資訊。

在 中建立 SaaS 產品 AWS Marketplace

身為 AWS Marketplace 賣方，您可以將軟體新增為服務 (SaaS) 產品 AWS Marketplace。這包括根據您的帳單模型建立 SaaS 產品，並將其與適當的 AWS Marketplace API 操作整合。若要在 中將軟體做為 SaaS 產品銷售 AWS Marketplace，請遵循下列廣泛步驟：

- 在 中建立 SaaS 產品 AWS Marketplace。
- 將 SaaS 訂閱、合約或合約與pay-as-you-go產品整合 AWS Marketplace。
- 測試訂閱、合約或合約與 pay-as-you-go 產品的整合 AWS Marketplace。
- 提交您的產品以啟動。

下列程序說明如何在 中建立 SaaS 產品 AWS Marketplace。

建立 SaaS 產品

建立 SaaS 產品

1. 決定列出 SaaS 產品

   擁有您想要銷售的 SaaS 產品 AWS Marketplace。檢閱並了解如何 規劃您的 SaaS 產品。
2. 決定定價和優惠類型

   SaaS 產品有三種優惠類型：訂閱、合約和具有隨需pay-as-you-go的合約。您選擇的優惠類型會影響您整合 SaaS 產品的方式 AWS Marketplace。如需詳細資訊，請參閱規劃您的定價。

3. 收集資產

   收集提交產品所需的資產。資產包括：

   - 產品標誌 URL – 可公開存取的 Amazon S3 URL，其中包含您所提供產品的標誌的清晰影像。
   - 最終使用者授權合約 (EULA) URL – 您的產品必須具有提供 PDF 檔案的 EULA。您必須提供 Amazon S3 儲存貯體的連結，客戶可以在產品 AWS Marketplace 頁面上檢閱 EULA。
   - 產品註冊 URL – 這是在 中成功訂閱您的產品後，買方重新導向的 URL AWS Marketplace。
   - 您產品的中繼資料 – 您可以在 的產品建立精靈中提供中繼資料 AWS Marketplace 管理入口網站。
   - 您產品的支援資訊 – 此資訊包含您產品支援管道的電子郵件地址和 URLs。

4. 提交您的產品以進行整合

   使用您的賣方帳戶和管理 AWS Marketplace 入口網站來 在 上建立初始 SaaS 產品頁面 AWS Marketplace。 AWS Marketplace 會將您的產品發佈為有限產品，這表示它只能用於整合和測試。您可以在產品概觀頁面上使用您的產品程式碼和 Amazon Simple Notification Service (SNS) 主題。

   > **Note**
   >
   > 您的產品必須維持較低價格，您和 AWS Marketplace 賣方營運團隊才能測試您的產品，而不會產生高昂的成本。當您請求產品的公開可見性時，我們會要求您提供產品的實際價格。

5. 與 整合 AWS Marketplace

   您的產品必須支援客戶加入和使用您的產品，包括在授予他們存取權之前驗證其訂閱，在某些情況下，還必須測量其用量。與 的整合 AWS Marketplace 方式取決於您用於產品的優惠類型。如需根據優惠類型整合的詳細資訊，請參閱下列主題：

   - 訂閱整合
   - 合約整合
   - 與隨需pay-as-you-go整合的合約

   將產品與 整合的最後一步 AWS Marketplace 是進行測試，以確保整合正常運作。

6. 產品測試和合約取消

完成整合程序後，我們建議您訂閱自己的產品，以評估並確認客戶體驗。建立測試訂閱可讓您：

- 檢閱產品資訊
- 檢查可用的購買選項
- 處理測試付款
- 驗證您自己的產品網站連結

> **Important**
>
> 先取消您的測試訂閱，再將產品的可見性從限制變更為公有。如果您需要協助，請提交支援票證，以聯絡 Marketplace Seller Operations 團隊。

7. 提交您的產品以啟動

驗證整合並準備好讓產品上線後，請選擇更新可見性。 AWS Marketplace 賣方營運團隊會檢閱您的產品並更新價格，之後可見性才能更新為公開。

> **Note**
>
> AWS Marketplace Seller Operations 使用手動程序來驗證和更新 SaaS 產品。此程序需要 7 到 10 個工作天才能將可見性更新為公有，如果團隊發現錯誤，則會需要更長的時間。如需計時的詳細資訊，請參閱本指南中的計時和期望。

在 上建立初始 SaaS 產品頁面 AWS Marketplace

您可以使用軟體即服務 (SaaS) 應用程式中繼資料，使用 在 AWS Marketplace 目錄中建立初始 SaaS 產品頁面 AWS Marketplace 管理入口網站。然後，您也可以新增產品資訊、產品部署詳細資訊和公開優惠詳細資訊。或者，您可以將帳戶新增至允許清單以測試產品。如需詳細資訊，請參閱下列程序。

建立初始 SaaS 產品頁面

1. 登入 AWS Marketplace 管理入口網站。
2. 選擇建立 SaaS 產品，然後選擇 SaaS 產品。
3. 產生 SaaS 產品 ID 和程式碼。您也可以新增選用標籤，以支援標籤型授權。

> **Note**
>
> 如需標籤型授權的相關資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的使用標籤控制對 AWS 資源的存取。

4. 使用自助式體驗來建立 AWS Marketplace 清單。新增產品資訊、產品部署詳細資訊和公開優惠詳細資訊。您也可以選擇將帳戶新增至允許清單，以測試產品。

> **Note**
>
> 如果您需要在完成步驟之前結束工作階段，請選擇儲存並結束選項，將目前的選擇儲存到預備區域。此選項會建立請求，以驗證您提供的資訊。驗證您的請求時，您無法編輯產品。如果您的請求成功，您可以選擇繼續建立產品，以繼續建立您的產品。
> 如果您的請求不成功，是因為驗證錯誤，這會顯示在產品請求日誌中。選取請求以檢視錯誤，然後在動作下選擇複製到新的 以更正錯誤並重新提交請求。若要更新先前的步驟，請開啟產品詳細資訊頁面並提交變更請求。

> **Note**
>
> 在測試期間，您的價格預設為每個維度 0.001 美元。此價格可讓您在有限狀態下測試產品，而不會產生大筆帳單。您會在公開產品時提供實際價格。

5. 選擇提交。然後， AWS Marketplace 驗證資訊。如果驗證成功， 會以有限狀態 AWS Marketplace 發行產品。驗證成功後，您可以預覽、整合和測試您的產品。

> **Note**
>
> 驗證進行中時，您無法編輯產品。當您的產品最初發佈時，只有 AWS 帳戶 用於建立產品和 AWS Marketplace 賣方營運團隊測試帳戶的 才能存取它。如果您從 SaaS 產品頁面檢視產品，您可以選擇在 上檢視 AWS Marketplace，以檢視 AWS Marketplace 買方將出現在 中的產品詳細資訊。其他 AWS Marketplace 使用者看不到此詳細資訊清單。

在 中設定 SaaS 產品設定 AWS Marketplace

在 中建立軟體即服務 (SaaS) 產品之後 AWS Marketplace，您可以修改許多產品設定。下列各節說明如何提交變更請求和修改產品設定，例如更新定價詳細資訊、產品可見性和其他設定。

主題

- 管理變更請求
- 更新產品資訊
- 更新架構詳細資訊
- 更新 AWS 帳戶 IDs的允許清單
- 更新產品可見性
- 更新定價條款
- 新增定價維度
- 更新定價維度
- 限制定價維度
- 決定買方如何存取您的產品
- 設定快速啟動
- 依國家/地區更新可用性
- 更新產品的退款政策
- 更新最終使用者授權合約 (EULA)

管理變更請求

在自助式清單中，您可以使用變更請求來變更您的產品。您可以在請求索引標籤 AWS Marketplace 管理入口網站 的 上找到您目前的請求。 您可以透過位於導覽列下的請求變更下拉式清單提出新請求。

為 SaaS 產品建立變更請求

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 從產品索引標籤中，從下拉式清單中選取 SaaS。
3. 提交請求後，便會開始處理。變更請求會經歷下列狀態：檢閱中、準備變更和套用變更。
4. 當請求的處理完成時，其狀態會變更為下列其中一個值：

    - 成功 – 此狀態表示已處理您請求的變更，且變更會反映在系統中。

- 失敗 – 此狀態表示請求發生錯誤，且變更未處理。如果狀態為失敗，您可以選取請求以尋找錯誤代碼，提供如何修正問題的建議。您可以對錯誤進行疑難排解，並建立新的變更請求。若要讓程序更快，您可以使用複製到新的請求函數，複製失敗請求的詳細資訊。您可以進行所需的變更並重新提交請求。

以更新開頭的變更請求將載入專案的目前詳細資訊。然後，您可以進行更新，以覆寫現有的詳細資訊。新增和限制請求對專門用於在每個請求成功後佈建的更新（在您選擇自助式體驗中的儲存和結束和提交動作之後）。這表示現有的訂閱者可以繼續使用產品，直到其訂閱或合約結束為止。不過，任何新的訂閱者都無法新增至處於受限狀態的產品。

更新產品資訊

建立產品後，您可能想要變更其中相關聯的資訊 AWS Marketplace。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 在 SaaS 產品頁面的 SaaS 產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新產品資訊。
4. 更新您想要變更的下列任何欄位：

    - 產品標題
    - SKU
    - 簡短描述
    - 長描述
    - 產品標誌 URL
    - 重點標示
    - 產品類別
    - 關鍵字
    - 產品影片 URL
    - 資源
    - 支援資訊

    > **Note**
    >
    > 如需標誌格式的詳細資訊，請參閱公司和產品標誌需求。

5. 若要更新產品資訊，請選擇提交。
6. 確認請求顯示在請求索引標籤上，且狀態為審核中。您可能需要重新整理頁面，才能查看您的新請求。

更新架構詳細資訊

若要接收產品部署所在的特殊指定 AWS，請在 AWS Marketplace 管理入口網站 (AMMP) 中選擇託管模式並上傳架構圖表，以更新產品的架構詳細資訊。如需 AWS 考慮部署在 上的託管模式清單 AWS，請參閱 指導方針。

更新架構詳細資訊

1. 登入 AWS Marketplace 管理入口網站。
2. 在產品功能表中，選擇 SaaS。
3. 在 SaaS 產品表格中，選擇您的產品。
4. 針對請求變更，選擇更新架構詳細資訊。
5. 選擇託管模式。如果您選擇 AWS Marketplace 考慮部署的託管模式 AWS，請完成下列兩個額外步驟。如需在 上部署之模式的詳細資訊指導方針， AWS請參閱本指南先前所述的 。
   a. 如果出現提示，請在架構圖區段中，選擇選擇檔案以 PNG 或 JPG 格式上傳架構圖。如需圖表的詳細資訊，請參閱 建立架構圖。
   b. 如果出現提示，請在應用程式平面中選擇應用程式執行的位置。如需詳細資訊，請參閱控制平面與應用程式平面。
6. 選擇更新封存詳細資訊。

請求完成後，請求狀態變更為成功。若要檢查請求狀態，請在 SaaS 產品資料表中選擇您的產品，然後選擇請求日誌索引標籤。

若要檢視您的評估結果，請在 SaaS 產品資料表中選擇您的產品，然後選擇架構詳細資訊索引標籤。

更新 AWS 帳戶 IDs的允許清單

您可以變更可在有限狀態下檢視產品的 AWS 帳戶 IDs 清單。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 在 SaaS 產品頁面的 SaaS 產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選取更新允許清單。清單顯示目前允許列出的 AWS 帳戶 IDs。

4. 在允許清單 AWS 帳戶欄位中，輸入 AWS 帳戶 IDs並使用逗號分隔。
5. 若要更新 AWS 帳戶 IDs 的允許清單，請選擇提交。

更新產品可見性

若要變更哪些買方可以在 中檢視您的 Quick Launch 體驗 AWS Marketplace，您可以使用更新可見性。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 在 SaaS 產品頁面中，選取要修改的產品。
3. 從請求變更下拉式清單中，選擇更新可見性。

> **Note**
>
> 您可以使用此變更請求，請求將產品從有限狀態移至公有狀態。不過，變更請求必須經過 AWS Marketplace 賣方營運團隊核准程序，才能移至公有。

4. 當您發佈至公有時，您會提供產品的實際價格。此價格會在您的清單核准公有可見性後套用。
5. 若要提交您的請求以供檢閱，請選擇提交。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

更新定價條款

若要變更 SaaS 產品上每個維度的定價，請使用更新定價條款。

> **Note**
>
> 任何維度的定價增加都會導致定價更新選項至少在接下來的 90 天內無法使用。如果同時更新價格降低和提高，請先更新價格降低。

1. 開啟 AWS Marketplace 管理入口網站並登入您的賣方帳戶。
2. 在 SaaS 產品頁面的 SaaS 產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選取更新公開優惠，然後選取更新定價條款。
4. 目前定價已預先填入 欄位。您可以刪除目前的價格，然後新增新的價格。

5. 若要提交您的請求以供檢閱，請選擇提交。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會更新為成功或失敗。

新增定價維度

您可以新增要用來為您的產品收費的維度。維度是基礎度量單位，您的買方在使用產品時需支付此度量單位的費用。

> **Note**
>
> 若要更新現有定價維度的名稱或描述，請參閱 the section called “更新定價維度”。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 從 SaaS 產品索引標籤中，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新定價維度，然後選擇新增定價維度。
4. 提供維度 API 識別符、顯示名稱和描述，將新維度新增至您的產品，然後選擇下一步。

   > **Note**
   >
   > API 識別符和名稱在所有維度中必須是唯一的。您無法在建立維度後變更 API 識別符和單位。

5. 定義您新增的每個維度的價格，然後選擇下一步以檢閱您的變更。

   > **Note**
   >
   > 您只能為為您的產品選取的定價模型新增維度（例如合約、用量或消耗合約）。對於有限產品，新新增維度的價格設定為 0.01 美元。您可以在產品準備好公開可見性時更新價格。

6. 選擇提交以提交您的請求以供檢閱。
7. 在請求索引標籤中，確認請求狀態為審核中。當請求完成時，狀態會變更為成功。

更新定價維度

您可以更新要用來為您的產品收費的維度。維度是買方在使用產品時需支付的基本度量單位。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 從 SaaS 產品索引標籤中，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新定價維度，然後選擇更新維度資訊。
4. 尋找您要更新的維度，然後選擇名稱或描述。
5. 提供新名稱或描述，然後選擇核取記號以確認您的更新。維度名稱必須是唯一的。
6. 選擇提交以提交您的請求以供檢閱。
7. 在請求索引標籤中，確認請求狀態為審核中。當請求完成時，狀態會變更為成功。

限制定價維度

您可以限制目前列於產品中的維度。此請求會從產品中移除選取的維度。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 從 SaaS 產品索引標籤中，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新定價維度，然後選擇限制定價維度。
4. 對於有限和公開產品，系統會提示您使用聯絡我們https://aws.amazon.com/marketplace/management/contact-us/按鈕來聯絡 AWS Marketplace 賣方營運團隊。使用表單，提供您要從產品清單中移除之維度的詳細資訊。

> **Note**
>
> 操作不一定能夠限制定價維度。

決定買方如何存取您的產品

您可以選擇下列其中一個選項，讓客戶如何存取您的產品：

- the section called “更新 SaaS URL 履行選項” – 客戶在 中訂閱您的產品後，會為其重新導向的網站使用 URL AWS Marketplace。
- the section called “設定快速啟動” – 客戶使用簡化的程序來設定和啟動您的產品。您可以為具有有限或公開可見性的現有產品完成此組態。

更新 SaaS URL 履行選項

若要更新用來履行 SaaS 產品的 URL，請使用更新履行選項索引標籤。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 在 SaaS 產品頁面的 SaaS 產品索引標籤上，選擇您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新履行選項，然後選擇編輯預設履行 URL。
4. 在履行 URL 欄位中，輸入 SaaS 產品履行選項的新 URL。
5. 若要提交您的請求以供檢閱，請選擇提交。
6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會更新為成功或失敗。

設定快速啟動

中列出的 SaaS 產品 AWS Marketplace 通常需要在訂閱買方的帳戶中部署 AWS 資源 （例如 IAM 角色）。快速啟動可讓您為買方提供使用 AWS CloudFormation 範本的引導step-by-step說明和資源部署。買方使用 CloudFormation 範本來設定和啟動產品。若要進一步了解快速啟動組態程序，請參閱啟用 SaaS 快速啟動實驗室。

若要設定客戶可用來啟動 SaaS 產品的快速啟動體驗，請使用履行選項索引標籤。

1. 在 SaaS 產品頁面的 SaaS 產品索引標籤上，選取您要修改的產品。

   > **Note**
   >
   > 若要設定快速啟動體驗，產品必須具有有限或公開可見性。

2. 在產品詳細資訊頁面上，選擇履行選項索引標籤。
3. 針對快速啟動，選擇啟用和設定按鈕。
4. 如需帳戶登入詳細資訊，請提供您的網站 URL，供買方登入或建立帳戶。此 URL 會在買方體驗中開啟新索引標籤。買方接著會登入或建立 帳戶，並返回 AWS Marketplace 以啟動範本。
5. 建立 AWS CloudFormation 範本。

   > **Tip**
   >
   > 在建立將 AWS 資源部署到買方 AWS 帳戶的 CloudFormation 範本時，請遵循 AWS Well-Architected Framework。如需詳細資訊和資源，請參閱 AWS Well-Architected 和 AWS Well-Architected Framework。

   針對AWS CloudFormation 範本，選擇新增 AWS CloudFormation 範本按鈕並提供下列資訊：

- 標題 – 提供 CloudFormation 部署的名稱。
- 描述 – 提供範本的描述。
- 堆疊名稱 – 提供堆疊的名稱。此名稱是 CloudFormation 中買方的堆疊名稱。
- CloudFormation 範本 URL – 提供 Amazon Simple Storage Service 將檢閱此範本。

> **Note**
>
> 為了簡化客戶的啟動程序，我們建議將與您組態程序相關聯的範本數量降至最低。理想情況下，您希望有一個範本部署使用產品所需的資源。有關 CloudFormation 範本的問題，請聯絡您的 AWS Marketplace 業務開發合作夥伴或AWS Marketplace 賣方營運團隊。

- 必要的 IAM 許可 – 提供部署 CloudFormation 範本所需的許可。如果您想要共用部署參數，這些參數會存放為買方AWS Secrets Manager的 中的秘密，您的政策必須包含下列動作：
  - `secretsManager:ListSecrets`
  - `secretsManager:DescribeSecret`
  - `secretsManager:ReplicateSecretToRegions`
  - `secretsManager:GetSecretValue`

> **Note**
>
> 如果您的產品需要賣方提供的 CloudFormation 部署參數（例如 API 金鑰和外部 IDs），請使用 `PutDeploymentParameter`操作與客戶共用參數。如需詳細資訊，請參閱 AWS Marketplace 部署服務 API 參考PutDeploymentParameter中的 。

6. （選用）對於手動組態指示，為想要手動設定產品的買方提供指示。請考慮包含產品入門指南和文件的連結。
7. 如需啟動詳細資訊，請提供買方在部署 CloudFormation 堆疊之後存取產品的 URL。
8. （選用）對於允許列出的快速啟動帳戶，提供逗號分隔的 清單 AWS 帳戶 ，以有限可見性檢視快速啟動體驗。
9. 選擇提交按鈕。快速啟動體驗的可見性有限，這表示只有您的帳戶和允許列出的帳戶才能看見。透過有限可見性，您可以在訂閱產品並選擇設定帳戶按鈕後，使用設定和啟動頁面來測試組態。

   - 若要從買方的角度預覽頁面，請選擇預覽。當兩個頁面都開啟時，您的組態變更將反映在預覽上。

10. 當您準備好時，您可以在 AWS Marketplace 目錄中發佈 Quick Launch 體驗。使用產品詳細資訊頁面上履行選項索引標籤上的更新快速啟動可見性按鈕。

    當您將可見性變更為公開時， AWS Marketplace 賣方營運團隊會檢閱組態、執行買方測試，並發佈體驗。

    > **Note**
    >
    > 如果您在啟用快速啟動體驗時需要支援，請聯絡AWS Marketplace 賣方營運團隊。

依國家/地區更新可用性

您可以定義可提供您產品的國家/地區。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 從 SaaS 產品索引標籤中，選取您要修改的產品。
3. 從請求變更下拉式清單中，選取更新公開優惠，然後選擇依國家/地區更新可用性。
4. 請選擇下列其中一個選項：
   - 所有國家 - 在所有支援的國家/地區提供。
   - 具有排除項目的所有國家/地區 – 適用於所有支援的國家/地區，但所選國家/地區除外。
   - 僅限允許列出的國家/地區 – 提供產品的特定國家/地區清單。
5. 選擇提交以提交您的請求以供檢閱。
6. 在請求索引標籤中，確認請求狀態為審核中。當請求完成時，狀態會變更為成功。

更新產品的退款政策

您可以使用更新退款政策來更新產品的退款政策。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 在 SaaS 產品頁面的 SaaS 產品索引標籤上，選取您要修改的產品。
3. 從請求變更下拉式清單中，選取更新公開優惠，然後選取更新退款政策。
4. 目前的退款政策詳細資訊會在文字方塊中提供。視需要檢閱和修改詳細資訊。提交請求會覆寫目前的退款政策。
5. 若要提交您的請求以供檢閱，請選擇提交。

6. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會更新為成功或失敗。

更新最終使用者授權合約 (EULA)

您可以為訂閱您產品的新使用者更新 EULA。

1. 開啟 AWS Marketplace 管理入口網站，並登入您的賣方帳戶。
2. 從 SaaS 產品索引標籤中，選取您要修改的產品。
3. 從請求變更下拉式清單中，選擇更新公開優惠，然後選擇更新 EULA。
4. 您可以選擇 AWS Marketplace (SCMP) 的標準合約或提交自訂 EULA。對於自訂 EULA，您必須提供合約的 Amazon Simple Storage Service (Amazon S3) URL。您的 Amazon S3 儲存貯體必須可公開存取。
5. 選擇提交以提交您的請求以供檢閱。
6. 在請求索引標籤中，確認請求狀態為審核中。當請求完成時，狀態會變更為成功。

將您的 SaaS 訂閱產品與 整合 AWS Marketplace

將您的產品與 整合 AWS Marketplace 是 中的一個步驟在 中建立 SaaS 產品 AWS Marketplace。若要將軟體整合為服務 (SaaS) 訂閱產品 AWS Marketplace，您必須撰寫程式碼，並示範其可以成功回應數個客戶案例。下列各節說明如何整合您的 SaaS 訂閱模型 AWS Marketplace。

> **Note**
>
> 開始之前，請確定您已為軟體software-as-a-service(SaaS) 產品選擇正確的定價模型 AWS Marketplace。如需詳細資訊，請參閱規劃您的 SaaS 產品。

主題

- 案例：您的服務會驗證新客戶
- 案例：計量器用量
- 案例：監控使用者訂閱的變更
- 案例：驗證客戶訂閱
- 測試您的 SaaS 訂閱產品整合

案例：您的服務會驗證新客戶

當客戶訂閱您的產品時，它們會重新導向至您的註冊 URL，這是具有臨時`x-amzn-marketplace-token`字符的 HTTP POST 請求。以下列方式回應此請求：

1. 呼叫 中的 `ResolveCustomer` API 操作`CustomerAWSAccountId`，`ProductCode`以交換`CustomerIdentifier`、 和 的字符 AWS Marketplace Metering Service。
2. `ProductCode` 在您的系統中保留 `CustomerIdentifier`、 `CustomerAWSAccountID`和 以供日後呼叫。您必須存放客戶是否擁有有效的訂閱，以及您需要的客戶相關資訊。
3. 為了回應請求，您必須顯示使用者第一次使用的體驗 （適用於您的服務）。

案例：計量器用量

當客戶開始使用您的服務時，您必須每小時傳送計量記錄。如需如何計量的詳細資訊，請參閱使用 SaaS 訂閱設定用量計量。

建議您使用 AWS CloudTrail 來監控活動，以確保帳單資訊已傳送至 AWS。傳送計量記錄時，請記住下列事項：

- 計量請求會每小時刪除重複。
- 每小時傳送的記錄是累積的。
- 我們強烈建議最佳實務是，即使過去一小時沒有記錄，您每小時都會傳送計量記錄，使用量為 0。

案例：監控使用者訂閱的變更

設定 Amazon Simple Queue Service (Amazon SQS) 佇列，並訂閱您產品的 Amazon SNS 主題。您的 SNS 主題資訊已包含在您在建立產品時從 AWS Marketplace 賣方營運團隊收到的電子郵件訊息中。如需詳細資訊，請參閱在 中建立 SaaS 產品 AWS Marketplace。透過訂閱 SNS 主題，您會收到客戶訂閱變更的通知，包括提供或撤銷特定客戶的存取權。

> **Note**
>
> Amazon SNS 主題 Amazon Resource Name (ARN) 看起來像 `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>`。

您必須回應的通知包括：

- `subscribe-success` – 客戶已訂閱，而且您可以成功根據其客戶 ID 進行計量。
- `unsubscribe-pending` – 客戶正在進行取消訂閱。您應該傳送任何上次計量記錄。
- `unsubscribe-success` – 客戶已取消訂閱。將不再接受客戶的計量記錄。遵循關閉客戶資源的實務，遵守您的保留政策。
- `subscribe-fail` – 客戶訂閱失敗。您不應該對照他們的客戶 ID 計量，或代表客戶建立資源。

案例：驗證客戶訂閱

代表客戶建立資源之前，請確認客戶應該可以存取您的產品。儲存您透過 Amazon SQS 收到的通知中客戶的最新狀態，以了解客戶是否具有存取權。

測試您的 SaaS 訂閱產品整合

將 SaaS 訂閱產品與 整合後 AWS Marketplace，您必須進行深入測試，以確保整合成功。下列程序概述驗證產品整合的步驟。

> **Note**
>
> 使用您自己的帳戶訂閱您的產品，並測試整合是否成功。價格可以暫時降低，以便您可以測試購買流程，而不會在這些帳戶中產生高費用。如需有關暫時降低價格或允許其他測試帳戶存取您產品的詳細資訊，請聯絡AWS Marketplace 賣方營運團隊。
> 產品啟動後，服務必須繼續回應新客戶的這些案例。

1. 訂閱您的產品，使用允許的 帳戶來測試客戶體驗。
2. 使用允許的帳戶訂閱之後，請確定帳戶已重新導向至註冊 URL，而且重新導向是包含臨時字符的 POST 請求。請確定您的應用程式仍保留客戶 ID，以供日後呼叫使用。這會測試 的一部分案例：您的服務會驗證新客戶。
3. 在上一個步驟中驗證測試帳戶後，將帳戶加入您的應用程式。例如，您可以讓測試客戶填寫表單來建立新的使用者。或者，提供他們其他後續步驟，以存取您的 SaaS 應用程式。這會測試 的一部分案例：您的服務會驗證新客戶。
4. 測試客戶加入後，請使用 中的 `BatchMeterUsage` API 操作，提出將計量記錄傳送至 AWS 以計費的請求 AWS Marketplace Metering Service。這會測試 案例：計量器用量。
5. 測試訂閱變更。可能的情況包括取消訂閱、訂閱成功和訂閱失敗。這會測試 案例：監控使用者訂閱的變更。

6. 驗證訂閱是否成功。收到測試帳戶的 Amazon SNS 通知並顯示成功的訂閱訊息後，即可開始計量。在您收到 Amazon SNS 通知 AWS Marketplace Metering Service 之前傳送到 的記錄，不會進行計量。這會測試 案例：驗證客戶訂閱。

   > **Note**
   >
   > 為了防止帳單問題，強烈建議以程式設計方式等待此通知，然後再代表您的客戶啟動資源。

7. 在您完成所有整合要求並測試解決方案之後，請通知 AWS Marketplace 賣方營運團隊。他們會透過 API `BatchMeterUsage`操作驗證您已成功傳送計量記錄，對解決方案執行一系列最終測試。

整合和測試完成後，您可以執行最終審核並在公開上列出您的產品 AWS Marketplace。如需詳細資訊，請參閱在 中建立 SaaS 產品 AWS Marketplace。

將您的 SaaS 合約產品與 整合 AWS Marketplace

將您的軟體即服務 (SaaS) 合約產品與 整合， AWS Marketplace 是 中的一個步驟在 中建立 SaaS 產品 AWS Marketplace。若要將軟體整合為服務 (SaaS) 合約產品 AWS Marketplace，您必須撰寫程式碼，並示範其可以成功回應數個客戶案例。下列各節說明這些案例、說明如何回應這些案例，並提供測試整合的概觀。

> **Note**
>
> 開始之前，請確定您已為 中的軟體即服務 (SaaS) 產品選擇正確的定價模型 AWS Marketplace。如需詳細資訊，請參閱規劃您的 SaaS 產品。

主題

- 案例：您的服務會驗證新客戶
- 案例：您的服務處理客戶請求
- 案例：監控使用者訂閱的變更
- 測試您的 SaaS 合約產品整合

案例：您的服務會驗證新客戶

當客戶訂閱您的產品時，它們會重新導向至您的註冊 URL，這是具有臨時`x-amzn-marketplace-token`字符的 HTTP POST 請求。以下列方式回應此請求：

1. 呼叫 中的 `ResolveCustomer` API 操作`CustomerAWSAccountId`，`ProductCode`以交換`CustomerIdentifier`、 和 的字符 AWS Marketplace Metering Service。
2. 透過呼叫 中的 `GetEntitlements` API 操作，確認客戶可存取的訂閱和數量 （ 如適用 ） AWS Marketplace Entitlement Service。
3. 將 `CustomerIdentifier`、 `CustomerAWSAccountId`和 保留`ProductCode`在您的系統中以供日後呼叫。存放客戶是否具有有效的訂閱，以及您需要的客戶相關資訊。
4. 為了回應請求，您必須顯示使用者的第一次使用體驗 （ 適用於您的服務 ）。

案例：您的服務處理客戶請求

當客戶對您的服務提出請求時，您必須以適當的動作或訊息回應下列案例：

- 它們在您的系統中沒有客戶 ID。這表示他們尚未訂閱。您應該告訴使用者如何訂閱。
- 它們具有客戶 ID，API `GetEntitlements`操作會傳回適當的權限。在這種情況下，您應該完成請求。
- 它們確實有客戶 ID，但 `GetEntitlements` API 操作不會傳回任何權利，或沒有足夠的數量來滿足請求。在這種情況下，您必須決定如何處理存取和管理他們的體驗。

案例：監控使用者訂閱的變更

設定 Amazon Simple Queue Service (Amazon SQS) 佇列，並訂閱您產品的 Amazon SNS 主題。您的 SNS 主題資訊已包含在您在建立產品時從 AWS Marketplace 操作團隊收到的電子郵件訊息中。如需詳細資訊，請參閱在 中建立 SaaS 產品 AWS Marketplace。透過訂閱 SNS 主題，您會收到客戶權利變更的通知，包括提供或撤銷特定客戶的存取權。

> **Note**
>
> SNS 主題 Amazon Resource Name (ARN) 看起來像 `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>`。

您必須回應的唯一通知是：

- `entitlement-updated` – 客戶權限已變更，您必須呼叫 `GetEntitlements` API 操作才能查看新狀態。更新您的客戶商店，如果適用 （例如，客戶的合約已失效），請遵循關閉客戶資源的實務，遵守您的保留政策。

> **Note**
>
> 如需其他資訊，請參閱 使用 檢查權利 AWS Marketplace Entitlement Service。

測試您的 SaaS 合約產品整合

將 SaaS 合約產品與 整合後 AWS Marketplace，您必須進行深入測試，以確保整合成功。下列程序概述驗證產品整合的步驟。

> **Note**
>
> 使用您自己的帳戶訂閱您的產品，並測試整合是否成功。價格可以暫時降低，以便您可以測試購買流程，而不會在這些帳戶中產生高費用。如需暫時降低價格或允許其他測試帳戶存取您產品的詳細資訊，請聯絡我們。
> 產品啟動後，服務必須繼續回應新客戶的這些案例。

1. 使用允許的 帳戶，透過取得您產品的合約來測試客戶體驗。
2. 帳戶簽訂合約後，請確定帳戶已重新導向至註冊 URL，且重新導向是包含臨時權杖的 POST 請求。請確定您的應用程式保留客戶 ID 以供日後通話使用，並正確處理客戶擁有的權限。這會測試的一部分案例：您的服務會驗證新客戶。
3. 在上一個步驟中驗證測試帳戶後，將帳戶加入您的應用程式。例如，您可以讓測試客戶填寫表單來建立新的使用者。或者，提供他們其他後續步驟，以存取您的 SaaS 應用程式。這會測試 的一部分案例：您的服務會驗證新客戶。
4. 如果 `GetEntitlements` API 操作未傳回任何權利，無論是在加入期間或持續驗證通過時，您的應用程式都必須正確管理無權使用者的存取和體驗。這會測試 案例：您的服務處理客戶請求。
5. 測試訂閱變更。確認您的應用程式正確處理取消訂閱、成功訂閱和失敗的訂閱案例。這會測試 案例：監控使用者訂閱的變更。
6. 完成所有整合要求並測試解決方案後，請通知 AWS Marketplace 營運團隊。然後，他們會驗證您已成功呼叫 `GetEntitlements` API 操作並充分加入新客戶，以測試解決方案。

整合和測試完成後，您可以執行最終審核並在公開上列出您的產品 AWS Marketplace。如需詳細資訊，請參閱在 中建立 SaaS 產品 AWS Marketplace。您也可以填寫退款申請表來取消測試訂閱。如需取消訂閱的詳細資訊，請參閱 the section called “請求產品退款”。

將您的 SaaS 合約與pay-as-you-go產品與 整合 AWS Marketplace

將您的產品與 整合 AWS Marketplace 是 中的一個步驟在 中建立 SaaS 產品 AWS Marketplace。若要將軟體即服務 (SaaS) 合約產品與 整合 AWS Marketplace，您必須撰寫程式碼，並示範它可以成功回應數個客戶案例。下列各節說明這些案例、說明如何回應這些案例，並提供測試整合的概觀。

> **Note**
>
> 開始之前，請確定您已為其中的軟體即服務 (SaaS) 產品選擇正確的定價模式 AWS Marketplace。如需詳細資訊，請參閱規劃您的 SaaS 產品。

主題

- 案例：您的服務會驗證新客戶
- 案例：您的服務處理客戶請求
- 案例：計量表用量
- 案例：監控使用者權利的變更
- 使用隨需pay-as-you-go整合測試您的 SaaS 合約

案例：您的服務會驗證新客戶

當客戶訂閱您的產品時，它們會重新導向至您的註冊 URL，這是具有臨時`x-amzn-marketplace-token`字符的 HTTP POST 請求。以下列方式回應此請求：

1. 呼叫 中的 `ResolveCustomer` API 操作`CustomerAWSAccountId`，`ProductCode`以交換`CustomerIdentifier`、 和 的字符 AWS Marketplace Metering Service。
2. 透過在 中呼叫 `GetEntitlements`動作，驗證客戶可存取的訂閱和數量 （ 如適用 ） AWS Marketplace Entitlement Service。
3. `ProductCode` 在您的系統中保留 `CustomerIdentifier`、 `CustomerAWSAccountId`和 以供日後呼叫。存放客戶是否具有有效的訂閱，以及您需要的客戶相關資訊。
4. 為了回應請求，您必須顯示使用者的第一次使用體驗 （ 適用於您的服務 ）。

案例：您的服務處理客戶請求

當客戶對您的服務提出請求時，您必須以適當的動作或簡訊回應下列案例：

- 它們在您的系統中沒有客戶 ID。這表示他們尚未訂閱。您應該向他們提供說明如何訂閱的訊息。
- 他們有客戶 ID，API `GetEntitlements`操作會傳回適當的權利。在此案例中，您應該完成請求。
- 它們確實有客戶 ID，但 `GetEntitlements` API 操作不會傳回任何權利，或沒有足夠的數量來滿足請求。在此案例中，您必須決定如何處理存取和管理他們的體驗。

案例：計量表用量

當客戶開始使用您的服務時，您必須每小時傳送計量記錄。如需如何計量的詳細資訊，請參閱 使用 SaaS 訂閱設定用量計量。

我們建議您使用 AWS CloudTrail 來監控活動，以確保帳單資訊已傳送至其中 AWS。傳送計量記錄時，請注意下列事項：

- 計量請求會在該小時刪除重複。
- 每小時傳送的記錄是累積的。
- 我們強烈建議最佳實務是，即使過去一小時沒有記錄，您仍每小時傳送計量記錄，使用量為 0。

案例：監控使用者權利的變更

設定 Amazon Simple Queue Service (Amazon SQS) 佇列，並訂閱您產品的 Amazon SNS 主題 - 有兩個 SNS 主題，一個用於權利變更，另一個用於訂閱變更。您的主題資訊包含在您在建立產品時從 AWS Marketplace 賣方營運團隊收到的電子郵件訊息中。如需詳細資訊，請參閱在 中建立 SaaS 產品 AWS Marketplace。透過訂閱 SNS 主題，您會收到客戶訂閱變更的通知，包括提供或撤銷特定客戶的存取權。

> **Note**
>
> 訂閱變更的 SNS 主題 Amazon Resource Name (ARN) 看起來像 `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>`。權利變更的 SNS 主題 ARN 看起來像 `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>`。

您必須回應的通知如下：

- `entitlement-updated`（在權利 SNS 主題中）– 客戶權利已變更，您必須呼叫 `GetEntitlements` API 操作以查看新狀態。更新您的客戶商店，如果適用（例如，客戶的合約已到期），請遵循關閉客戶資源的實務，遵守您的保留政策。
- `subscribe-success`（在訂閱 SNS 主題中）– 客戶已訂閱，您可以成功根據其客戶 ID 進行計量。
- `unsubscribe-pending`（在訂閱 SNS 主題中）– 客戶正在取消訂閱。您應該傳送任何上次計量記錄。
- `unsubscribe-success`（在訂閱 SNS 主題中）– 客戶已取消訂閱。將不再接受客戶的計量記錄。遵循您關閉客戶資源的實務，遵守您的保留政策。
- `subscribe-fail`（在訂閱 SNS 主題中）– 客戶訂閱失敗。您不應該針對其客戶 ID 計量，或代表客戶啟用資源。

> **Note**
>
> 如需其他資訊，請參閱 使用 檢查權利 AWS Marketplace Entitlement Service。

使用隨需pay-as-you-go整合測試您的 SaaS 合約

將合約與pay-as-you-go產品整合之後 AWS Marketplace，您必須進行深入測試，以確保整合成功。下列程序概述驗證產品整合的步驟。

> **Note**
>
> 使用您自己的帳戶訂閱您的產品，並測試整合是否成功。價格可以暫時降低，以便您可以測試購買流程，而不會在這些帳戶中產生高額費用。如需暫時降低價格或允許其他測試帳戶存取您產品的詳細資訊，請聯絡AWS Marketplace 賣方營運團隊。
> 產品啟動後，服務必須繼續回應新客戶的這些案例。

1. 使用允許的帳戶，透過取得您產品的合約來測試客戶體驗。
2. 帳戶簽訂合約後，請確定帳戶重新導向至註冊 URL，且重新導向是包含臨時字符的 POST 請求。請確定您的應用程式保留客戶 ID 以供未來通話使用，並正確處理客戶擁有的權利。這會測試 的一部分案例：您的服務會驗證新客戶。

3. 在上一個步驟中驗證測試帳戶之後，將帳戶加入您的應用程式。例如，您可以讓測試客戶填寫表單來建立新的使用者。或者，提供他們存取 SaaS 應用程式的其他後續步驟。這會測試 的一部分案例：您的服務會驗證新客戶。
4. 如果 `GetEntitlements` API 操作未傳回任何權利，無論是在加入期間或持續驗證通過時，您的應用程式都必須正確管理無權使用者的存取和體驗。這會測試 案例：您的服務處理客戶請求。
5. 測試客戶加入後，請使用 中的 `BatchMeterUsage` API 操作，提出將計量記錄傳送至 AWS 以用於計費的請求 AWS Marketplace Metering Service。這會測試 案例：計量表用量。
6. 測試訂閱變更。確認您的應用程式正確處理取消訂閱、訂閱成功和訂閱失敗的情況。這會測試 案例：監控使用者權利的變更。
7. 在您完成所有整合要求並測試解決方案之後，請通知 AWS Marketplace 賣方營運團隊。然後，他們會驗證您已成功呼叫 `GetEntitlements` API 操作並充分加入新客戶，以測試解決方案。他們也會確認您已成功使用 `BatchMeterUsage` API 操作傳送計量記錄。

整合和測試完成後，您可以執行最終審核並在公開上列出您的產品 AWS Marketplace。如需詳細資訊，請參閱在 中建立 SaaS 產品 AWS Marketplace。

部署無伺服器 SaaS 整合

建立 SaaS 解決方案之後，您可以將其與 上對應的清單整合 AWS Marketplace。若要這樣做，您可以使用 AWS CloudFormation 範本來建立無伺服器整合。整合可讓您註冊客戶、授予和撤銷存取權、更新客戶權利，以及報告計量用量。

如需建立無伺服器 SaaS 整合的詳細資訊，請參閱 整合您的 Saas 與 Serverless SaaS 整合參考，這是一個實驗室，說明如何部署 AWS CloudFormation 範本並完成相關任務。

規劃您的 SaaS 產品

將軟體即服務 (SaaS) 產品新增至 之前 AWS Marketplace，您必須先進行一些規劃。此步驟對於您產品的成功至關重要。缺乏規劃可能會導致帳單問題，或者您可能需要重新建立您的產品 AWS Marketplace。下列各節說明如何規劃 SaaS 產品，

> **Important**
>
> 在您設定產品大部分的設定之後，就無法再進行變更。如果您需要在 中建立產品後進行變更 AWS Marketplace，您可能需要使用正確的設定建立新的產品。

主題

- 規劃您的定價
- 規劃您的帳單整合
- 規劃您的 Amazon SNS 整合
- 規劃客戶將如何存取您的產品

規劃您的定價

SaaS 產品有三種定價模式 AWS Marketplace。為您的產品選擇正確的定價模型，是您在規劃產品時將做出的最重要決策。選擇錯誤的定價模型可以讓您恢復數週。定價模型會決定您客戶的付款選項，以及您需要寫入、測試和部署的帳單整合程式碼。如需不同類型的定價模型的相關資訊，請參閱 SaaS 產品定價。

> **Note**
>
> 所有 SaaS 定價模型都支援免費試用。如需詳細資訊，請參閱 SaaS 免費試用版。

規劃您的帳單整合

在 上擁有 SaaS 產品的其中一個優點 AWS Marketplace 是合併帳單。若要利用此利益，您必須整合 AWS Marketplace Metering Service 或 AWS Marketplace Entitlement Service，視您選擇的定價模型而定。這兩種服務可協助您確保帳單和用量報告的準確性。

規劃整合之後，您必須先測試與產品的整合，才能上線。如需整合和測試的詳細資訊，請參閱存取 AWS Marketplace 計量和權利服務 APIs。

規劃您的 Amazon SNS 整合

您可以訂閱 SaaS 產品的兩個 Amazon Simple Notification Service (Amazon SNS) 主題。如需詳細資訊，請參閱 SaaS 通知。這些訊息可協助您以程式設計方式處理由 AWS 或 客戶啟動的訂閱和合約變更。Amazon SNS 通知可以是程式設計觸發條件，可讓客戶在您的產品註冊網站上註冊新帳戶。他們也可以拒絕訂閱過期的客戶存取您的產品。根據您的程式設計處理這些通知的方式，您可以選擇客戶如何接收通知。

規劃客戶將如何存取您的產品

本節說明如何讓買方存取您的產品。

規劃您的 SaaS 產品註冊網站

購買 SaaS 產品的客戶需要存取它。您必須規劃並實作您希望客戶存取產品的方式。SaaS 產品支援下列存取選項：

- 快速啟動
- AWS PrivateLink
- 您自己的產品網站

若要使用您的註冊網站驗證 AWS Marketplace 客戶，請參閱 SaaS 客戶加入。

使用 Quick Launch 讓客戶存取您的產品

使用 Quick Launch 部署選項，以減少買方設定、部署和啟動產品所需的時間和資源。快速啟動可減少買方在過程中必須造訪的網站數量。如需詳細資訊，請參閱設定快速啟動。

AWS PrivateLink 使用 讓客戶存取您的 SaaS 產品

您可以使用 使用 透過 Amazon VPC 交付您的產品 AWS PrivateLink 將服務設定為 Amazon Virtual Private Cloud (Amazon VPC) 端點服務。您的客戶可以建立 VPC 端點， AWS 雲端 並透過虛擬網路存取您的軟體。或者，您可以透過您擁有和維護的網站提供軟體產品的存取權，讓客戶透過網際網路建立連線。

使用您自己的網站

您的 SaaS 產品託管在您的環境中，必須透過您管理和維護的公有端點，例如網站，透過網際網路存取。一般而言，您有一個網站，客戶可使用該網站來註冊您的產品、登入以使用產品，以及存取您產品的支援。

的 SaaS 產品指導方針 AWS Marketplace

AWS Marketplace 針對 上的所有軟體即服務 (SaaS) 產品和產品維護下列指導方針， AWS Marketplace 以向客戶提升安全、可靠且可靠的平台。下列各節提供 SaaS 產品的指導方針 AWS Marketplace。

提交時會檢閱所有產品及其相關中繼資料，以確保它們符合或超過目前的 AWS Marketplace 準則。這些指導方針經過審核和調整，以滿足我們不斷演進的安全要求。此外， AWS Marketplace 會持續審查產品，以確認它們符合這些準則的任何變更。如果產品不合規，我們可能會要求您更新產品，在某些情況下，您的產品可能暫時無法提供給新訂閱者，直到問題解決為止。

主題

- 產品設定準則
- 客戶資訊要求
- 產品使用準則
- 架構準則

產品設定準則

所有 SaaS 產品都必須遵守下列產品設定準則：

- 定價維度不能僅限於私有優惠。買方應該能夠訂閱公開產品的任何定價維度。
- 至少有一個定價維度的價格必須大於 0.00 美元。
- 所有定價維度都必須與實際軟體相關，且不得包含與軟體無關的任何其他產品或服務。
- 僅在 AWS GovCloud (US) 區域中提供的 SaaS 產品必須包含產品標題中的GovCloud某處。

客戶資訊要求

所有 SaaS 產品都必須遵守下列客戶資訊要求：

- SaaS 產品必須完全透過 上列出的維度計費 AWS Marketplace。
- 您可以隨時收集 SaaS 產品的客戶付款資訊，包括信用卡和銀行帳戶資訊。
- SaaS 註冊頁面必須包含買方電子郵件地址的輸入欄位。您可以包含其他欄位，例如：
  - 名稱
  - 郵遞區號
  - 電話號碼
  - 公司資訊
  - 產品設定偏好設定

  如果您計劃使用多種語言，您必須提供註冊頁面的英文檢視。

產品使用準則

所有 SaaS 產品都必須遵守下列產品使用準則：

- 訂閱 中的產品後 AWS Marketplace，客戶應該能夠在 SaaS 應用程式中建立 帳戶，並取得 Web 主控台的存取權。如果客戶無法立即存取應用程式，您必須提供訊息，其中包含他們何時將存取的特定指示。建立帳戶後，客戶必須收到通知，確認其帳戶已建立，並明確執行後續步驟。
- 如果客戶在 SaaS 應用程式中已有 帳戶，他們必須能夠從履行登陸頁面登入。
- 客戶必須能夠在 SaaS 應用程式中查看其訂閱的狀態，包括任何相關的合約或訂閱使用資訊。
- 客戶必須能夠輕鬆取得相關問題的協助，例如：使用應用程式、疑難排解和請求退款 （如適用）。必須在履行登陸頁面上指定支援聯絡選項。
- 產品軟體和中繼資料不得包含將使用者重新導向至其他雲端平台、其他產品、追加銷售服務或無法使用的免費試用優惠的語言 AWS Marketplace。

  如需 SaaS 產品免費試用的相關資訊，請參閱 在 中建立 SaaS 免費試用優惠 AWS Marketplace。
- 如果您的產品是另一個產品或其他 ISV 產品的附加元件，您的產品描述必須指出它擴展了另一個產品的功能，如果沒有它，您的產品具有非常有限的公用程式。例如，此產品擴展了 <product name> 的功能，如果沒有它，此產品的公用程式非常有限。請注意，<product name> 可能需要自己的授權才能使用此清單的完整功能。

架構準則

下列主題會列出並描述 SaaS 產品的架構準則。

主題

- 指導方針
- 建立架構圖

指導方針

> **Note**
>
> 下列指導方針自 2025 年 5 月 1 日起生效。

- 您可以發佈所有 SaaS 架構。
- 部署在 上的產品會在 AWS Marketplace 搜尋結果及其產品詳細資訊頁面中 AWS 收到特殊指定。若要 AWS Marketplace 讓 將您的產品視為「已部署」 AWS，您的產品必須完全執行 AWS。這包括應用程式和控制平面。

應用程式平面可以在賣方 AWS 帳戶、買方 AWS 帳戶或兩者中執行。如需詳細資訊，請參閱控制平面與應用程式平面白皮書。

產品用來傳輸、存放或處理應用程式資料的第三方服務，但內容交付網路 (CDNs)、網域名稱系統 (DNSs) 和公司身分提供者 (IdPs) 除外，也必須完全執行 AWS。

> **Note**
>
> 應用程式資料是屬於買方或為買方產生的資料。

產品用於安全、監控、資料複寫或遷移的代理程式或閘道可以在買方擁有的外部環境中執行 AWS，包括內部部署，但必須僅將資料傳送至 AWS 以進行儲存和分析。

您必須包含架構圖表以供檢閱。您無法公開圖表。如需詳細資訊，請參閱下一節建立架構圖中的 。

- 賣方可以發佈未完全執行的產品 AWS。
- 需要買方基礎設施中資源的應用程式必須遵循下列準則：
  - 若要被視為 SaaS 產品而非受管服務，您的控制平面—如 SaaS 架構基礎 AWS 白皮書所定義—必須位於您管理的基礎設施中。如需詳細資訊，請參閱 SaaS 與受管服務供應商白皮書。
  - 在產品描述中，您必須通知客戶，如果他們在 AWS Marketplace 交易之外產生 AWS 基礎設施費用，他們必須支付這些費用。
  - 您必須以安全的方式佈建資源，例如使用 AWS 安全字符服務 (AWS STS) 或 AWS Identity and Access Management (IAM)。
  - 建立授予應用程式許可的使用說明或部署範本時，必須遵循最低權限原則。
  - 您必須提供其他文件，說明所有佈建 AWS 的服務、IAM 政策陳述式，以及如何在客戶帳戶中部署和使用 IAM 角色或使用者。
  - 您必須提供指示或部署範本，讓買方能夠在其 AWS 帳戶中部署所需的資源。
  - 如果您提供 AWS CloudFormation 範本 (CFTs) 來將資源部署到買方 AWS 的帳戶，則他們必須遵守AWS Marketplace CFTs的政策。您必須遵循為買方啟用 SaaS Quick Launch 部署選項時提供的方法，將這些 CFTs 發佈為 SaaS 清單的一部分。SaaS SaaS Quick Launch 可讓您的買方更輕鬆地設定 SaaS 解決方案。
  - 如果 Amazon Machine Image (AMIs) 部署到買方的 AWS 帳戶中，則他們必須遵守AWS Marketplace AMIs的政策。您的 AMIs通過 管理入口網站 （賣方入口網站） 中的 AMI 掃描器。AWS Marketplace 請求您的產品公開時，您還必須聯絡 AWS Marketplace 操作並提供掃描結果的證明。

  - 如果容器映像部署到買方的 AWS 帳戶中，映像必須符合AWS Marketplace 容器的政策。您的容器映像可以託管在 外部 AWS，但必須在 Amazon Elastic Container Registry (Amazon ECR) 中掃描，並且沒有重大漏洞。請求您的產品公開時，您還必須聯絡 AWS Marketplace 操作，並提供容器通過掃描的證明。
- 從註冊為提供者並提交 SaaS 發佈請求 AWS 帳戶 的 成功呼叫 AWS Marketplace APIs。SaaS 定價模型會決定應呼叫哪些 APIs：
  - SaaS 合約 – 中的 GetEntitlements AWS Marketplace Entitlement Service。
  - 消耗的 SaaS 合約 – 中的 GetEntitlements AWS Marketplace Entitlement Service 和 AWS Marketplace Metering Service 中的 BatchMeterUsage。
  - SaaS 訂閱 – AWS Marketplace Metering Service 中的 BatchMeterUsage。
- 僅在 AWS GovCloud (US) 區域中提供的 SaaS 產品必須解釋其他 AWS 區域 和 AWS GovCloud (US) 區域之間的架構界限、產品的使用案例，以及不建議用於產品的工作負載。

如需 SaaS 架構的詳細資訊，請參閱 SaaS 架構基礎 AWS 白皮書。

建立架構圖

若要接收產品部署所在的特殊指定 AWS，請在 AWS Marketplace 管理入口網站中更新產品的架構詳細資訊。選取部署在 上的託管模式， AWS 並上傳 AWS 檢閱的架構圖。如需考慮部署的 AWS Marketplace 託管模式 AWS，請參閱上一節指導方針中的 。如果您的託管模式變更，您必須更新產品的架構詳細資訊。

> **Note**
>
> - 您無法公開圖表。
> - 您可以發佈未完全執行的產品 AWS。

建立圖表時，請使用下列條件：

- 將元件分組並標記為應用程式平面或控制平面的一部分。
- 對於 AWS 屬於產品核心業務邏輯的任何 外部元件，請將它們與應用程式平面分組。
- 元件可以代表低階詳細資訊（例如，運算執行個體和網路子網路）或高階服務（例如，資料分析平台）。
- 元件不需要識別所使用 AWS 服務或非AWS 服務的名稱。

- 將元件放在邏輯執行的位置。例如，在賣方的 AWS 帳戶中、買方的 AWS 帳戶、賣方的非AWS 環境或其他環境。
- 對於資料複寫或工作負載遷移產品，請包含所有支援的來源和目標環境。

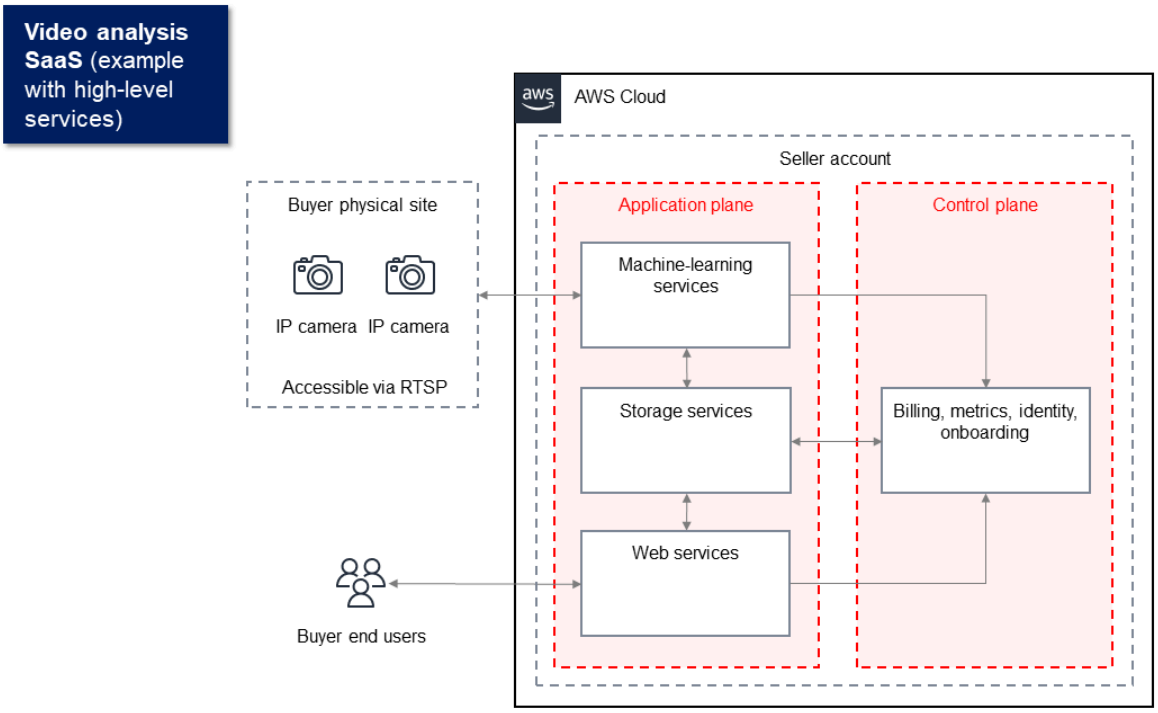
> **Note**
>
> 您用來更新 SaaS 產品的架構詳細資訊的架構圖表不會發佈，也不會公開提供給買方。

詳細資訊層級

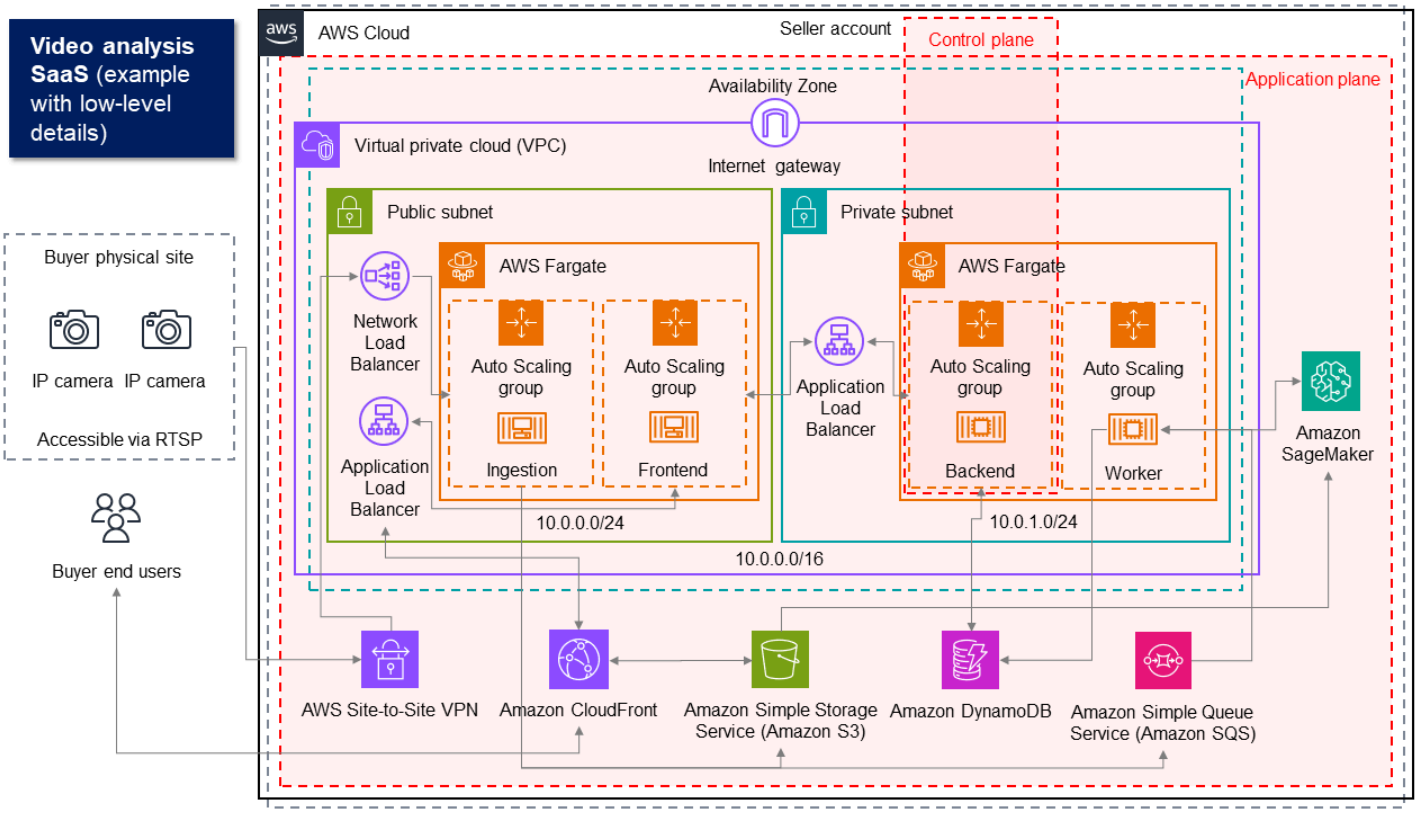
您可以建立高階圖表，顯示主要系統元件、包含基本資料流程，並著重於應用程式平面和控制平面服務。或者，您可以建立低階、詳細的圖表，以細分每個元件、顯示特定連線，並包含具有不同細節層級的技術規格。

下圖顯示假設性影片分析 SaaS 應用程式的架構。每個 顯示不同層級的詳細資訊。兩者都是可接受的。使用它們做為詳細資訊層級的範例，以包含在您自己的圖表中。

以下是高階圖表的範例。



以下是低階、詳細圖表的範例。



如需詳細資訊，請參閱什麼是架構圖表？。建立圖表後，請在 AWS Marketplace 管理入口網站 (AMMP) 中更新您的架構詳細資訊。如需詳細資訊，請參閱更新架構詳細資訊。

中的 SaaS 產品定價 AWS Marketplace

購買者購買軟體後，即服務 (SaaS) 產品 AWS Marketplace， AWS Marketplace 會為您提供其帳單識別符。您可以使用帳單識別符來呼叫 AWS Marketplace Entitlement Service 和 AWS Marketplace Metering Service。然後，客戶會在您的 AWS 環境中或透過您建立的虛擬私有雲端 (VPC) 端點連線存取產品。本主題提供 的 SaaS 定價模型清單 AWS Marketplace。

> **Note**
>
> 所有 SaaS 定價模型都支援免費試用。如需詳細資訊，請參閱 SaaS 免費試用版。

SaaS 定價模型

定價方式	描述
SaaS 訂閱	pay-as-you-go模型，我們針對買方每小時使用您的 SaaS 產品向買方收費。如需詳細資訊，請參閱SaaS 訂閱的定價。
SaaS 合約	買方會預先收到使用您軟體的帳單，或者您可以為他們提供彈性的付款排程。客戶也可以支付超出合約的額外用量。如需詳細資訊，請參閱軟體 SaaS 合約定價。
SaaS 合約與 pay-as-you-go	買方會預先收到使用您軟體的帳單，或者您可以為他們提供彈性的付款排程。除了合約價格之外，買方還需支付額外的計量費率。如需詳細資訊，請參閱軟體 SaaS 合約定價。

> **Note**
>
> 建立清單並將其發佈至有限之後，您就無法變更定價模型。

若要讓 SaaS 產品在 上可用 AWS Marketplace，請決定您要提供 SaaS 訂閱定價模型或 SaaS 合約定價模型。

SaaS 訂閱的定價

對於軟體即服務 (SaaS) 訂閱， 會根據您傳送給我們的計量記錄向您的客戶 AWS Marketplace 計費。所有費用都必須從客戶帳戶中部署的軟體每小時測量和報告。然後，所有用量都會每月計算，並使用與 AMI 型 AWS Marketplace 產品相同的機制每月計費。我們向客戶收取產品用量帳單的能力取決於接收您的計量記錄。您有責任確保產品計量記錄已成功傳輸和接收。

您必須先執行下列動作，才能發佈具有訂閱定價的 SaaS 產品：

1. 在 中建立新的 SaaS 產品 AWS Marketplace 管理入口網站，選擇新 SaaS 訂閱。
2. 使用必要資訊完成一般索引標籤中的欄位。記下產品代碼。

3. 在定價索引標籤的設定定價下，選取描述您產品定價最精確的類別。定價類別會顯示在 AWS Marketplace 網站上的客戶。您可以選擇頻寬 (GBps、MBps)、資料 (GB、MB、TB)、主機 （小時）、請求或使用者 （小時）。如果沒有預先定義的類別符合您的需求，您可以選擇更通用的單位類別。

   接著，定義您的定價維度。每個定價維度代表您可以設定每個單位價格的功能或服務。維度的範例包括使用者、掃描的主機和擷取的 GB 日誌。您最多可以定義 24 個維度。對於您定義的每個維度，您必須新增下列資訊：

   - Dimension API 名稱 – 將計量記錄傳送到 時使用的 API 名稱AWS Marketplace Metering Service。此名稱指出您的客戶使用的維度。此名稱會顯示在帳單報告中。名稱不需要便於讀取，因為您是唯一可存取報告的人。設定名稱後，您無法變更名稱。
   - 維度描述 – 面向客戶的陳述式，描述產品的維度。描述不能超過 70 個字元，而且應該易於使用。描述的範例為每小時的管理員，以及佈建的每 Mbps 頻寬。產品發佈後，您可以變更此描述。
   - 維度率 – 此產品每個 FCP 單位的軟體費用，以 USD 為單位。此欄位支援八個小數位數。

當 SaaS 訂閱結束時

客戶可以透過 取消訂閱您的 SaaS 訂閱產品 AWS Management Console。SaaS 訂閱結束程序的重點包括下列項目：

1. 您的 SaaS 產品會透過該客戶的 Amazon SNS 主題傳送`unsubscribe-pending`通知。
2. 您有一小時的時間來衡量客戶的任何剩餘用量。
3. 在此小時之後，您會收到`unsubscribe-success`通知。此時，您無法再傳送此客戶的計量記錄。

您可以自行決定如何為取消訂閱的客戶停用 SaaS 產品中的功能。例如，您的產品可能會完成客戶現有的工作，但無法建立工作。您可能想要向客戶顯示一則訊息，說明其用量已停用。客戶可以透過 重新訂閱您的產品 AWS Marketplace。

取消 SaaS 訂閱時

SaaS 訂閱取消程序的重點包括下列項目：

1. 客戶可以在 AWS Marketplace 網站的您的 Marketplace 軟體頁面取消其 SaaS 訂閱產品的訂閱。

   您的 SaaS 產品會透過該客戶的 Amazon SNS 主題傳送通知。
2. 您有一小時的時間來衡量客戶的任何剩餘用量。

3. 您會通知產品中的客戶取消正在進行中。如果客戶表示他們想要透過您的產品取消，請將客戶導向 AWS Marketplace。為了確保未來不會產生任何費用，客戶應該向 確認取消 AWS Marketplace。

軟體 SaaS 合約定價

針對軟體即服務 (SaaS) 合約，客戶會開始購買您的軟體，並與您達成協議。根據協議，客戶有權使用您的 SaaS 產品的指定數量。 AWS Marketplace 將這些權利傳達給您的 SaaS 應用程式。這是透過 AWS Marketplace Entitlement Service. 使用 SaaS 合約定價模型時，您的應用程式永遠不會傳送計量記錄。相反地，它會透過呼叫. AWS Marketplace Entitlement Service您可以定義使用關係類別、維度和合約長度。

AWS Marketplace 根據您與客戶之間的合約，預先或依據您定義的付款排程，向客戶開立帳單。在那之後，他們有權使用這些資源。對於超出合同的額外使用，您的軟件需要報告使用情況，並根據我們通過我們收到的計量記錄向客戶收到的計量記錄向客戶收取 AWS Marketplace 費用 AWS Marketplace Metering Service。

您必須先執行下列動作，才能發佈具有合約定價的 SaaS 產品：

1. 在中建立新的 SaaS 產品 AWS Marketplace 管理入口網站，然後選擇 [新增 SaaS 合約]。
2. 在「一般」標簽中填寫相應的字段，并包含必要的信息。記下產品代碼。
3. 在「訂價」頁標上：
   a. 針對「設定訂價」，選擇您要提供客戶的「合約期間」。您可以為每個合約期間輸入不同的價格。您可以選取下列一或多個選項：「每月」、「1 年」、「2 年」和「3 年」。如果您要建立私人選件，您可以選擇以月為單位的自訂期間 (最多 60 個月)。
   b. 針對 「選擇您要提供的合約類型」，請從下列選項中選擇您希望客戶購買產品的方式：
      - 買家可以選擇一個或多個提供的選項 — 客戶可以為您提供的每個定價範圍選擇一個數量。
      - 買家可以從提供的多個級別中選擇一個級別 — 客戶可以從包含不同功能集，服務和用量的選項中選擇一個級別。
   c. 選擇最準確地描述產品價格的使用單位類別。定價類別會在 AWS Marketplace 網站上顯示給客戶。您可以選擇頻寬 (GBps,MBps)、資料 (GB、MB、TB)、主機 (小時)、要求或使用者 (小時)。如果沒有任何預先定義的類別符合您的需求，您可以選擇更通用的「單位」類別。
4. 選擇類別後，請定義您的「定價範圍」。每個「訂價範圍」代表您可以為其設定每單價的功能或服務。維度範例包括使用者、已掃描的主機以及擷取的 GB 記錄。對於您定義的每個維度，您可以新增名稱、說明、價格和API名稱。系統會向客戶顯示名稱、價格和描述。您可以使用API名稱進行追蹤和報告， AWS Marketplace 如下所示：
   - 呼叫AWS Marketplace Entitlement Service以擷取客戶購買的維度。

- 呼叫AWS Marketplace Metering Service以指出客戶使用的維度。

對於合約中的每個定價維度，您可以選擇讓客戶在合約之上額外使用該維度的時間付費。您也可以新增其他維度 (不含合約價格)，客戶只需支付即可消費。

使用精靈為 SaaS 產品建立合約時，您必須為定價維度定義下列欄位：

- 維度API名稱 — 呼叫權利API時使用的名稱。此名稱會顯示在非面向外部的帳單報表和報表中。API名稱的長度上限為 15 個字元。設定名稱後，就無法變更。
- 維度顯示名稱：— 維度的客戶對應名稱。此名稱應該可協助客戶瞭解產品的維度。該名稱應該是用戶友好的，其最大長度為 24 個字符。此值可以變更。
- 維度說明：— 維度的面向客戶說明，提供有關產品維度的其他資訊。說明的長度上限為 70 個字元。
- 維度-每月價格 — 此維度 1 個月選項的每單位軟體費用。此場支持三位小數。
- 維度-1 年價格 — 此維度 12 個月選項的每單位軟體費用。此場支持三位小數。這不是每月的費用。價格必須符合 12 個月的一次性收費價格。
- 維度-2 年價格 — 此維度 24 個月選項的每單位軟體費用。此場支持三位小數。
- 維度-3 年價格 — 此維度 36 個月選項的每單位軟體費用。此場支持三位小數。

範例：資料儲存應用

	每月價格	12 個月價格	24 個月價格	額外使用的 P ay-as-you-go 價格
未加密資料 (GB)	每 GB 1.50 美元	每秒 16 美元	每公升 30 美元	每小時 0.1/GB
加密資料 (GB)	每秒 1.5 美元	每秒 16 美元	每公升	每小時 0.11 美元

範例：記錄監控產品

	每月價格	12 個月價格	額外使用的 P ay-as-you-go 價格
基本 (10 部受監控的主機，5 個受監控的容器)	一百元	1000 美元	

	每月價格	12 個月價格	額外使用的 P ay-as-you-go 價格
標準 (20 台受監控的主機，10 個受監控的容器)	$200	二千元	
Pro (40 部受監控的主機，監控 20 個容器)	400 美元	4000 美元	
每小時監控的其他主機			0.1 美元
每小時監控額外的容器			0.2 美元

> **Note**
>
> 價格可以是以下持續時間：1 個月，12 個月，24 個月或 36 個月。您可以選擇為您的產品提供一或多個這些選項。每個維度的持續時間必須相同。例如，假設您具有`ReadOnlyUsers`和`AdminUsers`維度。如果您提供的年度價格`ReadOnlyUsers`，您也必須提供年度價格。`AdminUsers`

SaaS 合約升級

除了較長的期限外，客戶可以將合約升級為較高價值之一。例如，他們可以升級到更高數量或更高價值的權利。客戶會獲得其現有合約的攤派貸方。客戶無法減少現有合約的規模。他們只能在續訂時減小尺寸，或取消續訂。

權利是由您的 SaaS 產品進行驗證，這會撥打. AWS Marketplace Entitlement Service

自動續約

當客戶透過 AWS Marketplace SaaS 合約購買您的產品時，他們可以同意自動續約合約條款。客戶會繼續支付每月或 1 年、2 年或 3 年的授權費用。客戶一律可以選擇修改續訂設定。他們可以根據不同的數量和持續時間取消續訂或續訂合同。

當 SaaS 合約結束時

SaaS 合約產品的合約到期日。合約結束時，會發生下列事件：

1. 您的 SaaS 產品會收到`entitlement-updated`通知，指出買家的權益已變更。會 AWS Marketplace Entitlement Service 傳回空白回應。
2. 您有 1 小時為客戶計量任何剩餘使用量。過了這段時間之後，您就無法再傳送此客戶的計量記錄。

當 SaaS 合同被取消時

SaaS 合同取消過程的要點包括以下內容：

1. 但是，客戶可以要求取消和退款 SaaS 合同產品 AWS 支援。

   客戶必須在 48 小時內通過申請退款 AWS 支援。

   全額或按比例退款通常在 3-5 個工作日內給予。
2. 您的 SaaS 產品會透過 Amazon SNS 主題為該客戶傳送通知。
3. 如有任何額外的使用費，您有一小時的時間向客戶傳送最後的計量記錄。
4. 您會從您的產品通知客戶取消作業正在進行中。如果客戶指出要透過您的產品取消，請將客戶導向至 AWS Marketplace。為了確保 future 不會產生任何費用，客戶應確認取消。 AWS Marketplace

在 中建立 SaaS 免費試用優惠 AWS Marketplace

身為 AWS Marketplace 賣方，您可以在 (AMMP) 中建立軟體即服務 AWS Marketplace 管理入口網站 (SaaS) 免費試用優惠。客戶可以使用 SaaS 免費試用選項，在做出大型購買決策之前評估軟體產品。客戶訂閱您的產品後，您的產品會執行與付費客戶相同的權利檢查。

每個 AWS 帳戶 只能對 SaaS 產品使用一次免費試用。免費試用期間授予的免費用量不會在 AWS 組織中連結帳戶之間共用。單一主要付款人帳戶中的不同連結帳戶可以建立自己的個別免費試用。

> **Note**
>
> - 如果您使用 Seller Data Delivery Service (SDDS)，您會在 Amazon Simple Storage Service 儲存貯體中收到協議詳細資訊試驗報告。報告包含協議詳細資訊，例如訂閱者名稱和 ID、優惠 ID，以及協議開始和結束日期。

- 建立新的訂閱時，賣方也會收到 Amazon Simple Notification Service (Amazon SNS) 通知。Amazon SNS 通知包含識別免費試用協議的 `isFreeTrialTermPresent` 旗標。
- 此外，訂閱免費試用的客戶會使用額外的字符 重新導向至您的註冊 URL`x-amzn-marketplace-offer-type=free-trial`。您可以使用權杖為使用免費試用的客戶建立唯一的註冊體驗。

建立 SaaS 免費試用優惠

賣方可以在 AWS Marketplace 管理入口網站 (AMMP) 中建立 SaaS 免費試用優惠。

建立 SaaS 免費試用優惠

1. 登入 AWS Marketplace 管理入口網站。
2. 在 上 AWS Marketplace 管理入口網站，選擇：
   - 建立或管理優惠
   - 優惠索引標籤
3. 在優惠頁面上，選擇公開免費試用索引標籤，以檢閱所有 SaaS 免費試用。
4. 選擇建立免費試用優惠。賣方可以為每個公有 SaaS 產品建立一個 SaaS 免費試用優惠。
5. 針對優惠基本概念，選取您的產品，然後選擇下一步。
6. 在免費試用設定中：
   a. 輸入免費試用長度的天數（天）。

      免費試用的持續時間從 7 到 90 天不等。

   b. 檢視現有公開優惠中的產品維度。

      您無法變更 SaaS 訂閱免費試用的產品維度。

      您可以為 SaaS 合約免費試用設定每個維度的數量限制，以及移除或新增維度。
7. 檢視服務協議。

   對於 EULA 版本，您可以選擇適用於 的標準合約 AWS Marketplace或自訂 EULA，然後選擇檢閱優惠。
8. 驗證並檢閱優惠的所有資訊，然後選擇建立優惠。

取消 SaaS 免費試用優惠

賣方可以隨時從 取消免費試用優惠 AWS Marketplace 管理入口網站。

取消 SaaS 免費試用優惠

1. 登入 AWS Marketplace 管理入口網站。
2. 在 上 AWS Marketplace 管理入口網站，選擇：
   - 建立或管理優惠
   - 優惠索引標籤
3. 在優惠頁面上，選取優惠。
4. 選擇檢視優惠。
5. 選擇取消優惠。

優惠取消後，此優惠的作用中協議會一直有效，直到過期為止。無法建立已取消優惠的新協議。

透過 將客戶加入您的 SaaS 產品 AWS Marketplace

透過軟體即服務 (SaaS) 訂閱和 SaaS 合約，您的客戶可以透過 訂閱您的產品， AWS Marketplace 但存取您 AWS 環境中的產品。訂閱產品後，您的客戶會被導向至您建立和管理的網站，做為 SaaS 產品的一部分，以註冊其帳戶並設定產品。

建立 SaaS 產品清單時，您會提供註冊登陸頁面的 URL。我們會在客戶訂閱後，使用該 URL 將客戶重新導向至您的註冊登陸頁面。在軟體的註冊登陸頁面上，您可以收集為客戶建立 帳戶所需的資訊。如果您打算透過電子郵件聯絡客戶以取得用量通知，建議您收集客戶的電子郵件地址。

註冊登陸頁面必須能夠識別和接受來自 的表單資料中的`x-amzn-marketplace-token`字符， AWS Marketplace 以及客戶的帳單識別符。然後，它應該將該字符值傳遞至 ， AWS Marketplace Metering Service 以解析唯一客戶 AWS 帳戶 ID、客戶識別符 （已於 2025 年 12 月 31 日之後棄用） 和對應的產品代碼。如需程式碼範例，請參閱「`ResolveCustomer` 程式碼範例」。

> **Note**
>
> 註冊字符會解析為特定訂閱的客戶，而每個產生的字符都有 4 小時的過期時段。只要呼叫者使用相同的字符呼叫 API，它就會繼續傳回相同的回應值，直到字符過期為止。

設定您的 SaaS 產品以接受新買方

您有責任正確設定 SaaS 軟體，以接受新客戶並適當計量新客戶。下列程序概述了識別、實作和計量新客戶存取軟體的一種建議方法：

1. 當客戶造訪您 AWS Marketplace 網站上的產品頁面時，他們選擇訂閱您的產品。
2. 客戶的 AWS 帳戶 已訂閱您的產品。這表示從您的產品傳送的訂閱和計量記錄會成為客戶 AWS 帳單的一部分。
3. 系統會為客戶產生註冊字符，其中包含其 AWS 帳戶 ID、客戶識別符和您的產品代碼。
4. 客戶會重新導向至軟體的註冊登陸頁面。此頁面必須能夠接受具有客戶識別符的字符。
5. 客戶的瀏覽器會將POST請求傳送至軟體的註冊登陸頁面 URL。請求包含一個POST參數 `x-amzn-marketplace-token`，其中包含客戶的註冊字符。從註冊網站的觀點來看，客戶已提交具有此參數的表單。註冊字符是不透明字串。如果優惠類型是免費試用，則第二個參數`x-amzn-marketplace-offer-type`的值為 `free-trial`，將會新增至請求。
6. 若要將此註冊字符兌換為客戶 AWS 帳戶 ID、客戶識別符和產品代碼，您的網站必須在 AWS Marketplace Metering Service 上呼叫 ResolveCustomer。如需 `ResolveCustomer` 呼叫的範例，請參閱 `ResolveCustomer` 程式碼範例。客戶識別符不是客戶的 AWS 帳戶 ID，但其在產品之間是通用的，應該儲存到內部來源做為客戶記錄的一部分。產品程式碼是 AWS 提供給您的 SaaS 產品的唯一字串。每個 AWS 產品都有一個唯一的產品代碼，該代碼會在註冊期間指派給您。
7. 指示客戶在您的產品中建立 帳戶或登入現有的 帳戶。

   > **Note**
   >
   > 如果設定或連結至您產品中的現有客戶帳戶需要您團隊的手動程序，您可以使用聯絡我們表單來收集客戶的聯絡資訊。在收集他們的聯絡資訊並解析他們的 AWS 帳戶 ID 和唯一的客戶識別符（如步驟 6 中所取得）之後，顯示客戶的通知訊息。在通知中，說明他們的帳戶正在設定中，並要求他們等待您聯絡他們。提供客戶預期的周轉時間和您的聯絡資訊。同時傳送具有相同詳細資訊的電子郵件訊息給客戶。

8. 客戶現在已使用該 SaaS 產品特定的登入資料登入您的網站。在您的帳戶資料庫中，您可以為每個客戶有一個項目。您的帳戶資料庫必須具有 AWS 帳戶 ID 的資料欄。驗證系統中沒有其他帳戶共用 AWS 帳戶 ID。
9. 在您的賣方註冊程序期間，您會訂閱 Amazon SNS 主題，當客戶訂閱或取消訂閱您的產品時通知您。以下是 JSON 格式的 Amazon SNS 通知，可通知您客戶動作：

- 權利通知 – 對於具有包含合約定價模型的產品，當買方建立新合約、升級合約、續約合約或合約過期時，您會收到通知。您的帳戶資料庫必須具有訂閱狀態的額外資料欄。如需詳細資訊，請參閱Amazon SNS 主題： aws-mp-entitlement-notification。
- 訂閱通知 – 對於具有任何定價模式的產品，包括合約和訂閱，當買方訂閱或取消訂閱產品時，您會收到通知。如需詳細資訊，請參閱Amazon SNS 主題： aws-mp-subscription-notification。

我們建議您使用 Amazon Simple Queue Service (Amazon SQS) 來擷取這些訊息。收到使用 的訂閱通知後subscribe-success，客戶帳戶已準備好進行計量。系統不會計量您在此通知之前傳送的記錄。如需如何執行此操作的資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的步驟 2：授予 Amazon SNS 主題傳送訊息至 Amazon SQS 佇列的許可。

> **Note**
>
> 除非您收到subscribe-success通知，否則請勿啟用產品訂閱。

10. 使用存放在資料庫中的 AWS 帳戶 ID，透過 AWS Marketplace Metering Service 來計量用量，或透過 檢查權利 AWS Marketplace Entitlement Service。

安全性和訂購

身為賣方，您有責任信任從 或系統簽署 AWS 的 AWS 帳戶 IDs。我們建議您立即解析註冊字符，因為它可能會在大約一小時後過期。解析註冊字符之後，請將 AWS 帳戶 ID 儲存為客戶瀏覽器工作階段上已簽章的屬性，直到註冊完成為止。

SaaS 產品的 Amazon SNS 通知

若要接收通知，您可以訂閱產品建立期間 AWS Marketplace 提供給您的 Amazon Simple Notification Service (Amazon SNS) 主題。這些主題提供有關客戶訂閱的變更和您產品的合約權利的通知。這可讓您知道何時提供和撤銷特定客戶的存取權。

> **Note**
>
> 在產品建立過程中，您將收到 SNS 主題的實際 Amazon Resource Name (ARN)。例如：`arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`

下列 Amazon SNS 主題適用於軟體即服務 (SaaS) 產品：

- Amazon SNS 主題： `aws-mp-entitlement-notification` – 本主題會在買方建立新合約、升級合約、續約合約或合約過期時通知您。這僅適用於定價模型包含合約 （ 也稱為 SaaS 合約和 SaaS 合約使用量 （ 超額 ）) 的產品。
- Amazon SNS 主題： `aws-mp-subscription-notification` – 本主題會在買方訂閱或取消訂閱產品時通知您，並包含`offer-identifier`適用於私有優惠的 和適用於 SaaS 免費試用版的免費試用旗標。這適用於所有定價模式，包括合約和訂閱 （ 也稱為 SaaS 訂閱、SaaS 合約和 SaaS 使用合約 （ 超額 ）。

若要進一步了解您回應這些通知的案例，請參閱下列主題：

- 將您的 SaaS 訂閱產品與 整合 AWS Marketplace
- 將您的 SaaS 合約產品與 整合 AWS Marketplace
- 將您的 SaaS 合約與pay-as-you-go產品與 整合 AWS Marketplace

Amazon SNS 主題： `aws-mp-entitlement-notification`

`aws-mp-entitlement-notification` 主題中的每個訊息都有下列格式。

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXX",
}
```

`<action-name>` 一律為 `entitlement-updated`。

> **Note**
>
> - 對於權利訊息，無論動作為何（新、升級、續約或過期），訊息都是相同的。GetEntitlement 需要後續呼叫 ，才能探索更新的內容。
> - 對於使用（超額）的 SaaS 合約，賣方會與 aws-mp-subscription-notification SNS 主題一起提供。這是賣方新增超額定價時收到的額外通知。當賣方取得新客戶時，而不是只取得 entitlement-updated（可能是指任何類型的動作），賣方會收到訂閱訊息，指出這是新客戶。
> - 對於未來日期的協議 (FDAs)，此主題會在協議開始日期（而非協議簽署日期）啟動。當權利發生後續變更時，例如取消、取代、續約或協議到期，也會啟動。

具有合約定價的產品（包括pay-as-you-go的合約）必須回應這些訊息。如需如何回應的詳細資訊，請參閱 案例：監控使用者訂閱的變更。

Amazon SNS 主題： **aws-mp-subscription-notification**

aws-mp-subscription-notification 主題中的每個訊息都有下列格式。

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123",
    "isFreeTrialTermPresent":"true"
}
```

只有在動作為 subscribe-success或 時， offer-identifier才會包含在通知中subscribe-fail。當動作為 unsubscribe-pending或 時，它不會包含在通知中unsubscribe-success。對於 2024 年 1 月之前建立的優惠，此識別符僅包含在私有優惠的通知中。對於 2024 年 1 月及更新版本建立的優惠，此識別符包含在所有優惠的通知中，包括私有優惠和公開優惠。

如需優惠類型的資訊，請參閱 DescribeEntity API 的回應或協議續約儀表板中的協議優惠可見性。

> **Note**
>
> 對於 DescribeEntity API，如果您 AWS 帳戶 在以該優惠規則為目標的面向帳戶中找到 ，則這是私有優惠。如果 AWS 帳戶 帳戶中沒有以該優惠規則為目標的面向為目標的 ，則這是公開優惠。

`isFreeTrialTermPresent` 屬性會指出買方的訂閱是否為免費試用。此屬性的 JSON 值不是布林資料類型。反之，值會轉換為字串資料類型。如需詳細資訊，請參閱 SaaS 免費試用。

`<action-name>` 會根據通知而有所不同。可能的動作包括：

- `subscribe-success` – 賣方何時可以開始傳送計量記錄`subscribe-success`的訊息發出訊號。如果買方接受以協議為基礎的優惠，此訊息會再次與新的 一起傳送`offer-identifier`。
- `subscribe-fail` – 如果產生`subscribe-fail`訊息，即使買方已從 轉換 AWS Marketplace 到賣方的 SaaS 登陸頁面，付款仍可能失敗。在允許使用產品之前，賣方應等待`subscribe-success`訊息。
- `unsubscribe-pending` – 當買方取消訂閱時，`unsubscribe-pending`會先傳送訊息。這表示賣方在完全取消買方之前，取得最終計量記錄的時間有限（約一小時）。
- `unsubscribe-success` – `unsubscribe-success` 訊息表示取消完成，之後將不再接受進一步的計量記錄。

> **Note**
>
> - 如果買方取消訂閱，然後在傳送最終`unsubscribe-success`訊息之前立即成功重新訂閱，則不會傳送最終`unsubscribe-success`訊息，而是`subscribe-success`傳送訊息。
> - 對於未來日期的協議 (FDAs)，`subscribe-success`動作會在協議開始日期（而非協議簽署日期）啟動。

具有訂閱定價的產品（包括pay-as-you-go的合約）必須回應這些訊息。如需如何回應的詳細資訊，請參閱下列主題：

- 將您的 SaaS 訂閱產品與 整合 AWS Marketplace
- 將您的 SaaS 合約與pay-as-you-go產品與 整合 AWS Marketplace

訂閱 SQS 佇列至 SNS 主題

我們建議訂閱 Amazon SQS 佇列至提供的 SNS 主題。如需建立 SQS 佇列和訂閱佇列至主題的詳細說明，請參閱《Amazon Simple Notification Service 開發人員指南》中的訂閱 Amazon SQS 佇列至 Amazon SNS 主題。

> **Note**
>
> 您只能從 AWS 帳戶 用來銷售產品的 訂閱 AWS Marketplace SNS 主題。不過，您可以將訊息轉送到不同的 帳戶。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的將 Amazon SNS 訊息傳送至不同帳戶中的 Amazon SQS 佇列。

輪詢 SQS 佇列以取得通知

將 SQS 佇列訂閱 SNS 主題後，訊息會儲存在 SQS 中。您必須定義持續輪詢佇列、尋找訊息並相應地處理它們的服務。

存取 AWS Marketplace 計量和權利服務 APIs

本節概述與 AWS Marketplace Metering Service 或 整合的程序 AWS Marketplace Entitlement Service，用於確保您的軟體即服務 (SaaS) 產品的客戶用量帳單和報告準確無誤。假設您已提交 SaaS 訂閱產品或已發佈至有限狀態的 SaaS 合約產品。在有限狀態下，您可以使用測試帳戶來驗證適當的組態和函數，但您的產品無法公開使用。

> **Note**
>
> 如果您的 SaaS 產品與其他受 AWS 管服務整合，以不同方式處理計量 （ 例如 Amazon SageMaker Ground Truth 或 AWS WAF)，則您不需要與 AWS Marketplace 計量服務整合。產品計量應該只發生在一個系統中，以避免重複向您的客戶收費。

主題

- 使用 SaaS 訂閱設定用量計量
- 使用 檢查權利 AWS Marketplace Entitlement Service
- SaaS 產品整合檢查清單

如需設定 AWS CLI和 登入資料的相關資訊，請參閱AWS Command Line Interface 《 使用者指南》中的設定 AWS CLI。如果您是 AWS Python SDK 的新手，請參閱 Boto 3 Quickstart。

使用 SaaS 訂閱設定用量計量

對於軟體即服務 (SaaS) 訂閱，您可以計量所有用量，然後 AWS 根據您提供的計量記錄向客戶計費。對於 SaaS 合約，您只能測量超出客戶合約權利的用量。下列各節提供如何設定 SaaS 產品用量計量的相關資訊。

當您的應用程式測量客戶的用量時，您的應用程式會提供累積的用量 AWS 數量。您的應用程式會測量您在建立產品時定義的定價維度，例如傳輸的 GB 或指定小時內掃描的主機。例如，如果您根據傳送至應用程式的資料量收費，您可以測量資料量，每小時傳送一次對應的計量記錄。 會使用計量資料以及您在建立產品時提供的價格來 AWS 計算客戶的帳單。

> **Note**
>
> 或者，您可以將用量分割為您追蹤的屬性。這些屬性會以標籤的形式公開給買方。這些標籤可讓買方檢視依標籤值分割為用量的成本。例如，如果您由 使用者收費，且使用者具有 `Department` 屬性，則可以使用索引鍵為 的標籤建立用量配置`Department`，每個值一個配置。這不會變更您報告的價格、維度或總用量，但可讓您的客戶依適合您產品的類別檢視其成本。如需詳細資訊，請參閱廠商計量標記 （ 選用 ）。

主題

- 每小時計量
- 設定您的產品以計量用量
- 廠商計量標記 （ 選用 ）

每小時計量

建議您 AWS 每小時向 報告所有客戶的用量，一次批次最多 25 個。這可讓客戶盡可能精細地了解其用量和成本。如果您在超過一小時的時段 （ 例如一天 ） 彙總用量，請注意下列考量。

- AWS 只有在收到來自您的計量記錄時， 才能向客戶收取產品用量的帳單。您有責任確保您產品的計量記錄已成功傳輸和接收。您可以使用 AWS CloudTrail 來驗證您傳送的記錄是否正確。您也可以使用資訊在一段時間內執行稽核。如需詳細資訊，請參閱使用 記錄 AWS Marketplace 計量 API 呼叫 AWS CloudTrail。

- 如果這是具有定價模式 "Subscription" （非定價模式 "Contract" 或 "Contract with Consumption") 的 SaaS，則買方可以隨時取消訂閱。當買方啟動此取消訂閱動作時，賣方會收到`unsubscribe-pending`通知，並有 1 小時的時間在最終`unsubscribe-success`通知之前傳送所有未報告的用量。第二次通知之後的任何內容都不會被接受。其他兩個定價模型具有根據訂閱時間設定的持續時間，買方無法在訂閱期間取消訂閱。他們只能關閉自動續約。如果未自動續約，則會在該持續時間結束時傳送相同的通知。
- 如果您未每小時傳送計量記錄，且發生應用程式或網路中斷，您的記錄將會進一步落後。如果在訂閱過期後還原應用程式或網路中斷，這可能會導致未報告的用量。
- 即使沒有要報告的用量，您也可以繼續每小時傳送計量記錄，如果該小時沒有要報告的用量，則記錄 0 的數量。請注意，在您報告買方的維度用量 0 或以上後，您就無法修改記錄。因此，最佳實務是報告前一小時的用量。在整個計費週期中，買方的 AWS Billing and Cost Management 主控台中會顯示回報的使用量。主控台會在個別明細項目中顯示產品的每個維度，其中包含消耗的單位總數，以及該計費週期到目前為止消耗的明細項目總成本，即使維度價格為每單位 0 美元。如果啟用，AWS 成本和用量報告 (CUR) 會顯示此詳細資訊，包括每單位費率。這些報告不會即時更新。
- 在發佈期間， AWS Marketplace 操作團隊會先測試 SaaS 應用程式是否成功傳送計量記錄，再允許產品發佈。一般而言，團隊會執行 SaaS 的模擬註冊，並確認收到計量記錄。

> **Note**
>
> 如果您的 SaaS 產品與以不同方式 （例如 Amazon SageMaker Ground Truth 或 AWS WAF) 處理計量的另一個 AWS 受管服務整合，則您不需要與 AWS Marketplace 計量服務整合。您產品的計量應該只發生在一個系統中，以避免重複向您的客戶收費。請注意， 目前 AWS Marketplace 不會發佈 AWS WAF 新產品。

設定您的產品以計量用量

您可以使用 AWS Marketplace Metering Service 中的 `BatchMeterUsage`操作來交付計量記錄 AWS。請謹記以下幾點：

- 我們要求賣方使用 `BatchMeterUsage`操作來使用批次處理。
- 我們刪除了該小時的重複計量請求。
  - 每個product/customer/hour/維度的請求都會刪除重複。
  - 您可以隨時重試任何請求，但如果計量不同的數量，則會對原始數量收費。
  - 如果您為相同的customer/dimension/hour傳送多個請求，則不會彙總記錄。

- 如果客戶訂閱您的產品，賣方可以傳送時間戳記最長為過去 6 小時的計量記錄。如果客戶取消訂閱，賣方必須在客戶取消訂閱後的 1 小時內傳送計量記錄。
- BatchMeterUsage 承載不得超過 1MB。選擇要在BatchMeterUsage請求中傳送的用量記錄數量，以免超過承載的大小。
- AWS Marketplace Metering Service 可在 AWS 一般參考中AWS Marketplace 端點和配額中 AWS 區域 列出的 中使用。根據預設，當您請求產品時，SaaS 計量產品會啟用美國東部 （維吉尼亞北部） 區域。如果您想要使用其他區域，請聯絡AWS Marketplace 賣方營運團隊。如需詳細資訊，請參閱 BatchMeterUsage。

如需程式碼範例，請參閱「SaaS 產品整合的程式碼範例」。

範例：主機掃描

您的產品會分析運算硬體是否有已知的安全漏洞。客戶手動啟動或排程其 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的這些掃描。當您的產品執行這些掃描時，它會增加每小時掃描的唯一主機數量。在此範例中，您的產品使用 Hosts 類別。您可以為掃描的主機類型宣告多個維度。例如，您可以針對小型、中型和大型主機收取不同的價格。

範例：日誌分析

您的 SaaS 產品會擷要客戶產品、報告趨勢和異常所產生的日誌。當客戶將日誌上傳至您的產品時，您會測量以 MB、GB 或 TB 為單位接收的資料數量。在每小時的第十分鐘，cron 任務會讀取前一小時每位客戶的此用量。任務會建置批次報告，並使用 BatchMeterUsage操作將其傳送至其中 AWS。在此範例中，您的產品使用 資料類別。您的產品也可以測量任何指定小時內存放的日誌資料量。在這種情況下，您的產品可以沿著兩個維度計量：每小時接收的資料和每小時存放的總資料。您可以繼續計量儲存的資料，直到客戶刪除此資料或資料過期為止。

廠商計量標記 （選用）

廠商計量標記可協助獨立軟體廠商 (ISVs) 為買方提供更精細的軟體用量洞察，並協助他們執行成本分配。

標記買方軟體用量的方法有很多。其中一種方法是先詢問您的買方，他們想要在成本分配中看到什麼。然後，您可以將用量分割到您追蹤買方帳戶的屬性。屬性的範例包括 Account ID、Cost Centers、 Business Unit和您產品的其他相關中繼資料。這些屬性會以標籤的形式公開給買方。使用標籤，買方可以在帳單 AWS 主控台 (https://console.aws.amazon.com/costmanagement/：//) 中檢視依標籤值分割為用量的成本。廠商計量標記不會變更您報告的價格、維度或總用量。它可讓您的客戶依適合您產品的類別檢視其成本。

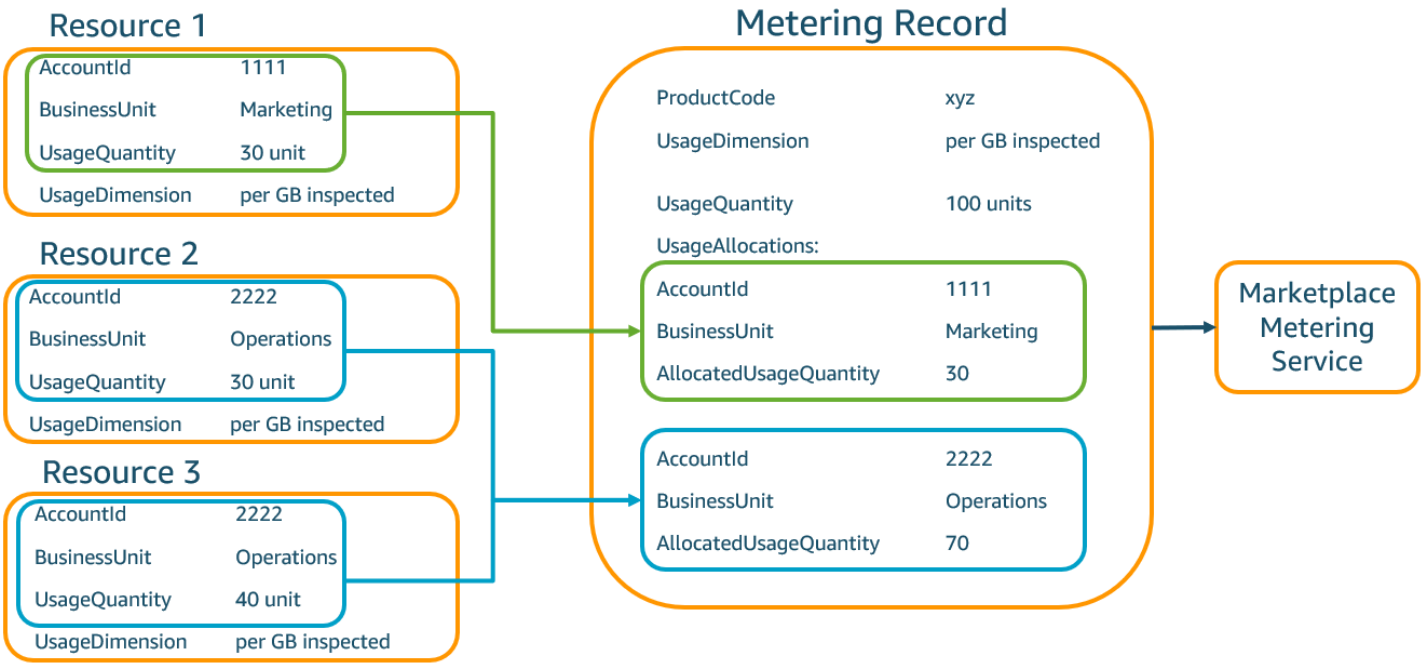
在常見的使用案例中，買方使用 訂閱您的產品 AWS 帳戶。買方也有許多與相同產品訂閱相關聯的使用者。您可以使用索引鍵為 的標籤建立用量配置Account ID，然後將用量配置給每個使用者。在此情況下，買方可以在帳單和成本管理主控台中啟用Account ID標籤，並分析個別使用者用量。

賣方體驗

賣方可以使用相同的標籤集彙總資源的計量記錄，而不是彙總所有資源的用量。例如，賣方可以建構包含不同 儲存貯體的計量記錄UsageAllocations。每個儲存貯UsageQuantity體代表一組標籤，例如 AccountId和 BusinessUnit。

在下圖中，資源 1 具有一組唯一的 AccountId 和 BusinessUnit標籤，並在計量記錄中顯示為單一項目。

資源 2 和資源 3 都有相同的AccountId標籤 2222和相同的BusinessUnit標籤 Operations。因此，它們會合併為計量記錄中的單一UsageAllocations項目。



賣方也可以將沒有標籤的資源合併為單一資源，UsageAllocation並將其做為 中的其中一個項目傳送UsageAllocations。

限制包括：

- 標籤數量 – 5
- 大小 UsageAllocations（基數）– 2，500
- 請求大小上限 – 1 MB

驗證包括：

- 標籤索引鍵和值允許的字元 – a-zA-Z0-9+ -=._：\/@
- 跨UsageAllocation清單的標籤上限 – 5
- 兩個 UsageAllocations 不能有相同的標籤（也就是標籤索引鍵和值的相同組合）。如果是這種情況，他們必須使用相同的 UsageAllocation。
- AllocatedUsageQuantity 的 總和UsageAllocation必須等於 UsageQuantity，也就是彙總用量。
- 承載大小上限不能超過 1 MB。這包括輸入屬性索引鍵（例如，、UsageRecordsAllocatedUsageQuantity、標籤）。

> **Note**
>
> 為了確保您沒有違反承載限制，請根據業務需求建立大小上限的範例請求物件、將物件轉換為 JSON 字串，並取得以位元組為單位的大小。請確定單一 API 呼叫不會超過 1 MB 的限制。例如，如果具有 1 的請求的大小UsageRecord上限為 200 KB，請勿UsageRecords在請求中傳送超過 5 個 (200KB * 5 = 1MB)。

買方體驗

下表顯示買方啟用 AccountId和BusinessUnit廠商標籤後買方體驗的範例。

在此範例中，買方可以在成本用量報告中看到配置的用量。廠商計量的標籤使用字首“aws:marketplace:isv”。買方可以在 AWS產生的成本分配標籤下的 Billing and Cost Management 中啟用它們。

成本用量報告的第一列和最後一列與賣方傳送到計量服務的內容相關（如賣方體驗範例所示）。

成本用量報告（簡化）

ProductCode	買方	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	網路：每個 (GB) 已檢查	70	2222	作業

ProductCode	買方	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	網路：每個 (GB) 已檢查	30	3333	財務
xyz	111122223333	網路：每個 (GB) 已檢查	20	4444	IT
xyz	111122223333	網路：每個 (GB) 已檢查	20	5555	行銷部門
xyz	111122223333	網路：每個 (GB) 已檢查	30	1111	行銷部門

如需程式碼範例，請參閱「BatchMeterUsage 使用用量配置標記程式碼範例 （選用）」。

使用 檢查權利 AWS Marketplace Entitlement Service

如果您的產品是 SaaS 合約產品，您的產品會使用 GetEntitlements API 呼叫 AWS Marketplace Entitlement Service 來擷取客戶的權利。您的產品應該針對 AWS Marketplace Entitlement Service 驗證該帳戶的後續用量。例如，如果客戶在帳戶中佈建 10 個使用者，您的產品應該檢查 AWS Marketplace 權利服務是否有該容量的權利。

若要驗證客戶對您的產品的權利，請使用 AWS Marketplace 權利服務中的 `GetEntitlements`操作。 AWS Marketplace 權利服務僅適用於美國東部 （維吉尼亞北部） 區域，可透過 `entitlement.marketplace.us-east-1.amazonaws.com`或 存取`entitlement-marketplace.us-east-1.api.aws`。

`GetEntitlements` 接受客戶 AWS 帳戶 ID、客戶識別符和維度做為篩選條件。 `ProductCode` 是必要的參數。操作會傳回分頁的權限清單。結果有一個 `ExpirationDate` 欄位，顯示權利有效的最短期間。如果客戶已設定自動續約，則 ExpirationDate 欄位中的日期是續約日期。

如需程式碼範例，請參閱「SaaS 產品整合的程式碼範例」。

擷取使用者動作的權限

下列範例可協助您進一步了解擷取使用者動作權利的程序。

範例：以使用者為基礎的產品

您提供的產品允許特定客戶存在多個帳戶。客戶可以造訪儀表板來佈建新使用者（例如，指派登入資料）。當客戶佈建新的使用者時，您的產品會呼叫 `GetEntitlements` 來驗證容量是否存在。如果沒有，您可以呼叫 AWS Marketplace Metering Service 來為其他使用者計費。

範例：資料儲存產品

您提供的產品可讓客戶以加密或未加密的形式存放特定數量的資料。客戶可以檢視儀表板，顯示產品中現有和配置的資料量。您的儀表板會透過 擷取分配金額`GetEntitlements`。

SaaS 產品整合檢查清單

在軟體即服務 (SaaS) 產品上線之前，您必須驗證您已完成必要的組態。您可以使用下列 SaaS 整合檢查清單來驗證您已完成所需的組態。

類別	需求
存取	已提交賣方註冊表單，其中包含要使用的 AWS Marketplace 所需 AWS 帳戶。
存取	完成賣方註冊，包括條款和條件、銀行帳戶，以及 W8 或 W9 稅務表單。
存取	已為註冊 AWS Marketplace 帳戶設定跨帳戶角色。
產品	完成 中的產品請求表單 AWS Marketplace 管理入口網站。
產品	在 AMMP 中建立產品精靈的備註索引標籤中提供用於測試 AWS 的帳戶 IDs。
產品	在產品索引標籤中提供 .txt 格式的 EULA URL。
產品	從 接收您的產品代碼和 Amazon SNS 主題資訊 AWS Marketplace。

類別	需求
產品	已訂閱 Amazon SNS 主題，並建立 Amazon SQS 佇列來訂閱 Amazon SNS 主題。
帳單解決方案	已驗證您可每小時將 SaaS 訂閱產品的計量記錄傳送給每個客戶的 `BatchMeterUsage` 操作。可以傳送計量記錄，供每個客戶用於 SaaS 合約產品的額外使用。
帳單解決方案	已驗證您可以從適用於 SaaS 合約產品的權限服務驗證客戶 AWS Marketplace 權限。
帳單解決方案	已驗證測試帳戶產生的帳單上，成本是否如預期顯示。
帳單解決方案	已針對無效的客戶 IDs和已取消的訂閱等情況進行測試。
產品	已將產品請求提交回 AWS Marketplace 進行發佈。
註冊	實作 HTTPS 註冊頁面，可接受 HTTP POST 請求。
註冊	驗證您可以接受新客戶註冊。
註冊	驗證您未將註冊字符存放在 Cookie 中。
註冊	已驗證您使用 `CustomerIdentifier` 從 AWS 字符`ResolveCustomer` 取得 `ProductCode` 和 。
註冊	已驗證您可以解決從 收到的註冊字符 AWS ，而不會延遲。
註冊	已測試您未封鎖向 Gmail 等電子郵件地址註冊。
註冊	已測試您可以接受未完成的註冊和多次註冊嘗試。

類別	需求
訂閱	測試您可以處理 `unsubscribe-pending` 和 `unsubscribe-success` 訊息。
訂閱	已驗證您在收到`unsubscribe-pending` 訊息後的一小時內傳送最終計量記錄。
安全	驗證 AWS 根帳戶沒有 API 金鑰、具有強式密碼，並與硬體多重要素驗證 (MFA) 裝置相關聯。所有管理存取都是透過使用 AWS Identity and Access Management (IAM) 建立的身分進行。沒有共用帳戶。
安全	驗證 IAM 角色是否用於所有程式設計的 Amazon Elastic Compute Cloud (Amazon EC2) 存取。登入資料不會硬式編碼為指令碼、標頭或原始程式碼。
安全	已驗證您維護完整的日誌和日誌合併。
安全	已驗證您有定義明確的公有和私有子網路邊界，可隔離應用程式服務和資料庫和檔案系統的存取。區分敏感資料和隔離公有和私有資料的不同資料類別定義。
安全	已驗證您在傳輸中和靜態時有私有資料加密，並已排定金鑰輪換。
安全	已驗證您擁有安全事件工具和存取權限，以及符合及時調查和復原的定期排程事件回應練習。
可靠性	已驗證系統可因應需求的變化、視需要向上和向下擴展，並採用負載平衡以確保高效能。系統也會視需要提供以邊緣為基礎的快取。
可靠性	指定經過驗證的復原時間和點目標，並定期排程災難復原。元件故障是透過自動觸發和通知自行修復。

上的 SaaS 產品報告 AWS Marketplace

AWS Marketplace 為您的軟體即服務 (SaaS) 產品產生報告，其中包含訂閱者、財務、用量和稅金的資料。下表提供有關可用報告的資訊。您可以使用報告來更深入地了解您的客戶群，並更了解您的財務，包括銷售和稅金。如需詳細資訊，請參閱the section called “賣方報告”。下表摘要說明 SaaS 產品的財務報告方式。

報告	SaaS 內容
每日業務報告	預付合約費用會顯示在費用區段中。 計量用量費用會顯示在用量區段中。
每月收入報告	預付合約費用會顯示在年度訂閱區段中。 計量用量費用會顯示在帳單和收入資料區段中。
銷售補償報告	預付合約費用和每月額外用量費用會顯示為個別明細項目。
客戶訂閱者報告	新的 SaaS 合約會出現在年度訂閱區段中。 新的 SaaS 訂閱會顯示在每小時/每月訂閱區段中。

SaaS 產品整合的程式碼範例

您可以使用下列程式碼範例，將軟體即服務 (SaaS) 產品與發佈和維護產品所需的 AWS Marketplace APIs 整合。如需詳細資訊，請參閱下列區段。

主題

- ResolveCustomer 程式碼範例
- GetEntitlement 程式碼範例
- BatchMeterUsage 程式碼範例
- BatchMeterUsage 使用用量配置標記程式碼範例 （選用）

ResolveCustomer 程式碼範例

下列程式碼範例適用於所有定價模型。Python 範例會交換 CustomerIdentifier、ProductCode和 的x-amzn-marketplace-token字符CustomerAWSAccountId。CustomerAWSAccountId 是與訂閱相關聯的 AWS 帳戶 ID。當您從重新導向時，此程式碼會在註冊網站上的應用程式中執行 AWS Marketplace 管理入口網站。重新導向是包含字符的 POST 請求。

如需 的詳細資訊ResolveCustomer，請參閱AWS Marketplace 《計量服務 API 參考》中的 ResolveCustomer。

```
# 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
```

回應範例

```
{
    'CustomerIdentifier': 'string',
    'CustomerAWSAccountId':'string',
    'ProductCode': 'string'
}
```

GetEntitlement 程式碼範例

下列程式碼範例適用於具有合約的 SaaS 產品，以及具有消耗定價模型的 SaaS 合約。Python 範例會驗證客戶是否具有作用中的權限。

如需 的詳細資訊GetEntitlement，請參閱AWS Marketplace 《權利服務 API 參考》中的 GetEntitlement。

```
# Import AWS Python SDK
import boto3

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

# Filter entitlements for a specific customerID
#
# productCode is supplied after the AWS Marketplace Ops team has published
# the product to limited
#
# customerID is obtained from the ResolveCustomer response
entitlement = marketplaceClient.get_entitlements({
    'ProductCode': 'productCode',
    'Filter' : {
        # Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
        'CUSTOMER_IDENTIFIER': [
            'customerID',
        ]
        # Option 2: Using CustomerAWSAccountID (preferred)
        # 'CUSTOMER_AWS_ACCOUNT_ID': [
        #     'awsAccountID',
        # ]
    },
    'NextToken' : 'string',
    'MaxResults': 123
})

# TODO: Verify the dimension a customer is subscribed to and the quantity,
# if applicable
```

回應範例

傳回的值對應於您在 中建立產品時所建立的維度 AWS Marketplace 管理入口網站。

```
{
```

```
   "Entitlements": [
      {
         "CustomerIdentifier": "string",
         "CustomerAWSAccountID": "string",
         "Dimension": "string",
         "ExpirationDate": number,
         "ProductCode": "string",
         "Value": {
            "BooleanValue": boolean,
            "DoubleValue": number,
            "IntegerValue": number,
            "StringValue": "string"
         }
      }
   ],
   "NextToken": "string"
}
```

BatchMeterUsage 程式碼範例

下列程式碼範例與使用定價模型的 SaaS 訂閱和合約相關，但不與不使用 的 SaaS 合約產品相關。Python 範例會將計量記錄傳送至 AWS Marketplace ，向您的客戶收取pay-as-you-go費用。

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited
#
# You can use either:
# - customerID from the ResolveCustomer response (deprecated after Dec 31, 2025)
# - AWS account ID of the buyer

# Import AWS Python SDK
import boto3
from datetime import datetime

# Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
usageRecord = [
    {
        'Timestamp': datetime(2015, 1, 1),
        'CustomerIdentifier': 'customerID',
```

```
        'Dimension': 'string',
        'Quantity': 123
    }
]

# Option 2: Using CustomerAWSAccountID (preferred)
# usageRecord = [
#     {
#         'Timestamp': datetime(2015, 1, 1),
#         'CustomerAWSAccountID': 'awsAccountID',
#         'Dimension': 'string',
#         'Quantity': 123
#     }
# ]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(
    UsageRecords=usageRecord,
    ProductCode='productCode'
)
```

如需 的詳細資訊BatchMeterUsage，請參閱AWS Marketplace 《計量服務 API 參考》中的 BatchMeterUsage。

回應範例

```
{
    'Results': [
        {
            'UsageRecord': {
                'Timestamp': datetime(2015, 1, 1),
                'CustomerIdentifier': 'string',
                'CustomerAWSAccountID': 'string',
                'Dimension': 'string',
                'Quantity': 123
            },
            'MeteringRecordId': 'string',
            'Status': 'Success' | 'CustomerNotSubscribed' | 'DuplicateRecord'
        },
    ],
    'UnprocessedRecords': [
        {
```

```
            'Timestamp': datetime(2015, 1, 1),
            'CustomerIdentifier': 'string',
            'CustomerAWSAccountID': 'string',
            'Dimension': 'string',
            'Quantity': 123
        }
    ]
}
```

BatchMeterUsage 使用用量配置標記程式碼範例 （選用）

下列程式碼範例與 SaaS 訂閱和具有用量定價模型的合約相關，但不適用於不使用 的 SaaS 合約產品。Python 範例會傳送具有適當用量分配標籤的計量記錄給 AWS Marketplace ，以向您的客戶收取 pay-as-you-go費用。

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited
#
# You can use either:
# - customerID from the ResolveCustomer response (deprecated after Dec 31, 2025)
# - AWS account ID of the buyer

# Import AWS Python SDK
import boto3
import time

# Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
usageRecords = [
    {
        "Timestamp": int(time.time()),
        "CustomerIdentifier": "customerID",
        "Dimension": "Dimension1",
        "Quantity": 3,
        "UsageAllocations": [
            {
                "AllocatedUsageQuantity": 2,
                "Tags": [
                    { "Key": "BusinessUnit", "Value": "IT" },
                    { "Key": "AccountId", "Value": "*********" },
```

```
                ]
            },
            {
                "AllocatedUsageQuantity": 1,
                "Tags": [
                    { "Key": "BusinessUnit", "Value": "Finance" },
                    { "Key": "AccountId", "Value": "*********" },
                ]
            },
        ]
    }
]

# Option 2: Using CustomerAWSAccountID (preferred)
# usageRecords = [
#     {
#         "Timestamp": int(time.time()),
#         "CustomerAWSAccountID": "awsAccountID",
#         "Dimension": "Dimension1",
#         "Quantity": 3,
#         "UsageAllocations": [
#             {
#                 "AllocatedUsageQuantity": 2,
#                 "Tags": [
#                     { "Key": "BusinessUnit", "Value": "IT" },
#                     { "Key": "AccountId", "Value": "*********" },
#                 ]
#             },
#             {
#                 "AllocatedUsageQuantity": 1,
#                 "Tags": [
#                     { "Key": "BusinessUnit", "Value": "Finance" },
#                     { "Key": "AccountId", "Value": "*********" },
#                 ]
#             },
#         ]
#     }
# ]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(
    UsageRecords=usageRecords,
    ProductCode="testProduct"
```

```
)
```

如需 的詳細資訊BatchMeterUsage，請參閱《 AWS Marketplace Metering Service API 參考》中的 BatchMeterUsage。

回應範例

```
{
    "Results": [
        {
            "Timestamp": "1634691015",
            "CustomerIdentifier": "customerID",
            "CustomerAWSAccountID": "awsAccountID",
            "Dimension": "Dimension1",
            "Quantity": 3,
            "UsageAllocations": [
                {
                    "AllocatedUsageQuantity": 2,
                    "Tags": [
                        { "Key": "BusinessUnit", "Value": "IT" },
                        { "Key": "AccountId", "Value": "*********" }
                    ]
                },
                {
                    "AllocatedUsageQuantity": 1,
                    "Tags": [
                        { "Key": "BusinessUnit", "Value": "Finance" },
                        { "Key": "AccountId", "Value": "*********" }
                    ]
                }
            ],
            "MeteringRecordId": "8fjef98ejf",
            "Status": "Success"
        }
    ],
    "UnprocessedRecords": [
        {
            "Timestamp": "1634691015",
            "CustomerIdentifier": "customerID",
            "CustomerAWSAccountID": "awsAccountID",
            "Dimension": "Dimension1",
            "Quantity": 3,
            "UsageAllocations": []
```

```
        }
    ]
}
```

使用 透過 Amazon VPC 交付您的產品 AWS PrivateLink

AWS Marketplace 支援 AWS 服務 AWS PrivateLink，可讓您使用 Amazon 網路，讓買方存取您透過銷售的產品 AWS Marketplace。下列各節概述使用 AWS PrivateLink 技術透過 Amazon Virtual Private Cloud (VPC) 端點設定和交付產品的程序。

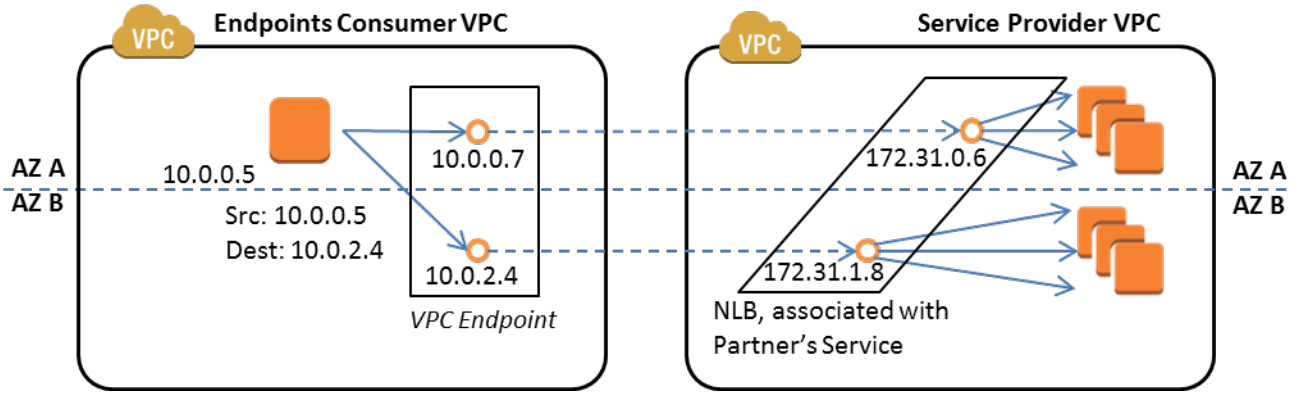
本主題假設您具備數個 AWS 服務和 AWS Marketplace 環境的工作知識。

主題

- 簡介
- 設定您的產品
- 將您的產品提交至 AWS Marketplace
- 買方存取 VPC 端點
- 附錄：檢查清單

簡介

身為 AWS Marketplace 賣方，您可以透過 Amazon VPC 端點讓買方存取您的服務。此方法可讓買方使用 AWS PrivateLink技術跨 Amazon 網路存取您的服務。如果您使用 AWS Marketplace 建立和交付此方案，買方可以在其中探索您的服務 AWS Marketplace。您的買方也可以在建立 VPC 端點的可用服務清單中找到您的產品。



VPC 端點是一種虛擬裝置，可讓 AWS 客戶在其 VPC 與其他 AWS 服務之間建立私有連線，而不需要透過網際網路、NAT 裝置、VPN 連線或 進行存取 AWS Direct Connect。您可以透過 建立端點服務 AWS Marketplace ，讓買方能夠使用此技術連線到您的服務。此連線方法對您的買方更安全，因為他們透過 Amazon 私有網路而非網際網路存取您的服務。

對於您要提供服務的每個區域，您可以建立或使用現有的資源來設定 VPC、設定服務執行個體、設定網路負載平衡器，以及透過建立服務端點向網路負載平衡器註冊服務。完成這些步驟並測試產品後，您可以將組態資訊提供給AWS Marketplace 賣方營運團隊。

AWS 建議您提供私有 DNS 名稱，供買方在建立 VPC 端點時使用。

買方建立 VPC 端點時，可以選擇啟用私有 DNS 名稱。透過選擇此選項，買方的 VPC 服務會設定私有託管區域。如果您提供私有 DNS 名稱，買方可以在設定 VPC 端點連線到您的服務時使用它。在買方的私有託管區域中，私有 DNS 名稱 (api.example.com) 會指向為端點服務建立的隨機產生的 DNS 名稱 (vpce-11111111111111111-yyyyyyyy.api.vpce.example.com)。買方的 EC2 執行個體會跨不同的 VPCs 呼叫相同的統一 DNS 名稱 (api.example.com：//)。此外，如果公有和私有 DNS 名稱相同，買方可以在從 VPC 內部或外部存取您的服務時使用相同的公有名稱。

如需透過 提供服務的協助 AWS Marketplace，您可以聯絡AWS Marketplace 賣方營運團隊。當 AWS Marketplace 買方訂閱您的服務並建立 VPC 端點時，您的服務會顯示在您的 AWS Marketplace 服務下。 AWS Marketplace Seller Operations 團隊使用易於使用的 DNS 名稱，以便在建立 VPC 端點時輕鬆探索您的服務。

您的產品會建立為軟體即服務 (SaaS) 產品。計量和計費與其他 AWS Marketplace SaaS 產品相同。

設定您的產品

若要將產品設定為可透過 Amazon VPC 端點使用：

1. 建立或使用現有的 Amazon VPC。
2. 為您的產品建立（或使用現有的） Amazon EC2 執行個體。
3. 在您提供產品的每個區域中建立網路負載平衡器。 AWS 建議您包含區域的所有可用區域 (AZs)。
4. 使用 Amazon VPC 主控台、CLI 或支援的 SDKs 來建立 VPC 端點服務。
5. 確認您可以透過網路負載平衡器存取服務。
6. 向 AWS Certificate Manager (ACM) 請求易用 DNS 名稱的憑證。ACM 發行憑證前，會先驗證您是否擁有或控制憑證請求中的網域名稱。
7. 將易於使用的 DNS 名稱子網域委派給 AWS Marketplace 賣方營運團隊提供給您的名稱伺服器，例如 api.vpce.example.com。在您的 DNS 系統中，您必須建立名稱伺服器 (NS) 資源記錄，將此子

網域指向 AWS Marketplace 賣方營運團隊提供的 Amazon Route 53 名稱伺服器，以便 DNS 名稱（例如 vpce-0ac6c347a78c90f8.api.vpce.example.com) 可公開解析。

8. 允許存取買方 AWS 的帳戶。

   注意：您可以使用支援的 SDK 或此 CLI 命令自動存取帳戶：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。

將您的產品提交至 AWS Marketplace

在將服務發佈至 的過程中 AWS Marketplace，您會與 AWS Marketplace Seller Operations 團隊合作。若要提交啟用 PrivateLink 的產品：

1. 將以下資訊透過電子郵件傳送給AWS Marketplace 賣方營運團隊：
   a. 用來建立端點的端點和 AWS 帳戶。端點類似於：com.amazonaws.vpce.us-east-1.vpce-svc-0daa010345a21646
   b. 您服務的易用 DNS 名稱。這是 AWS Marketplace 買方用來存取您產品的 DNS 名稱。
   c. 您用來請求憑證 AWS 的帳戶，以及買方用來存取 VPC 端點的私有 DNS 名稱。

      AWS Marketplace Seller Operations 團隊會驗證您公司的身分和 DNS 名稱，以用於您要註冊的服務（例如 api.vpce.example.com)。驗證後，DNS 名稱會覆寫預設的基本端點 DNS 名稱。

買方存取 VPC 端點

AWS Marketplace 建立 VPC 端點的買方可以在下列情況下探索您的服務：

- 您遵循本頁稍早所述的賣方程序來建立或使用現有產品。
- 買方訂閱您的服務。
- 您已將買方 AWS 的帳戶新增至允許的帳戶清單。

當買方建立 VPC 端點時，可以選擇將私有託管區域與其 VPC 建立關聯。託管區域包含服務預設私有 DNS 名稱的記錄集，該服務解析為其 VPC 中端點網路介面的私有 IP 地址。

任何買方託管的端點，包括 AWS Marketplace 服務，都可以提供許可給所有帳戶（「*」許可）。不過，當您使用此方法時，除非您依服務名稱搜尋，否則服務不會包含在描述呼叫或主控台中。若要在描述呼叫中顯示服務，服務必須將買方 AWS 的帳戶明確新增至允許清單。

若要存取您的服務，買方會執行下列動作：

1. 探索並訂閱您的 服務 AWS Marketplace。
2. 使用 AWS Command Line Interface (AWS CLI)、API 或 Amazon VPC 主控台來探索您的服務，然後建立 VPC 端點，以連線到其使用的子網路和 AZs 中的服務。端點會在子網路中顯示為彈性網路介面。本機 IP 地址和區域和區域 DNS 名稱會指派給端點。

用戶端 DNS 名稱	名稱
區域性	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

如果您提供預設私有 DNS 名稱，且買方在建立 VPC 端點時選擇啟用私有 DNS 名稱 （與私有託管區域相關聯），則買方會看到區域預設私有 DNS 名稱以連線至您的服務。

名稱	Alias (別名)	別名託管區域 ID	（備註）
api.example.com	vpce<0dc9a211a78c90f8>。https：//api.vpce.example.com	Z00AABBCCDD	IAD1 IAD2

附錄：檢查清單

使用以下檢查清單，確保您在提交產品給 AWS Marketplace 賣方營運團隊之前，先設定和測試您的產品。

產品建立檢查清單

- 建立 （或使用現有的） VPC，然後設定它。
- 在 VPC 內建立和設定網路負載平衡器。

- 透過建立 VPC 端點服務，向網路負載平衡器註冊服務。
- 將您用來設定 VPC 端點 AWS 的帳戶 ID 提供給 AWS Marketplace 賣方營運團隊。
- 將預設端點服務名稱（例如 com.amazonaws.vpce.us-east-1.vpce-svc-0bbb070044a2164) 提供給 AWS Marketplace 賣方營運團隊。
- 提供易於使用的服務 DNS 名稱（必要），以覆寫隨機產生的服務 DNS 名稱。針對您易於使用的服務 DNS 名稱所使用的子網域，向 ACM 請求 SSL 憑證。將這些憑證和您用來請求憑證 AWS 的帳戶 ID 提供給 AWS Marketplace 賣方營運團隊。
- 建議：提供私有 DNS 名稱。
- 建立程序來通知並允許 AWS Marketplace 買方使用 AWS PrivateLink 技術連線到您的服務。將買方 AWS 的帳戶 IDs 新增至允許的帳戶清單。

產品測試

- 確認您的服務已設定且可探索。
- 確認您的服務可透過網路負載平衡器探索。
- 確認買方可以建立 VPC 端點並存取您的服務。使用您擁有 AWS 的帳戶，而該帳戶不是您用來設定服務的帳戶。

中的專業服務產品 AWS Marketplace

身為賣方，您可以為 AWS Marketplace 買方提供專業服務。專業服務包括評估、遷移、支援、管理和訓練其他人如何使用 AWS 服務和產品的服務 AWS Marketplace。賣方會建立產品方案，描述他們提供的服務、與客戶協商以建立條款協議，然後透過 建立服務的自訂優惠 AWS Marketplace。

> **Note**
>
> 身為獨立軟體廠商 (ISV)、管道合作夥伴或諮詢合作夥伴，您也可以授權其他合作夥伴使用管道合作夥伴私有優惠來轉售您的專業服務產品。如需詳細資訊，請參閱建立頻道合作夥伴作為 ISV 的轉售機會。

買方可以在 AWS Marketplace 目錄中尋找專業服務產品，方法是選取類別下的專業服務、在交付方法下選擇專業服務，然後依發佈者、定價模型和定價單位縮小搜尋範圍。他們需為 AWS 帳單中的服務付費。他們可以使用 等工具 AWS Cost Explorer 來集中付款和管理其成本。

如需專業服務產品的詳細資訊，請參閱：

- 專業服務產品入門
- 提供專業服務產品的詳細資訊
- 專業服務產品的需求
- 建立頻道合作夥伴作為 ISV 的轉售機會

下列影片會進一步說明如何在 中管理專業服務產品 AWS Marketplace。

在 中管理專業服務產品 AWS Marketplace。

取得說明

如需專業服務產品的協助，請聯絡您的 業務開發合作夥伴 AWS Marketplace 或 AWS Marketplace 賣方營運團隊。

中的專業服務產品入門 AWS Marketplace

身為賣方，您可以為 AWS Marketplace 買方提供專業服務。專業服務包括評估、遷移、支援、管理和訓練其他人如何使用 AWS 服務和產品的服務 AWS Marketplace。您的產品定義會告知買方您提供的

服務，並說明為什麼他們應該為這些服務選擇貴公司。 AWS Marketplace 允許買方與您聯絡。您同意合約，然後建立私有優惠，允許買方以固定成本購買您的服務。下列各節說明如何開始使用專業服務產品，提供建立第一個產品並提供客戶所需的步驟。

下列影片會詳細說明如何在 中列出專業服務產品 AWS Marketplace。

主題

- 必要條件
- 建立專業服務產品
- 建立私有優惠
- 編輯產品資訊
- 編輯產品定價
- 編輯產品可見性
- 移除專業服務產品

必要條件

若要在 上銷售專業服務 AWS Marketplace，您必須完成下列先決條件：

- 有權存取 AWS Marketplace 管理入口網站。這是您用來註冊為賣方和管理您銷售產品的工具 AWS Marketplace。若要進一步了解如何存取 AWS Marketplace 管理入口網站，請參閱 賣方的 AWS Marketplace 政策和許可。
- 註冊為 AWS Marketplace 賣方，並提交您的稅務和銀行資訊。您必須提供非英國銀行帳戶才能接收支出。若要進一步了解如何成為賣方，請參閱 以 AWS Marketplace 賣方身分開始使用。
- 您必須擁有專業服務產品，才能提供與 AWS 服務或至少一項 中的公有產品相關的服務 AWS Marketplace。您的產品必須直接支援這些產品，或提供可驅動這些產品訂閱的服務。

> **Note**
>
> 您的產品必須列在這些主要類別中的至少一個：評估、實作、受管服務、進階支援或訓練。如需專業服務產品指導方針的詳細資訊，請參閱 上的專業服務產品需求 AWS Marketplace。

建立專業服務產品

下列程序說明如何在 中建立新的專業服務產品 AWS Marketplace 管理入口網站。

建立專業服務產品

1. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中，選取專業服務。此頁面會顯示您已建立的所有專業服務產品，以及您為建立或修改這些產品而提出的任何請求。
3. 在專業服務產品索引標籤上，選取建立專業服務產品。
4. 在建立產品頁面上，提供產品的資訊，然後選取提交。如需您必須提供的詳細資訊，請參閱 在 中提供專業服務產品的詳細資訊 AWS Marketplace。
5. （選用）從 的產品功能表中 AWS Marketplace 管理入口網站，選取專業服務，然後選擇請求索引標籤。確認您看到產品請求具有正確的產品標題，且請求狀態正在審核中。您的產品應該在幾分鐘內以有限預覽模式建立。

> **Note**
>
> 您可以隨時返回專業服務頁面的請求索引標籤，以查看請求的狀態。建立程序中的任何錯誤都會顯示在這裡。您可以選取請求以查看請求詳細資訊或修正錯誤。

當您的產品最初建立時，只有您的 AWS 帳戶（您用來建立產品的）才能存取它。如果您從專業服務頁面檢視產品，您可以選擇在 上檢視 AWS Marketplace，以檢視 AWS Marketplace 買方在 中顯示的產品詳細資訊。除非您向其他 AWS Marketplace 使用者提供私有優惠，否則此詳細資訊清單不適用於其他使用者。

若要了解如何公開提供產品，請參閱編輯產品可見性。

建立私有優惠

當潛在買方檢視您的產品時 AWS Marketplace，他們無法直接購買。當他們嘗試訂閱時，系統會重新導向他們向您請求私有優惠。 AWS Marketplace 會傳送電子郵件訊息到您的 AWS Marketplace 賣方帳戶根使用者電子郵件地址，通知您客戶已請求私有優惠。下列程序說明如何回應此請求。

> **Note**
>
> 當您透過 建立高於 250，000 美元的私有優惠時 AWS Marketplace 管理入口網站，可能需要額外的核准。如需詳細資訊，請透過 聯絡 Private Offer Success 團隊 (POST)、廠商財務成功 (VFS) 或賣方報告團隊AWS Marketplace 管理入口網站。

為專業服務產品建立私有優惠

1. 聯絡客戶以解決您對請求的任何問題。在 中建立私有優惠之前，先同意優惠條款 AWS Marketplace。買方沒有義務購買您的產品，因此在建立優惠之前達成共識是合理的。
2. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
3. 從功能表中選取優惠，然後選取建立私有優惠。
4. 在建立私有優惠頁面上，選取要為其建立私有優惠的產品。僅包含目前可用的產品。
5. 輸入您要延伸私有優惠的買方帳戶 ID。您可以在單一私有優惠中包含最多 25 個帳戶。如果買方使用請求優惠功能，您收到的電子郵件訊息會包含請求帳戶的買方帳戶 ID。
6. 選擇是否允許買方分期支付產品。一般而言，空頭合約會一次付款。較長的合約可能可以選擇分期付款，但這是您與買方簽訂協議的一部分。選取下一步。
7. 完成優惠詳細資訊，包括以下資訊：
   - 自訂優惠名稱 – 提供足夠的詳細資訊，讓您和客戶可以辨識優惠。包含您的公司或產品名稱以及產品的描述。請勿包含任何個人身分識別資訊，包括姓名、電話號碼或地址。
   - 協議結束日期 – 協議服務結束日期。例如，如果您提供 1 年的支援，請輸入服務提供日期後 1 年的日期。
   - 產品維度 – 您所提供服務的價格和單位，如下所示：
     - 一次性付款優惠 – 您可以列出每個維度及其相關聯的價格 （例如，您可以擁有稱為 Silver、Gold 和 Platinum 的維度）。買方可以選擇並支付其偏好設定的費用。
     - 包含付款排程的優惠 – 您必須選擇單一維度，並提供付款排程，其中包含每筆付款的金額和日期。

     > **Note**
     >
     > 如果您想要建立零美元優惠，您必須選取我想要啟用零美元價格進行確認。此預防措施有助於防止您不小心建立免費優惠。

   - 服務協議 – 定義您與客戶之服務協議的文件。您上傳的文件 （文字或 PDF 格式） 會一起附加到單一 PDF 文件，因此請確定不需要檔案名稱來了解內容。
   - 優惠過期日期 – 優惠過期的日期。這將決定買方必須接受優惠的時間長度，並且與何時提供專業服務無關。您可以在建立優惠之後延長優惠到期日。
8. 完成選項編輯後，請選取下一步。
9. 在檢閱優惠頁面上，確定優惠詳細資訊正確無誤，然後選擇建立優惠。

> **Note**
>
> 您的優惠可能需要一些時間才能發佈。發佈後，您可以在管理優惠頁面上檢視優惠。如果您需要編輯優惠（尚未接受），您可以從該頁面執行此操作。

10. 優惠發佈後，即可在管理私有優惠頁面上取得，從該優惠的動作選單中選取複製優惠 URL，然後在電子郵件訊息中將其傳送給買方接受。

編輯產品資訊

下列程序說明如何在 中編輯現有專業服務產品的產品資訊 AWS Marketplace 管理入口網站。

編輯產品資訊

1. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中，選取專業服務。此頁面會顯示您已建立的所有專業服務產品，以及您在建立或修改這些產品時未完成的任何請求。
3. 選取您要編輯的現有產品。然後，從請求變更功能表中，選取更新產品資訊。
4. 對詳細資訊進行變更。如需可編輯欄位的詳細資訊，請參閱 在 中提供專業服務產品的詳細資訊 AWS Marketplace。
5. 選取提交以建立請求。
6. （選用）如果您尚未在「專業服務」頁面的「請求」索引標籤上，請從「產品」選單中 AWS Marketplace 管理入口網站，選取「專業服務」，然後選擇「請求」索引標籤。確認您看到具有正確產品標題的請求，且請求狀態正在審核中。您的產品將在幾分鐘內更新為您所請求的變更。如果發生錯誤，您可以在這裡檢視，並在修正錯誤後重新提交編輯。

編輯產品定價

下列程序說明如何在 中編輯現有專業服務產品的定價資訊 AWS Marketplace 管理入口網站。

編輯產品定價

1. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中，選取專業服務。此頁面會顯示您已建立的所有專業服務產品，以及您為建立或修改這些產品而提出的任何請求。

3. 選取您要編輯的現有產品，然後從請求變更功能表中選取更新定價維度。

> **Note**
>
> 您只能透過 新增定價維度 AWS Marketplace 管理入口網站。若要修改或移除先前建立的維度，請連絡AWS Marketplace 賣方營運團隊提出您的請求。在您的請求中，包含產品 ID 以及您想要變更或移除哪些維度的詳細資訊。

4. 新增任何您想要的新定價維度。如需定價欄位的詳細資訊，請參閱 在 中提供專業服務產品的詳細資訊 AWS Marketplace。
5. 選取提交以建立請求。
6. （選用）從 的產品功能表中 AWS Marketplace 管理入口網站，選取專業服務，然後選擇請求索引標籤。確認您看到具有正確產品標題的請求，且請求狀態正在審核中。您的產品將在幾分鐘內更新為您所請求的變更。如果發生錯誤，您可以在這裡檢視，並在修正錯誤後重新提交編輯。

編輯產品可見性

根據預設，產品建立時可見性有限，只有您的帳戶才能看到新產品。您可以新增其他測試帳戶，或在中公開顯示產品 AWS Marketplace。下列程序說明如何在 中編輯現有專業服務產品的可見性 AWS Marketplace 管理入口網站。

編輯產品可見性

1. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中，選取專業服務。此頁面會顯示您已建立的所有專業服務產品。
3. 選取您要編輯或更新可見性的現有產品。然後，從請求變更功能表中，選取更新產品可見性。
4. 選取選項以更新您的產品可見性，然後選擇提交以提交您的請求以供審核。
5. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。如果狀態為失敗，請選取請求名稱以檢視找到的問題。

> **Note**
>
> 若要在公有 AWS Marketplace 目錄中顯示產品，賣方 AWS Marketplace 營運團隊需要產品審核，以確保產品符合產品準則（請參閱 上的專業服務產品需求 AWS Marketplace）。完成請求可能需要幾天的時間。

移除專業服務產品

下列程序說明如何從 移除現有的專業服務產品 AWS Marketplace 管理入口網站。

> **Note**
>
> 移除專業服務產品不會影響作用中的私有優惠。

移除產品

1. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。
2. 從產品功能表中，選取專業服務。此頁面會顯示您已建立的所有專業服務產品。
3. 選取您要移除的現有產品。然後，從請求變更功能表中，選取更新產品可見性。
4. 選取限制做為可見性選項，以更新您的產品可見性，然後選擇提交。
5. 確認請求索引標籤顯示請求狀態為審核中。當請求完成時，狀態會變成成功。

> **Note**
>
> 完成請求可能需要幾天的時間。具有作用中優惠的產品將移至限制狀態，直到完成最後一個作用中訂閱或合約，然後從中移除為止 AWS Marketplace。限制狀態表示現有使用者可以繼續使用產品。不過，大眾將無法再看到該產品，也無法供新使用者使用。

在 中提供專業服務產品的詳細資訊 AWS Marketplace

當您在 上發佈專業服務產品時 AWS Marketplace，您必須提供產品中繼資料。產品中繼資料包括可見性和定價的產品設定，以及為買方提供有關專業服務產品的詳細資訊的產品資訊。您可以編輯產品描述、其他資源、定價維度和產品可見性。以下各節提供您可以在準備或編輯產品詳細資訊時使用的資訊。

> **Note**
>
> 如需專業服務產品的指導方針和需求的相關資訊，請參閱 上的專業服務產品需求 AWS Marketplace。

主題

- 產品描述
- 其他資源
- 支援資訊
- 定價維度
- 產品可見性

產品描述

產品詳細資訊中的產品描述區段是您產品的核心。它會向您的潛在買方描述您的產品，以便他們可以做出購買決策。產品詳細資訊的本節包含下列資料：

- 產品標題 – 產品的名稱。這是用來識別您的產品；它會顯示在產品頁面和搜尋結果中。為您的產品提供有意義的名稱。它在 內必須是唯一的 AWS Marketplace。
- SKU –（選用）用於追蹤您的產品 AWS Marketplace。此資訊僅供您個人使用；買方不會看到此資訊。
- 簡短描述 – 產品的簡潔描述，顯示在產品 AWS Marketplace 目錄中產品標題的圖磚和下方。
- 長描述 – 較長的格式化描述，可向買方說明產品的詳細資訊。列出產品特有的產品功能、優點、用量和其他資訊。使用可用的格式設定，讓資訊更容易理解和掃描。
- 產品標誌 – 此欄位是公有 Amazon S3 URL，指向代表您產品的影像檔案。檔案必須是 .png、.jpg 或 .gif 格式、透明或白色背景、小於 5MB，且寬度和高度介於 110-10，000 像素之間。標誌會在產品提交期間上傳並存放於 中 AWS Marketplace。修改 URL 的內容不會在提交 AWS Marketplace 之後修改 中的標誌。

  > **Note**
  >
  > 您提供的 Amazon S3 URL 必須可公開使用。這是 檔案所在的 Amazon S3 儲存貯體屬性。如需詳細資訊，請參閱《Amazon Simple Storage Service 主控台使用者指南》中的如何編輯 Amazon S3 儲存貯體的公有存取設定？。

- 重點 – 一組一到三個關於您產品的短點，描述其主要特徵或差異。至少需要一個反白顯示。
- 產品類別 – 您提供的服務類型。您必須選擇至少一個，最多三個類別。有許多類別可供選擇，但專業服務產品必須至少包含下列其中一項：

評估

評估客戶目前的操作環境，為其組織尋找正確的解決方案。

實作

協助設定、設定和部署第三方軟體。

進階支援

存取專家的指導方針和協助，以滿足客戶的需求。

受管服務

代表客戶End-to-end環境管理。

訓練

由專家提供的量身打造的研討會、計劃和教育工具，以協助客戶員工學習最佳實務。

- 搜尋結果的關鍵字 – 提供買方可能用來搜尋您產品的最多三個關鍵字。您可以在逗號分隔清單中列出關鍵字，最多 250 個字元。
- 相關產品 – 選用 – 至少包含一個您的服務可搭配 或 支援的公 AWS Marketplace 有產品。在選擇要顯示在產品詳細資訊頁面或這些產品相關產品中的產品時， AWS Marketplace 會使用這些產品做為輸入。

其他資源

在產品詳細資訊的其他資源區段中，您可以提供已建立的資源連結，以協助客戶。這是線上儲存的一到三個可下載資源的選用集。資源的範例包括產品資訊表、白皮書或產品手冊。針對每個資源，提供資源的名稱和 URL。

支援資訊

本節是格式化的文字欄位，可讓您描述您為服務提供的支援。

客戶預期問題獲得支援，例如使用 服務、故障診斷和請求退款 （如適用）。您必須至少提供下列其中一項產品支援：電子郵件地址、電話號碼或支援 URL (URL)。包含買方在購買您的產品時可預期的支援層級說明。

定價維度

專業服務的定價維度是您提供的套件。例如，您可以提供銀卡、金卡和 Platinum 支援。或者，您可以提供 10、20 或 50 小時的諮詢。針對您要提供的每個維度（至少一個，最多 24 個），指定名稱和描述。當您直接與買方合作為產品建立私有優惠時，您可以設定這些維度的實際價格。

> **Note**
>
> 如需如何使用定價維度以及如何設定價格的詳細資訊，請參閱 建立私有優惠。

產品可見性

已發行產品只能在 AWS Marketplace 您自己的 帳戶、一小組測試帳戶或所有 AWS 帳戶中看見。根據預設，產品會在私有版本中發佈。若要變更產品可見性，請參閱 編輯產品可見性。

上的專業服務產品需求 AWS Marketplace

AWS Marketplace 會維護 上所有產品和優惠的需求 AWS Marketplace。這些要求有助於為我們的客戶提升安全、可靠且精心策劃的數位目錄。我們鼓勵賣方審查適用的其他控制和協定的實作，以滿足其特定產品的需求。

AWS Marketplace 提交時會檢閱所有產品及其相關中繼資料，以確保它們符合或超過目前的 AWS Marketplace 要求。我們會檢閱和調整這些要求，以符合我們的安全要求。此外， AWS Marketplace 會持續審查產品，以確認其符合這些要求的任何變更。如果產品不合規， AWS Marketplace 會與您聯絡以更新您的產品。在某些情況下，您的產品可能暫時無法提供給新訂閱者，直到問題解決為止。

主題

- 產品設定準則
- 客戶資訊要求
- 產品使用準則
- 架構準則

產品設定準則

所有專業服務產品都必須遵守下列產品設定準則：

- 所有定價維度都必須與使用 提供和計費的實際服務相關 AWS Marketplace。
- 您的產品必須列在下列其中一個類別中：評估、實作、受管服務、進階支援或訓練。
- 除了必要的專業服務類別之外，您的產品也應該選擇其他符合所提供服務的適當類別來正確分類。
- 您產品的標誌不得設計為與 AWS 標誌或來自不相關第三方的任何標誌混淆。
- 您的產品詳細資訊不得包含令人反感或明確的材料。他們必須遵守 https：//https://aws.amazon.com/aup/ 上提供的 AWS 可接受使用政策。
- 您的專業服務產品必須直接支援或提供與 上列出的第三方軟體產品相關的服務， AWS Marketplace 或協助客戶達成與採用或管理 相關的特定結果 AWS 雲端。

客戶資訊要求

所有專業服務產品都必須遵守下列客戶資訊要求：

- 專業服務產品必須完全透過列出的維度計費 AWS Marketplace。
- AWS Marketplace 您在任何時候都不允許收集 上專業服務產品清單的客戶付款資訊，包括信用卡和銀行帳戶資訊。
- AWS 與您的專業服務產品結合提供給您的任何訂閱者或潛在訂閱者資訊，必須僅根據 AWS Marketplace 賣方的條款與條件使用。

產品使用準則

所有專業服務產品都必須遵守下列產品使用準則：

- 客戶透過專業服務清單與您聯絡之後，您應該在兩個工作天內與他們聯絡。客戶接受私有優惠之後，您必須在兩天內聯絡他們或提供他們後續步驟，除非私有優惠另有說明。
- 客戶必須能夠輕鬆取得問題的協助，例如使用 服務、疑難排解和請求退款 （ 如適用 ）。必須在履行登陸頁面上指定支援聯絡選項。支援描述應包含有關客戶可預期支援層級的陳述式。
- 您產品的概觀應包含明確的價值主張、主要功能、詳細文件的連結，以及所提供服務的購買前和購買後支援的明確定義。
- 您的產品應該有清楚且直接的服務條款。

架構準則

所有專業服務產品都必須遵守下列架構準則：

- 專業服務產品必須與 AWS 服務或至少一個公 AWS Marketplace 有產品 （最多四個） 相關，其中產品會直接為這些相關產品提供服務，或提供可推動更多訂閱者訂閱這些相關產品的服務。
- 如果所提供的服務在客戶的基礎設施中需要額外資源，請遵循下列準則：
  - 以安全的方式佈建資源，例如使用 AWS Security Token Service 或 AWS Identity and Access Management (IAM)。
  - 提供其他文件，包括所有佈建 AWS 服務的描述、IAM 政策陳述式，以及如何在客戶帳戶中部署和使用 IAM 角色或使用者。
  - 在產品描述中包含通知，說明如果客戶產生額外的 AWS 基礎設施成本，與其 AWS Marketplace 交易分開，他們需負責支付額外的基礎設施費用。
  - 如果您的產品部署代理程式，請提供說明給客戶，說明如何在其 中部署代理程式 AWS 帳戶。

在 中為專業服務產品定價 AWS Marketplace

身為 AWS Marketplace 賣方，您可以接受專業服務產品的私有優惠。私有優惠是用來從中購買產品的協商條款 AWS Marketplace。身為賣方，您會與買方進行協商，以取得私有優惠。私有優惠是 中可用的定價模型之一 AWS Marketplace。透過賣方私有優惠，有可用於多年期和自訂持續時間合約的選項。本主題提供有關專業服務產品定價和優惠的詳細資訊。

如需多年期和自訂持續時間合約的詳細資訊，請參閱 為您的 AWS Marketplace 產品準備私有優惠和私有優惠分期方案。

每個產品只能設定一個價格。如需定價 AWS Marketplace 產品的詳細資訊，請參閱的產品定價 AWS Marketplace。

私有優惠的運作方式

您可以從 中的優惠頁面建立和管理私有優惠AWS Marketplace 管理入口網站。您可以為優惠指定產品，以產生唯一的 ID 和 URL。您將為私有優惠建立定價計劃、新增法律條款和銷售文件，並將優惠擴展至特定買方 AWS 帳戶。只有您為其建立優惠的帳戶才能看見優惠。

在您建立私有優惠並通知潛在買方之後，他們可以檢視並接受優惠。若要檢視優惠，買方必須登入 AWS 帳戶 接收優惠的 。

> **Note**
>
> 除非您將優惠擴展到其連結帳戶或其管理帳戶，否則買方無法檢視優惠。您無法在優惠中提供服務限制，因此除非產品有限制，否則買方可以按照他們想要的協商價格使用您產品的數量。

如需建立私有優惠的相關資訊，請參閱建立和管理私有優惠。

私有優惠會在賣方報告中追蹤。如需詳細資訊，請參閱報告私有優惠和賣方報告指南。

數據產品

AWS Data Exchange 是一項服務，可讓 AWS 客戶輕鬆地在 AWS 雲端. 身為供應商，AWS Data Exchange 無需建立和維護任何資料交付、權益或帳單技術。AWS Data Exchange 的供應商擁有安全、透明且可靠的通道，可以更有效地接觸客 AWS 戶並授予現有客戶訂閱。成為 AWS Data Exchange 供應商的程序需要幾個步驟來判斷資格。

資料產品具有下列部分：

- 產品詳細資料 — 此資訊可協助潛在訂閱者瞭解產品為何。這包括名稱、說明 (短期和長篇)、標誌影像，以及支援聯絡資訊。產品詳細信息由供應商填寫。
- 產品優惠 — 為了在 AWS Data Exchange 中提供產品，供應商必須定義公開方案。這包括價格和持續時間，數據訂閱協議，退款政策以及創建自定義優惠的選項。
- 資料集 — 產品可以包含一或多個資料集。資料集是一組以檔案為基礎的資料內容的動態集合。資料集是動態的，並使用修訂版本化。每個修訂可以包含多個資產。

如需詳細資訊，包括資格要求，請參閱 AWS Data Exchange 使用者指南中的在 AWS Data Exchange 上提供資料產品。

提交您的產品以在 上發佈 AWS Marketplace

您可以使用產品提交程序來提供您的產品 AWS Marketplace。產品可能很簡單，例如具有單一價格結構的單一 Amazon Machine Image (AMI)，或更複雜的 AWS CloudFormation 範本，以及複雜的定價選項和付款排程。您可以透過下列兩種方式 AWS Marketplace 管理入口網站 之一，定義您的產品方案並透過 提交：

- 使用產品索引標籤 – 對於較不複雜的產品，您可以使用產品索引標籤來完全定義和提交您的請求。
- 使用資產索引標籤 – 對於更複雜且需要更多定義的產品，您可以下載產品載入表單 (PLF)、新增產品詳細資訊，然後使用檔案上傳選項上傳完成的表單。

> **Note**
>
> 資料產品提供者必須使用 AWS Data Exchange 主控台來發佈產品。如需詳細資訊，請參閱《AWS Data Exchange 使用者指南》中的發佈新產品。

我們建議您先使用產品索引標籤來決定要使用的方法。下表列出您用來提交請求的組態和方法。第一欄是您產品的定價模型，另外三個欄則說明產品如何部署給客戶。

定價方式	使用單一節點 AMI 啟動的產品	使用 啟動的產品 AWS CloudFormation	做為軟體即服務 (SaaS) 啟動的產品
使用自有授權 (BYOL)	產品索引標籤	資產索引標籤	
免費	產品索引標籤	資產索引標籤	
按小時付費	產品索引標籤	資產索引標籤	
按年每小時付費	產品索引標籤	資產索引標籤	
每月付費	產品索引標籤	資產索引標籤	
每月每小時	資產索引標籤	資產索引標籤	

定價方式	使用單一節點 AMI 啟動的產品	使用 啟動的產品 AWS CloudFormation	做為軟體即服務 (SaaS) 啟動的產品
付費用量 (AWS Marketplace Metering Service)	產品索引標籤	資產索引標籤	
合約定價	產品索引標籤		
SaaS 訂閱			產品索引標籤
SaaS 合約			產品索引標籤
SaaS Legacy			資產索引標籤

您可以個別提交產品，或者，如果您使用產品載入表單，則可以同時提交多個產品或產品更新。您無法使用產品索引標籤來提交多個產品。如果您不清楚要提交哪些產品以及如何提交它們，請從使用產品索引標籤開始。如果您在提交時遇到任何問題，請聯絡AWS Marketplace 賣方營運團隊。

主題

- 使用產品索引標籤
- 公司和產品標誌要求
- 提交付費重新封裝軟體的要求
- 具有硬體元件的產品需求
- AWS CloudFormation推出的產品（免費或付費）或以用量為基礎的付費 AMI 產品
- 產品變更和更新
- 時間和期望
- 將 AMIs提交至 AWS Marketplace
- 最終檢查清單

使用產品索引標籤

若要存取產品索引標籤，請登入 AWS Marketplace 管理入口網站。從產品索引標籤中，根據您管理的產品類型，選擇伺服器、SaaS 或機器學習。

該產品類型的儀表板隨即出現，並顯示您目前的產品。如果您選擇請求索引標籤，儀表板會顯示任何未完成的請求和您已完成的請求歷史記錄。開始建立產品請求後，您可以儲存進行中的工作，並視需要在數個不同的工作階段中建立請求。

當您提交產品請求時， AWS Marketplace 團隊會對其進行檢閱。您可以在產品頁面上監控請求的狀態，了解您請求的產品類型。對於新產品，在您的請求核准發佈後，您會收到一個有限清單 URL，用於預覽和核准提交。在您核准提交之前，不會發佈您的產品優惠。

當您請求產品更新時，它會發佈，而不需要您檢閱和核准變更。這包括新增或移除版本，以及中繼資料變更。

您可以在請求索引標籤下追蹤請求的狀態。標籤會顯示下列其中一項：

- 草稿 – 您已開始請求程序，但尚未提交您的請求。
- 已提交 – 您已完成並提交請求，且請求正在審核中。
- 必要動作 – AWS Marketplace 團隊已檢閱您的請求，且需要更多資訊。
- 需要核准 – AWS Marketplace 團隊已建立您產品的有限清單 URL。您必須檢閱並核准或拒絕 URL， AWS Marketplace 才會發佈。如果您核准，則當網站發佈時，狀態會變更為待發佈。如果您拒絕，狀態會傳回至 Draft，以便您可以修改請求。
- 發佈待定 – 您已核准請求的模擬，並 AWS Marketplace 正在發佈您的產品。
- 已過期 – 您已啟動請求程序，但未在六個月內完成，因此請求已過期。

如果您有狀態為已提交的項目，您可以撤回提交。如果您有狀態為 Draft 的項目，您可以刪除請求。這可讓您重新開始。當您刪除草稿項目時，該項目會移至請求歷史記錄索引標籤。

若要在 中新增您的產品 AWS GovCloud (US) AWS 區域，您必須擁有 Active AWS GovCloud (US) 帳戶，並遵守 AWS GovCloud (US) 要求，包括 出口控制要求。

公司和產品標誌要求

您的公司標誌和產品的標誌必須符合下列 AWS Marketplace 準則，以便使用者體驗在瀏覽時一致 AWS Marketplace：

產品標誌規格 – 您的產品標誌影像應具有透明或白色背景，大小為 120 到 640 像素，比例為 1：1 或 2：1（寬）。

公司標誌規格 – 您的公司標誌影像應該具有透明背景，大小為 220 x 220 像素，允許在其中每一側填補 10 像素。

提交付費重新封裝軟體的要求

您必須先符合下列要求，才能提交重新封裝軟體的清單。在這種情況下，重新封裝的軟體包含開放原始碼 AMIs或由其他廠商建立的軟體，例如搭配 Windows 的 AMI。

要求

- 產品標題必須指出重新封裝所新增的值。產品標題的範例包括：強化的 <Product>、新增套件的 <Product>，或 <Product2> 上的 <Product1>。
- 產品標題不得包含 文件不支援的任何其他語言。例如，除非在您提供的產品詳細資訊中證實這些字詞，否則產品標題可能不會使用認證、原始或免費的字詞。
- 產品簡短描述必須包含概述產品費用的明確陳述式。簡短描述必須以片語開頭 此產品具有與之相關聯的費用...。例如，如果產品包含來自賣方的支援費用，則產品描述應說明：此產品具有與賣方支援相關的費用。
- 產品標誌必須與賣方註冊程序期間使用的公司標誌相同。只有在您使用官方軟體標誌時，產品標誌才會與您的公司標誌不同，因此您必須從原始軟體供應商取得明確許可。如果取得明確許可，則該文件的連結必須包含在變更請求的備註區段中，或是在使用產品載入表單時，在檔案上傳頁面的輸入簡短描述欄位中。
- 對於 AMI 產品，不得從原始產品重複使用 AMI 名稱。AMI 名稱必須以賣方名稱開頭，並遵循此格式：【Seller Name】 【name-given-to-ami】。

  如果 AMI 名稱不符合命名慣例，您可以從 AWS 主控台複製 AMI 並將其重新命名。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的複製 Amazon EC2 AMI。 Amazon EC2

如果付費清單適用於非由您公司建立的獨立軟體產品，且產品未新增任何智慧財產權，例如綁定其他軟體程式庫或新增特殊組態，則連同先前的要求，您也必須符合下列要求：

- 產品標題必須包含賣方名稱 （以及新增的值，如先前所述）。賣方名稱是賣方註冊期間使用的名稱。例如，<Product> 搭配 <seller> 的維護支援。
- 產品詳細描述的第一行必須以片語開頭 這是重新封裝的軟體產品，其中額外費用適用於... （或者，如果是開放原始碼，這是重新封裝的開放原始碼軟體產品，其中額外費用適用於...)。然後，長描述必須包含一個清晰的陳述式，總結您要收取的費用，以及描述這些功能的其他詳細資訊。例如，針對其他支援的開放原始碼產品收費的詳細描述可能開始為：這是重新封裝的開放原始碼軟體產品，其中 {SLA Details} 的支援需支付額外費用。

具有硬體元件的產品需求

不允許銷售硬體產品 AWS Marketplace。如果您提交的軟體產品需要硬體元件（例如 SIM 卡、智慧型裝置、IoT 裝置或感應器），您必須符合下列要求：

- 硬體元件無法販售 AWS Marketplace。
- 硬體元件的成本不能包含在產品的定價中。
- 清單的產品概觀區段必須包含下列陳述式：此清單可能需要的任何硬體都必須分別取得。如需詳細資訊，請參閱產品詳細資訊。

AWS CloudFormation推出的產品（免費或付費）或以用量為基礎的付費 AMI 產品

> **Note**
>
> 有些定價模型不再需要您使用本節所述的產品載入表單來發佈 AMI 與 CloudFormation 產品。當您在賣方入口網站的伺服器產品頁面上使用 CloudFormation 建立 Amazon Machine Image (AMI) 或 AMI，且未立即提示您下載產品載入表單時，請參閱建立 AMI 型產品並將 CloudFormation 範本新增至您的清單。

使用產品載入表單 (PLF) 來提交 AWS Marketplace 客戶使用 AWS CloudFormation 範本啟動的產品。PLF 可透過 取得 AWS Marketplace 管理入口網站。

您遵循這些廣泛的步驟來提交產品：

- 選擇定價模型。
- 下載 Microsoft Excel 試算表的產品載入表單 (PLF)。
- 填寫產品載入表單。
- 依照表單中的指示提交您的產品。

如需完成每個步驟的詳細資訊，請依列出的順序展開各區段。

選擇定價模型

您必須為您的產品選取定價模型。您選擇的模型會控制您在 PLF 中輸入的定價資訊。如需支援的定價模型清單，請參閱本指南的 AMI 產品定價 AWS Marketplace中的 。

下載新產品的 PLF

1. 啟動 AWS Marketplace 管理入口網站。
2. 在資產索引標籤的右側窗格中，選擇具有 CloudFormation 產品的單一 AMI 連結。

   表單會顯示在您的瀏覽器中。
3. 選擇下載檔案，然後在 Excel 中編輯檔案。

   —或—

   如果您有 Microsoft OneDrive，請選擇編輯副本。這會將 PLF 儲存到 OneDrive，您可以在該處編輯它。

> **Note**
>
> 試算表包含數個範例產品。您必須先刪除這些項目，才能提交表單。

下載現有產品的 PLF

下載表單

- 啟動 AWS Marketplace Dashboard，然後選擇下載產品載入表單。

試算表包含舊版產品的資料。保留該資料，並在下一個空白列新增新產品。

填寫 PLF

下列步驟說明如何完成產品載入表單 (PLF)。這些步驟適用於新的和現有的產品。

填寫表單

1. 在 SKU 退款和取消政策欄中，輸入與您產品相關的所有資訊。

> **Note**
>
> 在產品存取指示欄中，您必須提供詳細且明確的使用說明。請遵循本指南建立 的 AMI 和容器產品使用說明 AWS Marketplace中列出的要求。

2. 在輸入端點 URL 相對 URL 欄中，輸入 AMI 的必要資訊。

> **Important**
>
> 您必須與 AWS Marketplace 共用您的 AMI。若要這樣做，請遵循本指南將 AMIs提交至 AWS Marketplace中的步驟 。

3. 最終使用者授權合約 URL 欄提供標準 AWS Marketplace 協議的連結。您可以接受該協議，或輸入您偏好使用的 EULA 連結。如果您提供連結，則必須允許客戶下載 EULA，例如來自 Amazon S3 儲存貯體的連結。

   如需標準合約的詳細資訊，請參閱本指南在 中使用標準化合約 AWS Marketplace中的 。

4. 在 us-east-1 可用性中，讓 可用於所有未來的執行個體類型資料欄，輸入您打算使用的每個 AWS 區域**FALSE**下的 **TRUE**或 。

> **Note**
>
> GovCloud 區域有額外的需求。例如，您必須擁有 GovCloud 帳戶，才能使用 GovCloud 區域。如需詳細資訊，請參閱 AWS GovCloud 使用者指南中的設定。

5. 在建議執行個體類型欄中，接受建議的執行個體類型，或從清單中選擇另一個執行個體類型。確定執行個體類型可在您要使用的區域中使用。

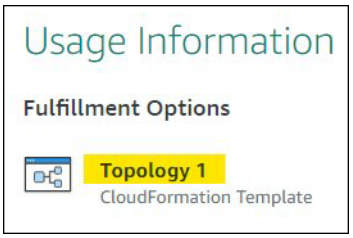
> **Note**
>
> - 大多數客戶接受建議的執行個體類型。
> - 您必須確保執行個體類型可在與產品相同的區域中使用。

6. 在建議執行個體類型和要包含的國家/地區之間的欄中，在您的執行個體類型**TRUE**下輸入 。這會啟用執行個體類型。**FALSE** 其餘類型的輸入 。如需執行個體類型的詳細資訊，請參閱 https：//https://aws.amazon.com/ec2/instance-types/。
7. 在要包含的國家/地區和要排除資料欄的國家/地區中，輸入您要包含或排除的任何國家/地區的兩個字母國家/地區代碼，例如美國。
8. 在定價模型欄中，輸入您產品的定價模型。

   以下清單說明定價模型和您必須完成的任何其他資料欄。

   - BYOL URL – 輸入授權 URL。不需要輸入定價資訊。
   - 每小時 – 輸入您設定為 TRUE 的任何執行個體類型的價格。將所有其他資料欄保留空白。與此相關的資料欄是從 a1.medium 每小時價格到 z1d.metal 每小時價格的資料欄
   - 每小時年度定價 – 填寫上一個步驟中列出的資料欄，以及從 a1.medium 年度價格開始到 z1d.metal 年度價格的資料欄。輸入您設定為 TRUE 的任何執行個體類型的價格。您可以將所有其他資料欄保留空白。
   - 用量 – 在 FCP 類別到 FCP Dimension24 速率欄中，輸入與用量維度相關的資訊。
   - 合約 – 在合約類別到合約 Dimension24 36-Month費率欄中，輸入與合約維度相關的資訊。
9. 在安全群組規則 1 到安全群組規則 12 欄中，輸入產品安全群組的相關資訊。遵循`tcp,#,#,0.0.0.0/0` 格式。例如，`tcp,22,22,0.0.0.0/0`針對 SSH 使用 ，`tcp,3389,3389,0.0.0.0/0`針對 RDP 使用 。
10. 在叢集 AWS 和資源拓撲 1：叢集和資源拓撲的標題 3：架構圖表 URL AWS 欄中，輸入您產品的 CloudFormationdata。您必須輸入下列資料：

    - 拓撲標題 – 部署或履行選項的標題。標題會出現在您產品的詳細資訊頁面上的履行選項區段。例如：

    - 在定價估算欄中，輸入包含值的 AWS 計算器連結。
    - 簡短和簡短描述 – 輸入部署選項的描述。
    - 範本 URL – 提供 Cloudformation 範本的可下載連結。
    - 架構圖 – 提供 CloudFormation 拓撲架構圖的可下載連結。每個部署選項都必須具有顯示堆疊啟動內容的不同圖表。圖表必須遵循 中列出的要求架構圖。

提交 PLF

下列步驟說明如何提交完成的 PLF。

1. 登入 AWS Marketplace 管理入口網站。
2. 在資產索引標籤上，選擇檔案上傳。
3. 在檔案上傳頁面上，上傳您的 PLF 和任何 AWS CloudFormation 範本。檔案上傳程式提供安全的傳輸機制和已提交檔案的歷史記錄。上傳者會自動通知 AWS Marketplace，其會檢閱您提交的政策和安全合規、軟體漏洞和產品可用性。如果團隊對請求有任何疑問或問題，他們會傳送電子郵件訊息給您。

更新您的產品

對於您使用產品載入表單 (PLF) 建立的產品，您也可以使用 PLF 對這些產品進行變更。您可以變更您完成的原始 PLF，或者，如果無法使用，您可以從新的 PLF 開始。就像使用產品索引標籤一樣，您可以新增新版本、移除現有版本，以及更新定價、執行個體類型、區域可用性和中繼資料。若要進行更新，您可以像準備新產品一樣準備任何更新的產品。產品更新準備好後，請依照下列步驟進行：

1. 使用您現有的 PLF，或啟動 AWS Marketplace 管理入口網站，然後在資產索引標籤上選擇檔案上傳。在產品載入表單和賣方指南下，您可以下載產品的 PLF。
2. 更新 PLF 中的產品。
3. 在資產索引標籤AWS Marketplace 管理入口網站下的 中，選擇檔案上傳。
4. 在檔案上傳頁面上，上傳您更新的 PLF 和任何 AWS CloudFormation 範本。檔案上傳程式提供安全的傳輸機制和已提交檔案的歷史記錄。上傳者會自動通知 AWS Marketplace 團隊開始處理您的請求。包含提交的說明（新增新版本、變更價格、變更中繼資料等）。

您的產品提交會檢閱政策和安全性合規、軟體漏洞和產品可用性。如果請求有任何疑問或問題， AWS Marketplace 團隊將透過電子郵件訊息與您聯絡。現有產品頁面的更新會直接處理和發佈，無需額外檢閱。

產品變更和更新

賣方可以隨時對其產品提交變更，並將如先前所述進行處理。不過，有些變更只能每 90 或 120 天進行，或在有待定變更時進行。範例包括價格變更和 AWS 區域 或執行個體類型變更。常見的變更包括：

- 新版本 – 軟體的新版本和修補程式或更新的推出。根據您的請求，我們可以通知訂閱您 AWS Marketplace 內容的客戶有關新版本的可用性，或代表您傳送升級指示。
- 中繼資料變更 – 產品資訊的變更（描述、URLs和使用說明）。
- 定價變更 – 定價金額的變更。請求完成後，會傳送通知給目前的客戶。傳送通知後，價格變更將於 90 天時段後的下個月一日開始生效。例如，如果您在 3 月 16 日 90 天後進行變更，大約是 6 月 16 日，但價格變更發生在下個月的第一天。變更的實際日期為 7 月 1 日。
- 定價模式變更 – 定價模式的變更（例如每小時、免費、每小時_年）。並非所有定價模型變更都受到支援，而所有變更模型的請求都必須經過 AWS Marketplace 團隊審核和核准。從免費模式到付費模式的任何變更都會對現有客戶產生重大影響。另一種方法是提議具有其他功能的新產品，並鼓勵目前的客戶遷移。
- 區域或執行個體變更 – 新增或移除執行個體類型或區域。
- 產品摘錄 – 從 移除產品頁面 AWS Marketplace ，以防止新客戶訂閱。請求完成後，會傳送通知給目前的客戶。

時間和期望

雖然我們努力盡快處理請求，但請求可能需要多次反覆運算，並由賣方和 AWS Marketplace 團隊進行審核。針對完成程序所需的時間，使用下列指引：

- 請求時間總計通常需要 2-4 個日曆週。由於對產品中繼資料和軟體進行多次反覆運算和調整，更複雜的請求或產品可能需要更長的時間。
- 我們需要在任何計劃事件或版本之前至少 45 天完成產品請求和 AMI，以便我們可以相應地排定請求的優先順序。

如果您對請求有任何疑問，請聯絡AWS Marketplace 賣方營運團隊。

將 AMIs提交至 AWS Marketplace

建立並提交至 的所有 AMIs AWS Marketplace 都必須遵守所有產品政策。我們建議在提交之前對您的 AMI 進行一些最終檢查：

- 從系統移除所有使用者登入資料；例如，所有預設密碼、授權金鑰、金鑰對、安全金鑰或其他登入資料。
- 確定根登入已停用或鎖定。僅允許 sudo 存取帳戶。

- 如果您要提交要部署到 AWS GovCloud (US) 區域的 AMI，則需要有作用中的 AWS GovCloud 帳戶，並同意 AWS GovCloud 要求，包括適用的出口控制要求。

AMI 自助式掃描

AWS Marketplace 在 中提供 AMI 掃描 AWS Marketplace 管理入口網站。對於新產品，系統會自動執行掃描，並在提交時提供結果。對於現有產品的新版本，請使用測試「新增版本」功能來啟動掃描。此程序通常在不到一小時的時間內完成，並在單一方便的位置提供全面的意見回饋。

使用自助式掃描開始共用和掃描您的 AMI

1. 導覽至 https：//https://aws.amazon.com/marketplace/management/products/server。
2. 在產品標題欄中，選擇產品。
3. 在產品詳細資訊頁面的版本區段中，選擇測試「新增版本」。
4. 在測試「新增版本」頁面上，輸入必要的版本資訊和驗證參數。
5. 選擇執行測試。
6. 檢視掃描結果。

成功掃描 AMI 之後，您可以依照目前的程序，透過上傳您的產品載入表單 (PLF) 將其提交給 AWS Marketplace 賣方營運團隊。如果您有任何問題，請聯絡AWS Marketplace 賣方營運團隊。

若要將您的 AMI 包含在自助式掃描清單中，AMI 必須位於 `us-east-1`（維吉尼亞北部） 區域，並由您的 AWS Marketplace 賣方帳戶擁有。如果您需要授予其他帳戶對 的存取權 AWS Marketplace 管理入口網站，您必須將這些帳戶註冊為賣方。如需詳細資訊，請參閱註冊為 AWS Marketplace 賣方。

AMI 複製和產品程式碼指派

提交您的 AMI 之後， 會為您指出軟體應該可在其中使用的每個區域 AWS Marketplace 建立複製的 AMIs。在此複製和發佈過程中， 會將產品代碼 AWS Marketplace 連接至複製的 AMIs。產品代碼用於控制存取和計量用量。所有提交都必須通過此 AMI 複製程序。

最終檢查清單

為了協助避免發佈產品時發生延遲，請在提交產品請求之前使用此檢查清單。

產品用量

- 生產就緒。
- 不會依時間或其他限制限制產品用量。
- 與一鍵式履行體驗相容。
- 軟體包含使用產品所需的一切，包括用戶端應用程式。
- 預設使用者使用隨機密碼和/或建立初始使用者需要驗證買方是否獲授權使用執行個體的唯一值，例如執行個體 ID。

免費或付費產品

- 使用產品不需要額外的授權。
- 付費的重新封裝軟體符合 AWS Marketplace 提交付費重新封裝軟體的要求。
- 買方不需要提供個人身分識別資訊（例如電子郵件地址）即可使用產品。

AMI 準備

- 使用硬體虛擬機器 (HVM) 虛擬化和 64 位元架構。
- 不包含任何已知的漏洞、惡意軟體或病毒。
- 買方具有 AMI 的作業系統層級管理存取權。
- 在 中使用測試「新增版本」 AWS Marketplace 管理入口網站 來掃描您的 AMI。

對於 Windows AMIs

- 使用最新版本的 `Ec2ConfigService`，如《Amazon EC2 使用者指南》中的使用 EC2Config服務設定Windows執行個體中所述。
- `Ec2SetPassword`、 `Ec2WindowsActivate`和 `Ec2HandleUserData`外掛程式已啟用，如《Amazon EC2 使用者指南》中的使用 EC2Config服務設定Windows執行個體中所述。
- 不存在訪客帳戶或遠端桌面使用者。

對於 Linux AMIs

- 根登入已鎖定並停用。
- 不包含授權金鑰、預設密碼或其他登入資料。
- 所有必要欄位皆已完成。
- 所有值都在指定的字元限制內。

- 所有 URLs載入時都不會發生錯誤。
- 產品映像至少為 110px 寬，比例介於 1：1 和 2：1 之間。
- 系統會為所有啟用的執行個體類型指定定價（適用於每小時、每月、每年）。
- 指定每月定價（適用於每小時_每月和每月定價模型）。

如果您對自動化 AMI 建置有任何疑問或意見，請聯絡AWS Marketplace 賣方營運團隊。

在 上行銷您的產品 AWS Marketplace

您可以提升 的意識 AWS Marketplace ，並直接將流量導向您的產品頁面，藉此為產品的成功做出貢獻 AWS Marketplace。下列各節提供的資訊和支援可協助您行銷已列出的產品或產品 AWS Marketplace。

主題

- 180 天 GTM 學院
- 宣布您產品的可用性
- AWS Marketplace 訊息
- 上的檢閱 AWS Marketplace
- 連結至 AWS Marketplace
- 使用 AWS Marketplace 品牌
- 了解以賣方 AWS 身分購買
- 在 上直接連結至您的產品 AWS Marketplace
- 新聞發佈
- AWS Marketplace 商標使用準則

180 天 GTM 學院

180 天的 GTM 學院適用於所有 AWS Marketplace 賣方，並提供自助go-to-market (GTM) 資源，協助您建置、啟用和追蹤產品的需求產生行銷活動 AWS Marketplace。您可以：

- 快速追蹤清單的需求。
- 將 AWS 和 AWS Marketplace 價值主張整合到您的訊息中，以改善行銷投資回報並增強客戶訊息。
- 您可以在 AWS Marketplace Go-to-Market 計劃指南中或在其中進行。

您可以從 的 Marketplace 資源區段存取 180 天 GTM 學院AWS Marketplace 管理入口網站。

宣布您產品的可用性

我們鼓勵您廣泛宣布您的 產品是否可用 AWS Marketplace。您可以透過新聞發佈、推文、部落格或任何其他偏好的媒體管道來執行此操作。我們已提供範例文字供您納入，以及使用我們商標和發行新聞發佈的指導方針和指示。

我們將在公開之前檢閱您的部落格、推文和其他非新聞發佈公告，以確保與 AWS 簡訊和品牌指導方針或語音保持一致。將您的審核請求提交給您的 AWS 客戶經理。審核最多需要 10 個工作天才能完成。當您張貼任何 推文、部落格或新聞發佈時，請通知我們，我們將盡最大努力重新張貼以提高其可見性。

AWS Marketplace 訊息

在客戶通訊中，您可能想要描述使用 購買產品的目的、目標和好處 AWS Marketplace。參考 時，請使用下列訊息 AWS Marketplace。

什麼是 AWS Marketplace？

AWS Marketplace 是一種線上商店，可讓客戶輕鬆尋找、比較並立即開始使用執行 的軟體和服務 AWS。的訪客 AWS Marketplace 可以使用 1-Click 部署快速啟動預先設定的軟體，並僅按小時或月份支付其使用量。 AWS 處理帳單和付款，以及軟體費用會顯示在客戶的 AWS 帳單上。

客戶為什麼要購物 AWS Marketplace？

尋找和部署軟體可能具有挑戰性。 AWS Marketplace 具有各式各樣的商業和免費 IT 和商業軟體，包括資料庫和應用程式伺服器、IoT 解決方案、開發人員工具和商業應用程式等軟體基礎設施。 AWS Marketplace 可讓客戶比較選項、閱讀評論，並快速找到他們想要的軟體。然後，他們可以使用 1-Click 或使用 將其部署到自己的 Amazon Elastic Compute Cloud 執行個體 AWS Marketplace 管理入口網站。

軟體價格清楚地張貼在網站上，客戶可以立即購買大多數軟體，其中付款工具已在 Amazon Web Services 的檔案中。軟體費用會顯示在與 AWS 基礎設施費用相同的每月帳單上。

軟體或 SaaS 賣方為什麼銷售 AWS Marketplace？

使用 AWS Marketplace，軟體和軟體即服務 (SaaS) 銷售者在 上執行的 產品， AWS 可以受益於提高客戶意識、簡化部署和自動計費。

AWS Marketplace 協助在 上執行的軟體和服務的軟體和 SaaS 賣方 AWS ，透過將他們的產品公開給數十萬個 AWS 客戶，從個別軟體開發人員到大型企業。

在 上銷售 AWS Marketplace 可讓獨立軟體廠商 (ISVs) 為其軟體新增每小時計費，而無需進行昂貴的程式碼變更。他們只需將 Amazon Machine Image (AMI) 上傳至 AWS ，並提供每小時成本。帳單由管理 AWS Marketplace，讓賣方承擔計量用量、管理客戶帳戶和處理付款的責任，讓軟體開發人員有更多時間專注於建置絕佳的軟體。

此外，客戶可以輕鬆部署軟體的預先設定映像，簡化新客戶的加入。

上的檢閱 AWS Marketplace

AWS Marketplace 可讓客戶提交您產品的評論。我們也為 G2 等聯合審查者提供能力，G2 是策劃獨立產品審查business-to-business市場，可在 上整合其聯合審查 AWS Marketplace。

AWS Marketplace 客戶檢閱必須符合買方使用者指南中列出的檢閱準則。檢閱提交在 上不會發佈，AWS Marketplace 直到檢閱提交後確認其符合我們的檢閱條件為止。如需檢閱準則的詳細資訊，請參閱產品檢閱。聯合審核組織使用自己的唯一程序來驗證其審核，並且在發行 AWS Marketplace 之前不會由 審核。如果您認為您產品的聯合審查不符合產品審查準則，或產品審查包含令人反感的內容，請聯絡AWS Marketplace 賣方營運團隊。

檢閱者也可以根據 5 星評分系統為您的產品提供星星評分。這些評分的平均值是為您的產品提供整體的星星評分。聯合審查也可以包含星星評分，但來自聯合審查的星星評分不會以 AWS 客戶星星評分進行平均。

以下是有關產品檢閱功能的其他重點：

- 您無法從中移除產品檢閱 AWS Marketplace。不過，只要評論符合管理產品評論的審查條件，您就可以對任何評論留下評論。
- 如果您認為審核不符合審核準則或包含令人反感的內容，您可以聯絡AWS Marketplace 賣方營運團隊並描述您的疑慮。
- AWS 在 中搜尋產品的客戶 AWS Marketplace 可以根據評分、已驗證的評論和外部來源的評論來搜尋和篩選結果。 AWS 客戶可以在搜尋結果中查看外部來源的評分以及 AWS 客戶評分。
- 您產品的聯合審查會自動新增至 ， AWS Marketplace 您無需支付任何費用。由於會自動新增檢閱，因此您不需要提交請求，即可新增聯合檢閱。
- 如果您的產品沒有任何聯合審核，您可以聯絡聯合審核者，並遵循他們的程序來審核您的產品。例如，使用 G2，您可以造訪其網站並宣告您的產品頁面，以開始其檢閱程序。

連結至 AWS Marketplace

您的公司可能有 Web 存在，其中描述和提升您的產品。我們建議您強調產品可在 上執行 AWS ，並且可以使用 購買 AWS Marketplace。為了簡化客戶探索和部署軟體的程序，我們提供了將客戶連結至產品的指示。

使用 AWS Marketplace 品牌

> **Note**
>
> 您必須是 AWS 合作夥伴，才能使用此處列出的指南和資產。如需成為 AWS 合作夥伴的詳細資訊，請參閱加入 AWS 合作夥伴網路。

AWS Marketplace 提供訊息指引和品牌資產，協助您公開您的軟體、資料或服務。如果您想要在 上提升清單 AWS Marketplace，您可以下載下列資源：

- AWS 合作夥伴創意與傳訊指南 2023
- 可用於 AWS Marketplace品牌資產

下載 資源

1. 登入 AWS Partner Central。
2. 將下列 URL 貼到瀏覽器的地址列，然後開啟頁面：AWS 合作夥伴創意與傳訊指南 2023。
3. 選擇 Download (下載)。
4. 針對品牌資產中的可用 AWS Marketplace，重複步驟 2 和 3。

了解以賣方 AWS 身分購買

使用 購買 AWS 可讓 AWS Marketplace 賣方在自己的網站上顯示和銷售 AWS Marketplace 產品。AWS Marketplace 賣方可以與隨 AWS Marketplace call-to-action(CTA) 按鈕和 APIs 整合，以在賣方擁有的網站上建置 AWS驅動型軟體探索和採購體驗。其中包括 AWS 賣方擁有的 Web 屬性，包括獨立軟體廠商 ISVs)、頻道合作夥伴 (CPs)、經銷商 (DPs)。

> **Note**
>
> 如需使用 AWS 功能購買的詳細資訊，請參閱使用 購買 AWS。

CTA 按鈕包括：

- 使用 購買 AWS

- 在 上檢視優惠 AWS
- 免費試用 AWS
- 請求私有優惠
- 請求示範

使用 購買 AWS 可提供下列優點：

- AWS 網站上的聯合品牌和探索：在您的網站上建立採用 技術的聯合品牌體驗 AWS Marketplace。讓客戶能夠從 AWS Marketplace 您網站上的 搜尋精選產品清單、篩選產品、存取檢閱、請求自訂定價，以及使用其 AWS 帳戶放心進行購買。
- AWS 客戶的採購速度：網站上的使用 購買 AWS 按鈕可讓客戶快速訂閱免費試用，或使用其 AWS 帳戶進行購買。透過 購買 AWS 可為客戶提供簡化的聯合品牌購買體驗 AWS Marketplace。
- 利用 中的現有操作 AWS Marketplace：透過您的網站將 的探索和採購優勢擴展 AWS Marketplace 到客戶，同時利用 中的現有程序 AWS Marketplace 來管理您的清單、支出、支援和洞見，而無需額外努力。

使用 AWS 先決條件購買

若要啟用使用 購買 AWS，您必須：

- 註冊為 中的賣方 AWS Marketplace。
- 成為 中的授權賣方或經銷商 AWS Marketplace ，並同意賣方的 AWS Marketplace 服務條款。
- 遵守AWS 商標準則。
- 遵守使用 AWS 商標購買授權。
- 在其中至少有一個公有 SaaS 產品清單 AWS Marketplace。

> **Note**
>
> 目前，使用 購買僅 AWS 支援 SaaS 產品。

- 遵守使用 AWS 品牌購買準則：
  - 登入您的 AWS Partner Central 帳戶，下載行銷工具組，其中包含品牌元素和行銷資產要遵循 AWS Marketplace 的品牌資產和規格。
  - 如需使用品牌元素和行銷資產的指示，請參閱使用 AWS 創意和簡訊購買準則。

使用 AWS 購買：以賣方身分入門

檢閱品牌變更

使用 購買 AWS 體驗啟動之前，請完成下列步驟以提交品牌審核請求：

1. 導覽至商業市場 AWS Marketplace 管理入口網站 並提交支援票證
   a. 選取以 購買 AWS做為類別
   b. 選取品牌檢閱做為子類別。
2. 在票證中提供下列項目：
   a. 您的主要電子郵件地址和賣方 ID。
   b. 在提供有關請求的詳細資訊區段中：
      i. 您將實作與 AWS 整合一起購買的 URL
      ii. 您的沙盒、即時網站或模型的連結，供 AWS Marketplace 營運團隊檢閱。如果您未包含此項目，操作團隊會在您提交票證後請求此項目。
3. AWS Marketplace 賣方營運團隊會根據使用 AWS 品牌購買準則來評估您的實作。他們將提供意見回饋以確保合規。您必須在啟動之前處理所有意見回饋並取得核准。

使用 AWS 報告購買

使用 AWS 指標購買 可讓賣方全面檢視其轉換漏斗。 AWS 賣方購買儀表板提供詳細的洞見，以衡量使用者參與度。賣方可以透過 存取具有 AWS 指標的購買AWS Marketplace 管理入口網站。

若要檢視使用 AWS 指標購買儀表板：

1. 登入 AWS Marketplace 管理入口網站。
2. 導覽至 Insights。
3. 按一下行銷。
4. 按一下使用 購買 AWS

或者，連結其帳戶的賣方可以透過 Partner Central AWS 存取指標。若要檢視這些指標，請登入 AWS Partner Central 並導覽至分析區段。

在 上直接連結至您的產品 AWS Marketplace

為您的網站或行銷資料新增深層連結，引導客戶前往特定的產品頁面 AWS Marketplace。使用下列連結格式：

```
https://aws.amazon.com/marketplace/pp/prodview-prodview ID
```

使用產品的唯一英數字元代碼取代 *prodview ID* AWS Marketplace。例如：

Example

```
https://aws.amazon.com/marketplace/pp/prodview-1234567890abcdef
```

當您在 aws.amazon.com/marketplace 上搜尋應用程式時，prodview ID 會出現在 URL 中。如需尋找 prodview ID 的協助，請聯絡您的客戶經理。

> **Note**
>
> 使用連結之前，請先進行測試，以確保連結引導您的客戶前往正確的頁面。

新聞發佈

建議您 AWS Marketplace 透過任何偏好的管道，在 上宣布產品的可用性。不過，Amazon AWS Marketplace 必須先檢閱並簽署參考的所有新聞發佈，才能發佈或公告。雖然我們鼓勵您公告，但我們不支援與 AWS Marketplace 賣方的聯合新聞發佈。我們將依case-by-case支援具有引號的新聞發佈 AWS。引號必須符合多種條件，包括但不限於：它宣布上列出的新產品或服務， AWS Marketplace 或包含 使用的客戶參考 AWS Marketplace。

您必須草擬所有新聞發佈。我們建議使用以下標題：【插入產品名稱】 現已推出 AWS Marketplace。為保持一致性，請使用本文件中的訊息。

新聞發佈應該：

- 清楚準確地描述公告與 Amazon.com 有何關聯
- 釐清您在 AWS 和 客戶的角色
- 以客戶為中心並強調客戶的好處

新聞發佈不應：

- 使用合作夥伴、合作夥伴關係或聯盟等術語來描述關係。我們偏好協議、團隊或關係。
- 除非事先同意，否則請包含來自 Amazon Web Services 主管的引號。
- 除非參考公司樣板中的網站，否則請包含任何銷售預測或使用商家的 .com。
- 將您的組織稱為 Amazon.com 的關聯，因為這可能會與我們的線上關聯計劃 Amazon Associates 混淆。
- 公開有關 Amazon.com 的專屬資訊，或參考我們的股票代號。

以文字格式提交給您的客戶經理，以檢閱您的新聞發佈。此外，使用任何商標之前，請檢閱 Amazon Web Services 商標準則。 AWS AWS Marketplace 商標特定的指導方針位於下一節。

AWS Marketplace 商標使用準則

這些準則適用於您在 Amazon.com， Inc. 和/或其附屬公司 (Amazon) 事先核准的資料中使用 AWS Marketplace 標誌和商標（分別為「商標」和統稱為「商標」）。始終需要嚴格遵守這些準則，任何使用商標違反這些準則的行為都會自動終止任何與您使用商標相關的授權。

1. 您可以僅將商標用於 Amazon 明確授權的目的，而且您的使用必須：(i) 遵守與 Amazon 之間有關使用任何商標（統稱「協議」）的所有協議up-to-date；(ii) 遵守本指南的up-to-date；以及 (iii) 遵守 Amazon 可能不時發行適用於使用商標的任何其他條款、條件或政策。
2. 我們會提供核准的商標映像供您使用。您不得以任何方式更改商標，包括但不限於變更商標的比例、顏色或字型，或從商標中新增或移除任何元素。
3. 您不得以任何暗示 Amazon 贊助或背書的方式使用商標，除非是依協議特別授權使用商標。
4. 您不得使用商標來詆毀 Amazon、其產品或服務，或以 Amazon 自行決定可能減少或以其他方式損害或損害商標中 Amazon 的信譽的方式。
5. 商標必須單獨顯示，商標的每一端與其他視覺、圖形或文字元素之間必須具有合理的間距。在任何情況下，商標都不應放置在干擾商標可讀性或顯示的任何背景上。
6. 您必須在任何顯示商標的資料中包含下列陳述式：「AWS Marketplace 和 AWS Marketplace 標誌是 Amazon.com， Inc. 或其附屬公司的商標。
7. 您確認，商標的所有權利都是 Amazon 的專屬資產，而您使用商標所產生的所有商譽都將符合 Amazon 的利益。您不會採取任何與 Amazon 在商標中的權利或所有權相衝突的動作。

Amazon 保留權利，可自行決定隨時修改這些準則和/或核准的商標，並針對任何使用採取適當動作，無需許可或不符合這些準則的任何使用。如果您對這些準則有任何疑問，請聯絡 trademarks@amazon.com 尋求協助，或來信至下列地址：

Amazon.com， Inc.，收件人：商標

郵政信箱 81226

華盛頓州西雅圖 98108-1226

AWS Marketplace 事件的賣方通知

您可以透過電子郵件、Amazon EventBridge 事件和 Amazon Simple Notification Service (Amazon SNS) 主題接收與 AWS Marketplace 產品相關的及時通知。 AWS Marketplace 提供。例如，您可以接收與私有優惠、專業服務產品的請求和重複掃描提醒相關的通知。本主題提供可用通知和事件選項的概觀。

- 電子郵件通知 – 在 中 AWS Marketplace ，賣方、買方和獨立軟體廠商 (ISVs) 可以接收電子郵件通知。 AWS Marketplace 使用您的根帳戶即時傳送自動電子郵件到您 AWS 帳戶的電子郵件，讓您了解 AWS Marketplace 優惠和協議。您也可以為通知新增自訂電子郵件別名，並取消訂閱電子郵件通知。如需詳細資訊，請參閱管理 AWS Marketplace 事件的電子郵件通知。
- Amazon SNS 主題 – 若要接收產品客戶訂閱變更的通知，您可以訂閱產品建立期間 AWS Marketplace 提供給您的 Amazon SNS 主題。例如，您可以知道客戶何時接受私有優惠。如需詳細資訊，請參閱您產品類型的 Amazon SNS 主題：
  - 軟體即服務 (SaaS) 產品
  - Amazon Machine Image (AMI) 產品
  - 容器產品
- EventBridge – 您可以使用 EventBridge 來接收 動作的事件 AWS Marketplace，例如建立優惠時。事件包含 ID、過期日期和產品詳細資訊等資訊。如需詳細資訊，請參閱 Amazon EventBridge 事件和《Amazon EventBridge 使用者指南》。

下列主題提供有關 中通知和事件的詳細資訊 AWS Marketplace。

主題

- 管理 AWS Marketplace 事件的電子郵件通知
- Amazon EventBridge 事件

管理 AWS Marketplace 事件的電子郵件通知

AWS Marketplace 會傳送電子郵件通知優惠、協議、訂閱、產品、安全性、帳單和付款，以及 Private Marketplace 的更新。獨立軟體廠商 ISVs)、 AWS Marketplace 頻道合作夥伴和客戶可以接收電子郵件通知。如需針對優惠和協議更新傳送的電子郵件通知範例和詳細資訊，請參閱 事件類型。

AWS Marketplace 會將電子郵件通知傳送至與您 AWS 帳戶的根使用者相關聯的電子郵件地址。若要更新與 AWS 您的帳戶相關聯的電子郵件地址，請參閱更新 AWS 您帳戶的主要聯絡人。您也可以為通知新增自訂電子郵件別名，並取消訂閱電子郵件通知。

> **Note**
>
> 如果您遺失 AWS Marketplace 電子郵件，請檢查您的垃圾郵件資料夾或調整電子郵件設定。來自 的電子郵件通知 AWS Marketplace 會從 傳送no-reply@marketplace.aws。Google 和 Yahoo 等供應商可能會篩選這些項目。如需說明，請參閱防止有效電子郵件前往 Yahoo Mail 中的垃圾郵件 (Google) 或封鎖和解除封鎖電子郵件地址。

主題

- 事件類型
- 欄位描述
- 管理 通知

事件類型

除了機器學習產品之外，所有產品和定價類型的電子郵件通知都支援下列事件類型。

Offers

下表顯示 優惠的事件。優惠是買方使用產品的一組條款。如需詳細資訊，請參閱 為您的 AWS Marketplace 產品準備私有優惠。

電子郵件	事件	收件人	Title	欄位
提供發佈的 CP 通知 CPPO	發佈 AWS Marketplace 頻道合作夥伴私有優惠。	頻道合作夥伴	AWS Marketplace 發佈的頻道合作夥伴私有優惠	客戶 AWS 帳戶 ID、ISV 名稱、ISV AWS 帳戶 ID、產品名稱、產品 ID、優惠 ID、機會 ID、優惠發佈日期、優惠過期日期

電子郵件	事件	收件人	Title	欄位
提供發佈的 ISV 通知 CPPO	發佈 AWS Marketplace 頻道合作夥伴私有優惠。	獨立軟體開發商	AWS Marketplace 發佈的頻道合作夥伴私有優惠	客戶 AWS 帳戶 IDs、通路合作夥伴、通路合作夥伴 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、機會 ID、優惠發佈日期、優惠過期日期、批發價格
OfferPublSVNotificationSCP-1.0	賣方發佈 AWS Marketplace 頻道合作夥伴私有優惠。	ISV 或 AWS Marketplace 頻道合作夥伴	已發佈的私有優惠	客戶 AWS 帳戶 IDs、產品名稱、產品 ID、優惠名稱、優惠 ID、優惠發佈日期、優惠過期日期、合約總價值
經銷商機會已過期	AWS Marketplace 通路合作夥伴銷售授權過期。	AWS Marketplace 頻道合作夥伴	優惠：檢視過期的銷售授權	ISV、產品名稱、產品 ID
經銷商機會已撤銷	ISV 撤銷 AWS Marketplace 通路合作夥伴銷售授權。	AWS Marketplace 頻道合作夥伴	優惠：檢視已停用的銷售授權	ISV、產品名稱、產品 ID

電子郵件	事件	收件人	Title	欄位
已建立經銷商機會	ISV 會為 AWS Marketplace 通路合作夥伴建立銷售授權或機會。	AWS Marketplace 頻道合作夥伴	優惠：為新的銷售授權建立私有優惠	ISV、產品名稱、產品 ID、銷售授權持續時間

協議

當買方接受公開優惠、私有優惠或 AWS Marketplace 通路合作夥伴私有優惠時，ISVs 和 AWS Marketplace 通路合作夥伴會收到電子郵件通知。接受優惠時會建立協議。也會針對協議生命週期事件傳送電子郵件通知。這些包括買方接受後的協議開始、取消、取代和失敗。下表顯示針對協議事件傳送的電子郵件。電子郵件通知目前適用於特定產品類型。您的特定產品類型可能還沒有此功能。

電子郵件	事件	收件人	Title	欄位
已接受優惠 – 私有或公有優惠	AWS Marketplace 客戶接受公有或私有優惠。	獨立軟體開發商	客戶接受了 AWS Marketplace 優惠。	客戶公司名稱、客戶 AWS 帳戶 ID、產品名稱、產品 ID、協議 ID、協議開始日期、協議結束日期、協議接受日期、購買金額
已接受優惠 – AWS Marketplace 通路合作夥	AWS Marketplace 客戶接受 AWS Marketpla	獨立軟體開發商	客戶接受 AWS Marketplace 了頻道合作夥伴	客戶公司名稱、客戶 AWS 帳戶 ID、通路合作夥伴名稱、通路合作夥伴 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期、協議接受日期、批發價格

電子郵件	事件	收件人	Title	欄位
伴私有優惠。	ce 頻道合作夥伴私有優惠。		私有優惠。	
已接受優惠 – AWS Marketplace 通路合作夥伴私有優惠。	AWS Marketplace 客戶接受 AWS Marketplace 頻道合作夥伴私有優惠。	AWS Marketplace 頻道合作夥伴	客戶接受 AWS Marketplace 了頻道合作夥伴私有優惠。	客戶公司名稱、客戶 AWS 帳戶 ID、ISV 名稱、ISV AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期、協議接受日期、批發價格、利潤、購買金額
協議已開始 – 私有優惠	合約或訂閱產品的 AWS Marketplace 協議已從未來開始日期的私有優惠開始。	獨立軟體開發商	AWS Marketplace 協議已開始	客戶 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期、購買金額

電子郵件	事件	收件人	Title	欄位
協議已開始 – AWS Marketplace 通路合作夥伴私有優惠	合約或訂閱產品的 AWS Marketplace 協議已從 AWS Marketplace 通路合作夥伴私有優惠開始。	獨立軟體開發商	AWS Marketplace 協議已開始。	客戶 AWS 帳戶 ID、通路合作夥伴名稱、通路合作夥伴 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期、批發價格
協議已開始 – 私有優惠	接受 AWS Marketplace 通路合作夥伴私有優惠會啟動合約或訂閱產品的協議。	AWS Marketplace 頻道合作夥伴	AWS Marketplace 協議已開始。	客戶 AWS 帳戶 ID、ISV 名稱、ISV AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期、利潤、購買金額
協議已取消 – 私有或公有優惠	取消私有或公開優惠協議。	獨立軟體開發商	AWS Marketplace 協議已取消。	客戶 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期

電子郵件	事件	收件人	Title	欄位
協議已取消 – AWS Marketplace 管道合作夥伴私有優惠	取消 AWS Marketplace 頻道合作夥伴私有優惠協議。	獨立軟體開發商	AWS Marketplace 協議已取消。	客戶 AWS 帳戶 ID、通路合作夥伴、通路合作夥伴 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期
協議已取消 – 管道合作夥伴私有優惠	取消 AWS Marketplace 頻道合作夥伴私有優惠協議。	AWS Marketplace 頻道合作夥伴	AWS Marketplace 協議已取消。	客戶 AWS 帳戶 ID、ISV、ISV AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID、協議開始日期、協議結束日期
合規失敗 – 記錄賣方	AWS Marketplace 協議因客戶付款失敗而失敗。	建立優惠的 ISV 或 AWS Marketplace 頻道合作夥伴	必要動作：AWS Marketplace 協議建立失敗	訂閱 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID

電子郵件	事件	收件人	Title	欄位
付款失敗 – 記錄賣方	AWS Marketplace 協議因客戶付款失敗而失敗。	建立優惠的 ISV 或 AWS Marketplace 頻道合作夥伴	AWS Marketplace 協議的付款失敗	客戶 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、協議 ID
已取代協議 – 私有優惠	AWS Marketplace 協議會由另一個協議取代。	獨立軟體開發商	已取代 AWS Marketplace 協議	公司名稱、客戶 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、新協議 ID、協議開始日期、協議結束日期、新購買金額
已取代協議 – 管道合作夥伴私有優惠	AWS Marketplace 協議會由另一個協議取代。	獨立軟體開發商	已取代 AWS Marketplace 協議	公司名稱、客戶 AWS 帳戶 ID、通路合作夥伴、通路合作夥伴 AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、新協議 ID、協議開始日期、協議結束日期、批發價格
已取代協議 – 管道合作夥伴私有優惠	AWS Marketplace 協議會由另一個協議取代。	AWS Marketplace 頻道合作夥伴	已取代 AWS Marketplace 協議	公司名稱、客戶 AWS 帳戶 ID、ISV 名稱、ISV AWS 帳戶 ID、產品名稱、產品 ID、優惠名稱、優惠 ID、新協議 ID、協議開始日期、協議結束日期、批發價格、利潤、新購買金額

支出

如果 ISVs和頻道合作夥伴的支出因銀行帳戶詳細資訊無效而暫停，則會收到電子郵件通知。

電子郵件	事件	收件人	Title	欄位
已暫停支出 – 銀行帳戶資訊無效	AWS Marketplace 由於銀行帳戶詳細資訊無效，會暫停向賣方支付款項。	ISV 和 AWS Marketplace 頻道合作夥伴	已暫停支出	資源 ARN

`Resource ARN` 欄位會顯示無效的銀行帳戶 Amazon Resource Number (ARN)。您可以在 AWS Marketplace 管理入口網站 (AMMP) 中新增銀行帳戶，以修正此問題。

新增銀行帳戶詳細資訊

在 AWS Marketplace 管理入口網站 (AMMP) 中新增銀行帳戶詳細資訊，以修正付款中的無效銀行帳戶 ARN 錯誤。

新增銀行帳戶詳細資訊

1. AWS Marketplace 管理入口網站 使用您的 AWS Marketplace 賣方帳戶登入 。
2. 在選單列上，選擇設定。
3. 選擇付款資訊索引標籤。
4. 在銀行帳戶區段中，選擇新增銀行帳戶。
5. 新增銀行帳戶詳細資訊並檢閱其準確性。如需詳細資訊，請參閱 步驟 3：提供銀行帳戶資訊。
6. 若要從買方接收付款，請在付款方法下，選擇新增付款方法。
7. 如果系統提示您驗證了解您的客戶 (KYC) 和銀行帳戶，請遵循驗證步驟。

8. 儲存您的變更。

欄位描述

下表顯示 Offers和 協議資料表中參考的欄位說明。

欄位	描述			
客戶公司名稱	訂閱者的公司名稱。			
客戶 AWS 帳戶 ID	訂閱產品的 AWS 帳戶 ID。			
ISV 名稱	賣方公司名稱。			
ISV AWS 帳戶 ID	賣方 AWS 帳戶的 ID。			
產品名稱	產品的標題。			
產品 ID	軟體產品的易記唯一識別符。			
優惠名稱	優惠的標題。			
優惠 ID	買方簽署的優惠的識別符。			
提供可見性	優惠是公有、私有或企業合約優惠。			
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。			

欄位	描述			
協議開始日期	客戶產品訂閱開始的日期，格式為 MM-DD-YYYY 。此日期可能與未來日期協議的接受日期不同。			
協議接受日期	客戶訂閱產品的日期，格式為 MM-DD-YYYY 。			
協議結束日期	合約到期的日期，格式為 MM-DD-YYYY 。對於計量或pay-as-you-go訂閱，此日期設定為 JAN-1-9999 。			
協議結束日期	合約到期的日期，格式為 MM-DD-YYYY 。對於計量或pay-as-you-go訂閱，此日期設定為 JAN-1-9999 。			

欄位	描述			
購買金額	協議的預估成本，也稱為合約總價值。這適用於 SaaS、專業服務和伺服器產品類型，以及合約或年度定價類型。			
管道合作夥伴公司名稱	以批發成本從 ISV 購買產品或服務以向客戶轉售的帳戶名稱。			
頻道合作夥伴 AWS 帳戶 ID	從 ISV 購買產品或服務以轉售給客戶的 AWS Marketplace 通路合作夥伴 AWS 帳戶 ID。			
批發價格	ISV 將產品轉售給 AWS Marketplace 通路合作夥伴的批發成本。			
貨幣代碼	與協議預估成本相關聯的優惠定價貨幣。			
新協議 ID	續約或升級協議的協議 ID。			
優惠發佈日期	賣方發佈優惠的日期。			

欄位	描述			
優惠過期日期	優惠過期的日期。			
機會 ID	已註冊機會的唯一識別符。			
銷售授權持續時間	經銷商獲授權使用折扣建立優惠的時間長度，如銷售授權中所指定。			

管理 通知

下列主題說明如何管理 事件的電子郵件通知。

新增或更新電子郵件地址

您可以使用 為自訂電子郵件通知新增最多 10 個電子郵件地址 AWS Marketplace 管理入口網站。

新增或更新電子郵件地址

1. 登入AWS Marketplace 管理入口網站。
2. 在設定中，選擇通知索引標籤。
3. 在自訂通知的電子郵件下，選擇新增電子郵件地址。
4. 如需收件人詳細資訊，請在電子郵件地址欄位中輸入自訂電子郵件地址。
5. （選用）選擇新增收件人以新增另一個電子郵件地址（最多共 10 個）。
6. 選擇提交。

從通知取消訂閱收件人

您可以移除電子郵件地址，讓收件人取消訂閱自訂電子郵件通知。

讓收件人取消訂閱事件通知

1. 登入AWS Marketplace 管理入口網站。

2. 從設定選擇通知索引標籤。
3. 在自訂通知的電子郵件下，選擇更新電子郵件地址。
4. 如需收件人詳細資訊，請選擇移除以移除電子郵件地址。
5. 選擇提交。

> **Note**
>
> 您也可以使用電子郵件中的連結取消訂閱。

Amazon EventBridge 事件

身為賣方，您可以使用 Amazon EventBridge 接收其中事件的通知 AWS Marketplace。例如，您可以在建立優惠 AWS Marketplace 時收到來自 的事件。事件包含 ID、過期日期和產品詳細資訊等詳細資訊。EventBridge 是一種事件匯流排服務，可用來將應用程式與來自各種來源的資料連線。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。下列各節提供 EventBridge 主控台中 Marketplace Catalog 服務下事件的詳細資訊。

主題

- 新優惠的事件
- 變更集的事件
- 安全性摘要報告的事件
- 支出的事件

本主題

賣方的動作	已接收的事件	相關主題
獨立軟體廠商 (ISV) 會建立優惠並使其可供購買	`Offer Released`	the section called “新優惠的事件”
頻道合作夥伴使用 ISV 的產品來建立優惠	`Offer Released`	the section called “新優惠的事件”
管道合作夥伴建立優惠	`Offer Released`	the section called “新優惠的事件”

賣方的動作	已接收的事件	相關主題
變更集成功	`Change Set Succeeded`	the section called “變更集的事件”
變更集失敗	`Change Set Failed`	the section called “變更集的事件”
變更集已取消	`Change Set Cancelled`	the section called “變更集的事件”
ISV 產品偵測到安全漏洞	`Products Security Report Created`	the section called “安全性摘要報告的事件”

新優惠的事件

當賣方建立優惠並使其可供購買時，他們可以接收具有下列詳細資訊類型的事件：`Offer Released`。

> **Note**
>
> 如需建立 EventBridge 規則的資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 規則。

以下是 ISV 所建立新優惠的範例事件內文。

```
{
  "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/bbbbaaaa-abcd-1111-abcd-666666666666",
      "title": "Product Title"
    },
    "manufacturer": {
      "accountId": "123456789012",
      "name": "Manufacturer Account Name"
    },
    "sellerOfRecord": {
      "accountId": "123456789012",
      "name": "Seller Account Name"
    },
    "targetedBuyerAccountIds": [
      "999988887777",
      "111122223333"
    ]
  }
}
```

以下是頻道合作夥伴使用 ISV 的產品來建立優惠時的範例事件內文。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Offer Released",
  "source": "aws.marketplacecatalog",
  "account": "123456789012",
  "time": "2023-08-26T00:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aws-marketplace:us-east-1:987654321098:AWSMarketplace/Offer/
offer-1234567890123"
  ],
```

```
  "detail": {
    "requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
    "catalog": "AWSMarketplace",
    "offer": {
      "id": "offer-1234567890123",
      "arn": "arn:aws:catalog:us-east-1:987654321098:Offer/offer-1234567890123",
      "name": "Offer Name",
      "expirationDate": "2025-08-26T00:00:00Z"
    },
    "product": {
      "id": "bbbbaaaa-abcd-1111-abcd-666666666666",
      "arn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
SaaSProduct/bbbbaaaa-abcd-1111-abcd-666666666666",
      "title": "Product Title"
    },
    "manufacturer": {
      "accountId": "123456789012",
      "name": "Manufacturer Account Name"
    },
    "sellerOfRecord": {
      "accountId": "987654321098",
      "name": "Seller Account Name"
    },
    "targetedBuyerAccountIds": ["999988887777", "111122223333"],
    }
  }
}
```

以下是頻道合作夥伴建立優惠時的範例事件內文。

```
{
  "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/bbbbaaaa-abcd-1111-abcd-666666666666",
      "title": "Product Title"
    },
    "manufacturer": {
      "accountId": "123456789012",
      "name": "Manufacturer Account Name"
    },
    "sellerOfRecord": {
      "accountId": "987654321098",
      "name": "Seller Account Name"
    },
    "targetedBuyerAccountIds": ["999988887777", "111122223333"],
    }
  }
}
```

變更集的事件

當變更集完成時，賣方、通路合作夥伴和私有市場管理員可以接收事件。當變更集完成且狀態為成功、失敗或取消時， AWS Marketplace 目錄 API 會傳送事件。這些事件的來源為 `aws.marketplacecatalog`，可能的詳細資訊類型值為 `Change Set Succeeded`、 `Change Set Failed`和 `Change Set Cancelled`。

> **Note**
>
> 如需變更集的資訊，請參閱《AWS Marketplace 目錄 API 參考》中的使用變更集。

每個事件都包含變更請求詳細資訊，例如變更集 ID、變更集名稱、事件詳細資訊類型、失敗代碼 （針對失敗的請求），以及請求的開始和結束時間。這可讓您監控變更集，而無需持續查詢`DescribeChangeSet`動作或檢查變更請求 AWS Marketplace 管理入口網站 的狀態。

> **Note**
>
> 如需建立 EventBridge 規則的資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 規則。

以下是Change Set Succeeded詳細資訊類型的事件內文範例。

```
{
  "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"
  }
}
```

以下是Change Set Failed詳細資訊類型的事件內文範例。

```
{
  "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"
  }
}
```

以下是Change Set Cancelled詳細資訊類型的事件內文範例。

```
{
  "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"
  }
}
```

安全性摘要報告的事件

當在賣方的產品上偵測到安全漏洞時，他們可以收到摘要報告事件，並定期提醒未解決的產品問題。這些事件的來源為 aws.marketplacecatalog，詳細資訊類型為 Products Security Report Created。

每個事件都包含偵測到問題的產品和版本計數摘要、影響最新版本數量的計數，以及需要解決以避免這些產品或版本的暫時限制的日期。

> **Note**
>
> 如需建立 EventBridge 規則的資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 規則。
> 如需管理安全事件的詳細資訊，請參閱部落格部落格文章中的如何改善產品目錄的安全性 AWS MarketplaceAWS 。

以下是Products Security Report Created詳細資訊類型的事件內文範例。

```
{
  "version": "0",
   "id": "01234567-0123-0123-0123-0123456789ab",
   "detail-type": "Products Security Report Created",
   "source": "aws.marketplacecatalog",
   "account": "123456789012",
   "time": "2023-10-31T00:00:00Z",
   "region": "us-east-1",
   "resources": [],
   "detail": {
     "numberOfProductsWithIssues": 1,
     "numberOfVersionsWithIssues": 1,
     "numberOfLatestVersionsWithIssues": 1,
     "newIssuesFound": true,
     "upcomingResolutionDueDate": "2023-12-01T00:00:00Z",
     "requestId": "533fa17d-3e97-5051-bcaf-1fae45fb3f8b"
   }
  }
```

支出的事件

當賣方銀行帳戶的支出因銀行帳戶詳細資訊無效而失敗時， AWS Marketplace ISVs和管道合作夥伴可能會收到事件。

在下列 JSON 事件程式碼中，這些事件`source`的值為 `aws.marketplace`，而`detail-type`值為 `Disbursement Paused`。此`resources`值顯示無效的銀行帳戶 Amazon Resource Number (ARN)。

```
{
"version": "0",
"id": "01234567-0123-0123-0123-0123456789ab",
"detail-type": "Disbursement Paused",
"source": "aws.marketplace",
"account":"<account id of end user>",
"time": "2022-11-01T13:12:22Z",
"region": "us-east-1",
"resources": ["arn:aws:payments:us-east-1:1234567890:paymentinstrument:123"],
"detail": {
"requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
"catalog": "AWSMarketplace"
 }
}
```

若要解決無效的銀行帳戶詳細資訊問題，請在 AWS Marketplace 管理入口網站中新增您的銀行帳戶詳細資訊。如需說明，請參閱新增銀行帳戶詳細資訊。

如需建立 Amazon EventBridge 規則的詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 中的規則。 EventBridge

中的賣方報告、資料饋送和儀表板 AWS Marketplace

身為 AWS Marketplace 賣方，您可以使用下列工具來收集和分析產品銷售的相關資訊：

- 自動 AWS Marketplace 產生並提供給所有已註冊 AWS Marketplace 賣家的報告。
- 您可以使用 API 來存取這些報告的區段。
- 提供其他客戶資訊的資料摘要，可用來識別報告中所列交易的客戶資訊。
- Amazon Amazon QuickSight 支援的儀表板，具有圖表、圖形和洞見，可協助您存取和分析財務資料。

AWS Marketplace 在報告、資料饋送和儀表板中提供盡可能多的資料，同時遵守下列各項：

- Amazon 保護客戶資料的標準和原則。
- 買方在購買產品時接受的條款與條件 AWS Marketplace。身為賣方，您在合約上必須安全地管理買方資料，並在買方要求時刪除資料。

> **Note**
>
> 若要取消報告的通知電子郵件，請聯絡AWS Marketplace 賣方營運團隊。

如需 中報告、資料摘要和儀表板的詳細資訊 AWS Marketplace，請參閱下列主題。

主題

- 中的賣方交付資料摘要 AWS Marketplace
- 中的賣方報告 AWS Marketplace
- 補充報告
- 賣方儀表板

中的賣方交付資料摘要 AWS Marketplace

AWS Marketplace 提供資料饋送作為一種機制，在賣方擁有的商業智慧工具之間，將結構化up-to-date產品和客戶資訊從 AWS Marketplace 系統傳送至賣方 Amazon S3 儲存貯體以進行 ETL（擷取、轉換

和載入）。資料摘要會收集逗號分隔值 (CSV) 檔案，並將其交付至您提供的加密 Amazon S3 儲存貯體。資料饋送會在一天內產生，並包含前一天 24 小時的資料。下列各節提供資料摘要的概觀，並說明如何存取和使用。後續各節說明每個資料饋送。

交易資料會以雙時間結構交付並附加，因此賣方可以沿著兩個時間軸存放和查詢資料，同時包含兩個時間戳

- 有效時間：當事實發生在真實世界時（您所知）
- 系統時間：當該事實記錄到資料庫時 (「當您知道時」)。

資料饋送會在前一天更新後每天午夜 UTC 交付，其中包含前一天 24 小時的資料。更新可由客戶訂閱、開立發票或 AWS 支付款項的客戶定義。

主題

- AWS Marketplace 資料饋送的儲存和結構
- 存取資料摘要
- 使用資料饋送收集和分析資料
- 資料饋送資料表概觀
- 資料饋送查詢範例
- 資料摘要

AWS Marketplace 資料饋送的儲存和結構

AWS Marketplace 提供資料饋送作為一種機制，在賣方擁有的商業智慧工具之間，將結構化up-to-date產品和客戶資訊從 AWS Marketplace 系統傳送至賣方 Amazon S3 儲存貯體以進行 ETL（擷取、轉換和載入）。本主題提供有關資料饋送結構和儲存體的詳細資訊。

資料饋送會收集逗號分隔值 (CSV) 檔案，並將其交付至您提供的加密 Amazon S3 儲存貯體。CSV 檔案具有下列特性：

- 它們遵循 4180 標準。
- 字元編碼是不含 BOM 的 UTF-8。
- 逗號用作值之間的分隔符號。
- 欄位會被雙引號逸出。
- \n 是換行字元。

- 日期會以 UTC 時區報告、以 ISO 8601 日期和時間格式報告，且在 1 秒內準確。
- 所有 *_period_start_date和 *_period_end_date值都包含在內，這表示 23:59:59是任何一天的最後一個可能時間戳記。
- 所有貨幣欄位前面都有貨幣欄位。
- 貨幣欄位使用句點 (.) 字元做為十進位分隔符號，而不使用逗號 (,) 做為千位分隔符號。

資料饋送的產生和存放方式如下：

- 資料饋送會在一天內產生，並包含前一天的 24 小時資料。
- 在 Amazon S3 儲存貯體中，資料饋送是使用下列格式按月組織：

  bucket-name/*data-feed-name_version*/year=*YYYY*/month=*MM*/data.csv
- 產生每日資料饋送時，它會附加到該月的現有 CSV 檔案。當新的月份開始時，會為每個資料饋送產生新的 CSV 檔案。
- 資料摘要中的資訊會從 2010/01/01 回填至 2020/04/30 （包含），並可在 year=2010/month=01子資料夾中的 CSV 檔案中取得。

  您可能會注意到指定資料饋送的當月檔案僅包含資料欄標頭，且沒有資料的情況。這表示該月沒有饋送的新項目。這可能會在更新頻率較低的資料饋送中發生，例如產品饋送。在這些情況下，資料可在回填資料夾中取得。
- 在 Amazon S3 中，您可以建立 Amazon S3 生命週期政策，以管理檔案在儲存貯體中保留多久。
- 您可以設定 Amazon SNS 在資料交付到加密的 Amazon S3 儲存貯體時通知您。如需如何設定通知的資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 Amazon SNS 入門。

資料的歷史化

每個資料饋送都包含記錄資料歷史記錄的資料欄。除了 之外valid_to，這些資料欄適用於所有資料饋送。它們被納入為常見的歷史記錄結構描述，有助於查詢資料。

資料欄名稱	描述
valid_from	相對於其他欄位的值，主索引鍵值對 有效的第一個日期。
valid_to	此欄僅顯示在地址資料饋送上，且一律為空白。
insert_date	將記錄插入資料饋送的日期。

資料欄名稱	描述
update_date	記錄上次更新的日期。
delete_date	此欄一律為空白。

以下顯示這些資料欄的範例。

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 和 update_date 欄位一起形成雙時間資料模型。valid_from 欄位命名為 時，會告訴您項目的有效期。如果項目已編輯，它可以在摘要中具有多個記錄，每個記錄都有不同的 update_date，但valid_from日期相同。例如，若要尋找項目的目前值，您可以從具有最新valid_from日期的記錄update_date清單中找到具有最新 的記錄。

在上述範例中，記錄最初是在 2018-12-12 建立。然後，它在 2019-03-29 變更（例如，如果記錄中的地址變更）。稍後於 2019-04-28 更正地址變更（因此 valid_from 不會變更，但 update_date會變更）。更正地址（罕見事件）會追溯變更原始valid_from日期的記錄，因此欄位不會變更。尋找最新記錄的查詢valid_from會傳回兩個記錄，具有最新記錄的 update_date會為您提供實際的目前記錄。

存取資料摘要

透過 AWS Marketplace，您可以使用資料饋送作為一種機制，在賣方擁有的商業智慧工具之間，從 Amazon S3 儲存貯體 ETL（擷取、轉換和載入）AWS Marketplace 的系統傳送結構化、up-to-date產品和客戶資訊。您需要將環境設定為接收加密 Amazon S3 儲存貯體的資料饋送。本主題說明如何存取和取消訂閱資料摘要。

主題

- 存取資料饋送
- 資料饋送政策
- 取消訂閱資料摘要

存取資料饋送

1. 使用 SQL 和 ETL （擷取、轉換、載入） 體驗來配置商業智慧或資料工程師。此人員也需要設定APIs的經驗。
2. 設定 Amazon Simple Storage Service 儲存貯體和資料摘要的訂閱。使用與您的 Marketplace 產品清單相關聯的 AWS 賣方帳戶 ID。若要這麼做，您可以觀看此 YouTube 影片或遵循下列步驟。

   影片和步驟說明如何使用有助於簡化組態的AWS CloudFormation 範本。

   a. 開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站，然後前往設定客戶資料儲存。

   b. 選擇使用 AWS CloudFormation 範本建立資源，在另一個視窗中開啟 AWS CloudFormation 主控台中的範本。

   c. 在範本中，指定下列項目，然後選擇下一步：

      - 堆疊名稱 – 您要建立以啟用資料饋送存取的資源集合。
      - Amazon S3 儲存貯體名稱 – 儲存資料摘要的儲存貯體。
      - （選用） Amazon SNS 主題名稱 – Amazon Simple Storage Service 儲存貯體時接收通知的主題。

   d. 在檢閱頁面上，確認您的項目，然後選擇建立堆疊。這會開啟具有 CloudFormation 狀態和詳細資訊的新頁面。

   e. 從資源索引標籤中，將下列資源的 Amazon Resource Name (ARNs) 從 CloudFormation 頁面複製到 AWS Marketplace 設定客戶資料儲存頁面上的欄位：

      - 用於儲存資料饋送的 Amazon S3 儲存貯體
      - AWS KMS 用於加密 Amazon S3 儲存貯體的金鑰
      - （選用） 當 將新資料 AWS 交付至 Amazon S3 儲存貯體時，用於接收通知的 Amazon SNS 主題

   f. 在設定客戶資料儲存頁面上，選擇提交。

   g. （選用） 編輯 CloudFormation 範本建立的政策。如需詳細資訊，請參閱資料饋送政策。

您現在已訂閱 資料摘要。下次產生資料摘要時，您可以存取資料。

3. 使用 ETL （擷取、轉換、載入） 操作將資料饋送連接到您的資料倉儲或關聯式資料庫。

> **Note**
>
> 資料工具具有不同的功能。您必須讓商業智慧工程師或資料工程師參與，才能設定整合以符合工具的功能。

4. 若要執行或建立 SQL 查詢，請設定資料饋送，以在資料工具中強制執行主要和外部金鑰。每個資料饋送都代表唯一的資料表，而且您必須使用實體關係來設定資料結構描述中的所有資料饋送。如需資料表和實體關係的詳細資訊，請參閱本指南資料饋送資料表概觀中的 。

5. 設定 Amazon Simple Notification Service 以自動重新整理您的資料倉儲或關聯式資料庫。您可以設定 Amazon SNS 通知，在將來自每個唯一摘要的資料交付至 Amazon S3 儲存貯體時傳送提醒。如果賣方資料工具支援此功能，則可以利用這些通知，在透過資料饋送接收新資料時自動重新整理賣方資料倉儲。請參閱《Amazon Simple Notification Service 開發人員指南》中的 Amazon SNS 入門。

   通知範例：

```
{
        "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]"
        }
```

   通知可以有下列`executionStatus`狀態：

   - `SKIPPED` – 賣方當天沒有新資料。
   - `COMPLETED` – 我們交付了包含新資料的摘要。
   - `FAILED` – 饋送交付發生問題。

6. 執行 SQL 查詢來驗證設定。您可以使用本指南中的範例查詢，或 GitHub 上的查詢，網址為 https：//https://github.com/aws-samples/aws-marketplace-api-samples/tree/main/seller-data-feeds/queries。

> **Note**
>
> 本指南中的範例查詢是針對 AWS Athena 撰寫的。您可能需要修改查詢，以便與工具搭配使用。

7. 決定商業使用者想要在何處使用資料。例如，您可以：

   - 從資料倉儲或 SQL 資料庫匯出 .csv 資料。
   - 將您的資料連接到視覺化工具，例如 PowerBI 或 Tableau。
   - 將資料映射到您的 CRM、ERP 或財務工具，例如 Salesforce、Infor 或 Netsuite。

如需 AWS CloudFormation 範本的詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的使用 AWS CloudFormation 範本。

資料饋送政策

當 CloudFormation 範本建立您的 Amazon S3 儲存貯體時，它會建立連接到該儲存貯體、 AWS KMS 金鑰和 Amazon SNS 主題的存取政策。這些政策允許 AWS Marketplace 報告服務使用資料饋送資訊寫入您的儲存貯體和 SNS 主題。每個政策都有如下所示的區段（此範例來自 Amazon S3 儲存貯體）。

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

```
        },
```

在此政策中， AWS Marketplace 會使用 `reports.marketplace.amazonaws.com`服務主體將資料推送至 Amazon S3 儲存貯體。您已在 CloudFormation 範本中指定 *amzn-s3-demo-bucket*。

當 AWS Marketplace 報告服務呼叫 Amazon S3 AWS KMS或 Amazon SNS 時，它會提供想要寫入儲存貯體的資料 ARN。若要確保寫入儲存貯體的唯一資料是代表您寫入的資料，您可以在 政策的條件`aws:SourceArn`中指定 。在下列範例中，您必須將 *account-id* 取代為 的 ID AWS 帳戶。

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

取消訂閱資料摘要

開啟 Web 瀏覽器並登入 AWS Marketplace 管理入口網站。然後，前往聯絡我們頁面，向 AWS Marketplace 賣方營運團隊提交取消訂閱請求。取消訂閱請求最多可能需要 10 個工作天才能處理。

使用資料饋送收集和分析資料

AWS Marketplace 提供資料饋送作為一種機制，在賣方擁有的商業智慧工具之間，將結構化up-to-date產品和客戶資訊從 AWS Marketplace 系統傳送至賣方 Amazon S3 儲存貯體以進行 ETL（擷取、轉換和載入）。當 Amazon S3 儲存貯體中有資料可用時，您可以透過下列方式使用資料饋送：

- 從您在 中建立的 Amazon S3 儲存貯體下載 .CSV 檔案存取資料摘要，以便您可以在試算表中檢視資料。
- 使用 ETL（擷取、轉換和載入）、SQL 查詢、商業分析工具來收集和分析資料。

  您可以使用 AWS 服務來收集和分析資料，或可執行 .CSV 型資料集分析的任何第三方工具。

如需收集和分析資料之資料摘要的詳細資訊，請參閱下列範例。

範例：使用 AWS 服務來收集和分析資料

下列程序假設您已將環境設定為接收資料摘要至 Amazon S3 儲存貯體，且儲存貯體包含資料摘要。

從資料饋送收集和分析資料

1. 從AWS Glue 主控台建立爬蟲程式，以連線至存放資料摘要的 Amazon S3 儲存貯體、擷取您想要的資料，並在 中建立中繼資料表 AWS Glue Data Catalog。

   如需 的詳細資訊 AWS Glue，請參閱 AWS Glue 開發人員指南。
2. 從 Athena 主控台，對 中的資料執行 SQL 查詢 AWS Glue Data Catalog。

   如需 Athena 的詳細資訊，請參閱《Amazon Athena 使用者指南》。
3. 從 QuickSight 主控台建立分析，然後建立資料的視覺效果。

   如需 QuickSight 的詳細資訊，請參閱《Amazon QuickSight 使用者指南》。

如需使用 AWS 服務在資料饋送中收集和分析資料之方式的詳細範例，請參閱 AWS Marketplace 部落格中的使用賣方資料饋送交付服務、Amazon Athena 和 QuickSight 建立賣方報告。

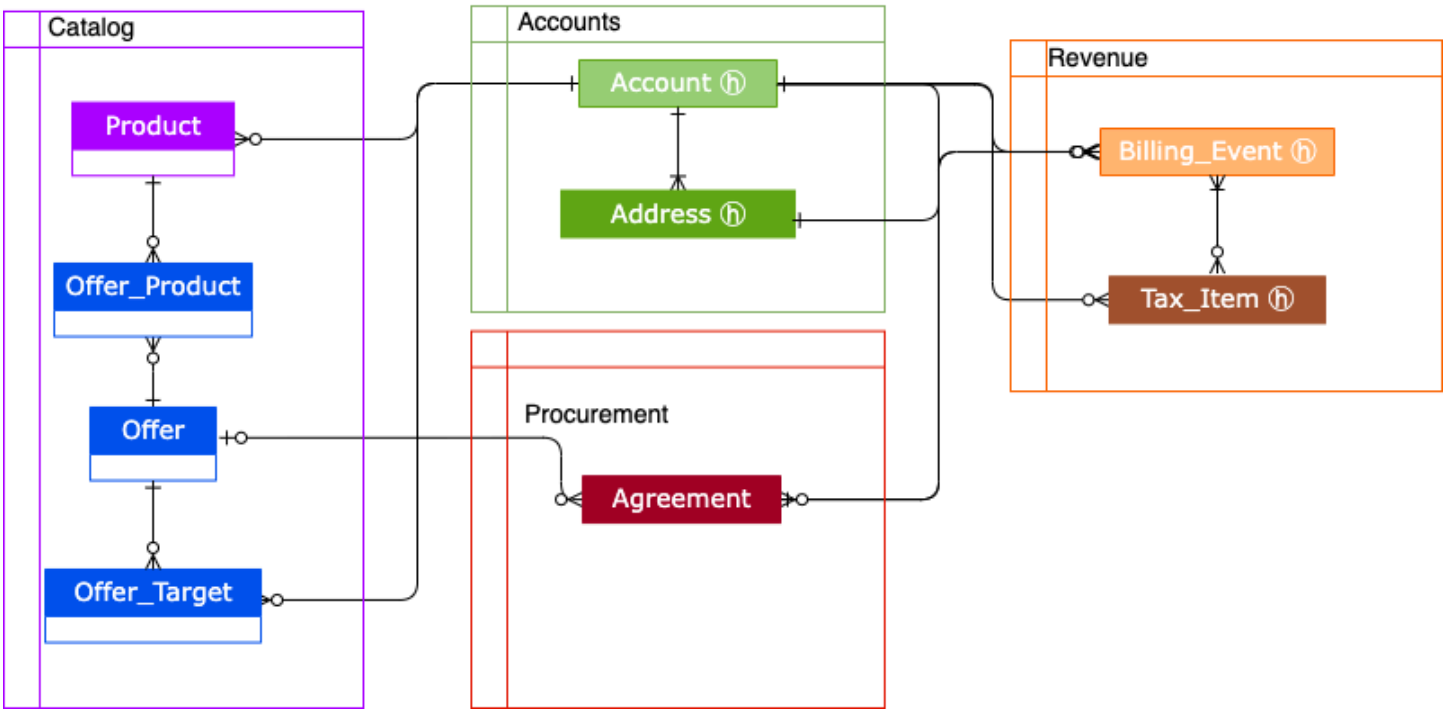
資料饋送資料表概觀

AWS Marketplace 以一組資料表的形式提供資料饋送，您可以合併為查詢提供更多內容。

AWS Marketplace 會在您的資料饋送中提供下列一般網域或感興趣的類別：

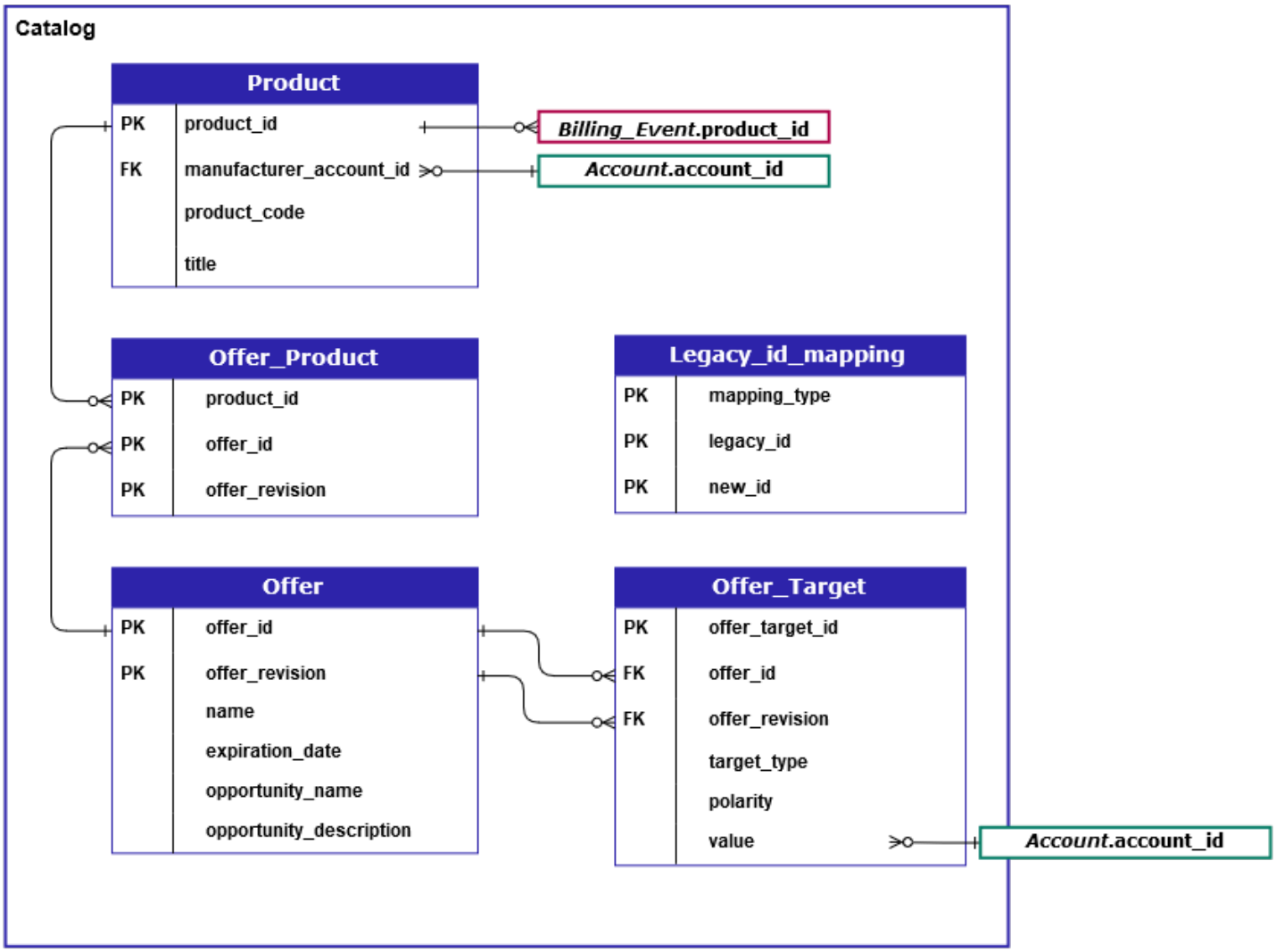
- 目錄 – 包含您帳戶中產品和優惠的相關資訊。
- 帳戶 – 包括提供或購買產品的帳戶相關資訊 AWS Marketplace （您自己的帳戶或您合作對象的帳戶，例如管道合作夥伴或買方）。
- 收入 – 包括有關帳單、支出和稅金的資訊。
- 採購 – 包含您建立做為記錄賣方的產品優惠協議相關資訊。

此圖表顯示目錄、帳戶和營收網域中的資料表。



目錄相關資料表

下圖顯示目錄網域中資料表之間的關係，以及資料表中的欄位。



`Product`、`Offer_Product`、`Offer_Target`、 `Offer`和 `Legacy_id_mapping`_tables 位於目錄網域中。

`Offer_Target` 資料表包含目標 `account_id` 的值欄位，但僅限於`target_type`值為時`account`。

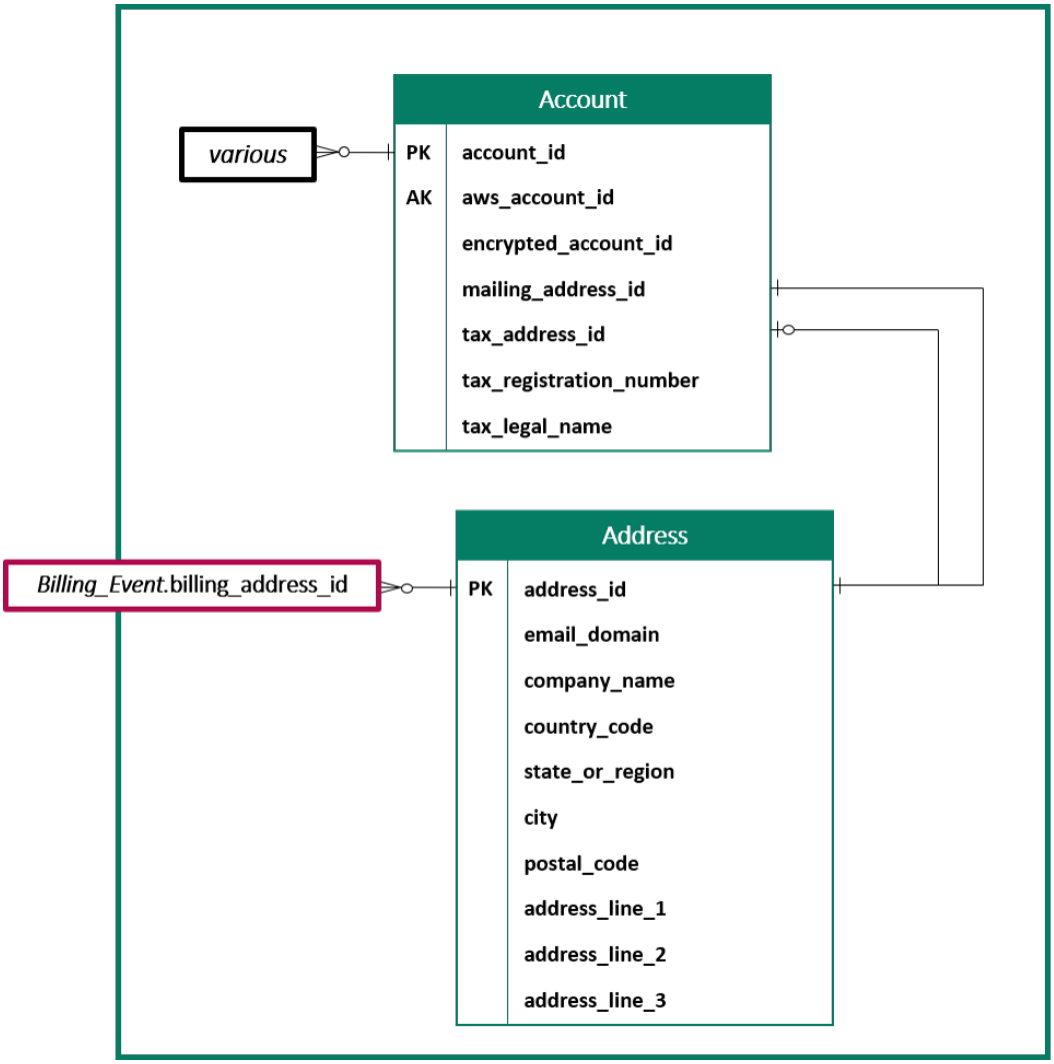
`Legacy_id_mapping` 資料表不會用於目前的資料。

> **Note**
>
> 如需這些資料表的詳細資訊，包括資料表中每個欄位的描述和可建立的聯結，請參閱下列主題：
>
> - 產品資料饋送
> - 提供產品資料饋送

- 提供資料摘要
- 提供目標資料饋送
- 舊版對映資料饋送

帳戶相關資料表

下圖顯示帳戶網域中 Account和 Address 資料表之間的關係，以及資料表中的欄位。



Note

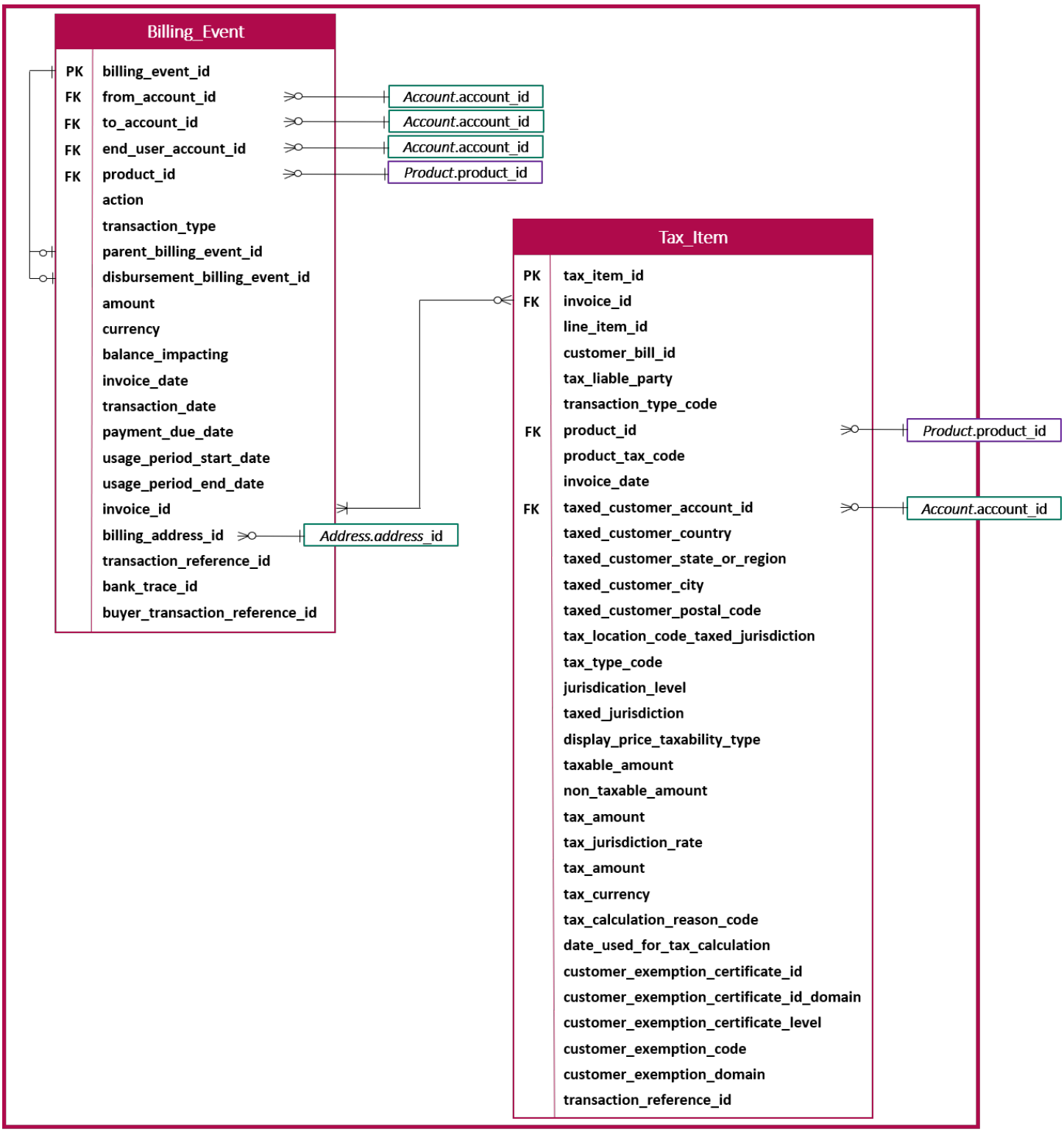
如需這些資料表的詳細資訊，包括資料表中每個欄位的描述，以及您可以建立的聯結，請參閱下列主題：

- 帳戶資料饋送

- 位址資料饋送

營收相關資料表

下圖顯示營收網域中 Billing_Event 和 Tax_Item 資料表之間的關係，以及資料表中的欄位。Billing_Event 資料表包含有關支出以及帳單事件的資訊。


Billing_Event
PK billing_event_id
FK from_account_id
FK to_account_id
FK end_user_account_id
FK product_id
action
transaction_type
parent_billing_event_id
disbursement_billing_event_id
amount
currency
balance_impacting
invoice_date
transaction_date
payment_due_date
usage_period_start_date
usage_period_end_date
invoice_id
billing_address_id
transaction_reference_id
bank_trace_id
buyer_transaction_reference_id
Account.account_id
Account.account_id
Account.account_id
Product.product_id
Address.address_id
Tax_Item
PK tax_item_id
FK invoice_id
line_item_id
customer_bill_id
tax_liable_party
transaction_type_code
FK product_id
product_tax_code
invoice_date
FK 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
jurisdication_level
taxed_jurisdiction
display_price_taxability_type
taxable_amount
non_taxable_amount
tax_amount
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
Product.product_id
Account.account_id

> **Note**
>
> 如需這些資料表的詳細資訊，包括資料表中每個欄位的描述，以及您可以建立的聯結，請參閱下列主題：
>
> - 帳單事件資料摘要
> - 稅務項目資料饋送

採購相關資料表

下圖顯示採購網域中協議資料表中的欄位。

> **Note**
>
> 如需這些資料表的詳細資訊，包括資料表中每個欄位的描述，以及可建立的聯結，請參閱本指南協議資料饋送中的 。

以下各節提供每個網域的實體關係 (ER) 圖表。每個 ER 圖表會顯示資料表和每個資料表中的欄位，以及您可以用來聯結資料表的欄位。

> **Note**
>
> 本節中的 ER 圖表不包含所有資料饋送的常用欄位。如需常見欄位的詳細資訊，請參閱AWS Marketplace 資料饋送的儲存和結構。

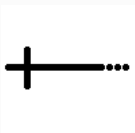
下表說明 ER 圖表中使用的符號。

符號	描述
PK	主索引鍵 – 資料表的主索引鍵。與 `valid_from` 和 `update_date` 欄位搭配使用時，它是唯一的。如需將這些欄位一起使用的詳細資訊，請參閱 資料的歷史化。如果多個欄位標記為主索引鍵，則欄位會一起形成主索引鍵。

符號	描述
	外部索引鍵 – 代表不同資料表中主索引鍵的欄位。在資料表中不一定是唯一的。 **Note** 在某些情況下，如果目前資料表中的記錄在外部資料表中沒有對應的記錄，則外部索引鍵可以為空白。
	備用索引鍵 – 可用來做為資料表中索引鍵的索引鍵。遵循與主索引鍵相同的唯一性規則。
	連接器 – 欄位之間的行代表連線，這是兩個可用來聯結資料表的欄位。該行的尾端代表連線類型。此範例代表one-to-many連線。

連接器類型

下表顯示每個連接器可以擁有的端點類型。

連接器類型	描述
	1 到 n – 具有此端點的連接器代表聯結，該聯結在這端只有一個值。
	零或一到 n – 具有此端點的連接器表示聯結，聯結的此端具有零或一個值。

連接器類型	描述
	零或大於 n – 具有此端點的連接器代表聯結，此端具有零、一或多個值。
	一或多個 n - 具有此端的連接器代表聯結，而聯結的此端具有一或多個值。

資料饋送查詢範例

本節提供使用 提供之資料摘要的複雜查詢範例 AWS Marketplace。這些範例類似於中的賣方報告 AWS Marketplace您從 取得的 AWS Marketplace 管理入口網站。您可以自訂這些查詢來建立您需要的其他報告。

查詢範例

- 協議和續約
- 計費收入
- 未收集或已分配的發票
- 已課稅發票
- 依產品分配
- 銷售補償報告

協議和續約

若要尋找您的協議和續約資料，您可以執行一組查詢，如下列範例所示。查詢會彼此建置，以建立協議和續約儀表板、精細資料區段。您可以使用範例，如下所示，或針對您的資料和使用案例進行自訂。

查詢中的註解說明查詢的功能，以及如何修改它們。

```
    Query currently under development.
```

計費收入

若要尋找發票資料，您可以執行一組查詢，如下列範例所示。這些查詢會彼此建置，以建立計費營收報告。您可以使用範例，如下所示，或針對您的資料和使用案例進行自訂。

查詢中的註解說明查詢的功能，以及如何修改它們。

```
-- 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 CPPO, 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
),
```

```
billed_revenue as (
  select
    ------------------
    -- Invoice Info --
    ------------------
    buyer_invoice_date as Invoice_Date,
    Payment_Due_Date as Payment_Due_Date,
    concat(
      'Net ',
      case
        when abs(date_diff('Day', Payment_due_date, buyer_invoice_date))>180 then
 '180+'
        else cast(abs(date_diff('Day', Payment_due_date, buyer_invoice_date)) as
 varchar)
        end,
      ' days'
    ) as payment_terms,
    buyer_invoice_id as Invoice_ID,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when seller_invoice_id = '' then null else seller_invoice_id end,
      'Not applicable') as Listing_Fee_Invoice_ID,

    ---------------------------
    --End user Information --
    ---------------------------
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when End_User_Company_Name = '' then null else End_User_Company_Name end,
      'Not available') as End_User_Company_Name,
    End_User_AWS_Account_ID,
    End_User_Encrypted_Account_ID,
    End_User_Email_Domain,
    End_User_City,
    End_User_State as End_User_State_or_Region,
    End_User_Country,
    End_User_Postal_Code,
    End_User_Address_ID,

    ---------------------------
    --Subscriber Information --
    ---------------------------
    case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'
```

```
      when Subscriber_Company_Name is null or Subscriber_Company_Name = '' then 'Not
provided'
      else Subscriber_Company_Name
      end as Subscriber_Company_Name,
   case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'
      else Subscriber_AWS_Account_ID
      end as Subscriber_AWS_Account_ID,
   case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'
      else Subscriber_Encrypted_Account_ID
      end as Subscriber_Encrypted_Account_ID,
   case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'
      when Subscriber_Email_Domain is null or Subscriber_Email_Domain = '' then 'Not
provided'
      else Subscriber_Email_Domain
      end as Subscriber_Email_Domain,
   case
      when Agreement_id is null or Agreement_ID = '' then 'Not available'
      when Subscriber_City is null or Subscriber_City = '' then 'Not provided'
      else Subscriber_City
      end as Subscriber_City,
   case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'
      when Subscriber_State is null or Subscriber_State = '' then 'Not provided'
      else Subscriber_State
      end as Subscriber_State_or_Region,
   case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'
      when Subscriber_Country is null or Subscriber_Country = '' then 'Not provided'
      else Subscriber_Country
      end as Subscriber_Country,
   case
      when Agreement_Id is null or Agreement_ID = '' then 'Not available'
      when Subscriber_Postal_Code is null or Subscriber_Postal_Code = '' then 'Not
provided'
      else Subscriber_Postal_Code
      end as Subscriber_Postal_Code,
   case
      when Agreement_ID is null or Agreement_ID = '' then 'Not available'
      when Subscriber_Address_ID is null or Subscriber_Address_ID = '' then 'Not
provided'
      else Subscriber_Address_ID
```

```
      end as Subscriber_Address_ID,

   ----------------------
   -- Procurement Info --
   ----------------------
    -- product title at time of invoice. It is possible that the title changes over
time and therefore there may be multiple product titles mapped to a single product id.
   coalesce(
     --empty value in Athena shows as '', change all '' value to null
     case when Product_Title = '' then null else Product_Title end,
     'Not provided') as Product_Title,
   -- offer name at time of invoice. It is possible that the name changes over time
therefore there may be multiple offer names mapped to a single offer id.
   case
     when Agreement_Id is null or Agreement_ID = '' then 'Not available'
     when (Offer_Name is null or Offer_Name = '') and Offer_Target = 'Public' then
'Not applicable'
     else Offer_Name
     end as Offer_Name,
   case
     when Agreement_Id is null or Agreement_ID = ''
     then 'Not available'
     else Offer_ID
     end as Offer_ID,
   -- offer target at time of invoice.,
   case
     when Agreement_Id is null or Agreement_ID = '' then 'Not available'
     else Offer_Target
     end as Offer_Visibility,
   coalesce(
     --empty value in Athena shows as '', change all '' value to null
     case when Agreement_ID = '' then null else Agreement_ID end,
     'Not available') as Agreement_ID,
   Agreement_Start_Date,
   Agreement_Acceptance_Date,
   Agreement_End_Date,
   Usage_Period_Start_Date,
   Usage_Period_End_Date,

   -----------------------
   -- Disbursement Info --
   -----------------------
   case
     when Disbursement_Flag = 'Yes' then 'Disbursed'
```

```
      when Disbursement_Flag = 'No' then 'Not Disbursed'
      else 'Other'
      end as Disbursement_Status,
    last_disbursement_date as disbursement_date,
    case
      when Disbursement_Flag = 'No' then 'Not applicable'
      when disburse_bank_trace_id_list is null or disburse_bank_trace_id_list = '' then
 'Not available'
      else disburse_bank_trace_id_list
    end as disburse_bank_trace_id,

    --------------
    -- Revenues --
    --------------
    -- We are rounding the sums using 2 decimal precision
    -- Note that the rounding method might differ between SQL implementations.
    -- The monthly revenue report is using RoundingMode.HALF_UP. This might create tiny
 discrepancies between this SQL output
    -- and the legacy report
    round(gross_revenue_invoiced,2) as Gross_Revenue,
    round(gross_refund_invoiced,2) as Gross_Refund,
    round(aws_rev_share_invoiced,2) as Listing_Fee,
    round(aws_refund_share_invoiced,2) as Listing_Fee_Refund,
    truncate(
      case
        when gross_revenue_invoiced != 0 then abs(aws_rev_share_invoiced/
gross_revenue_invoiced)
        when gross_refund_invoiced != 0 then abs(aws_refund_share_invoiced/
gross_refund_invoiced)
        else 0
      end
      ,4) as Listing_Fee_Percentage,
    round(seller_tax_share_invoiced,2) as Seller_Tax_Share,
    round(seller_tax_share_refund_invoiced,2) as Seller_Tax_Share_Refund,
    round(aws_tax_share_invoiced,2) as AWS_Tax_Share,
    round(aws_tax_share_refund_invoiced,2) as AWS_Tax_Share_Refund,
    round(aws_tax_share_listing_fee_invoiced,2) as AWS_Tax_Share_Listing_Fee,
    round(aws_tax_share_refund_listing_fee_invoiced,2) as
 AWS_Tax_Share_Refund_Listing_Fee,
    round(cogs_invoiced,2) as Wholesale_cost,
    round(cogs_refund_invoiced,2) as Wholesale_cost_Refund,
    round(seller_net_revenue,2) as Seller_Net_Revenue,
    currency as Currency,
```

```
   substring(internal_buyer_line_item_id,1,strpos(internal_buyer_line_item_id,'-')-1)
as Transaction_Reference_ID,
   broker_id as AWS_seller_of_record,

   -----------------
   -- Resale info --
   -----------------
   case
     when Opportunity_Id is null or Opportunity_Id = '' then
       case
         when Offer_Target = 'Public' then 'Not applicable'
         when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
         else null
       end
     else Opportunity_Id
   end as Resale_authorization_ID,
   case
     when Offer_Opportunity_Name is null or Offer_Opportunity_Name = '' then
       case
         when Offer_Target = 'Public' then 'Not applicable'
         when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not
null and Agreement_Id != '') then 'Not applicable'
         else null
       end
     else Offer_Opportunity_Name
   end as Resale_authorization_name,
   case
     when Offer_Opportunity_Description is null or Offer_Opportunity_Description = ''
then
       case
        when Offer_Target = 'Public' then 'Not applicable'
        when (Offer_Target is null or Offer_Target = '') and (Agreement_Id is not null
and Agreement_Id != '') then 'Not applicable'
         else null
       end
     else Offer_Opportunity_Name
   end as Resale_authorization_description,
   case
     when (Reseller_AWS_Account_ID is not null and Reseller_AWS_Account_ID != '')
       and (Reseller_Company_Name is null or Reseller_Company_Name = '') then 'Not
available'
     when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
       and (opportunity_id is null or opportunity_id = '') then 'Not applicable'
```

```
      when (select seller_account_id from seller_account) <>
manufacturer_aws_account_id
        and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
      else Reseller_Company_Name
    end as Reseller_Company_Name,
    case
      when (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '')
        and (Opportunity_Id is null or Opportunity_Id = '') then 'Not applicable'
      when (select seller_account_id from seller_account) <>
manufacturer_aws_account_id
        and (Reseller_AWS_Account_ID is null or Reseller_AWS_Account_ID = '') then 'Not
applicable'
      else Reseller_AWS_Account_ID
    end as Reseller_AWS_Account_ID,

    -----------------------
    -- Payer Information --
    -----------------------
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when Payer_Company_Name = '' then null else Payer_Company_Name end,
      'Not available') as Payer_Company_Name,
    Payer_AWS_Account_ID,
    Payer_Encrypted_Account_ID,
    Payer_Email_Domain,
    Payer_City,
    Payer_State as Payer_State_or_Region,
    Payer_Country,
    Payer_Postal_Code,
    Payer_Address_ID,

    ---------------------
    -- ISV Information --
    ---------------------
    manufacturer_aws_account_id as ISV_Account_ID,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when Manufacturer_Company_Name = '' then null else Manufacturer_Company_Name
end,
      'Not available') as ISV_Company_Name,

    ---------------------
    -- Products info --
```

```
    ---------------------
    Legacy_Product_ID,
    coalesce(
      --empty value in Athena shows as '', change all '' value to null
      case when Product_ID = '' then null else Product_ID end,
      'Not provided') as Product_ID,
    Product_Code
  from
    line_items_with_window_functions_enrich_offer_product_address_name as line
  where disbursement_id_or_invoiced = '<invoiced>'

)

select *
from billed_revenue
where invoice_date >= date_add('DAY', -90, current_date)
--where invoice_date between cast('2023-01-01' as timestamp) and cast('2024-03-01' as
 timestamp)
```

未收集或已分配的發票

若要尋找未收集或已分配的發票，您可以執行一組查詢，如下列範例所示。這些查詢會彼此建置，以建立集合和支出報告。您可以使用範例，如下所示，或針對您的資料和使用案例進行自訂。

查詢中的註解說明查詢的功能，以及如何修改它們。

```
-- 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 CPPO, 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'
      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)
```

已課稅發票

若要尋找已課稅發票，您可以執行一組查詢，如下列範例所示。這些查詢會彼此建置，以建立稅務報告。您可以使用範例，如下所示，或針對您的資料和使用案例進行自訂。

查詢中的註解說明查詢的功能，以及如何修改它們。

```
-- 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)
```

依產品分配

若要尋找產品分配的金額，您可以執行一組如下所示的查詢。此範例與支出報告賣方報告相當。

這些範例查詢會彼此建置，以建立具有支出的產品詳細資訊最終清單。它也會說明如何在特定時間點取得產品資訊。您可以如所示使用此範例，或針對您的資料和使用案例進行自訂。

查詢中的註解說明查詢的功能，以及如何修改它們。

> **Note**
>
> 執行此查詢時，我們假設使用兩個時間軸 `valid_from`和 `update`資料欄來擷取資料。如需軸的詳細資訊，請參閱 AWS Marketplace 資料饋送的儲存和結構。

```
-- 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;
```

銷售補償報告

若要依客戶尋找計費收入，您可以執行一組類似以下的查詢。此範例與銷售薪酬報告賣方報告相當。這些範例查詢會彼此建置，以建立客戶詳細資訊的最終清單，其中包含針對軟體用量向每位客戶計費的總金額。您可以使用如下所示的查詢，或為您的資料和使用案例自訂查詢。

查詢中的註解說明查詢的功能，以及如何修改它們。

> **Note**
>
> 執行此查詢時，我們假設擷取的資料使用兩個時間軸（`valid_from`和 `update`欄）。如需詳細資訊，請參閱AWS Marketplace 資料饋送的儲存和結構。

```
    -- 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 CPPO 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;
```

資料摘要

AWS Marketplace 提供多種資料摘要，以協助賣方收集和分析產品銷售的相關資訊。資料摘要可供所有已註冊的 AWS Marketplace 賣家使用。由於資料饋送是在一天內產生，因此它們包含最新的可用資料。

下列各節說明每個資料饋送。

主題

- 帳戶資料饋送
- 位址資料饋送
- 協議資料饋送
- 帳單事件資料摘要
- 舊版對映資料饋送
- 提供資料摘要
- 提供產品資料饋送
- 提供目標資料饋送
- 產品資料饋送
- 稅務項目資料饋送

帳戶資料饋送

此資料饋送提供您與之互動之所有帳戶的相關資訊：您自己、與您合作的任何通路合作夥伴、買家、付款人和所有稅務帳戶。

帳戶資料是不可變的，而且與版本號碼沒有關聯。系統會附加對欄位的變更，因此此資料饋送可能有數個具有相同值`account_id`和不同`valid_from`值的列。如需資料歷程記錄欄位的資訊，請參閱資料的歷史化。

帳戶資料摘要每 24 小時會重新整理一次，因此每天都有新資料可用。

下表說明資料饋送欄的名稱和說明。

資料欄名稱	描述
account_id	帳戶的全域唯一識別碼 (GUID)。 可用於聯結至Product、Offer_Target 、Billing_Event 和Tax_Item資料饋送中的欄位。如需可用於聯結之欄位的相關資訊，請參閱這些資料摘要。
aws_account_id	賣家 AWS 帳戶的帳 AWS 戶號碼，在 AWS 分區中是唯一的。
加密帳號	您應用程式個別購買者的唯一加密 ID。的值encrypted_account_id 是由 AWS Marketplace 計量服務使用，例如，作為CustomerIdentifier ResolveCustomer 動作所傳回的值。
郵件地址 ID	此帳戶的郵寄地址參照。
地址識別碼	此帳戶的稅捐地址參考。
稅務登記號	對於非美國帳戶，則為此帳戶的稅務登記號碼。
稅務法律名稱	對於非美國帳戶，則為合法公司名稱。這是稅捐商業發票上使用的名稱。

帳戶資料饋送範例

以下顯示帳戶資料饋送的範例。為了便於閱讀，不會顯示數據歷史記錄列。如需資料歷程記錄欄位的資訊，請參閱資料的歷史化。

account_id	aws_account_id	加密帳號	郵件地址 ID	地址識別碼	稅務登記號	稅務法律名稱
XK0 CSmiAm6PQ 4QqEog9ii	444456660 000	ZF7 oMzhe GWpH	25O3K46EN 6 斐濟克斯 8E3 eViOf			

account_id	aws_account_id	加密帳號	郵件地址 ID	地址識別碼	稅務登記號	稅務法律名稱
aochIzuPlkMfba7a1oDlZ			kaOiPalUiofjyFa			
7 年 5 聚合物 81Vx9ji04O1FF888W8jwTRoeEwTurbiQi	555567679999	373 Q8VvuQUqm	5OJ6RF2GVH2VJ9HFQIM800M vTjSzuLEHmyFY5Lr42s8	5OJ6RF2GVH2VJ9HFQIM800M vTjSzuLEHmyFY5Lr42s8	SE823935083345	
VIeGa2t9j3MuxioH9wc8lsndXXCgGCGUreeXriocM5	73739998888	8 SPxAYmi8MwX	NLUc5UeiMlGFTrDWCoftDPhDUF1oaSd8xgl5QM8Db7	{{} 瑜格威 0 hBYBiWMhndGU4/fMggmuoTC2j7Pm8ZKKNNyT	DE469558025	

位址資料饋送

> **Important**
>
> AWS Marketplace 將於 2023 年 12 月終止使用賣家交付資料饋送服務交付的 PiFe ed。使用 AddressFeed _V1 資料饋送來滿足您的地址資料需求。如果您有任何問題或需要支援，請聯絡AWS Marketplace 賣家營運團隊。

此資料饋送會為您與之互動的所有帳戶提供聯絡資訊：您自己、與您合作的任何通路合作夥伴、買家、付款人和所有稅務帳戶。每次發生新交易時，都會掃描交易的客戶地址，如果該地址不在您的資料饋送中，則會在資料饋送檔案中新增一個新項目。

地址數據是不可變的。

地址資料饋送會每 24 小時重新整理一次，因此每天都有新資料可用。

下表說明資料饋送欄的名稱和說明。

資料欄名稱	描述
位址識別碼	地址的唯一密鑰。 可用於從欄位上的Billing_Event 資料饋送或從mailing_address_id 或tax_address_id 欄位上的Account資料饋送聯結。billing_address_id
電子郵件網域	此帳戶所記錄之電子郵件地址的網域。
company_name	此帳戶登記的公司名稱。
國家代碼	這個地址的 ISO 3166 英文字母 2 國家/地區代碼。
州或地區	此地址登記的州或地區。
城市	這個地址登記的城市。
郵政編碼	此地址的檔案中的郵遞區號。
地址行 _1	這個地址的第一行地址。
地址行 2	此地址登記的第二行地址。
地址列 _3	這個地址登記的地址的第三行。

位址資料饋送範例

以下顯示位址資料饋送的範例。在資料摘要中，此資訊會顯示在單一表格中。為了便於閱讀，這裡的數據顯示在兩個表中，並且不顯示數據歷史記錄列。如需資料歷程記錄欄位的資訊，請參閱資料的歷史化。

位址識別碼	電子郵件網域	company_name	國家代碼	州或地區	城市	郵政編碼
V5nHby 0WMHNDGU TC2J7 PM 8 克里尼 BiYogwy AfMggmuo	一個 .com	马特奥·杰克逊公司	DE		漢堡	67568
G68xDB 頻率 DVHZF WGW6YF5 YOS0A6 尼斯夫 Wh5 ViLjf	B.com	瑪麗大公司	美國	OH	代頓	57684
NLUC5 GFTRW 福特夫特波多夫 1OASD8xgl 5QM8db7 UeiMl	C.com	我們的賣家	美國	NY	紐約	89475

地址行 _1	地址行 2	地址列 _3
	十九樓	

協議資料饋送

此資料摘要提供有關協議的資訊，這是提案者（記錄賣方）與接受者 (AWS 買方）之間簽訂的合約，以開始使用產品。此資料摘要提供您所建立做為記錄賣方之產品優惠協議的相關資訊。

協議資料饋送每 24 小時重新整理一次。

下表列出並說明資料饋送中的項目。

資料欄	描述
valid_from	相對於其他欄位的值，主索引鍵值對 有效的第一個日期。
insert_date	將記錄插入資料饋送的日期。
update_date	記錄上次更新的日期。
delete_date	此欄一律為空白。
Status	協議的目前狀態。支援的狀態包括： • 作用中 – 部分或全部的協議條款均有效。 • 已終止 – 由於 AWS 起始的終止事件，協議在其預先同意的結束日期之前終止。最常見的是付款失敗。 • 已取消 – 接受者選擇在其結束日期之前終止協議。 • 已過期 – 協議在其預先同意的結束日期結束。 • 已取代 – 已使用替代優惠取代協議。 • 續約 – 使用自動續約等功能，將協議續約為新協議。 • 已封存 – 協議已結束；未指定其結束、終止、取消或過期的確切原因。 • 已轉返 – 由於發生錯誤，協議修訂已轉返。舊版現已啟用。僅適用於非作用中的協議修訂。

資料欄	描述
	• 已取代 – 協議修訂不再處於作用中狀態，且另一個協議修訂現在處於作用中狀態。僅適用於非作用中的協議修訂。
estimated_agreement_value	客戶在整個協議生命週期中可能支付的已知金額總計。
currency_code	客戶可能在協議生命週期內支付的總已知金額貨幣。
agreement_id	協議的唯一識別符。
提案者_帳戶_id	提議此 PurchaseAgreement 的賣方，以賣方帳戶的全域唯一識別碼 (GUID) 表示。可用於加入帳戶資料饋送。
acceptor_account_id	接受此 PurchaseAgreement 的買方，以買方帳戶的全域唯一識別碼 (GUID) 表示。可用於加入帳戶資料饋送。
offer_revision_at_acceptance	對應至本合約之優惠的易記 ID。可用於加入優惠和優惠目標資料饋送。
start_time	協議開始的日期和時間。
end_time	協議結束的日期和時間。pay-as-you-go協議的欄位為 null，沒有結束日期。
acceptance_time	接受優惠或建立協議的日期和時間。 • 可以回溯bring-your-own-license協議的日期 • 如果使用未來日期協議功能建立協議，則可能與 start_date 不同
意圖	上次修改協議時的買方意圖。
preceding_agreement_id	先前協議的協議 ID。

資料欄	描述
status_reason_code	協議狀態變更的原因。

協議資料饋送範例

vali m	agr _id	提案者 _帳戶 _id	acc acc d	offe	offe isio acc e	sta e	enc	acc et_t	意圖	pre _ag t_id	stat	stat asc	esti _ag t_va	cur cod
202 1 20 58 00. 0	agr 44c 78a 5t	88a xxx xxx xxx x	88a xxx xxx xxx x	offe krx xxx x	1	202 1 20 58 00. 0	202 1 20 58 00. 0	202 1 20 58 00. 0	新		AC		1,0	美元

帳單事件資料摘要

此資料摘要提供帳單事件的相關資訊，包括發票和付款。

例如，您可以使用此資料饋送來了解買方的發票開立時間和內容。您也可以使用範例 SQL 查詢來分析來自此資料饋送的資料。

此資料饋送包含與您是記錄賣方的帳單事件相關的資訊。對於透過管道合作夥伴所簽訂的協議，此資料饋送包含製造商與記錄賣方之間帳單事件的相關資訊。

帳單事件資料饋送會每 24 小時重新整理一次，因此新的資料每天可用。

帳單事件資料不可變。

下表說明資料饋送資料欄的名稱和描述。

資料欄名稱	描述
billing_event_id	帳單事件的識別符。此 ID 在賣方環境中是唯一的。
from_account_id	啟動帳單事件的帳戶。如果 transaction_type 是 SELLER_REV_SHARE ，則它是買方的付款人帳戶。這是帳戶資料饋送的外部金鑰。 可用於加入 account_id 欄位上的Account資料饋送。
to_account_id	接收產品交易金額的帳戶。這是帳戶資料饋送的外部金鑰。 可用於加入 account_id 欄位上的Account資料饋送。
end_user_account_id	使用產品的帳戶。此帳戶可能與買方和付款人帳戶不同。 可用於加入 account_id 欄位上的Account資料饋送。
product_id	產品的識別符。這是產品資料饋送的外部金鑰。 可用於加入 product_id 欄位上的Product資料饋送。
動作	此事件的動作類型。可能的值如下： • INVOICED – 已開立金額的發票給買方。 • FORGIVEN – 買方已開立金額的發票，並已 AWS 還原費用。 • DISBURSED – 賣方已支付此金額。這可以包含一個月的發票，也可以是隨需支出。

資料欄名稱	描述
transaction_type	交易的類型。如需範例，請參閱 稅務案例。可能的值如下： • SELLER_REV_SHARE – 正數；這是賣方與買方簽訂的協議中設定的價格。 • SELLER_TAX_SHARE – 正數；這是新增至以SELLER_REV_SHARE 涵蓋賣方積欠的稅金。 • AWS_REV_SHARE – 負金額；這是列出費用。 • AWS_TAX_SHARE – 正數；這是除了 之外 AWS 所收取的稅金。 SELLER_REV_SHARE 此金額不會影響賣方的餘額。此金額不會支付，且是為了賣方了解開立給買方的稅金，並代表賣方匯給當局。 • *transaction_type* _REFUND – 買方請求的退款金額。 • *transaction_type* _CREDIT – 買方的 AWS 點數。 • BALANCE_ADJUSTMENT – AWS 為解決發票問題而進行的調整。 • DISBURSEMENT – 如果 的值action是 DISBURSED ，而 的值balance_impacting 是 1，則這是支付給賣方的金額。如果 的值action為 INVOICED，則此記錄會否定parent_billing_event_id 記錄的全部或部分。在此情況下，disbursement_billing_event_id 會顯示相關的支出，且 的值balance_impacting 為 0。 • DISBURSEMENT_FAILURE – 否定交易。

資料欄名稱	描述
parent_billing_event_id	當 的值broker_id 為 時AWS_INC，的值action為 DISBURSED 或 FORGIVEN，而 的值transaction_type 為 DISBURSEMENT 時，parent_billing_event_id 會參考啟動此帳單事件billing_event_id 的原始 。如果 action有另一個值，則此欄位為 null。 當 的值broker_id 為 時AWS_EUROPE ，會parent_billing_event_id 參照下列案例billing_event_id 啟動此帳單事件的原始 ： • 的值action為 FORGIVEN或 INVOICED，而 的值 transaction_type 為 AWS_REV_SHARE AWS_REV_SHARE_REFUND 、 或 SELLER_REV_SHARE_REFUND 。 • 的值action為 DISBURSED ，而 的值transaction_type 為 ANY (不含 DISBURSEMENT_FAILURE)。 • 的值transaction_type 為 AWS_TAX_SHARE 、SELLER_TAX_SHARE 、AWS_TAX_SHARE_REFUND 或 SELLER_TAX_SHARE_REFUND 。 • 的值action為 DISBURSED ，而 的值transaction_type 為 DISBURSEMENT_FAILURE 。 當 的值broker_id 為 時AWS_EUROPE ，會parent_billing_event_id 參考下列情況下先前成功支付帳單事件billing_event_id 的原始 ：

資料欄名稱	描述
	• 的值action為 DISBURSED ，而 的值transaction_type 為 DISBURSEMENT_FAILURE 。 當 的值broker_id 為 時AWS_EUROPE ，此欄位在所有剩餘的案例為 null。
disbursement_billing_event_id	當 的值action為 DISBURSED 且下列其中一項為 true 時，相關的支出： • transaction_type like ('SELLER%') • transaction_type like ('AWS%') 在所有其他情況下，此值為 null。
amount	帳單事件金額。
currency	ISO 639 貨幣代碼。
balance_impacting	在計算賣方支出時，是否會考慮金額。值 0 表示金額僅供參考，對餘額沒有影響。值 1 表示此金額在決定賣方支出時已納入考量。
invoice_date	發票建立的日期。
payment_due_date	當 的值action為 時INVOICED，發票的到期日。
usage_period_start_date	記錄中期間的開始日期。
usage_period_end_date	記錄中期間的結束日期。
invoice_id	AWS 發票 ID。

資料欄名稱	描述
billing_address_id	地址資料摘要中的付款人帳單地址參考。 可用於加入 address_id 欄位上的Address資料饋送。
transaction_reference_id	可讓您從下列報告交叉參考資料的識別符： • 支出報告 • 每月計費營收報告 • 銷售薪酬報告
bank_trace_id	對於支出交易 (transaction_type =is DISBURSEMENT 且動作為 DISBURSED)，由銀行指派的追蹤 ID。追蹤 ID 可用來與賣方銀行提供的報告建立關聯。
broker_id	促進交易的業務實體識別符。可能的值如下： • AWS_INC – 的識別符 AWS， Inc.（位於美國）。 • AWS_EUROPE – Amazon Web Services EMEA SARL 的識別符（以盧森堡為基礎）。 • AWS_JAPAN • AWS_AUSTRALIA • AWS_KOREA • NULL – 之前沒有明確 的交易broker_id 是由 協助AWS_INC。

資料欄名稱	描述
buyer_transaction_reference_id	使用 GROUP BY或 SQL 中的視窗函數建構，將帳單摘要中所有相關記錄分組在一起的識別符。這些相關記錄可以包含買方發票、賣方發票，以及列示費用的增值稅 VATs)。

稅務案例

買方和賣方所在國家和州的稅務模型會決定如何收集和匯出稅金。以下是可能的情況：

- 稅金由 收集和匯出 AWS。在這些情況下， `transaction_type`為 `AWS_TAX_SHARE`。
- 稅務由賣方收取 AWS、支付給賣方，並由賣方向稅務機關提交。在這些情況下，`transaction_type`為 `SELLER_TAX_SHARE`。
- 稅金不會由 收取 AWS。賣方必須計算稅金，並將其匯給稅務機關。在這些情況下， AWS Marketplace 不會執行稅務計算或接收稅務資訊。賣方支付收入份額的稅金。

帳單事件資料饋送的範例

本節顯示開立發票時和一個月後帳單事件資料期間的範例。請注意本節中所有資料表的下列各項：

- 在資料饋送中，`billing_event_id`值為 40 個字元的英數字串。此處顯示為兩個字元字串，以方便閱讀。
- 在資料饋送中，此資訊會顯示在單一資料表中。為了便於閱讀，資料會顯示在此處的多個資料表中，而且不會顯示所有資料欄。

針對本節中的範例，假設下列事項：

- Arnav 是買方。
  - 他的帳戶 ID 為 737399998888。
  - 他位於法國，受到市場協調人員法律的約束。如需詳細資訊，請參閱 Amazon Web Service Tax Help。
  - 他購買`prod-o4grxfafcxxxx`該產品，並按每月用量開立 120.60 美元的發票。
  - 他在該月內支付了發票。
- Jane 是製造商。

  - 她的帳戶 ID 是 111122223333。
- Paulo 是記錄的賣方。
  - 他的帳戶 ID 是 777788889999。
  - 他居住在堪薩斯州，不受市場促進者法律的約束。

記錄賣方的帳單事件資料摘要

身為記錄賣方，Paulo 會向買方 Arnav 開立發票。

下表顯示 Paulo 在開立 Arnav 發票時，資料摘要中的相關資訊。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	動作	transaction_type
I0	737399998888	777788889999	737399998888	prod-o4grxfafcxxxx	已開立發票	SELLER_REV_SHARE
I1	737399998888	AWS	737399998888	prod-o4grxfafcxxxx	已開立發票	AWS_TAX_SHARE
I2	777788889999	111122223333	737399998888	prod-o4grxfafcxxxx	已開立發票	SELLER_REV_SHARE
I3	777788889999	AWS	737399998888	prod-o4grxfafcxxxx	已開立發票	AWS_REV_SHARE

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		100	美元	2018-12-31T00：00：00Z	781216640

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		20.6	美元	2018-12-31T00：00：00Z	781216640
		-80	美元	2018-12-31T00：04：07Z	788576665
		-0.2	美元	2018-12-31T00：04：07Z	788576665

下表顯示在 Arnav 支付發票後，月底 Paulo 資料饋送中的相關資訊。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	動作	transaction_type
I10	737399998888	777788889999	737399998888		分發	SELLER_REV_SHARE
I12	777788889999	111122223333	737399998888		分發	SELLER_REV_SHARE
I13	777788889999	AWS	737399998888	prod-o4grxfafcxxxx	分發	AWS_REV_SHARE
I14	AWS	777788889999			分發	支出

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
I0	I14	-100	美元	2018-12-31T00：00：00Z	781216640
I2	I14	80	美元	2018-12-31T00：04：07Z	788576665
I3	I14	0.2	美元	2018-12-31T00：04：07Z	788576665
		19.8	美元		

製造商的帳單事件資料摘要

下表顯示 Paulo 發票 Arnav 時 Jane 資料饋送中的相關資訊。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	動作	transaction_type
I5	777788889999	111122223333		prod-o4grxfafcxxxx	已開立發票	SELLER_REV_SHARE
I6	777788889999	111122223333		prod-o4grxfafcxxxx	已開立發票	SELLER_TAX_SHARE
I7	111122223333	AWS		prod-o4grxfafcxxxx	已開立發票	AWS_REV_SHARE

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		73.5		2018-12-31T00：04：07Z	788576665
		6.5		2018-12-31T00：04：07Z	788576665
		-7.35		2018-12-31T00：04：07Z	788576665

下表顯示支付發票後，當月月底 Jane 資料饋送中的相關資訊。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	動作	transaction_type
I30	777788889999	111122223333		prod-o4grxfafcxxxx	分發	SELLER_REV_SHARE
I31	777788889999	111122223333		prod-o4grxfafcxxxx	分發	SELLER_TAX_SHARE
I32	111122223333	AWS		prod-o4grxfafcxxxx	分發	AWS_REV_SHARE
I33	AWS	111122223333			分發	支出

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
I5	I33	-73.5	美元		
I6	I33	-6.5	美元		
I7	I33	7.35	美元		
		72.65	美元		

查詢範例

如 中所述使用資料饋送收集和分析資料，您可以使用 Athena 對在受管 Amazon S3 儲存貯體中收集並儲存為資料饋送的資料執行查詢。本節提供一些您可能這麼做的常見方式範例。所有範例都假設使用單一貨幣。

範例 1：發票金額，包括稅金

若要了解已開立多少買方發票，包括稅金，您可以執行查詢，如下列範例所示。

```
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'
  );
```

範例 2：代表賣方向買方開立的發票金額

若要了解代表賣方開立多少買方發票，您可以執行查詢，如下列範例所示。

```
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'
;
```

範例 3： AWS 可以代表賣方收取的金額

若要了解 AWS 可代賣方收取多少金額，減去任何退款、抵用金和寬容帳戶，您可以執行查詢，如下列範例所示。

```
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')
;
```

範例 4：賣方可以收集的金額

若要了解賣方可以收集多少，您可以執行查詢，如下列範例所示。此範例會移除 AWS 收集的列出費用和稅金，並新增任何異常餘額調整。

```
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')
;
```

您也可以使用下列查詢來收集相同的資訊，如下列範例所示。

```
SELECT sum(amount) FROM billing_event
WHERE
  balance_impacting = 1
  and action in ('INVOICED','FORGIVEN')
;
```

下列範例顯示相同的資訊，但僅限於 2018 年交易，並假設所有買方都已支付其發票。

```
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')
;
```

範例 5：支出金額

若要了解已支付的金額，您可以執行查詢，如下列範例所示。

```
select sum(amount) FROM billing_event
WHERE
  action ='DISBURSED'
  and transaction_type like 'DISBURSEMENT%'
;
```

範例 6：待支付的金額

若要了解待支付的金額，您可以執行查詢，如下列範例所示。此查詢會移除已支付的金額。

```
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
      )
    )
;
```

取得相同資訊的另一個方法是執行查詢以取得賣方的餘額，如下列範例所示。

```
SELECT sum(amount) FROM billing_event
WHERE
 balance_impacting = 1
;
```

下列查詢延伸了我們的範例。它會將結果限制為 2018 年交易，並傳回交易的其他詳細資訊。

```
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
);
```

範例 7：發票集的餘額

若要了解一組發票的總和，您可以執行查詢，如下列範例所示。

```
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;
```

舊版對映資料饋送

此資料饋送列出產品 ID 和優惠 ID 如何對應至舊版全域唯一識別碼 (GUID)。舊版 GUID 用於舊版報表中，而新的 ID 則用於資料摘要和 AWS Marketplace API 中。

此資料饋送會提供您作為記錄賣方建立的所有產品，以及您獲授權轉售的所有產品的相關資訊。

舊版對應資料饋送會每 24 小時重新整理一次，因此每天都有新資料可用。

下表說明資料饋送欄的名稱和說明。

資料欄名稱	描述
對映類型	這是產品 ID 還是優惠 ID。
法律識別碼	此產品或優惠的舊版 ID。
新的識別碼	此產品或優惠的易記 ID。此 ID 會用作主索引鍵和所有目前的 API 動作。

舊版對映資料饋送範例

以下顯示舊版對映資料饋送的範例。為了便於閱讀，不會顯示數據歷史記錄列。如需資料歷程記錄欄位的資訊，請參閱資料的歷史化。

對映類型	法律識別碼	新的識別碼
提供	8a806c74-DBD6-403e-9362	優惠-達克斯恩弗溫
產品	1368541d-890b-四 b6c-9bb9-4a55306ab642	項目-o4 格拉法克 XXXY
提供	88d8382-6b3a-4c75-8345-a627b552f5F1	提供-吉斯姆勒 5 倍壓縮

提供資料摘要

優惠資料摘要提供您已建立做為記錄賣方之所有優惠的相關資訊。如果單一優惠有多個修訂，則所有修訂都會包含在資料饋送中。

當您進行優惠修訂，且公開欄位中的資料變更時，會在相同主索引鍵的資料饋送中建立新的記錄 (offer_id加上 offer_revision)。不過， valid_from 欄位具有不同的值。如需資料饋送歷史記錄資料欄的詳細資訊，請參閱 資料的歷史化。

優惠資料饋送每 24 小時重新整理一次，因此每天都有新的資料可用。

下表提供資料摘要資料欄的名稱和描述。

資料欄名稱	描述
offer_id	優惠的易記識別符。 可用來加入Offer_Product 資料饋送offer_id的欄位。
offer_revision	優惠修訂。此欄位和 offer_id 欄位結合以形成主索引鍵。

資料欄名稱	描述
	透過 offer_id， 可用來加入Target_Offer 資料摘要的 offer_id和 offer_revision 欄位。
name	優惠的賣方定義名稱。
expiration_date	優惠到期的日期和時間。
opportunity_name	連結至本優惠的任何機會資料。如果優惠受限於 AWS 機會，則此欄位會填入。
opportunity_description	連結至本優惠的任何描述性資訊。如果優惠受限於 AWS 機會，則此欄位會填入。
seller_account_id	賣方帳戶的全域唯一識別碼 (GUID)。可用於與帳戶資料饋送中的 account_id 欄位聯結。
opportunity_id	只有在經銷商銷售您的產品時，才會填入機會的識別符。opportunity_id 如果產品相同，則由不同管道合作夥伴（或賣方）建立的所有優惠都相同。

優惠資料饋送範例

以下顯示優惠資料饋送的範例。為了方便讀取，不會顯示資料歷史記錄資料欄。如需資料歷史記錄欄位的資訊，請參閱 資料的歷史化。

offer_id	offer_revision	name	expiration_date	opportunity_name	opportunity_description	seller_account_id	opportunity_id
offer-dacpxznflfwin	1	企業合約計劃優惠	9999-01-01T00：00：00Z				

offer_id	offer_revision	name	expiration_date	opportunity_name	opportunity_description	seller_account_id	opportunity_id
offer-gszhmle5npzip	1	賣方建立的私有優惠	2020-10-31T00：00：00Z				
offer-hmzhyle8nphlp	1	企業合約計劃優惠	9999-01-01T00：00：00Z				

提供產品資料饋送

一個優惠可以有幾個產品，一個產品可以包含在不同的優惠中。此資料饋送會列出選件與產品之間關係的相關資訊。

此資料饋送會提供您作為記錄賣方建立之所有產品優惠的相關資訊。

當您在報價中新增或移除產品時，您可以建立選件版本。

選件產品資料饋送會每 24 小時重新整理一次，因此每天都有新資料可用。

下表說明資料饋送欄的名稱和說明。如需有關資料饋送歷史記錄欄的資訊，請參閱資料的歷史化。

資料欄名稱	描述		
報價識別碼	此優惠的易記識別碼。 可用於聯結至資Offer料饋送的offer_id欄位。		
優惠修訂	與offer_id欄位結合以形成選件修訂的外部索引鍵。		

資料欄名稱	描述		
product_id	產品的易記識別碼，這是此優惠公開之產品的外部索引鍵。 可用於聯結至資Product料饋送的product_id 欄位。		

提供產品資料饋送範例

以下顯示「提供產品資料饋送」的範例。

報價識別碼	優惠修訂	product_id
優惠-達克斯恩弗溫	10	产品-o4 格拉法夫 XXXX
提供-吉斯姆勒 5 倍壓縮	24	項目-o4 格拉法克 XXXY

提供目標資料饋送

此資料饋送會針對您建立為記錄賣方的所有選件，列出選件修訂的目標。如果單一選件具有多個修訂，則所有修訂都會包含在資料饋送中。

當您進行選件修訂且公開欄位中的資料發生變更時，會在相同主索引鍵 (offer_id加號offer_revision) 的資料饋送中建立新記錄，但valid_from欄位的值不同。

選件目標資料饋送會每 24 小時重新整理一次，因此每天都有新資料可用。

下表說明資料饋送欄的名稱和說明。

資料欄名稱	描述
提供目標 ID	摘要的主索引鍵。

資料欄名稱	描述
優惠 ID + 優惠修訂	選件的識別碼和修訂。這兩個資料欄會參考此目標相關的選件。 可用來聯結至資Target料饋送的offer_id和offer_revision 欄位。
目標類型	指出選件收件者是否為BuyerAccounts ，表示非公開選件或ParticipatingPrograms 。
極性	指出選件是否要對target_type 。可接受的值如下： • PositiveTargeting — 此優惠適用於此target_type . • NegativeTargeting — 此優惠不適用於此target_type 。
value	代表目標的字串： AWS 帳戶 ID 或可與選件搭配使用的程式。例如，AWS Marketplace（SCMP）的標準合約，或AWS Marketplace 現場演示程序（FDP）。

選件目標資料饋送範例

以下顯示選件目標資料饋送的範例。為了便於閱讀，不會顯示數據歷史記錄列。如需資料歷程記錄欄位的資訊，請參閱資料的歷史化。

提供目標 ID	報價識別碼	優惠修訂	目標類型	極性	value
925ddc73f6a373b7d5544E3210610803b600	優惠-達克斯恩弗溫	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract

提供目標 ID	報價識別碼	優惠修訂	目標類型	極性	value
471ff22ae3165278f1fb960d3e14517bcd601	提供-吉斯姆勒 5 倍壓縮	1	ParticipatingPrograms	PositiveTargeting	FieldDemonstration
511ff22adfj65278f1fb960d3e14517bcd6e602	提供-吉斯姆勒 5 倍壓縮	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract

產品資料饋送

此資料饋送會提供您作為記錄賣方建立的所有產品，以及您獲授權轉售的所有產品的相關資訊。

產品數據是可變的。這表示當您變更下列其中一個欄位的值時，會在資料饋送中建立具有不同`valid_from`欄位值的新記錄。如需有關資料饋送歷史記錄欄的詳細資訊，請參閱資料的歷史化。

產品資料摘要每 24 小時會重新整理一次，因此每天都有新資料可用。

下表說明資料饋送欄的名稱和說明。

資料欄名稱	描述
product_id	產品的易記識別碼。 可用於聯結至`Account`、`Billing_Event` 和`Offer_Product` 資料饋送的`product_id` 欄位。
制造商帳戶 ID	產品擁有者的識別碼。這是「帳戶」資料饋送的外部索引鍵。 可用於聯結至資`Account`料饋送的`account_id` 欄位。

資料欄名稱	描述
產品代碼	用於測量產品的現有權益產品代碼。此值也可用於將資料與報表聯結，或參考 AWS Marketplace 計量服務中提供的內容。
標題	產品的標題。

產品資料饋送範例

以下顯示選件目標資料饋送的範例。為了便於閱讀，不會顯示數據歷史記錄列。如需資料歷程記錄欄位的資訊，請參閱資料的歷史化。

product_id	制造商帳戶 ID	產品代碼	標題
产品-o4 格拉法夫 XXXX	555568000000	產品代碼 _1	產品一
項目-t3 格拉法夫克 XXXY	444457000000	產品代碼 _2	產品二
項目-x8 法夫克西	666678000000	產品代碼 _3	產品 3

稅務項目資料饋送

此資料摘要提供客戶發票的稅務計算資訊。

指定客戶發票 (line_item_id) 的特定產品 () 可能會有多個明細項目 (product_id)invoice_id，每個稅務司法管轄區可能會有一或多個明細項目。例如，針對使用不同 AWS 實體 （ 例如，美國和愛爾蘭 ） 不同 AWS 區域規則的客戶，這會發生這種情況。若要進一步了解銷售的 AWS 收集銷售稅、增值稅或 GST 的位置，並將此類稅金以 AWS， Inc. 的名稱匯給當地稅務機關，請參閱 Amazon Web Service Tax Help。

稅務項目資料饋送會每 24 小時重新整理一次，因此新資料每天可用。

稅務項目資料不可變。

下表說明資料饋送資料欄的名稱和描述。如需資料歷史記錄資料欄的資訊，請參閱 資料的歷史化。

資料欄名稱	描述
tax_item_id	稅務項目記錄的唯一識別符。
invoice_id	AWS 發票 ID。您可以使用此值搭配 的值`product_id` 來尋找相關的稅務帳單事件。
line_item_id	客戶帳單明細項目的唯一識別符。退款交易具有與其轉送稅務交易相同的明細項目 ID。
customer_bill_id	客戶帳單的唯一識別符。買方可以與賣方共用此識別符，以協助識別和解決稅務計算問題。
tax_liable_party	`AWS` 或 `Seller`。如果賣方是納稅方，則會收取稅金。如果 AWS 是負稅方，則會收取銷售稅並由 匯出 AWS。如需詳細資訊，請參閱AWS Marketplace 賣方與稅務收集。 如果未收取任何稅金，此處不會顯示任何值。賣方需要判斷是否已針對每個發票收取一些稅金，因為賣方需負責收取稅金。
transaction_type_code	交易的類型。可能的值如下： • `AWS` – 轉送稅務交易 • `REFUND` – 全部或部分退款 • `TAXONLYREFUND` – 僅退稅 退款交易會與其原始轉送交易共用明細項目 ID。
product_id	產品的外部金鑰。 可用於加入 `product_id` 欄位上的`Product`資料饋送。
product_tax_code	用來識別產品稅務屬性的標準程式碼。賣方在建立或修改產品時選擇屬性。

資料欄名稱	描述
invoice_date	發票建立的日期。
taxed_customer_account_id	對已課稅之帳戶實體的外部金鑰。 可用於加入 `account_id` 欄位上的`Account`資料饋送。
taxed_customer_country	用於稅務計算之地址的 ISO 3166 alpha 2 國家代碼。
taxed_customer_state_or_region	用於稅務計算的州、區域或省。
taxed_customer_city	用於稅務計算的城市。
taxed_customer_postal_code	用於計算稅金的郵遞區號。
tax_location_code_taxed_jurisdiction	與已課稅位置相關聯的頂點地理碼。
tax_type_code	套用至交易的稅金類型。可能的值為 `None`、`Sales` 和 `SellerUse` 。
jurisdiction_level	用於稅務位置的地址的司法管轄區層級。可能的值為 `State`、`County`、`City` 和 `District`。
taxed_jurisdiction	稅務司法管轄區的名稱。
display_price_taxability_type	買方看到的價格是否包含或不含稅金。所有 AWS Marketplace 優惠皆不含稅金。
taxable_amount	在這個司法管轄區層級，應稅的交易金額。
nontaxable_amount	在此司法管轄區層級，不可課稅的交易金額。
tax_jurisdiction_rate	在此司法管轄區層級套用的稅率。
tax_amount	在此司法管轄區層級收取的稅金。
tax_currency	上述金額的 ISO 4217 alpha 3 貨幣代碼。

資料欄名稱	描述
tax_calculation_reason_code	交易是應稅交易、不可課稅交易、豁免交易，還是零評分交易，由司法管轄區層級組織。
date_used_for_tax_calculation	用於計算交易稅金的日期。
customer_exemption_certificate_id	豁免憑證的憑證 ID。
customer_exemption_certificate_id_domain	憑證存放在 Amazon 系統上的位置。
customer_exemption_certificate_level	提供豁免的司法管轄區層級。
customer_exemption_code	指定豁免的程式碼；例如 RESALE。
customer_exemption_domain	如果可用，則用於擷取客戶豁免資訊的 Amazon 系統。
transaction_reference_id	可讓您從下列報告交叉參考資料的識別符： • 支出報告 • 每月計費營收報告 • 銷售薪酬報告

> **Note**
>
> 自 2021 年 8 月 5 日起， AWS Marketplace 國際市場主持人銷售稅將在稅務項目資料饋送中包含項目。這表示自 2021 年 8 月 5 日起，帳單事件資料饋送中的每個 AWS_TAX_SHARE 和 SELLER_TAX_SHARE 記錄預期在稅務項目資料饋送中都有對應的記錄。

稅務項目資料饋送範例

以下顯示稅務項目資料饋送的範例。在資料饋送中，此資訊會以單一資料表顯示。為了便於讀取，資料會顯示在此處的多個資料表中，而且不會顯示所有資料欄。

tax_item_id	invoice_id	line_item_id	customer_bill_id
6p2ni6tu041xagvhbyanbgxl3xameha16txjoav_0001	781216640	710000000000000000000	221000000000000000
6p2ni6tu041xagvhbyanbgxl3xameha16txjoav_0002	781216640	530000000000000000000	221000000000000000
flr4jobxjzww8czdsrq4noue2uxd56j39wxw0k7_0001	250816266	764000000000000000000	572000000000000000
gfkjjobxjzw56jgkrsrqgjtk52uxd56j39wgj567d_0002	280336288	764000000000000000000	572439000000000000
wwk1qpvb8ran3geiw8e3mp6dgs2qj7wpkuwhgk1_0001	451431024	993000000000000000000	123000000000000000
wwk1qpvb8ran3geiw8e3mp6dgs2qj7wpkuwhgk1_0002	451431024	993000000000000000000	312000000000000000
fnohdid8kwgqq9lvii2k30spn3ftgwihbe8h75x_0001	229987654	921000000000000000000	639000000000000000

tax_liable_party	transaction_type_code	product_id	product_tax_code	invoice_date
賣方	AWS	prod-o4grxfafcxxxx	AWSMP_SOFTWARE_RA	2018-12-31T00：00：00Z

tax_liable_party	transaction_type_code	product_id	product_tax_code	invoice_date
賣方	AWS	prod-o4grxfafcxxxx	AWSMP_SOFTWARE_RA	2018-12-31T00：00：00Z
賣方	AWS	prod-t3grxfafcxxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00：00：00Z
賣方	退款	prod-t3grxfafcxxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00：00：00Z
賣方	AWS	prod-x8faxxfafcxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00：00：00Z
賣方	TAXONLYREFUND	prod-x8faxxfafcxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-05-31T00：00：00Z
AWS	AWS	prod-wghj8xfafrhgj	AWS_REMOTE_ACCESS_SOFTWARE	2019-07-31T00：00：00Z

taxed_customer_account_id	taxed_customer_country	taxed_customer_state_or_region	taxed_customer_city	taxed_customer_postal_code
VIeGa2t9j3MuxioH9wc8lsndXXCgGCGUreeXriocM5	美國	GA	密爾頓	48573-4839

taxed_customer_account_id	taxed_customer_country	taxed_customer_state_or_region	taxed_customer_city	taxed_customer_postal_code
VIeGa2t9j3MuxioH9wc8lsndXXCgGCGUreeXriocM5	美國	GA	密爾頓	48573-4839
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	美國	NC	杜拉姆	27517-4834
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	美國	NC	杜拉姆	27517-4834
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	美國	TX	不適用	75844-1235
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	美國	TX	休士頓	75844-1235
192a0421313e41f069b52962ed7babf716291b688	美國	CT	新 HAVEN	06002-2948

tax_location_code_taxed_jurisdiction	tax_type_code	jurisdiction_level	taxed_jurisdiction	display_price_taxability_type	taxable_amount	nontaxable_amount
460473664	銷售	State	GA	獨家	100	0
66301164	銷售	County	富爾頓	獨家	0	100
692938178	SellerUse	State	NC	獨家	58.1	523.8
692938178	SellerUse	State	NC	獨家	-58.1	523.8
356794387	銷售	State	TX	獨家	1105.14	0
528887443	銷售	City	休士頓	獨家	-36	0
171248162	銷售	State	CT	獨家	0	114.55

tax_jurisdication_rate	tax_amount	tax_currency	tax_calculation_reason_code	date_used_for_tax_calculation
0.206	20.6	美元	應稅	2018-10-31T00：00：00Z
0	0	美元	NonTaxable	2018-10-31T00：00：00Z
0.1	5.8	美元	應稅	2018-07-31T00：00：00Z
0.1	-5.8	美元	應稅	2018-07-31T00：00：00Z
0.06	66.3	美元	應稅	2018-07-31T00：00：00Z

tax_jurisdication_rate	tax_amount	tax_currency	tax_calculation_reason_code	date_used_for_tax_calculation
0.01	-0.36	美元	NonTaxable	2018-07-31T00：00：00Z
0	0	美元	豁免	2019-06-30T00：00：00Z

中的賣方報告 AWS Marketplace

Important

在 2024 年 8 月 30 日， AWS Marketplace 將停止數個報告和資料集：

- Marketplace 舊版逗號分隔 (csv) 賣方報告
- Commerce Analytics Service (CAS) API
- 關聯的電子郵件通知

檢查報告頁面是否有受影響的報告，或者如果您有任何問題，請洽詢在 5/30/2024 發出通訊的 AWS 管理員。

AWS Marketplace 提供報告，其中包含有關產品用量、買方、帳單和付款資訊的資訊。報告可供所有已註冊的 AWS Marketplace 賣方使用。

以下是報告產生的一些重點：

- 報告是每天、每週或每月產生，取決於報告。
- 報告會在 UTC 的 00：00 產生，並涵蓋前一天的 24：00 UTC。
- 報告會以 .csv 檔案產生。
- 您可以設定 Amazon SNS 在資料交付到加密的 Amazon S3 儲存貯體時通知您。設定通知後， 會將通知 AWS 傳送至與您註冊 AWS 帳戶相關聯的電子郵件地址 AWS Marketplace。

如需如何設定通知的資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 Amazon SNS 入門。

若要取消接收通知電子郵件，請聯絡AWS Marketplace 賣方營運團隊。

- 若要了解每個報告，您可以下載範例報告。

存取報告

AWS Marketplace 提供兩種設定報告的方式：

- 使用 API 界面。使用 AWS Marketplace Commerce Analytics Service 存取產品和客戶資料 可讓您透過 API 界面自動存取報告中的資料。您可以自動擷取您的資訊，並下載報告的一部分，而不是整個報告。服務會以非同步方式將資料傳回至 Amazon Simple Storage Service 中的檔案，而不是像傳統 API 一樣直接傳回資料。資料會以機器可讀取的格式交付，讓您可以將資料匯入或整合到您的系統中。
- 使用 中的報告儀表板AWS Marketplace 管理入口網站。此儀表板提供先前報告期間的報告。

您可以使用 AWS Identity and Access Management (IAM) 許可控制對報告的存取。

可用的 AWS Marketplace 賣方報告

下列報告可在 中使用 AWS Marketplace：

- 每日業務報告
- 每日客戶訂閱者報告
- 支出報告
- 每月計費營收報告
- 銷售薪酬報告

每日業務報告

每日業務報告提供下列方面的資料：

- AWS 客戶每天如何使用您的產品
- 該用量的預估收入。

只有在相關資訊可用時，您才會收到此報告。

如果您未收到此報告且認為應該收到，請聯絡AWS Marketplace 賣方營運團隊。當您想要取消報告的電子郵件通知時，您也可以聯絡該團隊。

您可以在 存取此報告AWS Marketplace 管理入口網站。如果您已註冊 the section called “AWS Marketplace Commerce Analytics 服務”，您也可以使用 AWS SDK 存取您的報告。

您可以為每位客戶使用唯一的識別符，以識別客戶隨時間和跨報告。識別符可讓您追蹤客戶用量模式，以便估算客戶支出、深入了解免費試用用量，以及年度用量趨勢。

發佈排程

此報告每天在 UTC 00：00 發佈，涵蓋前一天 UTC 00：00 到 23：59。排程的任何例外狀況都會在每日業務報告一節的簡介中記下。

主題

- 第 1 節：執行個體類型的用量
- 第 2 節：費用
- 第 3 節：免費試用轉換
- 第 4 節：新執行個體
- 第 5 節：新產品訂閱者
- 第 6 節：已取消的產品訂閱者

第 1 節：執行個體類型的用量

本節列出客戶使用的每個執行個體類型的資料列。例如，當客戶在某個執行個體類型上使用產品，在不同的執行個體類型上使用相同的產品時，報告會包含兩個執行個體類型中的每一個。

資料欄名稱	描述
客戶參考 ID	不是帳戶 ID 的唯一識別符。它有助於追蹤客戶的用量、收入和訂閱。 客戶參考 ID 排定於 2026 年 3 月 31 日棄用。請改用客戶 AWS 帳戶 ID。
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。

資料欄名稱	描述
使用者的狀態	與訂閱產品的帳戶相關聯的帳單地址狀態。
使用者的國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
產品標題	產品的標題。
產品代碼	產品的唯一識別符。
執行個體類型	與產品用量相關聯的執行個體類型：例如 t2.micro。
用量單位	客戶在報告期間使用的用量單位數。
用量單位類型	計量客戶用量的測量單位。例如，小時或天。
方案描述	產品方案的描述。例如，產品提供每小時用量、免費試用用量或年度用量。
預估營收	產品用量的預估收入。帳單會在月底完成。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則項目為 USD。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
客戶國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

第 2 節：費用

本節包含與 產品相關聯的費用型交易：例如，每年、每月、SaaS 合約產品費用和資料產品訂閱費用。本節中的資料涵蓋產生報告前 72 小時的 24 小時期間。例如，如果報告是在 5 月 24 日產生，則資料涵蓋 5 月 21 日的 24 小時期間。

資料欄名稱	描述
客戶參考 ID	不是帳戶 ID 的唯一識別符。它有助於追蹤客戶的用量、收入和訂閱。 客戶參考 ID 排定於 2026 年 3 月 31 日棄用。請改用客戶 AWS 帳戶 ID。
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
使用者的狀態	與訂閱產品的帳戶相關聯的帳單地址狀態。
使用者的國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
產品標題	產品的標題。
產品代碼	產品的唯一識別符。
Amount (數量)	使用費。如果有退款，則此值為負數。如果此項目適用於 an AWS Marketplace SaaS 合約，則金額代表維度的費用，而不是整個合約的費用。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則項目為 USD。
費用描述	費用原因：例如，月費、年費或退款。
客戶國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
客戶狀態	與計費帳戶相關聯的帳單地址狀態。
客戶城市	與計費帳戶相關聯的帳單地址城市。

資料欄名稱	描述
客戶郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
客戶電子郵件網域	與收費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 `liu-jie@example.com` ，則項目為 `example.com` 。
Start Date (開始日期)	an AWS Marketplace SaaS 合約或資料產品訂閱的開始日期。
結束日期	an AWS Marketplace SaaS 合約或資料產品訂閱的結束日期。
數量	合約指定之維度的單位數。
維度	合約指定的維度。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

第 3 節：免費試用轉換

本節列出免費試用開始、轉換和取消的資料，並涵蓋先前的 24 小時期間。

資料欄名稱	描述
產品標題	產品的標題。
產品代碼	代表產品的唯一識別符。
新的免費試用	在報告期間啟動的新免費試用數量。

資料欄名稱	描述
目前免費試用總數	作用中免費試用訂閱的總數。
轉換後的免費試用	在報告期間，從免費試用移至付費用量的訂閱總數。
非轉換的免費試用	結束免費試用且未轉換為付費用量的訂閱總數。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。

第 4 節：新執行個體

本節列出新 EC2 執行個體和執行個體類型的資料，並涵蓋先前的 24 小時期間。

資料欄名稱	描述
客戶參考 ID	不是帳戶 ID 的唯一識別符。它有助於追蹤客戶的用量、收入和訂閱。 客戶參考 ID 排定於 2026 年 3 月 31 日棄用。請改用客戶 AWS 帳戶 ID。
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
使用者的狀態	與訂閱產品的帳戶相關聯的帳單地址狀態。
使用者的國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
產品標題	產品的標題。
產品代碼	產品的唯一識別符。
Type	Amazon EC2 執行個體類型。
計數	EC2 執行個體的數量。

資料欄名稱	描述
客戶國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
客戶狀態	與計費帳戶相關聯的帳單地址狀態。
客戶城市	與計費帳戶相關聯的帳單地址城市。
客戶郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
客戶電子郵件網域	與收費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 `liu-jie@example.com` ，則項目為 `example.com` 。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

第 5 節：新產品訂閱者

本節列出新買方的資料，並涵蓋先前的 24 小時期間。

資料欄名稱	描述
客戶參考 ID	不是帳戶 ID 的唯一識別符。它有助於追蹤客戶的用量、收入和訂閱。 客戶參考 ID 排定於 2026 年 3 月 31 日棄用。請改用客戶 AWS 帳戶 ID。
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。

資料欄名稱	描述
使用者的狀態	與訂閱產品的帳戶相關聯的帳單地址狀態。
使用者的國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
產品標題	產品的標題。
產品代碼	產品的唯一識別符。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
客戶國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
客戶狀態	與計費帳戶相關聯的帳單地址狀態。
客戶城市	與收費帳戶相關聯的帳單地址城市。
客戶郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
客戶電子郵件網域	與計費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 `liu-jie@example.com` ，則項目為 `example.com` 。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

第 6 節：已取消的產品訂閱者

本節列出買方取消的資料，並涵蓋先前的 24 小時期間。

資料欄名稱	描述
客戶參考 ID	不是帳戶 ID 的唯一識別符。它有助於追蹤客戶的用量、收入和訂閱。 客戶參考 ID 排定於 2026 年 3 月 31 日棄用。請改用客戶 AWS 帳戶 ID。
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
使用者的狀態	與訂閱產品的帳戶相關聯的帳單地址狀態。
使用者的國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
產品標題	產品的標題。
產品代碼	產品的唯一識別符。
訂閱日期	訂閱開始的日期。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
客戶國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
客戶狀態	與計費帳戶相關聯的帳單地址狀態。
客戶城市	與收費帳戶相關聯的帳單地址城市。
客戶郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
客戶電子郵件網域	與計費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 `liu-jie@example.com` ，則項目為 `example.com` 。

資料欄名稱	描述
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

每日客戶訂閱者報告

此報告會列出購買您產品之客戶的資料。此報告不會指定目前或過去的用量，只會指定客戶訂閱您的產品。只有在相關資訊可用時，您才會收到此報告。如果您未收到此報告且認為應該收到，請聯絡AWS Marketplace 賣方營運團隊。您也可以聯絡 AWS Marketplace 賣方營運團隊，取消此報告的電子郵件通知。

您可以在 存取此報告AWS Marketplace 管理入口網站。如果您已註冊 the section called “AWS Marketplace Commerce Analytics 服務”，您也可以使用 AWS SDK 存取您的報告。

報告有兩個部分：一個用於每小時和每月訂閱，另一個用於年度訂閱。報告包含訂閱您產品之所有客戶的 AWS 帳戶 IDs 清單。

發佈排程

此報告每天在 UTC 00：00 發佈，涵蓋前一天 UTC 00：00 到 23：59。

主題

- 第 1 節：每小時和每月訂閱
- 第 2 節：可變長度訂閱

第 1 節：每小時和每月訂閱

本節列出截至 UTC 前一天 23：59：59 的所有用量型訂閱的資料。

資料欄名稱	描述
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
產品標題	產品的標題。
產品 ID	軟體產品的唯一識別符。
產品代碼	軟體產品的唯一識別符。
訂閱開始日期	訂閱的開始日期，格式為 YYYY-MM-DD 。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
經銷商帳戶 ID	頻道合作夥伴經銷商的唯一識別符。
經銷商帳戶名稱	頻道合作夥伴經銷商的名稱。

第 2 節：可變長度訂閱

本節列出截至 UTC 前一天 23：59：59 為止所有以費用為基礎的訂閱資料。

資料欄名稱	描述
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
產品標題	產品的標題。
產品 ID	軟體產品的唯一識別符。

資料欄名稱	描述
產品代碼	軟體產品的唯一識別符。此資訊也可作為 Amazon EC2 執行個體中繼資料的一部分取得。
訂閱 ID	訂閱的 ID。
訂閱數量	客戶購買的授權總數。
訂閱類型	訂閱的類型。
訂閱意圖	此項優惠是舊版優惠的升級或續約。
優惠 ID	買方簽署之優惠的識別符。
訂閱開始日期	客戶訂閱產品的日期，格式為 YYYY-MM-DD 。
先前的優惠 ID	如果有升級或續約優惠之前優惠的 ID。
提供可見性	優惠是公有、私有或企業合約優惠。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
經銷商帳戶 ID	頻道合作夥伴經銷商的唯一識別符。
經銷商帳戶名稱	頻道合作夥伴經銷商的名稱。

支出報告

Important

此報告將於 2024 年 8 月 30 日無法使用。您可以使用這些替代方案：

- 集合和支出儀表板

- GitHub 上的 collections_and_disbursements_report SQL 查詢。

支出報告提供自上一次支出以來，我們收集並支出到您的銀行帳戶之資金的相關資訊。支出可能包括客戶為您的產品訂閱付款或退款，以及向客戶收取或退款的一些稅金。在向客戶收取資金之前，您不會收到資金的支出。不同的客戶有不同的付款條件 AWS，因此每個未收帳年齡類別中的部分資金可能不是由客戶支付。

退款會顯示為負金額，因為在您授權退款之後，退款會退回給您的客戶。

此報告可在報告索引標籤 AWS Marketplace 管理入口網站 下的 上取得。若要建立與此類似的報告，您可以使用 AWS Marketplace 中的賣方交付資料摘要 AWS Marketplace，包括 依產品分配 作為基本報告，以自訂 來滿足您的需求。

> **Note**
>
> 若要取消此報告的通知電子郵件，請聯絡AWS Marketplace 賣方營運團隊。

發佈排程

此報告會在開始付款後 3-5 天發佈，以將資金轉移到您的銀行。一般而言，這是每月收到付款的賣方的報告。如果未啟動支出，則不會產生支出報告。

主題

- 第 1 節：依產品分配的金額
- 第 2 節：依客戶地理位置分配的金額
- 第 3 節：依執行個體時數分配的金額
- 第 4 節：未收取資金的存留期
- 第 5 節：已分配資金的存留期
- 第 6 節：逾期資金的存留期
- 第 7 節：未收款明細

第 1 節：依產品分配的金額

本節列出依產品分配的資料。

資料欄名稱	描述
產品	產品的標題。
產品代碼	產品的唯一識別符。
SellerRev	向客戶收取產品用量或費用的金額。
AWSRefFee	AWS Marketplace 費用的金額。
SellerRevRefund	如果在資料涵蓋期間處理任何退款，向客戶退款的訂閱成本金額。
AWSRefFeeRefund	如果在資料涵蓋期間處理任何退款，將會退還 AWS Marketplace 的費用金額。
SellerRevCredit	AWS Marketplace 放置在客戶帳戶的 AWS 點數。
AWSRefFeeCredit	AWS Marketplace 放置在您帳戶的 AWS 點數。
淨額	我們分配給您的總資金。此欄等於 SellerRev 欄減去 AWSRefFee 欄。向客戶提供退款時，此欄是負數，等於 SellerRevRefund 欄減去 AWSRefFeeRefund 欄。
交易參考 ID	交易的唯一識別符，可協助您關聯跨 AWS Marketplace 報告的交易。
SellerUSSalesTax	為此交易計費的美國銷售和使用稅的總金額。
SellerUSSalesTaxRefund	如果已處理退款，則針對此交易所退款的美國銷售和使用稅的總金額。
客戶 AWS 帳戶號碼	支付費用的帳戶 ID。
客戶國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
客戶狀態	與收費帳戶相關聯的帳單地址狀態。

資料欄名稱	描述
客戶城市	與收費帳戶相關聯的帳單地址城市。
客戶郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
客戶電子郵件網域	與收費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 `liu-jie@example.com` ，則項目為 `example.com` 。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與支付費用的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的使用情況、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

第 2 節：依客戶地理位置分配的金額

本節依客戶的地理位置列出支出的資料。

資料欄名稱	描述
和解 ID	支出的唯一識別符。
和解期間開始日期	支出期間的開始日期和時間。
和解期間結束日期	支出期間的結束日期和時間。
存款日期	支出發生的日期和時間。
已分配金額	支出的總金額。
國家代碼	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。

資料欄名稱	描述
狀態或區域	與收費帳戶相關聯的帳單地址狀態。
City	與收費帳戶相關聯的帳單地址城市。
郵遞區號	與軟體收費帳戶相關聯的帳單地址郵遞區號。
按稅務位置的淨額	依稅務位置分配給賣方的總資金，減去 AWS Marketplace 費用、退款以及美國銷售和使用稅。
依稅務位置的總金額	依稅務位置分配給賣方的總資金。
賣方美國銷售稅	代表賣方為此交易計費的美國銷售和使用稅總額。（也就是說，美國銷售和稅務報告中的相關記錄會顯示「稅務責任方」=「賣方」。）
賣方美國銷售稅退款	如果已處理退款，則代表賣方收取此類稅金時，此交易所退款的美國銷售和使用稅的總金額。（也就是說，美國銷售和稅務報告中的相關記錄會顯示「稅務責任方」=「賣方」。）

第 3 節：依執行個體時數分配的金額

本節列出依 Amazon EC2 執行個體時數分配的資料。

資料欄名稱	描述
產品	產品的標題。
產品代碼	產品的唯一識別符。
用量類型描述	用量的描述，包括優惠類型、區域和執行個體類型。
速率	優惠類型、區域和執行個體類型的每小時費率。

資料欄名稱	描述
使用者計數	使用優惠類型、區域和執行個體類型的唯一客戶數量。
執行個體時數	執行個體針對優惠類型、區域和執行個體類型所耗用的時數。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。

第 4 節：未收取資金的存留期

本節列出依存留期整理的未收集資金資料。未收取的資金可能包括尚未到期的金額。

資料欄名稱	描述
未收集 (< 31 天待定）	計費但未收取少於 31 天的資金總計。
未收集 (31–60 天待定）	31-60 天之間計費但未收取的資金總計。
未收集 (61–90 天待定）	61-90 天之間計費但未收取的資金總計。
未收集 (91–120 天待定）	在 91–120 天之間計費但未收取的資金總計。
未收集 (> 120 天待定）	已計費但未收取超過 120 天的資金總計。
未收集 （整體）	所有已計費但未收取的資金總計。

第 5 節：已分配資金的存留期

本節列出自上一個支出以來已收集資金的資料。

資料欄名稱	描述
已收集 (< 31 天待定）	在 0–31 天範圍內計費的已收取資金總計。

資料欄名稱	描述
已收集 (31–60 天待定）	在 31–60 天範圍內計費的已收取資金總計。
已收集 (61–90 天待定）	在 61–90 天範圍內計費的已收取資金總計。
已收集 (91–120 天待定）	在 91–120 天範圍內計費的已收取資金總計。
已收集 (> 120 天待定）	在超過 120 天範圍內計費的已收取資金總計。
已收集 （整體）	所有已收取資金的總計。

第 6 節：逾期資金的存留期

本節列出已累積且由客戶支付，但尚未根據客戶協議支付之資金的資料 AWS。

資料欄名稱	描述
逾期 (< 31 天）	過去 0-31 天內累積且到期，但客戶尚未付款的資金總計。
逾期 (31–60 天）	過去 31-60 天內累積且到期，但客戶尚未付款的資金總計。
逾期 (61–90 天）	過去 61-90 天內已累積但客戶尚未付款的資金總計。
逾期 (91–120 天）	過去 91–120 天內累積且到期，但客戶尚未付款的資金總計。
逾期 (> 120 天）	過去 121 天或更長時間內累積且到期，但客戶尚未付款的資金總計。
逾期 （整體）	已累積且到期，但客戶尚未付款的資金總計。

第 7 節：未收款明細

本節列出所有未收取的資金，依付款到期日排序。

資料欄名稱	描述
付款人 AWS 帳號	軟體收費的 帳戶。
產品代碼	產品的唯一識別符。
總收入	使用 產品或使用 產品的費用。
AWS 收入共享	在結算時從計費金額中扣除 AWS 的費用金額。
總退款	交易的任何退款金額總計。
AWS 退款共享	交易退款 AWS 的費用部分。
淨收入	此交易的計費淨額、減去 AWS 費用、退款，以及美國銷售和使用稅。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則項目為 USD。
AR 期間	交易的月份和年份，格式為 YYYY-MM。
交易參考 ID	代表交易的唯一識別符，您可以用來關聯跨 AWS Marketplace 報告的交易。
機會名稱	已註冊機會的唯一識別符。
機會描述	已註冊機會中的任何中繼資料。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與支付費用的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的使用情況、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。
付款到期日	付款到期日，格式為 YYYY-MM-DD 。

每月計費營收報告

Important

此報告的三個部分將於 2024 年 8 月 30 日無法使用。您可以針對第 1-3 節使用這些替代方案：

- 計費營收儀表板
- GitHub 上的 billed_revenue_report SQL 查詢。

第 4 節：具有彈性付款排程的合約將於 2024 年 8 月 30 日之後繼續提供。

每月計費收入報告為您提供有關每月計費收入的授權資訊，以供會計和其他財務報告之用。此報告顯示針對您產品的每小時、每年或每月用量向客戶 AWS 計費的總金額。報告有四個部分：每小時用量和每月費用的計費金額、可變長度訂閱、欄位示範用量和彈性付款。

Important

此報告中的金額僅反映我們向客戶計費的收入，而不是我們收集的金額。

此報告可在報告索引標籤 AWS Marketplace 管理入口網站 下的 上取得。如果您已註冊商業分析服務，您可以使用 API AWS Marketplace 呼叫來提取此報告的區段。如需詳細資訊，請參閱the section called “AWS Marketplace Commerce Analytics 服務”。

Note

若要取消此報告的通知電子郵件，請聯絡AWS Marketplace 賣方營運團隊。

發佈排程

此報告會在每月 15 日的 UTC 時間 00：00 每月發佈。

帳單和收入資料涵蓋下個月的 15 號到 14 號。

長期訂閱涵蓋該月第一天 UTC 00：00 到上個月最後一天 UTC 23：59。

例如，5 月 15 日發佈的報告涵蓋 UTC 時間 4 月 1 日至 4 月 30 日 23：59 的 UTC。

主題

- 第 1 節：帳單和營收資料
- 第 2 節：可變長度訂閱
- 第 3 節： AWS 欄位示範使用方式
- 第 4 節：具有彈性付款排程的合約

第 1 節：帳單和營收資料

本節列出所收集之用量帳單、退款、費用和美國銷售和使用稅的資料。

資料欄名稱	描述
客戶參考 ID	不是帳戶 ID 的唯一識別符。它有助於追蹤客戶的用量、收入和訂閱。 客戶參考 ID 排定於 2025 年 12 月 31 日棄用。請改用客戶 AWS 帳戶 ID。
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
Country	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
State	與計費帳戶相關聯的帳單地址狀態。
City	與計費帳戶相關聯的帳單地址城市。
郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
產品標題	產品的標題。
產品代碼	產品的唯一識別符。
客戶計費金額	向客戶收取產品用量或每月費用的金額。
AWS 列出費用	要從計費金額中扣除 AWS Marketplace 的費用金額。

資料欄名稱	描述
退款金額	如果在資料涵蓋期間處理任何退款，則向客戶退款的訂閱成本總和。
AWS 費用退款	如果在資料涵蓋期間處理任何退款， AWS Marketplace 則退款的費用部分。
成本	經銷商的貨物成本：例如，經銷商在銷售您的產品時向您支付的費用。
合作夥伴收入金額	交易的帳單總額，不含 AWS Marketplace 費用、退款，以及美國銷售和使用稅。
貨幣	交易的貨幣。例如，如果交易是美元，則項目為 USD。
交易參考 ID	交易的唯一識別符，可協助您關聯跨 AWS Marketplace 報告的交易。
美國銷售稅客戶帳單金額	代表賣方為此交易計費的美國銷售和使用稅總額。（也就是說，美國銷售和稅務報告中的相關記錄會顯示「稅務責任方」=「賣方」。）
美國銷售稅退款金額	如果已處理退款，則代表賣方收取此類稅金時，針對此交易所退款的美國銷售和使用稅金總量。（也就是說，美國銷售和稅務報告中的相關記錄會顯示「稅務責任方」=「賣方」。）
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
客戶電子郵件網域	與收費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 *liu-jie@example.com* ，則項目為 `example.com` 。
機會名稱	已註冊機會的唯一識別符。
機會描述	已註冊機會的中繼資料。

資料欄名稱	描述
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

第 2 節：可變長度訂閱

本節列出以費用為基礎的費用資料。

資料欄名稱	描述
客戶參考 ID	不是帳戶 ID 的唯一識別符。它有助於追蹤客戶的用量、收入和訂閱。 客戶參考 ID 排定於 2025 年 12 月 31 日棄用。請改用客戶 AWS 帳戶 ID。
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
Country	與收費帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
State	與計費帳戶相關聯的帳單地址狀態。
City	與計費帳戶相關聯的帳單地址城市。
郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
產品標題	產品的標題。
產品代碼	產品的唯一識別符。
訂閱數量	指定為可變長度訂閱購買一部分的授權總數。

資料欄名稱	描述
訂閱開始日期	可變長度訂閱購買的開始日期。
訂閱結束日期	可變長度訂閱購買的結束日期。
訂閱執行個體類型	與可變長度訂閱購買相關聯的執行個體類型。
客戶計費金額	為用量、每月費用或兩者收費的金額。
AWS 列出費用	從計費金額中扣除 AWS Marketplace 的費用金額。
退款金額	如果在資料涵蓋期間處理任何退款，退款給客戶的總金額。
AWS 費用退款	如果在資料涵蓋期間處理任何退款，則退還的部分 AWS Marketplace 費用。
成本	經銷商的貨物成本：例如，經銷商在銷售您的產品時向您支付的費用。
合作夥伴收入金額	此交易的帳單總額，不含 AWS Marketplace 費用、退款以及美國銷售和使用稅。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則項目為 USD。
交易參考 ID	交易的唯一識別符，可協助您關聯跨 AWS Marketplace 報告的交易。
美國銷售稅客戶帳單金額	代表賣方為此交易計費的美國銷售和使用稅總額。（也就是說，美國銷售和稅務報告中的相關記錄會顯示「稅務責任方」=「賣方」。）
美國銷售稅退款金額	如果已處理退款，則代表賣方收取此類稅金時，此交易所退款的美國銷售和使用稅總量。（也就是說，美國銷售和稅務報告中的相關記錄會顯示「稅務責任方」=「賣方」。）

資料欄名稱	描述
客戶電子郵件網域	與計費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 `liu-jie@example.com` ，則項目為 `example.com` 。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
合約開始日期	an AWS Marketplace SaaS 合約的開始日期。
合約結束日期	an AWS Marketplace SaaS 合約的結束日期。
機會名稱	已註冊機會的唯一識別符。
機會描述	已註冊機會的中繼資料。
解決方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別符。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。
付款人地址 ID	代表客戶地址的唯一識別符。

第 3 節： AWS 欄位示範使用方式

區段列出您產品的資料 AWS 現場示範用量。您可以設定您的產品，讓我們向潛在客戶展示您的產品。任何來自示範的使用都會列在此處。

資料欄名稱	描述
產品標題	產品的標題。
產品代碼	產品的唯一識別符。

資料欄名稱	描述
執行個體類型	與欄位示範相關聯的 Amazon EC2 執行個體類型。
用量單位	與產品相關聯的用量單位數量。
用量單位類型	與用量單位計數相關聯的用量單位：例如，小時。

第 4 節：具有彈性付款排程的合約

本節列出您在上一個報告期間以彈性付款排程建立之所有合約的資料。

資料欄名稱	描述
客戶 AWS 帳戶號碼	收費的 AWS 帳戶 ID。
客戶國家/地區	與收費付款人帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 alpha-2 標準。
客戶狀態	與收費付款人帳戶相關聯的帳單地址狀態。
客戶城市	與收費付款人帳戶相關聯的帳單地址城市。
客戶郵遞區號	與收費付款人帳戶相關聯的帳單地址郵遞區號。
客戶電子郵件網域	與收費付款人帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 *liu-jie@example.com*，則項目為 example.com 。
使用者參考 ID	收費的付款人帳戶帳戶。
AWS 使用者帳戶號碼	訂閱產品的帳戶 ID。
產品 ID	產品的唯一識別符。
產品標題	產品的標題。
產品類型	產品的類型。

資料欄名稱	描述
AWS Marketplace 優惠 ID	買方簽署之優惠的識別符。
合約建立日期	合約建立日期，即帳戶訂閱優惠的日期。
合約過期日期	合約到期的日期。
合約總值 (USD)	合約的總值，以 USD 為單位。
付款數量	為合約排程的付款數量。
發票日期	發票的建立日期。
發票金額 (USD)	發票上以 USD 計費的金額。
付款人參考 ID	不是帳戶 ID 的唯一識別符。它與費用計費的帳戶相關聯。它有助於追蹤客戶在所有 AWS Marketplace 財務報告中的用量、收入和訂閱。

銷售薪酬報告

Important

此報告將於 2024 年 8 月 30 日無法使用。您可以使用以下替代方法：

- 已計費收入儀表板
- 上的帳單報表查詢SQL。 GitHub

該報告會列出每月計費收益，以及不符合標準規定的其他客戶資訊the section called “每月計費營收報告”。此報告 AWS 會列出產品每小時、每年或每月使用量向客戶收取的總金額。

Important

此報表中的金額僅反映向客戶開立帳單的收入，而非從客戶收取的金額。

銷售補償報告以及在本計劃中與您分享的資訊構成 Amazon 的機密資訊，根據我們與您簽訂的保密協議，或者，如果沒有此類協議，則構成 AWS Marketplace 賣家的條款與條件。您只能透過公司名稱、地理位置和 AWS 帳戶 ID 將 AWS Marketplace 收入映射至代表，來使用此資訊來補償您的業務代表。您可以與需要了解它的員工共享此信息，以了解支付給他們的佣金來源。您使用和共享這些信息必須遵守我們與您的保密協議中的義務以及 AWS Marketplace 賣家的條款和條件，包括但不限於 AWS Marketplace 賣家條款和條件第 3.8 節。

若要建立與此類似的自己報表，您可以使用 AWS Marketplace 中的賣方交付資料摘要 AWS Marketplace，包括銷售補償報告做為基礎報表來自訂以符合您的需求。

出版日程

該報告每月發布，在每月的第十五天 00:00 UTC 發布。該報告涵蓋該月最後一個日曆日UTC的 00:00 UTC 至 23:59 之前一個曆月。舉例來說，5 月 15 日發表的報告涵蓋了 4 月 1 日凌晨零時UTC至四月三十日晚上 11 時 59 UTC 分。

帳單收入

此報告的「已計費收入」區段包含前一個日曆月的用量和收費費用。以下是欄名稱和說明。

> **Note**
>
> 在本報告中，上市費用是根據分級上市費用決定的交易收益（授權經銷商轉售的授權轉售產品除外）的百分比。

資料欄名稱	描述
客戶 AWS 帳號	收費的帳戶。
Country	與支付費用的帳戶相關聯的兩個字元國家/地區代碼。此報告使用 ISO 3166-1 阿爾法 2 標準。
State	帳單地址是與費用支付的帳戶相關聯的狀態。
City	與支付費用的帳戶相關聯的帳單地址城市。
邮政编码	與支付費用之帳戶相關聯的帳單地址郵遞區碼。

資料欄名稱	描述
電子郵件域	與收費的帳戶相關聯的電子郵件網域。例如，如果電子郵件地址是liu-jie@example.com ，則項目為example.com 。
產品代碼	產品的唯一識別碼。
產品標題	產品的標題。
總收入	使用產品所支付的金額或使用產品的月費。
AWS 收益分成	結算時從帳單金額中扣除的 AWS 費用金額。它會出現在the section called “支出報告”.
退款總額	交易的任何退款總金額。
AWS 退款分享	交易退還的 AWS 費用部分。
淨收入	針對此交易計費的淨金額，扣除 AWS 手續費、退款，以及美國銷售和使用稅。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則輸入為USD。
擴增實境期	交易的月份和年份，格式為YYYY-MM。
交易參照識別碼	代表交易的唯一識別碼，可用來關聯 AWS Marketplace 報表之間的交易。
機會名稱	已註冊商機的唯一識別碼。
機會說明	已註冊商機中的任何中繼資料。
方案標題	解決方案的名稱。
解決方案 ID	解決方案的唯一識別碼。
付款人參考 ID	不是帳戶 ID 的唯一識別碼。它與支付費用的帳戶相關聯。它有助於跟踪所有 AWS Marketplace 財務報告中客戶的使用情況，收入和訂閱。

資料欄名稱	描述
付款人地址 ID	代表客戶地址的唯一識別碼。

補充報告

AWS Marketplace 透過 將補充報告交付中的賣方交付資料摘要 AWS Marketplace給賣方擁有的 Amazon S3 帳戶，這些帳戶連接到與賣方 AWS Marketplace 清單相關聯的 AWS Marketplace 賣方帳戶 ID。如需詳細資訊，請參閱建立目的地 Amazon Simple Storage Service 儲存貯體。

如果前一天有新訂閱者，補充報告會在 UTC 每天 16：00 發佈。這些報告涵蓋前一天 13：59 UTC 到次日的 16：01 UTC。

協議詳細資訊報告

協議詳細資訊報告可協助您支援軟體即服務 (SaaS) 合約免費試用的客戶。報告包含協議詳細資訊，例如訂閱者名稱、訂閱者 ID、優惠 ID、協議開始和協議結束日期。

只有在相關資訊可用時，您才會收到此報告。如果您在認為應該時未收到此報告，請聯絡AWS Marketplace 賣方營運團隊。

您可以透過與 AWS Marketplace 賣方帳戶 ID 相關聯的 Amazon S3 儲存貯體存取此報告。

下表列出協議詳細資訊報告的資料欄名稱和描述。

SaaS 合約免費試用報告資料

名稱	描述
`vendor_display_name`	銷售產品的廠商名稱。
`vendor_aws_account_id`	與銷售產品的廠商相關聯的識別。
`subscriber_aws_account_id`	與訂閱產品的 AWS 帳戶 相關聯的識別。
`customer_id`	軟體產品的唯一識別符。
`product_title`	產品的標題。
`offer_id`	買方簽署之優惠的識別符。

名稱	描述
offer_visibility	指出優惠是公有、私有或企業合約優惠。
reseller_name	頻道合作夥伴經銷商的名稱。
reseller_aws_account_id	頻道合作夥伴經銷商的唯一識別符。
agreement_id	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
agreement_acceptance_date	接受協議的日期。
agreement_start_date	協議的開始日期。
agreement_end_date	協議的結束日期。對於以 go/subscriptions 方式計量/付款，此設定為 1-JAN-9999。
is_free_trial_offer	指出優惠或協議是否為免費試用優惠的旗標。
is_upgraded_after_free_trial	指出協議是否已升級至付費合約的旗標。
total_contract_value	合約的總值。

賣方儀表板

AWS Marketplace 提供由 Amazon QuickSight 支援的儀表板，其中包含圖表、圖形和洞見，可協助您存取和分析財務、銷售和行銷資料。賣方儀表板包括：

the section called “財務操作的儀表板”

- the section called “計費收入儀表板” – 提供用於會計和其他財務報告目的之計費收入的相關資訊。
- the section called “集合和支出儀表板” – 提供自上一次支出以來 AWS ，收集並撥付至銀行帳戶的資金資訊。
- the section called “稅務儀表板” – 提供賣方交易稅金的相關資訊。

銷售操作的儀表板

- the section called “協議和續約儀表板” – 在簽署協議後的 24 小時內提供協議和續約的相關資訊 AWS Marketplace。
- the section called “用量儀表板” – 為使用 SaaS 和伺服器用量型產品的客戶提供視覺化和精細資料。
- 行銷儀表板 – 提供多個儀表板，協助您追蹤行銷資料。

行銷儀表板

- 客戶協議儀表板 – 提供有關訂閱您產品的協議和客戶的資料。
- 列出效能儀表板 – 提供 AWS Marketplace 清單中流量和使用者行為的資料。
- 搜尋效能儀表板 – 提供有關套用至 AWS Marketplace 清單之關鍵字的資料。

儀表板可供具有適當許可的 AWS Marketplace 賣方使用。

存取儀表板

根據預設，賣方帳戶的 AWS Marketplace 系統管理員可以存取 中 Insights 索引標籤上的所有儀表板 AWS Marketplace 管理入口網站。系統管理員可以建立 AWS Identity and Access Management (IAM) 政策，為賣方公司中的其他使用者提供特定儀表板的存取權。

> **Note**
>
> 在 2023 年 9 月，我們將不再支援存取由舊版 IAM 許可啟用的賣方儀表板。在下列程式碼範例中，使用新的 Amazon Resource Name (ARN) 格式更新您的 IAM 許可。

如需建立政策的相關資訊，請參閱建立 IAM 政策。

儀表板政策

使用下列其中一個政策來提供部分或全部儀表板的存取權。

下列範例提供目前和未來 AWS Marketplace 資源的存取權，包括儀表板和報告。它使用所有目前和未來的資料饋送：

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

您也可以包含一或多個儀表板ARNs 來提供存取權。下列範例示範如何提供所有儀表板的存取權：

```
{
  "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"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/CustomerAgreements_V1"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/ListingPerformance_V1"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/SearchPerformance_V1"
    ]
  }]
}
```

若要移除對儀表板的存取，請從政策的 Resource區段中刪除它。

> **Note**
>
> 如需建立 AWS Identity and Access Management (IAM) 政策的詳細資訊，請參閱AWS Identity and Access Management 《使用者指南》中的建立 IAM 政策。

財務操作的儀表板

AWS Marketplace 提供多個儀表板，協助您追蹤財務資料。

主題

- 計費收入儀表板
- 集合和支出儀表板
- 稅務儀表板

計費收入儀表板

帳單收入儀表板提供所有帳單銷售的相關資訊 AWS Marketplace。此報告預計可節省時間，且比舊版每月計費營收報告提前約 45 天提供。舊版報告會在每個月的第 15 天傳送，這會延遲上個月帳單的可見性。

若要開啟此儀表板，請登入 AWS Marketplace 管理入口網站，選擇 Insights、財務操作，然後選擇計費營收索引標籤。

主題

- 重新整理計費營收儀表板的頻率
- 第 1 節：控制項
- 第 2 節：篩選條件
- 第 3 節：關鍵指標
- 第 4 節：趨勢
- 第 5 節：明細
- 第 6 節：精細資料

重新整理計費營收儀表板的頻率

儀表板會在 UTC 每天午夜更新。如果發票是在 5 月 10 日 UTC 午夜之前建立，則 5 月 11 日的更新會顯示發票。如果發票是在 5 月 10 日 UTC 午夜之後建立，則 5 月 12 日的更新會顯示發票。如果從上游系統收到的最新發票或支出資料延遲，最新的資料可能會延遲 1-2 天，以反映儀表板。

如果您沒有看到預期的客戶發票，請在聯絡 AWS Marketplace 業務開發聯絡人之前，使用下列程序。

為計費營收儀表板的遺失客戶發票進行疑難排解

1. 使用下列一或多個工具，確認客戶已接受優惠：
   - 每日客戶訂閱者報告
   - AWS Marketplace 事件通知
   - AWS Marketplace 管理入口網站 協議索引標籤
2. （適用於私有優惠）在您確認客戶接受優惠之後，請檢閱優惠發票排程和金額：
   - 如需私有優惠，請參閱 AWS Marketplace 管理入口網站 優惠索引標籤。
   - 對於管道合作夥伴私有優惠，請查看 AWS Marketplace 管理入口網站 合作夥伴索引標籤，以在合作夥伴機會中檢視自訂付款排程。
3. 考慮交易是否使用用量定價模型，其中會在每個月的第二天和第三天針對上個月的用量向客戶收費。

第 1 節：控制項

儀表板的此區段提供篩選條件，以精簡您的計費營收儀表板資料。例如，您可以從AWS Marketplace事件通知中選取欄位的篩選條件，以確認特定客戶帳戶 ID、訂閱者公司名稱或優惠 ID 的帳單。您也可以將篩選條件新增至分析，例如您想要包含在視覺效果中的日期範圍。在控制項中選取的篩選條件會更新指標、趨勢、明細和精細資料區段中顯示的資料。

如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的在 QuickSight 上篩選資料。 QuickSight

控制項描述

控制項名稱	描述
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者公司名稱	訂閱產品的帳戶名稱。

控制項名稱	描述
產品標題	產品的標題。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
AWS 記錄賣方	促進交易之商業實體的識別符。可能的值包括： • AWS_INC： AWS， Inc. 的識別符（位於美國）。 • AWS_EUROPE： AWS EMEA SARL 的識別符（以盧森堡為基礎）。 • AWS_澳洲： AWS Australia Pty Ltd 的識別符 • AWS_JAPAN： AWS 日本 G.K 的識別符。 • AWS_KOREA： AWS 韓國的識別符。
付款人 AWS 帳戶 ID	支付費用的帳戶 ID。
付款人公司名稱	收取費用之帳戶的商業名稱。
經銷商公司	授權販售軟體製造商產品之經銷商帳戶的商業名稱。
經銷商 AWS 帳戶 ID	從 ISV 以批發形式購買產品或服務以向客戶轉售的帳戶 ID。
轉售授權 ID	已註冊機會的唯一識別符。
轉售授權名稱	已註冊機會的唯一名稱。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。

控制項名稱	描述
訂閱者狀態或區域	與訂閱產品的帳戶相關聯的帳單地址狀態或區域。
交易參考 ID	交易的唯一識別符，可協助您關聯 AWS Marketplace 舊版報告的交易。

第 2 節：篩選條件

儀表板的這個區段提供篩選條件，根據優惠的貨幣和發票日期範圍來精簡記錄。您可以使用發票日期範圍來選取帳單日期是在指定日期之前或之後，還是在日期範圍內。選取的篩選條件會更新指標、趨勢、明細和精細資料區段中顯示的資料。預設發票日期範圍設定為提取過去六個月的資料。

第 3 節：關鍵指標

儀表板的此區段會顯示金鑰效能指標 (KPI)，以視覺化方式呈現使用所選篩選條件的金鑰計費營收數字比較。針對指定發票日期範圍和貨幣的總營收、總退款、列示費用、批發成本（如適用）和賣方淨營收，會顯示 KPI。您可以更新篩選條件，以查看不同貨幣和日期的指標。

第 4 節：趨勢

儀表板的本節提供指定日期範圍的計費營收趨勢。您可以依指定的日期彙總檢視趨勢，例如日、month-over-month、quarter-over-quarter或year-over-year，以深入了解計費收入。計費收入趨勢資訊可在彙總層級或依提供可見性類型提供：

- 帳單趨勢 – 提供總收入、賣方淨收入、批發成本（如適用）的快照，以及從日期彙總篩選條件中所選日期範圍一段時間內的退款。
- 依提供可見性的計費趨勢 – 透過提供跨私有優惠、公開優惠（或自助式）和企業計劃隨時間變化的可見性類型，提供優惠計數和總收入的快照。

第 5 節：明細

儀表板的本節提供跨訂閱者、記錄賣方、訂閱者地理位置和產品標題的業務相關重要指標。您可以依總收入、付款人計數、訂閱者計數、總退款、列示費、賣方淨收入和批發成本進行篩選。

第 6 節：精細資料

儀表板的這個區段會顯示所有計費銷售，包括針對您產品的每小時、每年或每月用量向客戶 AWS 計費的總金額。 使用以下三個頻率向客戶 AWS 計費：

- 訂閱接受日期（預付帳單）
- 自訂付款排程（彈性付款排程器所建置的私有優惠）
- 上個月用量在當月第二天和第三天的計量用量

> **Note**
>
> 在 2021 年 4 月 1 日之前建立的發票可能沒有相關聯的協議 ID、優惠 ID、訂閱者 AWS 帳戶 ID 或訂閱者公司名稱。

向客戶計費後 24 小時，客戶費用會顯示在此精細資料區段中。例如，如果客戶是在 11 月 3 日收費，則儀表板會在 11 月 4 日報告此發票。如需如何從 QuickSight 資料表匯出和下載資料的詳細資訊，請參閱 QuickSight 使用者指南中的從視覺效果匯出資料。

精細資料描述

資料欄	描述
發票日期	客戶收到產品訂閱帳單的日期。
付款到期日	付款到期日，格式為 YYYY-MM-DD。
付款條件	客戶的 AWS發票付款條件。
發票 ID	指派給已計費之 AWS 發票的 AWS ID。
列出費用發票 ID	透過 AWS EMEA SARL、日本或澳洲法人實體（記錄賣方）交易 AWS Marketplace 訂閱時，銷售的運算子需要向賣方收取賣方清單費用的增值稅 (VAT)，並給予符合稅務規範的發票。對於適用的交易，列出費用上評估之 VAT 的發票 ID 與軟體或產品訂閱發票 ID 不同。來自 AWS， Inc. 的交易值為 "Missing_listing_fee_invoice_id"，因為清單費用發票 ID 不適用。
批發發票 ID	指派給不可支付發票的 AWS ID，該發票代表在頻道合作夥伴私人優惠 (CPPO) 中 ISV 和頻道合作夥伴之間的銷售。對於公開優惠和

資料欄	描述
	Marketplace Private 優惠 (MPOs)，此欄位將是「不適用」。
訂閱者公司名稱	訂閱產品的帳戶名稱。
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者電子郵件網域	與訂閱產品之帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 liu-jie@example.com，則項目為 example.com。
訂閱者城市	與訂閱產品的帳戶相關聯的帳單地址城市。
訂閱者狀態或區域	與訂閱產品的帳戶相關聯的帳單地址狀態。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的帳單地址國家/地區。
訂閱者郵遞區號	與訂閱產品的帳戶相關聯的帳單地址郵遞區號。
產品標題	產品的標題。
優惠名稱	優惠的賣方定義名稱。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
協議開始日期	客戶產品訂閱開始的日期，格式為 MM-DD-YYYY。如果這是未來日期的協議，則此日期可能與接受日期不同。
協議接受日期	客戶訂閱產品的日期，格式為 MM-DD-YYYY。

資料欄	描述
協議結束日期	合約到期的日期，格式為 MM-DD-YYYY。對於計量/pay-as-you-go訂閱，此日期設定為 JAN-1-9999。
用量期間結束日期	產品使用期間的結束日期。
使用期間開始日期	產品使用期間的開始日期。
支出狀態	與發票相關聯的狀態，以確認 AWS 未將任何資金、部分或所有資金支付到您的銀行帳戶。可能的狀態為：已撥付、部分撥付、失敗、未撥付。
付款日期	AWS 開始向賣方銀行支付款項的日期。
支付銀行追蹤 ID	對於支出，追蹤 ID 由銀行指派。支出銀行追蹤 ID 可用於關聯賣方銀行提供的存款通知和報告中發票 AWS Marketplace 的報告。
總收入	向客戶收取產品用量或每月費用的金額。
總退款	如果在資料涵蓋期間處理任何退款，則退還給客戶的訂閱成本總和。
列出費用	要從計費金額中扣除 AWS Marketplace 的費用金額。
列出費用退款	如果在資料涵蓋期間處理任何退款，則退還的部分 AWS Marketplace 費用。
列出費用百分比	要從計費金額中扣除 AWS Marketplace 的費用百分比。
賣方稅務共享	為此交易計費的美國銷售和使用稅的總金額。
賣方稅金分攤退款	如果已處理退款，則此交易的美國銷售和使用退稅總金額。

資料欄	描述
AWS 稅務共享	代表賣方為此交易計費的美國銷售和使用稅金總量。
AWS 退稅	如果已處理退款，則代表賣方收取此類稅金時，此交易的美國銷售和使用退稅總金額。
批發成本	僅適用於頻道合作夥伴的私有優惠。經銷商的貨物成本。例如，經銷商在銷售製造商的產品時向製造商支付的費用。批發成本是清單價格乘以折扣百分比。
批發成本退款	僅適用於頻道合作夥伴的私有優惠。經銷商商品的退款成本。
批發賣方稅單	ISV 和頻道合作夥伴在頻道合作夥伴私人優惠 (CPPO) 中銷售的稅金，其中賣方需承擔稅金。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發賣方稅務共享退款	ISV 和通路合作夥伴在通路合作夥伴私有優惠 (CPPO) 中銷售的稅金退款，賣方應負責該優惠。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發其他賣方稅務共享	ISV 和頻道合作夥伴在頻道合作夥伴私人優惠 (CPPO) 中銷售的稅金，其中賣方需承擔稅金。當交易涉及的其他賣方負有責任時，會填入此欄位。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發其他賣方稅務共享退款	ISV 和通路合作夥伴在通路合作夥伴私有優惠 (CPPO) 中銷售的稅金退款，賣方應負責該優惠。當交易涉及的其他賣方負有責任時，會填入此欄位。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。

資料欄	描述
批發 AWS 稅共享	頻道合作夥伴私有優惠 (CPPO) 中 ISV 和頻道合作夥伴之間銷售的稅金 AWS ，其中 需納稅。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發 AWS 稅共享退款	ISV 和頻道合作夥伴在 的頻道合作夥伴私人優惠 (CPPO) 中銷售的稅金退款 AWS 需負稅金。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
賣方淨收入	交易計費的總金額，扣除 AWS Marketplace 費用、退款和美國銷售和使用稅。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則項目為 USD。
交易參考 ID	代表交易的唯一識別符，可用來關聯跨 AWS Marketplace 報告的交易。
AWS 記錄賣方	促進交易之商業實體的識別符。可能的值如下： • AWS_INC：AWS， Inc. 的識別符（位於美國） • AWS_EUROPE： AWS EMEA SARL 的識別符（以盧森堡為基礎） • AWS AUSTRALIA： AWS Australia Pty Ltd 的識別符 • AWS_JAPAN： AWS 日本 G.K 的識別符。 • AWS_KOREA： AWS 韓國的識別符
轉售授權 ID	已註冊機會的唯一識別符。
轉售授權名稱	已註冊機會的唯一名稱。
轉售授權描述	已註冊機會的 ISV 定義描述。

資料欄	描述
轉售公司名稱	以批發成本從 ISV 購買產品或服務以向客戶轉售的帳戶名稱。
付款人 AWS 帳戶 ID	從 ISV 以批發成本購買產品或服務以向客戶轉售的帳戶 ID。
付款人電子郵件網域	與收費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 liu-jie@example.com，則項目為 example.com。
付款人城市	與收費帳戶相關聯的帳單地址城市。
付款人狀態或區域	與計費帳戶相關聯的帳單地址狀態。
付款人國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。
付款人郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。
ISV 帳戶 ID	產品或服務擁有者的識別符。
ISV 公司名稱	產品或服務擁有者的商業名稱。
產品 ID	軟體產品的易記唯一識別符。
已分配的淨收入	分配給賣方的交易總金額。如果金額不等於「賣方淨收入」，則這是部分付款。
未分配的淨收入	交易未分配給賣方的總金額。如果金額不是零，則客戶應支付餘額。

集合和支出儀表板

集合和支出儀表板提供有關支付給銀行帳戶之資金的資料。它也提供所有發票的清單，其中已傳送部分付款、完整付款或發生付款失敗。該清單不包括未轉換支出的發票。若要檢閱未支出的發票，請使用計費收入儀表板。

支出包括客戶付款、您產品訂閱的已結清退款，以及向客戶收取或退款的一些稅金。儀表板上的退款會顯示為負金額，因為在您授權退款之後，款項會傳回給客戶。

集合和支出儀表板可讓您更快速地存取客戶支出。與舊版支出報告相比，預計可節省大約四天，該報告會在支出傳送後五天建立。

> **Note**
>
> 客戶有不同的付款條件 AWS，因此未收款存留期類別中的部分資金可能不是由客戶支付。

集合和支出儀表板提供操作和財務程序的資訊。儀表板會每天重新整理。如需詳細資訊，請參閱下列主題。

若要開啟儀表板，請登入 AWS Marketplace 管理入口網站，選擇 Insights、財務操作，然後選擇集合和支出索引標籤。

主題

- 重新整理集合和支出儀表板的頻率
- 第 1 節：控制項
- 第 2 節：篩選條件
- 第 3 節：關鍵指標
- 第 4 節：趨勢
- 第 5 節：明細
- 第 6 節：精細資料

重新整理集合和支出儀表板的頻率

集合和支出儀表板只會在北美工作日更新。您可以預期在收到銀行存款後的一天內看到已支付發票。

第 1 節：控制項

儀表板的這個區段提供篩選條件來精簡儀表板資料。例如，您可以從AWS Marketplace 事件通知中選取欄位的篩選條件，以確認特定客戶帳戶 ID、訂閱者公司名稱或優惠 ID 的支出。您也可以依支出狀態進行篩選，以了解所有已支付給您的發票，或開立和未支付發票。您可以將篩選條件新增至分析，例如您想要包含在視覺效果中的日期範圍。在控制項中選取的篩選條件會更新指標、趨勢、明細和精細資料區段中顯示的資料。

如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的在 QuickSight 上篩選資料。 QuickSight

控制項描述

控制項名稱	描述
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者公司名稱	訂閱產品的帳戶名稱。
產品標題	產品的標題。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
AWS 記錄賣方	促進交易之商業實體的識別符。可能的值包括： • AWS_INC： AWS， Inc. 的識別符（位於美國） • AWS_EUROPE： AWS EMEA SARL 的識別符（以盧森堡為基礎） • AWS_澳洲： AWS Australia Pty Ltd (AWS 澳洲）的識別符 • AWS_JAPAN： AWS 日本 G.K 的識別符。 • AWS_KOREA： AWS 韓國的識別符
支出狀態	與發票相關聯的狀態，以確認 AWS 未向銀行帳戶支付任何資金、部分或所有資金。可能的狀態為：已撥付、部分撥付、失敗、未撥付。
付款人 AWS 帳戶 ID	支付費用的帳戶 ID。
付款人公司名稱	收取費用之帳戶的商業名稱。
經銷商公司	授權販售軟體製造商產品之經銷商帳戶的商業名稱。

控制項名稱	描述
經銷商 AWS 帳戶 ID	從 ISV 以批發形式購買產品或服務以向客戶轉售的帳戶 ID。
轉售授權 ID	已註冊機會的唯一識別符。
轉售授權名稱	已註冊機會的唯一名稱。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。
訂閱者狀態或區域	與訂閱產品的帳戶相關聯的帳單地址狀態或區域。
交易參考 ID	交易的唯一識別符，可協助您關聯 AWS Marketplace 舊版報告的交易。
支付銀行追蹤 ID	對於支出，追蹤 ID 由銀行指派。銀行追蹤 ID 可以將賣方銀行提供的存款通知和報告與報告中的 AWS Marketplace 發票相關聯。

第 2 節：篩選條件

儀表板的這個區段提供篩選條件，根據優惠貨幣和兩個不同的日期維度來精簡記錄，無論日期欄位值是在指定日期之前或之後，還是在日期範圍內。日期維度是付款到期日或上次付款日期。支出日期會將結果限制在指定日期範圍內支出的發票。付款到期日包含到期日在指定範圍內的發票，無論付款日期為何。日期類別篩選條件會更新指標、趨勢、明細和精細資料區段中顯示的資料。預設日期類別是上次分配日期，並提取過去六個月的資料。

第 3 節：關鍵指標

儀表板的這個區段會顯示關鍵績效指標 (KPI)，以視覺化已分配和未分配收入數字之間的比較。針對總收入、淨收入、批發成本 （如果適用）、已分配金額和指定篩選條件未分配金額，會顯示 KPI。

第 4 節：趨勢

儀表板的此區段提供指定日期範圍的支出和逾期趨勢檢視。您可以依指定的日期彙總檢視趨勢，例如依日、月、季或年，以深入了解您的 AWS Marketplace 集合運作狀態。趨勢檢視包括下列項目：

- 支出趨勢 – 提供平均支出天數和相關淨收入的快照。趨勢會測量發票日期和支出日期之間的天數，以報告收集效率。您可以從日期彙總篩選條件中選取日期範圍。
- 已支付款項的存留期 – 提供淨收入的快照，以及依標準帳齡應收儲存貯體（例如未到期、1-30 天和 31 至 60 天）分類的已支付發票計數。趨勢會測量付款到期日和付款日期之間的天數，以報告付款是否在客戶的付款期限內。
- 未支付款項的存留期 – 提供淨收入的快照，以及未結和未付發票的計數，依逾期儲存貯體（例如未到期、1-30 天和 31 到 60 天）整理。未分配的資金可能包括尚未到期的金額。趨勢會測量今天日期與付款到期日之間的天數，以顯示傳入的應收款項。

第 5 節：明細

儀表板的此區段可讓您依優惠 ID、產品標題、付款人公司名稱、訂閱者公司名稱、經銷商名稱（如果他們參與通路合作夥伴私有優惠）、付款人地理位置和訂閱者地理位置來檢視應收款項。使用明細來衡量每個類別的已分配應收款項與未分配應收款項。

第 6 節：精細資料

儀表板的這個區段會依產品、客戶和優惠詳細資訊顯示所有支出和未收取的資金。

> **Note**
>
> 在 2021 年 4 月 1 日之前建立的發票可能沒有相關聯的協議 ID、優惠 ID、訂閱者 AWS 帳戶 ID 或訂閱者公司名稱。

如需有關如何從 QuickSight 資料表匯出和下載資料的資訊，請參閱 QuickSight 使用者指南中的從視覺效果匯出資料。

精細資料描述

資料欄	描述
發票日期	客戶收到產品訂閱帳單的日期。
付款到期日	付款到期日，格式為 YYYY-MM-DD。
付款條件	客戶的 AWS 發票付款條件。
發票 ID	指派給已計費之 AWS 發票的 AWS ID。

資料欄	描述
列出費用發票 ID	透過 AWS EMEA SARL、日本或澳洲法人實體（記錄賣方）交易 AWS Marketplace 訂閱時，銷售市場營運商（例如，AWS EMEA SARL）需要向賣方收取賣方清單費用的增值稅。對於適用的交易，列出費用上評估之 VAT 的發票 ID 與軟體或產品訂閱發票 ID 不同。
批發發票 ID	指派給不可支付發票的 AWS ID，代表 ISV 和頻道合作夥伴在頻道合作夥伴私人優惠 (CPPO) 中的銷售。對於公開優惠和 Marketplace Private 優惠 (MPOs)，此欄位將是「不適用」。
訂閱者公司名稱	訂閱產品的帳戶名稱。
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者電子郵件網域	與訂閱產品之帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 liu-jie@example.com，則項目為 example.com。
訂閱者城市	與訂閱產品的帳戶相關聯的帳單地址城市。
訂閱者狀態或區域	與訂閱產品的帳戶相關聯的帳單地址狀態。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的帳單地址國家/地區。
訂閱者郵遞區號	與訂閱產品的帳戶相關聯的帳單地址郵遞區號。
產品標題	產品的標題。
優惠名稱	優惠的賣方定義名稱。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。

資料欄	描述
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
協議開始日期	客戶產品訂閱開始的日期，格式為 MM-DD-YYYY。如果這是未來日期的協議，則此日期可能與接受日期不同。
協議結束日期	合約到期的日期，格式為 MM-DD-YYYY。 對於計量/pay-as-you-go訂閱，此日期設定為 JAN-1-9999。
協議接受日期	客戶訂閱產品的日期，格式為 MM-DD-YYYY。
用量期間結束日期	產品使用期間的結束日期。
使用期間開始日期	產品使用期間的開始日期。
支出狀態	與發票相關聯的狀態，用於確認 自上一次付款以來 AWS 已收集資金並將其分配給您的銀行帳戶。已收集並支付相關發票的已支付款項。尚未收集和支付相關發票的未支付款項。
付款日期	AWS 開始向賣方銀行支付款項的日期。
支付銀行追蹤 ID	對於支出，追蹤 ID 由銀行指派。銀行追蹤 ID 可用來關聯賣方銀行提供的存款通知，並在報告中報告發票 AWS Marketplace 。
總收入	向客戶收取的產品用量或每月費用的金額。
總退款	如果在資料涵蓋期間處理任何退款，則退還給客戶的訂閱成本總和。
列出費用	要從計費金額中扣除 AWS Marketplace 的費用金額。

資料欄	描述
列出費用退款	如果在資料涵蓋期間處理任何退款，則退還的部分 AWS Marketplace 費用。
列出費用百分比	要從計費金額中扣除 AWS Marketplace 的費用百分比。
賣方稅務共享	為此交易計費的美國銷售和使用稅的總金額。
賣方稅金分攤退款	如果已處理退款，則此交易的美國銷售和使用退稅總金額。
AWS 稅務共享清單費用	AWS 作為市場營運商，需要以自己的名稱收取賣方銷售的增值稅。此金額將對應至向您的 AWS Marketplace 賣方帳戶收取的增值稅金額。
AWS 稅務共享退稅清單費用	AWS 作為市場營運商，需要以自己的名稱退還賣方銷售的增值稅。此金額將對應至退回至您 AWS Marketplace 賣方帳戶的 VAT 金額。
批發成本	僅適用於頻道合作夥伴的私有優惠。經銷商的貨物成本。例如，經銷商在銷售製造商的產品時向製造商支付的費用。批發成本是清單價格乘以折扣百分比。
批發成本退款	僅適用於頻道合作夥伴的私有優惠。經銷商商品的退款成本。
批發賣方稅分攤	ISV 和頻道合作夥伴在頻道合作夥伴私有優惠 (CPPO) 中銷售的稅金，其中賣方需承擔稅金。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發賣方稅務共享退款	ISV 和通路合作夥伴在通路合作夥伴私有優惠 (CPPO) 中銷售的稅金退款，賣方應負責該優惠。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。

資料欄	描述
批發其他賣方稅務共享	ISV 和頻道合作夥伴在頻道合作夥伴私有優惠 (CPPO) 中銷售的稅金，其中賣方需承擔稅金。當交易涉及的其他賣方負責時，會填入此欄位。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發其他賣方稅務共享退款	ISV 和通路合作夥伴在通路合作夥伴私有優惠 (CPPO) 中銷售的稅金退款，賣方應負責該優惠。當交易涉及的其他賣方負責時，會填入此欄位。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發 AWS 稅分攤	ISV 和頻道合作夥伴在 的頻道合作夥伴私人優惠 (CPPO) 中銷售的 AWS 稅負稅金。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
批發 AWS 稅共享退款	ISV 和頻道合作夥伴在 的頻道合作夥伴私人優惠 (CPPO) 中銷售的稅金退款 AWS 需承擔稅金。對於公開優惠和 Marketplace Private 優惠 (MPPOs)，此欄位將是「不適用」。
賣方淨收入	交易計費的總金額，扣除 AWS Marketplace 費用、退款和美國銷售和使用稅。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則項目為 USD。
交易參考 ID	代表交易的唯一識別符，您可以用來關聯跨 AWS Marketplace 報告的交易。

資料欄	描述
AWS 記錄賣方	促進交易之商業實體的識別符。可能的值如下： • AWS_INC： AWS， Inc. 的識別符（位於美國） • AWS_EUROPE： AWS EMEA SARL 的識別符（以盧森堡為基礎） • AWS Australia Pty Ltd AWS（澳洲） • AWS 日本 G.K. • AWS_韓國
轉售授權 ID	已註冊機會的唯一識別符。
轉售授權名稱	已註冊機會的唯一名稱。
轉售授權描述	已註冊機會的 ISV 定義描述。
經銷商公司名稱	以批發成本從 ISV 購買產品或服務以向客戶轉售的帳戶名稱。
經銷商 AWS 帳戶 ID	從 ISV 以批發成本購買產品或服務以向客戶轉售的帳戶 ID。
付款人公司名稱	收取費用的帳戶名稱。
付款人 AWS 帳戶 ID	支付費用的帳戶 ID。
付款人電子郵件網域	與收費帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 liu-jie@example.com，則項目為 example.com。
付款人城市	與收費帳戶相關聯的帳單地址城市。
付款人狀態或區域	與計費帳戶相關聯的帳單地址狀態。
付款人國家/地區	與收費帳戶相關聯的兩個字元國家/地區代碼。
付款人郵遞區號	與收費帳戶相關聯的帳單地址郵遞區號。

資料欄	描述
ISV 帳戶 ID	產品或服務擁有者的識別符。
ISV 公司名稱	產品或服務擁有者的商業名稱。
產品 ID	軟體產品的易記唯一識別符。
支出狀態	與發票相關聯的狀態，以確認 AWS 未向銀行帳戶支付任何資金、部分或所有資金。可能的狀態為：已撥付、部分撥付、失敗、未撥付。
已分配的淨收入	分配給賣方的交易總金額。如果金額不等於「賣方淨收入」，則這是部分付款。
未分配的淨收入	交易未分配給賣方的總金額。如果金額不是零，則客戶應支付餘額。
支出期間	描述應收款項範圍的類別（例如，未到期、1-30 天和 31 到 60 天）。

稅務儀表板

稅務儀表板提供美國和國際交易稅務的視覺化和精細資料 AWS Marketplace。

若要開啟此儀表板，請登入 AWS Marketplace 管理入口網站，選擇洞見、行銷，然後選擇列出效能索引標籤。

主題

- 重新整理稅務儀表板的頻率
- 第 1 節：控制項
- 第 2 節：篩選日期
- 第 3 節：指標
- 第 4 節：稅務趨勢
- 第 5 節：明細
- 第 6 節：精細資料
- 美國豁免代碼

- 歐洲豁免代碼

重新整理稅務儀表板的頻率

儀表板會每日更新。如果從上游系統收到的最新交易資料延遲，最新的資料可能會有 1 到 2 天的延遲，以便反映在儀表板上。

第 1 節：控制項

儀表板的本節提供篩選條件，以精簡您的稅務資料。您也可以將篩選條件新增至分析，例如您想要包含在視覺效果中的日期範圍。在控制項中選取的篩選條件會更新指標、趨勢、明細和精細資料區段中顯示的資料。

如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的在 QuickSight 上篩選資料。 QuickSight

控制項描述

控制項名稱	描述
發票 ID	指派給向客戶收取費用的 AWS ID。
已稅客戶 AWS 帳戶 ID	為產品訂閱徵稅的帳戶 AWS ID。
AWS 記錄賣方	促進交易之商業實體的識別符。
稅務客戶國家/地區	與已課稅客戶相關聯的兩個字元國家/地區代碼
已課稅的客戶州或區域	與已課稅客戶相關聯的帳單地址狀態或區域。
稅務客戶城市	與已稅客戶相關聯的帳單地址城市
應稅	可課稅、不可課稅和退稅的來源交易。
稅務責任方	AWS 或 Seller。如果賣方是納稅方，則會收取稅金。如果 AWS 是稅務責任方，則會收取銷售稅並由 傳送 AWS。如需詳細資訊，請參閱AWS Marketplace 賣方的稅務說明。 如果未收取稅金，則不會顯示任何值。賣方必須判斷是否已為每個發票收取一些稅金，因為賣方需負責收取稅金。

控制項名稱	描述
稅務類型	套用至交易的稅金類型。可能的值為 None、 Sales和 SellerUse 、Tax registration type 、 VAT（增值稅）CNPJ、、IGST、CT。

第 2 節：篩選日期

儀表板的這個區段提供篩選條件，根據指定日期範圍的發票日期來精簡記錄。在發票日期中選取的篩選條件會更新指標、趨勢、明細和精細資料區段中顯示的資料。例如，如果您選取日期篩選條件 Past 30 days，則所有區段都會反映與過去 30 天內課稅發票相關聯的資料。

第 3 節：指標

儀表板的此區段會顯示關鍵績效指標 (KPI)，以視覺化與稅務相關的指標，包括應稅金額、非應稅金額、稅金、應稅發票數量，以及交易總數。您可以在篩選條件區段中更新日期篩選條件，以更新日期範圍。

第 4 節：稅務趨勢

儀表板的本節提供指定日期範圍的稅務趨勢。您可以依指定的日期彙總檢視趨勢，例如每日、month-over-month、quarter-over-quarter或year-over-year，以深入了解稅務。提供下列稅務趨勢資訊：

- 稅務趨勢 - 金額 – 依發票日期提供所選期間應稅和稅務金額的快照。
- 稅務趨勢 - 發票數量 – 依發票日期提供所選期間內發票數量的快照。

第 5 節：明細

儀表板的這個區段提供您業務的稅務指標，依發票 ID、稅務類型、產品標題和客戶地理位置。

第 6 節：精細資料

儀表板的這個區段會顯示組織交易的精細稅務資料 AWS Marketplace。

精細資料描述

資料欄	描述
發票 ID	指派給已計費之 AWS 發票的 AWS ID。這可以是買方商業發票、賣方批發發票或賣方列示費用發票。
列出費用發票 ID	指派給發票的 AWS ID，其中賣方清單費用和增值稅會向賣方收費。僅適用於 ISVs。
明細項目 ID	明細項目的唯一識別符。退款交易具有與其轉送稅務交易相同的明細項目 ID。
客戶帳單 ID	客戶帳單 ID
稅務責任方	AWS 或 Seller。如果賣方是納稅方，則會收取稅金。如果 AWS 是稅務責任方，則會收取銷售稅並由 傳送 AWS。如需詳細資訊，請參閱AWS Marketplace 賣方的稅務說明。 如果未收取稅金，則不會顯示任何值。賣方必須判斷是否已為每個發票收取一些稅金，因為賣方需負責收取稅金。
交易類型代碼	交易的類型代碼。數值包含： • AWS – 轉送稅務交易。 • REFUND – 全部或部分退款。 • TAXONLYREFUND – 僅退稅。 退款交易會與其原始轉送交易共用明細項目 ID。
產品 ID	軟體產品的易記唯一識別符。
產品標題	購買的產品名稱。

資料欄	描述
產品稅碼	用來識別產品稅務屬性的標準程式碼。您可以在建立或修改產品時選擇屬性。
發票日期	針對產品訂閱向客戶計費和課稅的日期。
已稅客戶 AWS 帳戶 ID	已課稅客戶的 AWS 帳戶 ID。
稅務客戶國家/地區	與已課稅客戶相關聯的兩個字元國家/地區代碼。
已課稅的客戶州或區域	與已課稅客戶相關聯的帳單地址狀態或區域。
稅務客戶城市	與已課稅客戶相關聯的帳單地址城市。
稅務客戶郵遞區號	用於稅務計算的郵遞區號。
稅務類型	套用至交易的稅金類型。可能的值為 None、Sales和 SellerUse 、Tax registration type 、VAT、CNPJ、IGST、CT。
管轄權層級	用於稅務位置的地址的司法管轄區層級。可能的值為 State、County、City 和 District。
稅務管轄區	已課稅司法管轄區的名稱。
顯示價格應稅類型	向客戶顯示的價格的可課稅類型。所有 AWS Marketplace 優惠皆為專屬。
稅務管轄權費率	在司法管轄區層級套用的稅金。
稅金	在管轄權層級收取的稅金。
貨幣	已收取稅金的貨幣。例如，如果交易以美元為單位，則項目為 USD。
法定稅額	以管轄層級收取的法定貨幣顯示的稅金。
法律稅貨幣	以已收取稅金的法定貨幣顯示的貨幣。

資料欄	描述
稅務計算原因代碼	交易是應稅交易，而不是應稅交易、豁免交易，還是零評分交易，由司法管轄區層級組織。
用於稅務計算的日期	用於計算交易稅金的日期。
客戶免稅憑證 ID	豁免憑證的憑證 ID。
客戶免稅憑證 ID 網域	憑證存放在 Amazon 系統中的位置。
客戶豁免憑證層級	提供豁免的司法管轄區層級。
客戶豁免代碼	指定豁免的程式碼。例如 RESALE。
交易參考 ID	交易的唯一識別符，可協助您關聯跨 AWS Marketplace 報告的交易。
AWS 記錄賣方	促進交易之商業實體的識別符。
買方發票 ID	指派給 AWS 發票的 AWS ID，其中已向買方收費。賣方可以將批發發票或列出費用發票與對應的買方費用建立關聯。

> **Note**
>
> 總稅額和總買方費用已移至計費收入儀表板中的AWS 稅務份額和總收入欄位。

美國豁免代碼

豁免代碼	描述
501C	501C 組織
AP	農業生產
CO	慈善組織

豁免代碼	描述
DIRECT	直接付款許可
DISTSBT	Distributor
DP	阿拉伯聯合大公國/外交部
DV	已停用的退役軍人
EDI	教育機構
FG	聯邦政府
HCP	醫療保健供應商
HO	醫院（非營利或州）
INSUR	保險
IPM	IPM
LB	程式庫
MPU	多個使用點豁免；數位產品/軟體豁免
NA	美國土著
NAI	美國原住民
NP	非營利組織
NPR	非營利宗教組織
OT	其他
重新銷售	經銷商
SBE	小型企業豁免
SLG	州/地方政府

歐洲豁免代碼

豁免代碼	描述
SpecialOrg	免收增值稅的組織。
SpecialZone	免稅的區域。
SplitPayment	買方會直接將發票上所述的任何 VAT 付款給稅務機關。

銷售操作的儀表板

AWS Marketplace 提供多個儀表板，協助您追蹤銷售資料。

主題

- 協議和續約儀表板
- 用量儀表板

協議和續約儀表板

此儀表板會在簽署協議的 24 小時內提供有關協議和續約的資訊 AWS Marketplace。此儀表板可協助您追蹤即將到期的訂閱。

若要開啟儀表板，請啟動 AWS Marketplace 管理入口網站，選擇 Insights 索引標籤，然後選擇銷售操作。

主題

- 第 1 節：控制項
- 第 2 節：選取日期類別
- 第 3 節：指標
- 第 4 節：訂閱趨勢
- 第 5 節：明細
- 第 6 節：精細資料

第 1 節：控制項

本節提供篩選條件，以精簡您的協議和續約資料。您可以從下列篩選條件中選取 。

> **Note**
>
> 如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的在 QuickSight 上篩選資料。
> QuickSight

控制項描述

控制項名稱	描述
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者公司名稱	訂閱產品的帳戶名稱。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
CPPO 旗標	是/否標記，指出是否使用管道合作夥伴私有優惠達成協議。如果是，記錄的賣方是管道合作夥伴。如果否，記錄賣方是軟體製造商（獨立軟體廠商）。
ISV AWS 帳戶 ID	產品或服務擁有者的帳戶。
ISV 公司名稱	產品或服務擁有者的商業名稱。
舊版產品 ID	軟體產品的舊版唯一識別符。
產品標題	產品的標題。
轉售授權 ID	ISV 提供給頻道合作夥伴的轉售授權 ID。
轉售授權名稱	ISV 提供給頻道合作夥伴的轉售授權名稱。

控制項名稱	描述
經銷商 AWS 帳戶 ID	從 ISV 以批發形式購買產品或服務以向客戶轉售的帳戶 ID。
經銷商公司名稱	授權販售軟體製造商產品之經銷商帳戶的商業名稱。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。
訂閱者狀態或區域	與訂閱產品的帳戶相關聯的帳單地址狀態或區域。
訂閱者城市	與訂閱產品的帳戶相關聯的帳單地址城市。

第 2 節：選取日期類別

儀表板的本節提供篩選條件，根據指定日期範圍的協議結束日期、協議結束日期或協議開始日期來精簡記錄。在選取日期類別中選取的篩選條件會更新指標、趨勢、明細和精細資料區段中顯示的資料。

如需最近簽署協議的相關資訊，您可以依協議開始日期進行篩選。如果您有興趣了解協議續約，您可以使用協議結束日期進行篩選。

第 3 節：指標

儀表板的此區段會顯示金鑰效能指標 (KPI)，以視覺化方式比較金鑰協議資料。針對作用中協議的數量和結束協議的數量，會顯示 KPI。您可以在選取日期類別欄位中更新日期條件，以更新日期範圍。

第 4 節：訂閱趨勢

儀表板的本節提供指定日期範圍的訂閱趨勢。您可以依指定的日期彙總檢視趨勢，例如每日、month-over-month、quarter-over-quarter或year-over-year，以深入了解協議。協議和續約趨勢資訊可在彙總層級取得。

- 訂閱趨勢 – 根據選取的日期彙總篩選條件和選取的日期類別篩選條件，提供作用中協議數量和已結束協議數量的快照。若要檢視協議開始日期的訂閱趨勢，您可以在選取日期類別篩選條件中選取協議開始日期。 若要檢視協議結束日期的訂閱趨勢，請在選取日期類別篩選條件中選取協議結束日期。

- 續約明細 – 提供已結束或即將在特定天數內結束的續約快照。您可以檢視數個時段的續約明細，例如 1 到 30 天前、31 到 60 天前或超過 60 天前的協議。您也可以篩選即將在零到 30 天內、31 到 60 天內結束的協議，以此類推。

第 5 節：明細

儀表板的這個區段為作用中訂閱者、ISVs和經銷商提供跨公司名稱的業務指標。您可以檢視協議 IDs、結束協議 IDs、優惠 IDs、續約協議 IDs和訂閱者的數目。

第 6 節：精細資料

儀表板的本節顯示協議、優惠、產品、協議提案者、訂閱者、轉售授權、經銷商和 ISVs精細資料。

> **Note**
>
> 如需如何從 QuickSight 資料表匯出和下載資料的詳細資訊，請參閱 QuickSight 使用者指南中的從視覺效果匯出資料。

精細資料描述

資料欄	描述
訂閱者公司名稱	訂閱者公司名稱。
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者電子郵件網域	與訂閱產品之帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 liu-jie@example.com，則項目為 example.com。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的帳單地址國家/地區。
訂閱者狀態或區域	與訂閱產品的帳戶相關聯的帳單地址狀態。
訂閱者城市	與訂閱產品的帳戶相關聯的帳單地址城市。
訂閱者郵遞區號	與訂閱產品的帳戶相關聯的帳單地址郵遞區號。

資料欄	描述
產品標題	產品的標題。
產品 ID	軟體產品的易記唯一識別符。
舊版產品 ID	軟體產品的舊版唯一識別符。
優惠名稱	優惠的賣方定義名稱。
優惠 ID	買方簽署之優惠的識別符。
提供可見性	優惠是公有、私有或企業合約優惠。
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
協議開始日期	客戶產品訂閱開始的日期，格式為 MM-DD-YYYY。如果這是未來日期的協議，則此日期可能與接受日期不同。
協議接受日期	客戶訂閱產品的日期，格式為 MM-DD-YYYY。
協議結束日期	合約到期的日期，格式為 MM-DD-YYYY。 對於計量/pay-as-you-go訂閱，此日期設定為 JAN-1-9999。

資料欄	描述
Status	協議的目前狀態。支援的狀態包括： • 有效 – 部分或全部的協議條款均有效。 • 已終止 – 由於 AWS 起始的終止事件，協議在其預先同意的結束日期之前終止。最常見的是付款失敗。 • 已取消 – 接受者選擇在其結束日期之前終止協議。 • 已過期 – 協議在其預先同意的結束日期結束。 • 已取代 – 已使用替代優惠取代協議。 • 續約 – 使用自動續約等功能，將協議續約為新協議。 • 已封存 – 協議已結束；未指定其結束、終止、取消或過期的確切原因。 • 已轉返 – 由於發生錯誤，協議修訂已轉返。舊版現已啟用。僅適用於非作用中的協議修訂。 • 已取代 – 協議修訂不再處於作用中狀態，且另一個協議修訂現在處於作用中狀態。僅適用於非作用中的協議修訂。 • 已轉返 – 由於發生錯誤，協議修訂已轉返。舊版現已啟用。僅適用於非作用中的協議修訂。
預估協議值	協議的預估成本，也稱為「合約總價值」或「購買金額」。這適用於下列產品類型：SaaS、專業服務和伺服器，以及合約或年度定價類型。
貨幣代碼	與協議估計成本相關聯的優惠定價貨幣。
經銷商公司名稱	以批發成本從 ISV 購買產品或服務以向客戶轉售的帳戶名稱。
經銷商 AWS 帳戶 ID	從 ISV 以批發成本購買產品或服務以向客戶轉售的帳戶 ID。

資料欄	描述
轉售授權 ID	已註冊機會的唯一識別符。
轉售授權名稱	已註冊機會的唯一名稱。
轉售授權描述	已註冊機會的 ISV 定義描述。
CPPO 旗標	QuickSight 中的計算欄位。
協議意圖	填入目前的協議是先前 agreement_id 欄中提供的先前協議的續約或升級。可能的值包括： • 升級：對於私有優惠，升級是任何修改（升級、降級、修改、擴展、續約）的廣泛術語。對於公開優惠，升級是變更條款（使用不同的優惠）以產生新協議的特定協議，並封存特定協議。 • 續約：僅適用於公開優惠。當續約協議的開始日期為特定協議的結束日期，且優惠 ID 尚未變更，但協議 ID 已變更時。例如，協議 A 已變成協議 B。 • 自動續約：僅適用於公開優惠。當目前協議結束時建立新協議時。買方已開啟自動續約。 • 新增：買方已接受條款（使用優惠）並訂閱新協議。
先前的協議 ID	如果續約或升級本合約，則為協議參考。
下一個協議 ID	如果目前協議續約或升級，則下一個協議的協議 ID。
先前的優惠 ID	如果目前協議已續約或升級，則與先前協議相關聯的優惠 ID。
下一個優惠 ID	如果目前協議已續約或升級，則與下一個協議相關聯的優惠 ID。
下一個協議 ID 接受日期 (s)	買方接受下一個協議的日期。

資料欄	描述
提案者 AWS 帳戶 ID	產品或服務擁有者的識別符。
提案者公司名稱	產品或服務擁有者的商業名稱。
ISV AWS 帳戶 ID	賣方的 AWS 識別符。
ISV 公司名稱	賣方的商業名稱。
協議結束 （以天為單位）	Quicksight 中的計算欄位。協議從今天結束的剩餘天數。
協議結束期間	QuickSight 中的計算欄位。協議將在其中結束的期間。

用量儀表板

用量儀表板可為使用 SaaS 和伺服器用量型產品的客戶提供視覺化和精細資料。 AWS Marketplace 賣方可以使用此儀表板來追蹤使用型產品的客戶耗用量，以對產品支援、定價、從公開轉換為私有優惠以及產品終止做出決策。儀表板提供過去 6 個月的資料，這是滾動時段。

若要開啟儀表板，請登入 AWS Marketplace 管理入口網站，選擇 Insights，選擇銷售操作，然後選擇用量索引標籤。

> **Note**
>
> 此儀表板會顯示建立優惠時指定之所有維度索引鍵的用量。例如，對於 Amazon Machine Image (AMI) 產品，維度索引鍵是執行個體類型，而且優惠中指定的所有執行個體類型都會顯示在此儀表板中，即使定價為 0 美元。若要查看在建立優惠時未指定的維度的產品用量，請考慮重新發佈產品以包含您需要的維度。

主題

- 重新整理用量儀表板的頻率
- 第 1 節：控制項
- 第 2 節：篩選條件
- 第 3 節：指標

- 第 4 節：趨勢
- 第 5 節：明細
- 第 6 節：精細資料

重新整理用量儀表板的頻率

儀表板會在每天太平洋標準時間下午 4 點 (UTC 午夜） 更新。請注意，從上游資料來源接收用量資料並可能遇到延遲，您可以參考用量日期和用量報告日期，以清楚了解用量何時發生，相較於儀表板上報告的時間。

第 1 節：控制項

儀表板的本節提供篩選條件，以精簡您的用量資料。例如，您可以從下列篩選條件中選取 。

> **Note**
>
> 如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的在 QuickSight 中篩選資料。
> QuickSight

控制項描述

控制項名稱	描述
最終使用者公司名稱	使用產品的帳戶名稱。
最終使用者 AWS 帳戶 ID	使用產品的帳戶 ID。
最終使用者國家/地區	與使用產品的帳戶相關聯的兩個字元國家/地區代碼。
產品標題	產品的標題。
產品代碼	用來計量產品的現有權利產品代碼。此值也用於將資料與報告聯結，或參考計量服務中提供 AWS Marketplace 的內容。
優惠 ID	買方簽署之優惠的識別符。

控制項名稱	描述
提供可見性	優惠是公有、私有或企業合約優惠。
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
維度鍵	與產品用量相關聯的資源類型。維度鍵適用於 SaaS 和伺服器用量型產品。
訂閱者公司名稱	訂閱產品的帳戶名稱。
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的兩個字元國家/地區代碼。
經銷商公司名稱	授權銷售產品製造商產品的經銷商帳戶名稱。
經銷商 AWS 帳戶 ID	從 ISV 以批發形式購買產品或服務以向客戶轉售的帳戶 ID。
轉售授權 ID	從 ISV 以批發形式購買產品或服務以向客戶轉售的帳戶 ID。
CPPO 旗標	是/否標記，指出是否使用管道合作夥伴私有優惠達成協議。如果是，記錄的賣方是管道合作夥伴。如果否，記錄賣方是產品製造商（獨立軟體廠商）。

第 2 節：篩選條件

儀表板的這個區段提供篩選條件，根據使用日期來精簡記錄。這些篩選條件中選取的值會更新指標、趨勢、明細和精細資料區段中顯示的資料。預設選擇是提取過去 6 個月用量的資料。

第 3 節：指標

儀表板的此區段會顯示關鍵效能指標 (KPI)，以視覺化與消耗相關的指標：預估用量單位、使用量的客戶，以及使用量的產品。您可以透過更新篩選條件區段中的使用日期條件來更新日期範圍。請注意，金鑰指標會顯示所有單位類型的資料。

第 4 節：趨勢

儀表板的本節提供指定日期範圍的使用趨勢。您可以依指定的日期彙總檢視趨勢，例如每日、month-over-month、quarter-over-quarter或year-over-year，以深入了解用量。您也可以選取用量單位類型，以圖形方式檢視其用量趨勢。

第 5 節：明細

儀表板的這個區段提供您企業在公司名稱、產品標題、維度索引鍵和所選單位類型的優惠 IDs等方面的預估用量指標。您也可以選取要檢視的項目數量。

第 6 節：精細資料

儀表板的此區段顯示用量、優惠、產品、訂閱者、付款人、最終使用者、轉售授權、經銷商和獨立軟體供應商 (ISVs) 的精細資料。請注意，精細資料表會顯示所有單位類型的資料。

應視為預估收入，直到月底帳單完成為止。以用量為基礎的發票會在下個月的第二天或第三天針對上個月的用量提供給買方（例如，在 11 月 1 日至 11 月 30 日期間計量用量的客戶，將在 12 月 2 日或 12 月 3 日收到用量的發票）。計量用量可能會在實際用量日期的幾天後到達此儀表板，因此用量日期和用量報告日期可能不同。這表示您可能需要在當月每天造訪儀表板，以取得當月up-to-date追蹤。如需授權客戶帳單資訊，請參閱財務操作索引標籤中的計費收入儀表板。

> **Note**
>
> 如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的從視覺效果匯出資料。

精細資料描述

資料欄	描述
使用日期	客戶產品使用的日期。
用量報告日期	在洞察儀表板中顯示客戶產品消耗的日期。
最終使用者公司名稱	使用產品的帳戶名稱。
最終 AWS 使用者帳戶 ID	使用產品的帳戶 ID。

資料欄	描述
最終使用者電子郵件網域	與使用產品之帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 abc@example.com，則項目為 example.com。
最終使用者城市	與使用產品的帳戶相關聯的城市。
最終使用者狀態或區域	與使用產品的帳戶相關聯的狀態或區域。
最終使用者國家/地區	與使用產品的帳戶相關聯的兩個字元國家/地區代碼。
最終使用者郵遞區號	與使用產品的帳戶相關聯的帳單地址郵遞區號。
產品標題	產品的標題。
舊版產品 ID	產品的舊版唯一識別符。
產品 ID	產品的易記唯一識別符。
產品代碼	用來計量產品的現有權利產品代碼。此值也用於將資料與報告聯結，或參考 中提供的內容 AWS Marketplace Metering Service。
優惠 ID	買方簽署之優惠的識別符。
優惠名稱	優惠的賣方定義名稱。
提供可見性	優惠是公有、私有或企業合約優惠。
協議 ID	提案者與接受者之間簽署的協議的唯一協議資料饋送參考，以開始使用產品。
協議接受日期	當客戶訂閱產品時，UTC 的日期時間戳記。
協議開始日期	客戶產品訂閱開始時，UTC 的日期時間戳記。如果這是未來日期的協議，則此日期可能與接受日期不同。

資料欄	描述
協議結束日期	合約到期的日期，以 UTC 為單位。對於計量/按 pay-as-you-go訂閱，此日期設定為 9999 年 1 月 1 日上午 12：00。
維度鍵	與產品用量相關聯的資源類型。維度鍵適用於 SaaS 和伺服器用量型產品。
區域	買方部署 Amazon EC2 執行個體的區域。
預估用量	為產品記錄的用量數量。
用量單位類型	記錄用量的單位類型。
預估收入	產品用量的收入。應視為預估收入，直到月底帳單完成為止。以用量為基礎的發票會在下個月的第二或第三個月提供給買方，以供上個月使用。
貨幣	交易的貨幣。例如，如果交易以美元為單位，則項目為 USD。
訂閱者公司名稱	訂閱產品的帳戶名稱。
訂閱者 AWS 帳戶 ID	訂閱產品的帳戶 ID。
訂閱者電子郵件網域	與訂閱產品之帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 abc@example.com，則項目為 example.com。
訂閱者城市	與訂閱產品的帳戶相關聯的帳單地址城市。
訂閱者狀態或區域	與訂閱產品的帳戶相關聯的帳單地址狀態。
訂閱者國家/地區	與訂閱產品的帳戶相關聯的帳單地址國家/地區。
訂閱者郵遞區號	與訂閱產品的帳戶相關聯的帳單地址郵遞區號。
付款人公司名稱	為產品付費的帳戶名稱。

資料欄	描述
付款人 AWS 帳戶 ID	為產品付費的帳戶 ID。
付款人電子郵件網域	與為產品付費之帳戶相關聯的電子郵件網域。例如，如果電子郵件地址為 abc@example.com，則項目為 example.com。
付款人城市	與已支付產品之帳戶相關聯的帳單地址城市。
付款人狀態或區域	與已支付產品之帳戶相關聯的帳單地址狀態。
付款人國家/地區	與已支付產品之帳戶相關聯的帳單地址國家/地區。
付款人郵遞區號	與支付產品之帳戶相關聯的帳單地址郵遞區號。
經銷商公司名稱	以批發成本從 ISV 購買產品或服務以向客戶轉售的帳戶名稱。
經銷商 AWS 帳戶 ID	從 ISV 以批發成本購買產品或服務以向客戶轉售的帳戶 ID。
轉售授權 ID	已註冊的轉售機會的唯一識別符。
轉售授權名稱	已註冊的轉售機會的唯一名稱。
轉售授權描述	已註冊轉售機會的描述。
CPPO 旗標	是/否標記，指出是否使用管道合作夥伴私有優惠達成協議。如果是，記錄的賣方是管道合作夥伴。如果否，記錄賣方是產品製造商 (ISV)。
ISV 公司名稱	產品或服務擁有者的名稱。
ISV AWS 帳戶 ID	產品或服務擁有者的識別符。

行銷儀表板

AWS Marketplace 提供多個儀表板，協助您追蹤行銷資料。

主題

- 客戶協議儀表板
- 列出效能儀表板
- 搜尋效能儀表板
- 了解使用 AWS 儀表板購買

客戶協議儀表板

客戶協議儀表板提供訂閱您產品的協議和客戶的概觀 AWS Marketplace。儀表板提供新協議和作用中協議的資料，以及客戶趨勢和設定檔。

> **Note**
>
> - 若要解除鎖定此儀表板，您必須註冊 AWS Marketplace Seller Prime 計畫。
> - 若要開啟此儀表板，請登入 AWS Marketplace 管理入口網站，選擇洞見、行銷，然後選擇客戶協議索引標籤。

如需使用 AWS Marketplace 儀表板的詳細資訊，請參閱本節賣方儀表板稍早的 。

主題

- 第 1 節：篩選條件
- 第 2 節：日期篩選條件深入探討
- 第 3 節：公有和私有優惠協議
- 第 4 節：客戶指標

第 1 節：篩選條件

您可以使用下列篩選條件來精簡協議資料。

控制項名稱	描述
產品標題	產品的標題。

控制項名稱	描述
日期篩選條件	包括過去 30、60 和 90 天、過去 12 個月 (TTM) 和年初至今 (YTD) 的資料。選擇自訂以定義特定的開始和結束日期。
客戶產業	為訂閱賣方產品的客戶 AWS 定義的產業。
客戶客群	為訂閱賣方產品的客戶 AWS 定義的客群。

如需篩選的詳細資訊，請參閱《Amazon QuickSight 使用者指南》中的在 QuickSight 上篩選資料。Amazon QuickSight

第 2 節：日期篩選條件深入探討

此日期篩選條件適用於客戶協議儀表板上的所有指標。

例如，當您使用預設year-to-date篩選條件值時，會出現模糊指標：

指標	描述
公開優惠協議計數	在 YTD 內至少有 1 個有效日的公開優惠可見性協議數量。
私有優惠協議計數	在 YTD 內至少有 1 個有效日的提供可見性，具有私有優惠的協議數量。
新的公開優惠協議計數	在 YTD 內接受日期的公開優惠可見性協議數量。
新的私有優惠協議計數	以私有優惠做為優惠可見性，且接受日期在 YTD 內的協議數量。
作用中的客戶	在 YTD 中至少有 1 個有效協議的唯一客戶數量，識別為訂閱者 AWS 帳戶 ID。作用中協議定義為在期間內至少有 1 個作用中日的協議。
新的付費客戶	唯一客戶的數量，識別為訂閱者 AWS 帳戶 ID，他們在 YTD 內的第一個計費月份。

第 3 節：公有和私有優惠協議

儀表板的這個區段會顯示協議的概觀。關鍵績效指標 KPIs) 包括公開優惠協議的數量、私有優惠協議的數量、新的公開優惠協議的數量，以及新的私有優惠協議的數量。您可以查看磁碟區和百分比的year-over-year或period-over-period變更。您可以透過更新篩選條件區段中的日期篩選條件來更新日期範圍。

協議是在提案者（產品或服務擁有者）與接受者（客戶）之間簽訂的合約，以開始使用產品。

> **Note**
>
> 新的協議指標包括作用中、已過期、已取消和已終止的協議。

下表列出並說明協議狀態。

狀態	描述
Active (作用中)	部分或所有協議條款均有效。
已過期	協議在其預先同意的結束日期結束。
已取消	接受者選擇在協議結束日期之前終止協議。
已終止	由於 AWS 啟動的終止事件，例如付款失敗，協議在其預先同意的結束日期之前結束。
已續約	協議已使用自動續約等功能續約為新協議。
已取代	協議已使用替代優惠取代。

第 4 節：客戶指標

儀表板的本節提供指定日期範圍的客戶趨勢。KPIs包括作用中的客戶數量和新的付費客戶數量。

- 每月趨勢 – 根據圖表中的篩選條件，提供作用中客戶數量和新付費客戶數量的每月檢視。
- 客戶指標 – 根據選擇的指標篩選條件，提供作用中客戶數量和新付費客戶數量的資料。您可以選取客戶客群或產業，以了解每個客群如何對總客戶指標做出貢獻。

> **Note**
>
> 協議具有指定狀態。如需狀態的相關資訊，請參閱上一節中的表格。

下表列出並說明指標。

指標	描述
公開優惠協議	識別為具有公開優惠可見性之協議 IDs 的協議數量；協議長度至少與所選日期範圍重疊一天；協議開始日期與協議結束日期之間會識別長度。
私有優惠協議	識別為具有私有優惠可見性之協議 IDs 的協議數量；協議長度至少與所選日期範圍重疊一天；協議開始日期與協議結束日期之間會識別長度。
新的公開優惠協議	識別為具有公開優惠可見性之協議 ID 的協議數量；協議接受日期落在選取的日期範圍內。
新的私有優惠協議	識別為具有私有優惠可見性之協議 IDs 的協議數量；協議接受日期落在選取的日期範圍內。
作用中的客戶	識別為至少具有 1 個作用中協議的訂閱者 AWS 帳戶 IDs 的客戶數目；協議長度至少與選取的日期範圍重疊一天
新的付費客戶	在所選日期範圍內，識別為第一個計費月份的訂閱者 AWS 帳戶 IDs 的客戶數量。

如需協議和收入的詳細資訊，請參閱本節中的 計費收入儀表板協議和續約儀表板和 。

列出效能儀表板

清單效能儀表板提供 AWS Marketplace 清單的概觀和詳細資訊。儀表板提供流量資料，以及客戶訂閱產品所採取的步驟。儀表板也提供依行銷管道的詳細流量明細。

> **Note**
>
> - 若要解除鎖定此儀表板，您必須註冊 AWS Marketplace Seller Prime 計畫。
> - 若要開啟此儀表板，請登入 AWS Marketplace 管理入口網站，選擇洞見、行銷，然後選擇列出效能索引標籤。

如需使用 AWS Marketplace 儀表板的詳細資訊，請參閱本節賣方儀表板稍早的 。

主題

- 第 1 節：篩選條件
- 第 2 節：日期篩選條件深入探討
- 第 3 節：摘要指標
- 第 4 節：每月趨勢和漏斗轉換
- 第 5 節：依頻道的流量趨勢
- 第 6 節：每月流量和協議指標
- 第 7 節：唯一訪客的 Web 流量來源
- 第 8 節：Web 流量來源明細

第 1 節：篩選條件

您可以使用下列篩選條件來精簡您的資料。

控制項名稱	描述
產品標題	產品的標題。
日期	包括過去 30、60 和 90 天、過去 12 個月 (TTM) 和年初至今 (YTD)。您可以選擇自訂來定義特定的開始和結束日期。

如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的在 QuickSight 上篩選資料。 QuickSight

第 2 節：日期篩選條件深入探討

此日期篩選條件適用於列出效能儀表板上的所有指標。

例如，當您使用預設year-to-date篩選條件值時，會顯示下列指標：

指標	描述
Unqiue 訪客	在 YTD 中造訪 AWS Marketplace 清單頁面的唯一個人數目。
新的公開優惠協議	在 YTD 內至少有 1 個有效日的公開優惠可見性協議數量。
新的唯一訪客	在 YTD 中第一次造訪 AWS Marketplace 列出頁面的唯一個人數目。
傳回唯一訪客	曾造訪 AWS Marketplace 列出頁面的唯一個人數量，這些頁面先前在所選年份開始之前曾造訪過，並在 YTD 中再次造訪過。

第 3 節：摘要指標

儀表板的此區段會顯示流量和公開優惠協議的摘要視覺化。關鍵績效指標 (KPIs) 包括唯一訪客的數量、新公開優惠協議的數量、新唯一訪客的數量，以及傳回唯一訪客的數量。您可以查看磁碟區和百分比的year-over-year或period-over-period變更。您可以透過更新篩選條件區段中日期欄位中的日期條件來更新日期範圍。

指標	描述
唯一訪客	已造訪 AWS Marketplace 清單頁面的唯一個人數目。
協議	提案者（產品或服務擁有者）與接受者（客戶）之間簽訂的合約，以開始使用產品。
新的公開優惠協議	公開優惠為優惠可見性，且接受日期在所選日期範圍內的協議數量。 包含作用中、過期、取消和終止的狀態。
狀態	作用中 - 部分或所有協議條款均有效。已過期 - 協議在其預先同意的結束日期結束。已取消 - 接受者選擇在其結束日期之前終止協議。已終止 - 由於 AWS 啟動的終止事件，例如付款失敗，合約在其預先同意的結束日期之前

指標	描述
	終止。已續約 - 協議已使用自動續約等功能續約為新協議。已取代 - 已使用替代優惠取代協議。

第 4 節：每月趨勢和漏斗轉換

儀表板的本節提供指定日期範圍的流量和協議趨勢。關鍵績效指標包括唯一訪客數量和新協議數量，參考新的公開優惠協議。每月唯一訪客 - 提供唯一訪客數量的每月檢視。每月唯一訪客的總數可以超過摘要指標區段中唯一訪客的總數，因為一個唯一訪客可以造訪幾個月。每月新協議 - 提供新公開優惠協議數量的每月檢視。Funnel 轉換 - 提供step-by-step轉換檢視。其中包括列出頁面、採購頁面、組態頁面和履行頁面。您可以選取唯一訪客、頁面檢視和唯一帳戶，如同長條所示。當您將滑鼠游標暫留在每個長條上時，您可以看到定義為目前長條值的轉換百分比，以先前長條的百分比表示。

指標	描述
唯一訪客	已造訪 AWS Marketplace 清單頁面的唯一個人數目。
頁面檢視	Marketplace AWS 清單頁面的造訪次數。
唯一帳戶	與訪客相關聯的唯一 AWS 帳戶 ID 數目。此指標僅適用於訪客驗證之後。

第 5 節：依頻道的流量趨勢

儀表板的這個區段可讓您深入了解每月流量趨勢。您可以選擇唯一訪客和頁面檢視做為長條顯示的內容。您可以在頻道篩選條件中選取單一頻道或多個頻道，以查看所選頻道的每月趨勢（每月）。

第 6 節：每月流量和協議指標

儀表板的這個區段為唯一訪客、新協議、轉換率和總協議提供每月檢視。轉換率定義為新的協議除以每月的唯一訪客。

資料表中的資料代表整個月份，即使您選取部分月份的日期範圍也一樣。例如，如果您篩選資料以顯示 2024 年 1 月 15 日至 2024 年 3 月 15 日期間，資料表仍會包含 2024 年 1 月 1 日至 2024 年 3 月 31 日的完整每月資料。

資料表中的資料會依月份排序，最近一個月會先顯示。您可以選擇按任何可用的資料欄排序資料表，方法是按一下對應的資料欄標頭，並使用 選項以遞增或遞減順序排序資料。

第 7 節：唯一訪客的 Web 流量來源

儀表板的此區段提供依行銷來源深入了解唯一訪客的詳細資訊。這些類別包括賣方有機探索、賣方領導的行銷活動、 AWS 有機探索和 AWS 領導的行銷活動。這些類別是透過與訪客或造訪相關聯的追蹤代碼來識別。

流量來源	描述
賣方有機探索	透過呼叫 網站上的動作和探索點或應用程式內通知抵達清單的唯一訪客。
賣方主導的行銷活動	透過付費廣告或促銷管道抵達清單的唯一訪客。
AWS 有機探索	透過 AWS 管道抵達清單的唯一訪客，例如有機 AWS 網站流量、 AWS console(s) 搜尋和 AWS Marketplace 搜尋。
AWS 行銷活動	透過 AWS 領導的行銷活動和付費促銷活動抵達清單的唯一訪客。

第 8 節：Web 流量來源明細

Web 流量來源明細表提供驅動流量之行銷活動的精細明細。

- 流量來源 - AWS包含領導和賣方領導為兩個主要類別。
- 流量類型 - 包括賣方有機探索、賣方主導的行銷活動、 AWS 有機探索和主導 AWS的行銷活動。
- 管道 - 包括各種行銷策略，包括付費顯示、付費搜尋、付費社交媒體、電子郵件、有機、內部網站流量外部連結和其他。
- 發佈者 - 指發佈行銷活動的實體，包括 Facebook、Google、LinkedIn AWS、Seller-led. 等。
- 提升 - 指已產生專用追蹤碼的特定行銷活動。範例可以是 `psm_linked_post_free-trial-global-ver-a`。

指標	描述
歸因協議	每次提升造訪所貢獻的協議總數。
轉換率	歸因協議除以每次提升的唯一訪客。

搜尋效能儀表板

搜尋效能儀表板提供有關您的 AWS Marketplace 清單在 AWS Marketplace 搜尋體驗中如何執行的指標。您可以使用儀表板來改善清單的可探索性和效能。儀表板會追蹤印模、點擊和點擊率的數量。它也會追蹤您提供和客戶使用的熱門關鍵字，並根據常見的客戶搜尋提供關鍵字建議。

儀表板的洞見可協助您精簡產品清單，以改善可見性、將關鍵字與客戶偏好設定配對，以及將更符合資格的流量推向您的企業 AWS Marketplace。

若要開啟儀表板，請啟動 AWS Marketplace Management Portal 並前往 Insights 索引標籤。

如需控制儀表板存取的詳細資訊，請參閱本節賣方儀表板稍早的 。

第 1 節：存取儀表板

若要解除鎖定此儀表板，您必須是 AWS Marketplace Seller Prime 計畫的成員。

註冊 Seller Prime 計畫後，賣方帳戶的系統管理員可以使用 中的 Insights 索引標籤 AWS Marketplace 管理入口網站 來啟動搜尋效能儀表板。

系統管理員也可以建立 AWS Identity and Access Management (IAM) 政策，為賣方公司中的其他使用者提供特定儀表板的存取權。

> **Note**
>
> 自 2023 年 9 月起，我們不再支援存取由舊版 IAM 許可啟用的賣方儀表板。使用新的 Amazon Resource Name (ARN) 格式，如下列範例所示來更新您的 IAM 許可。

如需建立政策的詳細資訊，請參閱AWS Identity and Access Management 《 使用者指南》中的建立 IAM 政策。

使用 政策來控制存取

使用下列其中一個政策來提供清單效能儀表板的存取權。

下列範例提供所有目前和未來 AWS Marketplace 資源的存取權，包括儀表板和報告，無論目前或未來的資料饋送。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

下列範例透過包含其 ARN 來提供搜尋效能儀表板的存取權。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/SearchPerformance_V1"
  }]
}
```

第 2 節：篩選條件

儀表板的本節提供下列篩選條件。使用它們來精簡您的資料。

控制項名稱	描述
產品標題	產品的標題。

控制項名稱	描述
	此篩選條件會影響印模、點擊、點擊率和搜尋關鍵字建議。它不會影響依產品類別資料表的熱門搜尋關鍵字。
日期篩選條件	您可以篩選過去 30、60 和 90 天、尾隨 12 個月 (TTM)，以及年初至今 (YTD)。您可以選擇自訂以使用特定的開始和結束日期。 此篩選條件會影響印模、點擊和點擊率。它不會影響AWS Marketplace 搜尋關鍵字建議或依產品類別資料表排名前列的搜尋關鍵字。

如需篩選的詳細資訊，請參閱《Amazon QuickSight 使用者指南》中的在 Amazon QuickSight 上篩選資料。 Amazon QuickSight

第 3 節： AWS Marketplace 搜尋效能

儀表板的這個區段會顯示有關 AWS Marketplace 清單可探索性和效能的關鍵效能指標 (KPIs)。儀表板提供印模數量、點擊次數和點擊率的 KPIs。

您可以看到磁碟區和百分比的year-over-year或period-over-period變更。您也可以變更日期範圍。

指標	描述
印象	您的清單出現在前 20 個搜尋結果中的次數 AWS Marketplace。
點按數	使用者從 AWS Marketplace 搜尋結果中按一下清單的次數。
點擊率 (CTR)	點按次數與印模次數之間的比率。衡量您的清單在從搜尋結果吸引客戶方面的效果。

第 4 節：曝光、點擊和點擊率的每月趨勢

儀表板的本節提供指定日期範圍的每月趨勢。它提供印模次數、點擊次數和點擊率的 KPIs。

第 5 節：AWS Marketplace 搜尋關鍵字建議

儀表板的本節提供列出您目前關鍵字的資料表，以及每個清單的建議關鍵字。

欄位名稱	描述
產品標題	產品的標題。 連結會帶您前往 AWS Marketplace Management Portal 中的產品概觀頁面，您可以在其中管理和編輯您的清單。
目前的關鍵字	您在 AWS Marketplace Management Portal 的更新產品資訊區段中提供的關鍵字。
效能最佳的關鍵字	出現在前五名客戶的關鍵字會搜尋您的產品。
AWS-建議的關鍵字	客戶在搜尋類似產品時最常使用的關鍵字。
預估流量增加 %	如果您採用建議的關鍵字，潛在的流量增益。不受日期篩選條件影響。

Note

雖然關鍵字很重要，但其他因素可能會影響搜尋結果。

- 清單最佳化
  - 最佳化清單的其他部分，例如簡短和長描述。
  - 請考慮提供免費試用等功能，以增強您清單的吸引力。
- 將關鍵字與清單和目標客戶配對
  - 盡可能將清單與常用關鍵字配對。
  - 僅使用與您的產品及其功能相關的關鍵字。否則，您的清單可能會出現在更多搜尋中，但使用者將選擇較少。
- 了解關鍵字變化
  - 請記住，類似術語，例如「git」和「gitops」可以視為相同的搜尋術語。
  - 評估任何變化的相關性，並相應地調整關鍵字。
- 更新關鍵字

  1. 在 中 AWS Marketplace 管理入口網站，開啟產品選單，然後選取產品類別。
  2. 選擇您產品的連結，選擇產品資訊索引標籤，然後導覽至提供產品資訊頁面。
  3. 向下捲動至可AWS Marketplace 探索性區段，尋找 AWS Marketplace 搜尋結果的關鍵字，並調整您的關鍵字。

第 6 節：依產品類別排名前列的搜尋關鍵字

每個產品類別中與您的相似產品的前五個客戶搜尋關鍵字。此儀表板上的其他產品和日期篩選條件不會影響此資料。

了解使用 AWS 儀表板購買

使用儀表板購買 AWS提供 Web 流量、業務開發和協議的概觀，這些客戶會使用call-to-action按鈕選擇您的購買 AWS，並在其上造訪您的使用採購購買 AWS頁面 AWS Marketplace。

如需有關存取儀表板的資訊，請參閱本指南賣方儀表板稍早的 。

主題

- 第 1 節：篩選條件
- 第 2 節：摘要指標
- 第 3 節：頁面檢視和協議總數的點選和趨勢資料總數
- 第 4 節：AWS 登入頁面捨棄率
- 第 5 節：購買與 AWS 採購頁面上的參與
- 第 6 節：購買 AWS 採購頁面上的參與指標趨勢資料

第 1 節：篩選條件

本節提供下列資料篩選條件。

篩選器名稱	描述
產品標題	產品的標題。
產品 ID	軟體產品的易記唯一識別符。
日期篩選條件	依過去 30、60 或 90 天、過去 12 個月或年初至今進行篩選。選擇自訂以輸入其他開始和結束日期。

如需篩選的詳細資訊，請參閱 QuickSight 使用者指南中的在 QuickSight 上篩選資料。 QuickSight

第 2 節：摘要指標

本節顯示使用call-to-action按鈕購買 AWS和使用採購購買 AWS頁面的相關資訊。

下表列出並說明指標。若要更新日期範圍，請在篩選條件區段中變更日期條件。

指標	描述
點按按鈕總數	賣方網站上的「使用 AWS 購買」等call-to-action按鈕的點選總數。
具有 AWS 頁面檢視的購買數量	使用採購購買 AWS頁面的造訪總數。這包括重複造訪。
購買與 AWS 頁面的協議總數	在採購購買 AWS頁面上建立的協議總數。

第 3 節：頁面檢視和協議總數的點選和趨勢資料總數

本節提供指定日期範圍的每月或每週趨勢。下表列出並說明 KPIs：

指標	描述
點按按鈕總數	賣方網站上的「call-to-action」按鈕的點選總數。
頁面檢視	總頁面會造訪購買 AWS 採購頁面。使用依清單顯示資料，在每月和每週檢視之間進行選擇。
協議總數	在購買 AWS 採購頁面上建立的協議總數。使用依清單顯示資料，在每月和每週檢視之間進行選擇。

第 4 節：AWS 登入頁面捨棄率

本節顯示 AWS 登入頁面的捨棄率。 費率是登入和建立協議的使用者與僅登入的使用者人數的比率。使用依清單顯示資料，依週或月進行篩選。

第 5 節：購買與 AWS 採購頁面上的參與

本節提供指定日期範圍的摘要指標和趨勢資料。下表列出並說明指標。使用篩選條件區段中的日期欄位來更新日期範圍。

指標	描述
唯一訪客的總數	唯一採購頁面使用者的總數。
退信率	檢視一個網站頁面後登入和離開的使用者與造訪多個頁面的使用者的比例。
平均使用者停留時間，以分鐘為單位	使用者登入到該使用者離開頁面之間的時間。

第 6 節：購買 AWS 採購頁面上的參與指標趨勢資料

圖表顯示第 5 節中指標的每月或每週趨勢。使用依清單顯示資料，在每月和每週檢視之間進行選擇。

AWS Marketplace 廠商洞見

AWS Marketplace Vendor Insights 是一種功能，可簡化組織執行的軟體風險評估，以保護他們信任且符合其標準的採購軟體。透過 AWS Marketplace Vendor Insights，買方可以近乎即時地從單一主控台監控產品的安全描述檔。 AWS Marketplace Vendor Insights 可以簡化買方的採購程序，並可能提高賣方的銷售額。它提供軟體產品安全和合規資訊的儀表板，以減少買方的評估工作。

AWS Marketplace Vendor Insights 儀表板中的所有安全與合規資訊，都是以從下列來源收集的證據為基礎：

- 賣方自我鑑證，包括 AWS Marketplace Vendor Insights 安全性自我評定和共識評估計畫問卷 (CAIQ)
- 產業標準稽核報告 （ 例如，國際標準化組織 ISO 27001)
- AWS Audit Manager，可自動從賣方的生產環境收集證據

AWS Marketplace Vendor Insights 會收集有關產品的合規成品和安全控制資訊，並將其呈現在儀表板中。儀表板會從賣方的自我評估、稽核報告的證據，以及來自 的即時證據中取得資料 AWS 帳戶。此資料會饋送至安全控制，然後饋送至儀表板供買方檢閱。即時證據是持續更新來自多個來源的資料以呈現最新資訊的方法。 AWS Config 已在賣方環境中啟用。有關組態、啟用備份和其他資訊的資料會自動更新。例如，假設產品的存取控制符合規範，且 Amazon Simple Storage Service (Amazon S3) 儲存貯體已公開。儀表板會顯示控制項的狀態從合規變更為不確定。

您必須先在 中設定基準資源和基礎設施， AWS 帳戶 才能使用 AWS Marketplace Vendor Insights。設定完成後， AWS Marketplace Vendor Insights 可以收集資訊，並產生軟體即服務 (SaaS) 產品的安全設定檔 AWS Marketplace。

目錄

- 了解 AWS Marketplace 廠商洞見
- 設定 AWS Marketplace 廠商洞見
- 檢視您的 AWS Marketplace Vendor Insights 設定檔
- 在 AWS Marketplace 供應商洞察中管理快照
- 在 AWS Marketplace Vendor Insights 中控制存取權

了解 AWS Marketplace 廠商洞見

AWS Marketplace Vendor Insights 會收集您產品的合規成品和安全控制資訊，並將其呈現在儀表板中。儀表板會從產品擁有者的自我評定、稽核報告的證據，以及即時證據中擷取資料 AWS 帳戶。此資料會饋送至安全控制，然後饋送至儀表板供買方檢閱。

儀表板會顯示由 AWS Marketplace Vendor Insights 從多個安全控制類別收集的證據型資訊。這可提供近乎即時的安全性描述檔檢視的洞見，並減少買方和賣方之間的討論。買方可以在幾個小時內驗證賣方的資訊是否完成評估。 AWS Marketplace Vendor Insights 為賣方提供一種機制，讓安全和合規狀態資訊自動保持在up-to-date。他們可以隨需與買方共用，無需隨機回應問卷。

AWS Marketplace Vendor Insights 會從三個來源收集以證據為基礎的資訊：

- 您的廠商自我評估 – 支援的自我評估包括 AWS Marketplace Vendor Insights 安全性自我評估和共識評估計畫問卷 (CAIQ)。
- 您的生產帳戶 – 在多個控制項中，25 個控制項支援從您的生產帳戶收集即時證據。每個控制項的即時證據是透過使用一或多個 AWS Config 規則來評估 AWS 資源的組態設定來產生。 會 AWS Audit Manager 擷取證據並準備供 AWS Marketplace Vendor Insights 使用。加入 AWS CloudFormation 範本會自動化啟用即時證據收集所需的必要步驟。 AWS Config 在賣方環境中已啟用 。有關組態、啟用備份和其他資訊的資料會自動更新。例如，假設產品的存取控制符合規範，且 Amazon S3 儲存貯體已公開。儀表板會顯示控制項的狀態從合規變更為不確定。
  - 開啟 AWS Config 和服務 AWS Audit Manager 。
  - 建立 AWS Config 規則和 AWS Audit Manager 自動評估。
  - 佈建 AWS Identity and Access Management (IAM) 角色，讓 AWS Marketplace Vendor Insights 可以提取評估結果。
- 您的 ISO 27001 和 SOC2 Type II 報告 – 控制項類別會映射到國際標準化組織 (ISO) 和系統和組織控制 (SOC2) 報告中的控制項。當您與 AWS Marketplace Vendor Insights 共用這些報告時，它可以從這些報告中擷取相關證據，並在儀表板上呈現。

設定 AWS Marketplace 廠商洞見

下列程序說明在 AWS Marketplace 軟體即服務 (SaaS) 清單上設定 AWS Marketplace Vendor Insights 的高階步驟。

在 SaaS 清單上設定 AWS Marketplace Vendor Insights

1. the section called “建立安全性描述檔”.

2. （選用）the section called “上傳憑證”。
3. the section called “上傳自我評定”.
4. （選用）the section called “啟用 AWS Audit Manager 自動化評估”。

建立安全性描述檔

安全設定檔可讓您的買方詳細了解軟體產品的安全狀態。安全性描述檔使用相關聯的資料來源，包括自我評估、認證和 AWS Audit Manager 自動化評估。

> **Note**
>
> 您可以建立有限數量的安全性設定檔。若要建立更多安全性設定檔，請請求提高配額。如需詳細資訊，請參閱 中的AWS 服務配額AWS 一般參考。

建立安全性設定檔

1. 使用可存取 AWS Marketplace 賣方帳戶的 IAM 使用者或角色登入。
2. 選擇產品，然後選取 SaaS 以導覽至 SaaS 產品頁面。
3. 選擇產品。
4. 選擇廠商洞見索引標籤，然後選擇聯絡支援以新增安全性設定檔。
5. 完成表單，然後選擇提交。

   Seller AWS Marketplace Operations 團隊將建立安全性設定檔。當安全性設定檔準備就緒時，他們將傳送通知電子郵件訊息給表單上識別的收件人。

上傳憑證

認證是一種資料來源，可在多個維度提供產品安全狀態的證據。 AWS Marketplace Vendor Insights 支援下列認證：

- FedRAMP 認證 – 驗證是否符合美國政府雲端安全標準
- GDPR 合規報告 – 示範遵循一般資料保護法規 (GDPR) 要求，保護個人資料和個人隱私權
- HIPAA 合規報告 – 示範遵循健康保險流通與責任法案 (HIPAA) 法規，以保護受保護的健康資訊
- ISO/IEC 27001 稽核報告 – 確認符合國際標準化組織 (ISO)/國際電工委員會 (IEC) 27001，強調資訊安全標準

- PCI DSS 稽核報告 – 示範符合 PCI 安全標準委員會設定的安全標準
- SOC 2 類型 2 稽核報告 – 確認符合服務組織控制 (SOC) 資料隱私權和安全控制

上傳憑證

1. 在 Vendor Insights 索引標籤上，導覽至資料來源區段。
2. 在憑證下，選擇上傳憑證。
3. 在憑證詳細資訊下，提供請求的資訊並上傳憑證。
4. （選用）在標籤下，新增標籤。

> **Note**
>
> 如需標籤的相關資訊，請參閱《標記 AWS 資源使用者指南》中的標記您的 AWS 資源。

5. 選擇上傳憑證。

> **Note**
>
> 認證會自動與目前的安全性設定檔建立關聯。您也可以關聯已上傳的憑證。在產品詳細資訊頁面上，選擇憑證下的關聯憑證，從清單中選擇憑證，然後選擇關聯憑證。
> 上傳認證之後，您可以使用產品詳細資訊頁面上的下載認證按鈕進行下載。您也可以使用更新認證按鈕來更新認證詳細資訊。

認證狀態會變更為 ValidationPending，直到驗證認證詳細資訊為止。處理資料來源期間和之後會出現替代狀態：

- 可用 – 資料來源已上傳且系統驗證已成功完成。
- AccessDenied – 資料來源的外部來源參考不再可供 AWS Marketplace Vendor Insights 讀取。
- ResourceNotFound – 資料來源的外部來源參考不再可供 VendorInsights 讀取。
- ResourceNotSupported – 已上傳資料來源，但尚未支援提供的來源。如需驗證錯誤的詳細資訊，請參閱 狀態訊息。
- ValidationPending – 資料來源已上傳，但系統驗證仍在執行中。在此階段，您沒有任何動作項目。狀態已更新為可用、ResourceNotSupported 或 ValidationFailed。
- ValidationFailed – 資料來源已上傳，但系統驗證因一或多個原因而失敗。如需驗證錯誤的詳細資訊，請參閱 狀態訊息。

上傳自我評定

自我評定是一種資料來源，可提供產品安全狀態的證據。 AWS Marketplace Vendor Insights 支援下列自我評定：

- AWS Marketplace Vendor Insights 自我評定
- 共識評估計畫問卷 (CAIQ)。如需詳細資訊，請參閱 Cloud Security Alliance 網站上的什麼是 CAIQ。

上傳自我評定

1. 在 https：//https://console.aws.amazon.com/marketplace 開啟 AWS Marketplace 主控台。
2. 在 Vendor Insights 索引標籤上，導覽至資料來源區段。
3. 在自我評定下，選擇上傳自我評定。
4. 在自我評定詳細資訊下，完成下列資訊：
   a. 名稱 – 輸入自我評定的名稱。
   b. 類型 – 從清單中選擇評估類型。

   > **Note**
   > 如果您選擇 Vendor Insights 安全性自我評定，請選擇下載範本以下載自我評定。為試算表中的每個答案選擇是、否或不適用。

5. 若要上傳已完成的評估，請選擇上傳自我評定。
6. （選用）在標籤下，新增標籤。

   > **Note**
   > 如需標籤的相關資訊，請參閱《標記 AWS 資源使用者指南》中的標記您的 AWS 資源。

7. 選擇上傳自我評定。

   > **Note**
   > 自我評定會自動與目前的安全性設定檔建立關聯。您也可以將已上傳的自我評定建立關聯。在產品詳細資訊頁面上，選擇在自我評定下關聯自我評定，從清單中選擇自我評定，然後選擇關聯自我評定。

上傳自我評定之後，您可以使用產品詳細資訊頁面上的下載自我評定按鈕進行下載。您也可以使用更新自我評定按鈕來更新自我評定詳細資訊。

狀態已更新為下列其中一項：

- 可用 – 資料來源已上傳且系統驗證已成功完成。
- AccessDenied – 資料來源的外部來源參考不再可供 VendorInsights 讀取。
- ResourceNotFound – 資料來源的外部來源參考不再可供 VendorInsights 讀取。
- ResourceNotSupported – 已上傳資料來源，但尚未支援提供的來源。如需驗證錯誤的詳細資訊，請參閱 狀態訊息。
- ValidationPending – 資料來源已上傳，但系統驗證仍在執行中。在此階段，您沒有任何動作項目。狀態已更新為可用、ResourceNotSupported 或 ValidationFailed。
- ValidationFailed – 資料來源已上傳，但系統驗證因一或多個原因而失敗。如需驗證錯誤的詳細資訊，請參閱 狀態訊息。

啟用 AWS Audit Manager 自動化評估

AWS Marketplace Vendor Insights 使用多個 AWS 服務 自動收集安全性設定檔的證據。

您需要下列 AWS 服務 和資源才能進行自動化評估：

- AWS Audit Manager – 為了簡化 AWS Marketplace Vendor Insights 設定，我們使用 AWS CloudFormation Stacks 和 StackSets，負責佈建和設定必要的資源。堆疊集會建立自動評估，其中包含由 自動填入的控制項 AWS Config。

  如需 的詳細資訊 AWS Audit Manager，請參閱AWS Audit Manager 《 使用者指南》 。
- AWS Config – 堆疊集會 AWS Config 部署一致性套件，以設定必要的 AWS Config 規則。這些規則允許 Audit Manager 自動化評估為該 中 AWS 服務 部署的其他 收集即時證據 AWS 帳戶。如需 AWS Config 功能的詳細資訊，請參閱 AWS Config 開發人員指南。

> **Note**
>
> 與後續月份 AWS Config 相比，您可能會在記錄 的初始月份注意到帳戶中的活動增加。在初始引導程序期間， 會 AWS Config 檢閱您帳戶中您選取要記錄 AWS Config 的所有資源。

如果您執行暫時性工作負載，您可能會看到 的活動增加， AWS Config 因為它會記錄與建立和刪除這些臨時資源相關的組態變更。暫時性工作負載是暫時使用因需要而載入和執行的運算資源。
暫時性工作負載的範例包括 Amazon Elastic Compute Cloud (Amazon EC2) Spot 執行個體 AWS Auto Scaling、Amazon EMR 任務和 AWS Lambda。為了避免增加執行暫時性工作負載的活動，您可以在 AWS Config 關閉的個別帳戶中執行這些類型的工作負載。此方法可避免增加組態記錄和規則評估。

- Amazon S3 – 堆疊集會建立下列兩個 Amazon Simple Storage Service 儲存貯體：
  - vendor-insights-stack-set-output-bucket-{account number} – 此儲存貯體包含來自堆疊集執行的輸出。Seller AWS Marketplace Operations 團隊會使用輸出來完成您的自動資料來源建立程序。
  - vendor-insights-assessment-reports-bucket-{account number} – AWS Audit Manager 發佈評估報告到此 Amazon S3 儲存貯體。如需發佈評估報告的詳細資訊，請參閱AWS Audit Manager 《 使用者指南》中的評估報告。

    如需 Amazon S3 功能的詳細資訊，請參閱《Amazon S3 使用者指南》。
- IAM – 加入堆疊集會在您的帳戶中佈建下列 AWS Identity and Access Management (IAM) 角色：
  - 部署`VendorInsightsPrerequisiteCFT.yml`範本時，它會建立管理員角色`AWSVendorInsightsOnboardingStackSetsAdmin`和執行角色`AWSVendorInsightsOnboardingStackSetsExecution`。堆疊集使用管理員角色，將所需的堆疊同時部署到多個 AWS 區域 堆疊。管理員角色會擔任執行角色，在 AWS Marketplace Vendor Insights 設定程序中部署必要的父堆疊和巢狀堆疊。如需自我管理許可的詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的授予自我管理許可。
  - 此`AWSVendorInsightsRole`角色提供 AWS Marketplace Vendor Insights 讀取 AWS Audit Manager 資源中評估的存取權。 AWS Marketplace Vendor Insights 會在您的 AWS Marketplace Vendor Insights 設定檔上的評估上顯示證據。
  - 為 AWS Marketplace Vendor Insights `AWSVendorInsightsOnboardingDelegationRole`提供列出和讀取`vendor-insights-stack-set-output-bucket`儲存貯體中物件的存取權。此功能可讓 AWS Marketplace 目錄操作團隊協助您設定 AWS Marketplace Vendor Insights 設定檔。
  - 此`AWSAuditManagerAdministratorAccess`角色提供管理存取權，以啟用或停用 AWS Audit Manager、更新設定和管理評估、控制項和架構。您或您的團隊可以擔任此角色來採取自動化評估的動作 AWS Audit Manager。

若要啟用 AWS Audit Manager 自動評估，您必須部署加入堆疊。

部署加入堆疊

為了簡化 AWS Marketplace Vendor Insights 設定，我們使用 AWS CloudFormation Stacks 和 StackSets，負責佈建和設定必要的資源。如果您有多個帳戶或多個 AWS 區域 SaaS 解決方案，StackSets 可讓您從中央管理帳戶部署入門堆疊。

如需 CloudFormation StackSets 的詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的使用 AWS CloudFormation StackSets。

AWS Marketplace Vendor Insights 設定要求您使用下列 CloudFormation 範本：

- `VendorInsightsPrerequisiteCFT` – 設定必要的管理員角色和許可，以在您的帳戶中執行 CloudFormation StackSets。在您的賣方帳戶中建立此堆疊。
- `VendorInsightsOnboardingCFT` – 設定所需的 AWS 服務 ，並設定適當的 IAM 許可。這些許可允許 AWS Marketplace Vendor Insights 收集在您 中執行之 SaaS 產品的資料， AWS 帳戶 並在您的 AWS Marketplace Vendor Insights 設定檔上顯示資料。在透過 StackSets 託管 SaaS 解決方案的賣方帳戶和生產帳戶中建立此堆疊。

建立 VendorInsightsPrerequisiteCFT 堆疊。

透過執行 `VendorInsightsPrerequisiteCFT` CloudFormation 堆疊，您可以設定 IAM 許可來開始加入堆疊集。

建立VendorInsightsPrerequisiteCFT堆疊

1. 檢閱並從 GitHub 網站上的 AWS Vendor Insights 範本範例儲存庫資料夾下載最新的`VendorInsightsPrerequisiteCFT.yml`檔案。
2. AWS Management Console 使用您的 AWS Marketplace 賣方帳戶登入 ，然後開啟位於 https：//https://console.aws.amazon.com/cloudformation 的 AWS CloudFormation 主控台。
3. 在 CloudFormation 主控台導覽窗格中，選擇堆疊，然後從下拉式清單中選擇建立堆疊和使用新資源 （ 標準 ）。（ 如果看不到導覽窗格，請在左上角選取並展開導覽窗格。）
4. 在指定範本下，選擇上傳範本檔案。若要上傳您下載`VendorInsightsPrerequisiteCFT.yml`的檔案，請使用選擇檔案。然後選擇下一步。
5. 輸入堆疊的名稱，然後選擇下一步。
6. （ 選用 ）視需要設定堆疊選項。

   選擇下一步。

7. 在 Review (檢閱) 頁面上，檢視您的選擇。若要進行變更，請在您要變更的區域中選擇編輯。您必須先選取功能區域中的確認核取方塊，才能建立堆疊。

   選擇提交。

8. 建立堆疊之後，請選擇資源索引標籤，並記下建立的下列角色：

   - `AWSVendorInsightsOnboardingStackSetsAdmin`
   - `AWSVendorInsightsOnboardingStackSetsExecution`

建立VendorInsightsOnboardingCFT堆疊集

透過執行 `VendorInsightsOnboardingCFT` CloudFormation 堆疊集，您可以設定所需的 AWS 服務並設定適當的 IAM 許可。這可讓 AWS Marketplace Vendor Insights 收集在您 中執行之 SaaS 產品的資料， AWS 帳戶 並將其顯示在您的 AWS Marketplace Vendor Insights 設定檔中。

如果您有多個帳戶解決方案，或如果您有單獨的賣方和生產帳戶，則必須在多個帳戶中部署此堆疊。StackSets 可讓您從建立先決條件堆疊的管理帳戶執行此操作。

堆疊集是使用自我管理許可進行部署。如需詳細資訊，請參閱AWS CloudFormation 使用者指南中的建立具有自我管理許可的堆疊集。

建立VendorInsightsOnboardingCFT堆疊集

1. 檢閱並從 GitHub 網站上的 AWS Vendor Insights 範本範例儲存庫資料夾下載最新的`VendorInsightsOnboardingCFT.yml`檔案。
2. AWS Management Console 使用您的 AWS Marketplace 賣方帳戶登入 ，然後開啟位於 https：//https://console.aws.amazon.com/cloudformation 的 AWS CloudFormation 主控台。
3. 在 CloudFormation 主控台導覽窗格中，選擇建立 StackSet。（如果看不到導覽窗格，請在左上角選取並展開導覽窗格。）
4. 在許可下，針對管理員角色選擇 IAM 角色名稱，然後從下拉式清單中選擇角色名稱的 AWSVendorInsightsOnboardingStackSetsAdmin。
5. 輸入 **`AWSVendorInsightsOnboardingStackSetsExecution`**做為 IAM 執行角色名稱。
6. 在指定範本下，選擇上傳範本檔案。若要上傳您下載`VendorInsightsOnboardingCFT.yml`的檔案，請使用選擇檔案，然後選擇下一步。
7. 提供下列 StackSet 參數，然後選擇下一步。

- `CreateVendorInsightsAutomatedAssessment` – 此參數會在您的 中設定 AWS Audit Manager 自動評估 AWS 帳戶。如果您有個別的管理和生產帳戶，則應該僅針對生產帳戶選取此選項，而非針對管理帳戶選取此選項。
- `CreateVendorInsightsIAMRoles` – 此參數會佈建 IAM 角色，允許 AWS Marketplace Vendor Insights 讀取您 中的評估資料 AWS 帳戶。
- `PrimaryRegion` – 此參數會設定 SaaS 部署 AWS 區域 的主要參數。這是在您的 中建立 Amazon S3 儲存貯體的區域 AWS 帳戶。如果您的 SaaS 產品僅部署到一個區域，則該區域即為主要區域。

8. 視需要設定 StackSet 選項。將執行組態保持為非作用中，然後選擇下一步。
9. 設定部署選項。如果您有多個帳戶解決方案，您可以將堆疊集設定為在多個帳戶和區域中部署為單一操作。選擇下一步。

> **Note**
>
> 如果您有多個帳戶解決方案，我們不建議將 部署到所有帳戶做為單一堆疊集。請密切注意步驟 7 中定義的參數。視您要部署的帳戶類型而定，您可能想要啟用或停用某些參數。StackSets 會將相同的參數套用至單一部署中的所有指定帳戶。您可以透過分組堆疊集中的帳戶來縮短部署時間，但您仍然需要針對多個帳戶解決方案部署多次。

> **Important**
>
> 如果您要部署到多個區域，您列出的第一個區域必須是 `PrimaryRegion`。將區域並行選項保留為循序的預設設定。

10. 在 Review (檢閱) 頁面上，檢視您的選擇。若要進行變更，請在您要變更的區域中選擇編輯。您必須先選取功能區域中的確認核取方塊，才能建立堆疊集。

    選擇提交。

    堆疊集每個區域大約需要 5 分鐘才能完成。

檢視您的 AWS Marketplace Vendor Insights 設定檔

您在 AWS Marketplace Vendor Insights 中的設定檔提供重要資訊，供買方在評估您的產品時使用。基於資料保護目的，我們建議您保護您的 AWS 帳戶 登入資料，並使用 AWS Identity and Access

Management (IAM) 設定個別使用者。透過該方法，每個使用者只會獲得履行其任務職責所需的許可。如需使用 IAM 建立使用者的詳細資訊，請參閱the section called “建立或使用群組”。

買方可以使用 AWS Marketplace Vendor Insights 儀表板來評估您的產品。在那裡，買方可以看到由您新增到設定檔的資料來源定義的產品概觀。安全設定檔由 10 個類別中的多個安全控制項定義。

用於定義資料的 10 個安全類別如下所示：

- 存取管理
- 應用程式安全
- 稽核、合規和安全性政策
- 業務彈性
- 資料安全與隱私
- 最終使用者裝置和行動安全
- 人力資源
- 基礎架構安全
- 風險管理和事件回應
- 安全性和組態政策

如需詳細資訊，請參閱 AWS Marketplace 買方指南中的了解控制類別。

透過設定和使用 AWS Marketplace Vendor Insights，您同意遵守 AWS 服務 條款和資料隱私權規則，以保持使用者資訊的隱私和安全。如需 AWS 資料隱私權條款的詳細資訊，請參閱資料隱私權常見問答集。如需服務條款的詳細資訊，請參閱 AWS 服務 詞彙。

以賣方身分檢視您的安全設定檔

完成自我評定並新增其他即時證據後，請務必以賣方的身分檢視您的設定檔。您會想要檢閱新增至設定檔的資訊。

> **Note**
>
> 在您請求 AWS Marketplace Vendor Insights 支援團隊更新其可見性之前，買方看不到此設定檔。支援團隊完成更新後，簽署保密協議 (NDA) 的買方即可存取安全性設定檔。
> 如果您想要從 AWS Marketplace Vendor Insights 設定檔刪除訂閱者的個人識別資訊 (PII) 資料，請聯絡 來啟動支援案例支援。

在買方檢視時檢視您的安全設定檔

1. 登入 AWS Management Console。
2. 前往入口網站中的 SaaS 產品頁面。
3. 選擇具有相關聯安全設定檔的產品。
4. 選取 Vendor Insights 索引標籤，然後選擇檢視最新發行的快照。
5. 在概觀索引標籤上，會顯示您上傳的所有憑證徽章。
6. 選取安全性和合規索引標籤，您可以在其中檢視從多個控制項收集的資料。若要檢視更多詳細資訊，請選擇每個控制集。

在 AWS Marketplace 供應商洞察中管理快照

快照是安全性設定檔的 point-in-time 狀態。在 AWS Marketplace 供應商洞察中，您可以使用快照在任何給定時間評估賣家的產品。身為賣家，您可以比較個人檔案在不同時間的安全狀態，或是比較不同安全性設定檔的最新快照，以支援您的決策。除了提供有關新鮮度和資料來源的透明度之外，快照還提供必要的安全性資訊。

在 AWS Marketplace 控制台的 AWS Marketplace 供應商見解快照摘要部分中，您可以查看有關創建和發布計劃的以下快照詳細信息：

- 上次建立的快照 — 上次為此設定檔建立的快照。
- 下一次排程建立 — 排定接下來建立的快照。
- 建立頻率 — 建立快照之間的時間長度或建立快照的頻率。
- 下一個排程版本 — 快照排定在接下來發行。
- 暫存時間 — 快照會暫存至少這段時間長度，然後有資格在快照釋放事件期間釋放。
- 發行頻率 — 發行事件之間的時間長度。

在 [快照清單] 區段中，快照狀態如下：

- 已發行 — 快照為公開狀態，可供具有此產品權限的使用者檢視。
- 擱置發行 — 快照已完成或處於必要的最短暫存期間，並排定於下一個發行版本。
- 私人 — 在安全性設定檔啟用之前建立的快照，或發生驗證錯誤且公眾無法看見。私人快照僅保留在賣家可見性中。

主題

- 建立快照
- 檢視快照
- 匯出快照
- 檢視最新發行的快照
- 延後快照釋放
- 變更快照清單的偏好設定

建立快照

若要為您的設定檔建立快照，請依照下列步驟執行。您每天最多可以建立 20 個快照。

1. 登入 AWS Management Console 並開啟AWS Marketplace 主控台。
2. 選擇供應商洞察。
3. 從供應商洞察中選擇產品。
4. 在產品設定檔頁面上，移至 [快照] 清單，然後選擇 [建立新快照]。
5. 會出現一則訊息，通知您快照排程將會變更。選擇建立。

> **Note**
>
> 建立新快照時，快照排程會變更。新快照的排程時間與您手動建立的快照相同。此訊息包含新排程。

新快照會在 30 分鐘內建立，並新增至快照清單。新的快照會建立為擱置中的發行版本狀態。在狀態變更為「已發行」之前，任何人都無法檢視新的快照。

檢視快照

若要檢視設定檔的快照，請依照下列步驟執行。

1. 登入 AWS Management Console 並開啟AWS Marketplace 主控台。
2. 選擇供應商洞察。
3. 從供應商洞察中選擇產品。
4. 在產品設定檔頁面上，移至 [快照] 清單，然後選擇您要檢視之快照的快照 ID。
5. 完成後，請選擇 [上一步] 結束快照檢視。

匯出快照

您可以匯出為 JSON 格式或 CSV 格式。若要匯出快照，請依照下列步驟執行。

1. 登入 AWS Management Console 並開啟AWS Marketplace 主控台。
2. 選擇供應商洞察。
3. 從供應商洞察中選擇產品。
4. 在產品設定檔頁面上，移至 [快照] 清單，然後選擇您要匯出之快照的快照 ID。
5. 選擇 Export (匯出)。
6. 從下拉式清單中，選擇 [下載 (JSON)] 或 [下載 (CSV)]。

檢視最新發行的快照

最新發行的快照是使用者用來檢視和評估產品健全狀況的快照。了解最新發行的快照中有哪些內容非常重要，以確保使用準確的資訊描繪產品。若要檢視您設定檔的最新快照，請依照下列步驟執行。

1. 登入 AWS Management Console 並開啟AWS Marketplace 主控台。
2. 選擇供應商洞察。
3. 從供應商洞察中選擇產品。
4. 在產品設定檔頁面上，移至 [快照] 清單，然後選擇您要檢視之快照的快照 ID。
5. 選擇檢視最新發行的快照。
6. 完成後，請選擇 [上一步] 結束快照檢視。

延後快照釋放

若要延遲快照釋放至您的設定檔，您可以延後特定快照 ID 的快照釋放。

1. 登入 AWS Management Console 並開啟AWS Marketplace 主控台。
2. 選擇供應商洞察。
3. 從供應商洞察中選擇產品。
4. 在產品設定檔頁面上，移至 [快照] 清單，然後選擇您要延後釋放之快照的快照 ID。
5. 從 [快照摘要] 中，選擇 [延後快照釋放]。
6. 會出現一則訊息，通知您快照排程將會變更。選擇「延後」。

會出現一則成功訊息，指出您已成功延遲此產品的快照發行。

變更快照清單的偏好設定

建立快照後，您可以在 [快照] 清單中變更快照檢視方式的偏好設定。

1. 登入 AWS Management Console 並開啟AWS Marketplace 主控台。
2. 選擇供應商洞察。
3. 從供應商洞察中選擇產品。
4. 在產品設定檔頁面上，移至 [快照] 清單，然後選擇您要變更之快照的快照 ID。
5. 選擇偏好設定圖示。您可以為快照自訂下列偏好設定：
   - 頁面大小 — 選取您要在每個頁面上列出多少快照：10 個資源、20 個資源或每頁 50 個資源。
   - 換行 — 選取用來換行的選項以檢視整個記錄。
   - 時間格式 — 選取您需要絕對、相對或 ISO。
   - 可見欄 — 選取您想要顯示快照詳細資料的選項：快照 ID、狀態和建立日期。

在 AWS Marketplace Vendor Insights 中控制存取權

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助您控制對 AWS 資源的存取。IAM 是 AWS 服務 您可以免費使用的 。如果您是管理員，您可以控制誰可以驗證 （登入） 和授權 （具有許可） 來使用 AWS Marketplace 資源。 AWS Marketplace Vendor Insights 使用 IAM 來控制對賣方資料、評估、賣方自我鑑證和產業標準稽核報告的存取。

控制誰可以在 中執行動作的建議方法是 AWS Marketplace 管理入口網站 使用 IAM 來建立使用者和群組。然後將使用者新增至群組，並管理群組。您可以將政策或許可指派給提供唯讀許可的群組。如果您有其他需要唯讀存取權的使用者，您可以將他們新增至您建立的群組，而不是為使用者新增許可。

政策是定義套用至使用者、群組或角色之許可的文件。許可決定使用者可以在 中執行的操作 AWS。政策通常允許存取特定動作，並且可以選擇性地授予特定資源允許的動作，例如 Amazon EC2 執行個體、Amazon S3 儲存貯體等。政策也可以明確拒絕存取。許可是政策中的陳述式，允許或拒絕存取特定資源。

> **Important**
>
> 您建立的所有使用者都使用其登入資料進行身分驗證。不過，它們使用相同的 AWS 帳戶。使用者所做的任何變更都可能會影響整個帳戶。

AWS Marketplace 已定義 許可，以控制具有這些許可的人員可在 中採取的動作 AWS Marketplace 管理入口網站。也有 AWS Marketplace 建立和管理的政策，結合數個許可。此AWSMarketplaceSellerProductsFullAccess政策可讓使用者完整存取 中的 產品 AWS Marketplace 管理入口網站。

如需可用動作、資源和條件索引鍵的詳細資訊，請參閱服務授權參考中的 AWS Marketplace Vendor Insights 的動作、資源和條件索引鍵。

AWS Marketplace Vendor Insights 賣家的許可

您可以在 AWS Marketplace Vendor Insights 的 IAM 政策中使用下列許可。您可以將許可合併為單一 IAM 政策，以授予您想要的許可。

CreateDataSource

CreateDataSource 允許使用者建立新的資料來源資源。支援的資料來源包括：

- SOC2Type2AuditReport
- ISO27001AuditReport
- AWSAuditManagerSecurityAutomatedAssessment
- FedRAMPCertification
- GDPRComplianceReport
- HIPAAComplianceReport
- PCIDSSAuditReport
- SecuritySelfAssessment

動作群組：讀寫

必要資源：無

建立 資源： DataSource

DeleteDataSource

DeleteDataSource 允許使用者刪除他們擁有的資料來源。資料來源必須與要刪除的任何設定檔取消關聯。如需詳細資訊，請參閱the section called “AssociateDataSource”。

動作群組：讀寫

必要資源： DataSource

GetDataSource

GetDataSource 允許使用者擷取資料來源的詳細資訊。資料來源的詳細資訊包括中繼資料資訊，例如關聯的時間戳記、原始建立參數，以及處理資訊，如果有的話。

動作群組：唯讀、讀寫

必要資源： DataSource

UpdateDataSource

UpdateDataSource 允許使用者更新資料來源的詳細資訊。詳細資訊包括中繼資料資訊，例如名稱和來源資訊（例如角色、來源 Amazon Resource Name (ARN) 和來源內容）。

動作群組：唯讀、讀寫

必要資源： DataSource

ListDataSources

ListDataSources 允許使用者列出他們擁有的資料來源。

動作群組：唯讀、唯讀、唯讀

必要資源：無

CreateSecurityProfile

CreateSecurityProfile 允許使用者建立新的安全性描述檔。安全性描述檔是一種資源，可用來管理快照產生的方式和時間。使用者也可以控制設定檔的狀態和適用條款，以控制買方存取快照的方式。

動作群組：唯讀、唯讀、唯讀

必要資源：無

建立 資源： SecurityProfile

ListSecurityProfiles

ListSecurityProfiles 允許使用者列出他們擁有的安全設定檔。

動作群組：唯讀、唯讀、唯讀

必要資源：無

GetSecurityProfile

CreateSecurityProfile 允許使用者取得安全設定檔的詳細資訊。

動作群組：唯讀和讀寫

必要資源： SecurityProfile

AssociateDataSource

AssociateDataSource 允許使用者將現有的 DataSource與 AWS Marketplace Vendor Insights 設定檔建立關聯。使用者可以透過將資料來源與設定檔建立關聯或取消關聯來控制快照的內容。

動作群組：讀寫

必要資源： SecurityProfile 和 DataSource

DisassociateDataSource

DisassociateDataSource 允許使用者取消現有 DataSource與 AWS Marketplace Vendor Insights 設定檔的關聯。使用者可以透過將資料來源與設定檔建立關聯或取消關聯來控制快照的內容。

動作群組：讀寫

必要資源： SecurityProfile 和 DataSource

UpdateSecurityProfile

UpdateSecurityProfile 允許使用者修改安全性描述檔屬性，例如名稱和描述。

動作群組：讀寫

必要資源： SecurityProfile

ActivateSecurityProfile

ActivateSecurityProfile 允許使用者設定安全設定檔Active的狀態。啟用安全設定檔後，可以在 Staged 狀態中建立新的快照，以便在符合其他條件時釋出它們。如需詳細資訊，請參閱the section called “UpdateSecurityProfileSnapshotReleaseConfiguration”。

具有至少一個Released快照Active的安全設定檔符合最終使用者的 AWS Marketplace Vendor Insights 探索資格。

動作群組：讀寫

必要資源： SecurityProfile

DeactivateSecurityProfile

DeactivateSecurityProfile 允許使用者設定安全設定檔Inactive的狀態。安全設定檔的此終端機狀態等同於從共用狀態移除設定檔。使用者只有在設定檔沒有作用中訂閱者時，才能停用安全設定檔。

動作群組：讀寫

必要資源： SecurityProfile

UpdateSecurityProfileSnapshotCreationConfiguration

UpdateSecurityProfileSnapshotCreationConfiguration 允許使用者定義快照建立組態的自訂排程。可以使用此動作覆寫每週建立的預設建立組態。

使用者可以使用此動作來變更排程，包括取消排程、將排程推遲到未來的日期，或提早啟動新的快照建立。

動作群組：讀寫

必要資源： SecurityProfile

UpdateSecurityProfileSnapshotReleaseConfiguration

UpdateSecurityProfileSnapshotReleaseConfiguration 允許使用者定義快照發行組態的自訂排程。使用此動作可以覆寫具有兩天預備期的每週版本預設建立組態，以供檢閱。

使用者可以使用此動作來變更排程，包括取消排程或將排程延遲到未來的日期。

動作群組：讀寫

必要資源： SecurityProfile

ListSecurityProfileSnapshots

ListSecurityProfileSnapshots 允許使用者列出他們擁有之安全設定檔的快照。

動作群組：唯讀、清單唯讀和讀寫

必要資源： SecurityProfile

GetSecurityProfileSnapshot

GetSecurityProfileSnapshot 允許使用者取得他們擁有之安全設定檔的快照。

動作群組：唯讀和讀寫

必要資源： SecurityProfile

TagResource

TagResource 允許使用者將新標籤新增至資源。支援的資源為 SecurityProfile和 DataSource。

動作群組：標記

選用資源： SecurityProfile 和 DataSource

UntagResource

UntagResource 允許使用者從資源中移除標籤。支援的資源為 SecurityProfile和 DataSource。

動作群組：標記

選用資源： SecurityProfile 和 DataSource

ListTagsForResource

ListTagsForResource 允許使用者列出資源的資源標籤。支援的資源為 SecurityProfile和 DataSource。

動作群組：唯讀

選用資源： SecurityProfile 和 DataSource

其他資源

IAM 使用者指南中的下列資源提供有關開始使用和使用 IAM 的詳細資訊：

- IAM 中的安全最佳實務
- 管理 IAM 政策
- 將政策連接至 IAM 使用者群組
- IAM 身分（使用者、使用者群組和角色）
- 在 AWS 帳戶中建立 IAM 使用者
- 建立 IAM 使用者群組
- 使用 政策控制對 AWS 資源的存取

AWS Marketplace 安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，該架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。第三方稽核人員定期檢測及驗證安全的效率也是我們 AWS 合規計劃的一部分。若要了解適用的合規計劃 AWS Marketplace，請參閱AWS 合規計劃範圍內的服務。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須負責其他因素，包括資料的敏感度、組織的需求，以及適用的法律和法規。

本文件可協助您了解如何在使用 時套用共同責任模型 AWS Marketplace。下列主題說明如何設定 AWS Identity and Access Management 來管理對 的存取， AWS Marketplace 以符合您的安全與合規目標。您也可以了解如何使用其他 AWS 服務來協助您監控和保護 AWS Marketplace 資源。

若要進一步了解您提供的產品的安全和其他政策 AWS Marketplace，請參閱下列主題：

- 的 AMI 型產品需求 AWS Marketplace
- 的容器型產品需求 AWS Marketplace
- 建立機器學習產品的需求和最佳實務
- 的 SaaS 產品指導方針 AWS Marketplace
- 上的專業服務產品需求 AWS Marketplace

> **Note**
>
> 若要了解資料產品在 AWS Data Exchange 上的安全性，請參閱《AWS Data Exchange 使用者指南》中的安全性。
>
> 若要了解 中買方的安全性 AWS Marketplace，請參閱《 AWS Marketplace 買方指南》中的 上的安全性 AWS Marketplace。

主題

- 控制 AWS Marketplace 管理入口網站存取

- 賣方的 AWS Marketplace 政策和許可
- AWSAWS Marketplace 賣方的 受管政策
- AWS Marketplace Commerce Analytics Service 帳戶許可
- Amazon SQS 許可
- AWS Marketplace 計量和權利 API 許可
- 搭配 使用服務連結角色進行轉售授權 AWS Marketplace
- 使用 記錄 AWS Marketplace 計量 API 呼叫 AWS CloudTrail

控制 AWS Marketplace 管理入口網站存取

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助您控制對 AWS 資源的存取。如果您是管理員，您可以控制誰可以驗證 （登入） 和授權 （具有許可） 使用 AWS Marketplace 資源。IAM 是 AWS 服務 您可以免費使用的 。

控制誰可以在 中執行動作的建議方法是 AWS Marketplace 管理入口網站 使用 IAM 來建立使用者和群組。然後將使用者新增至群組，並管理群組。例如，如果 John 應該被允許檢視您的產品，請為他建立使用者，並將他的使用者新增至您為唯讀存取建立的群組。您可以將政策或許可指派給提供唯讀許可的群組。如果您有其他需要唯讀存取權的使用者，您可以將他們新增至您建立的群組，而不是將許可新增至使用者。如果 John 的角色變更，而且他不再需要唯讀存取，您可以從群組中移除 John。

政策是定義套用至使用者、群組或角色之許可的文件。然後，許可會決定使用者可以在其中執行的動作 AWS。政策通常允許存取特定動作，並且可以選擇性地授予特定資源允許的動作，例如 Amazon EC2 執行個體、Amazon S3 儲存貯體等。政策也可以明確拒絕存取。許可是政策中的陳述式，允許或拒絕存取特定資源。您可以陳述任何類似：「A 具有執行 B 到 C 的許可」。 例如，Jane (A) 具有從 John 的 Amazon Simple Queue Service 佇列 (C) 讀取訊息 (B) 的許可。每當 Jane 向 Amazon SQS 傳送使用 John 佇列的請求時，服務會檢查她是否具有許可。它會進一步檢查請求是否符合 John 在許可中指定的條件。

> **Important**
>
> 您建立的所有使用者都使用其登入資料進行身分驗證。不過，它們使用相同的 AWS 帳戶。使用者所做的任何變更都可能會影響整個帳戶。

AWS Marketplace 已定義 許可，以控制具有這些許可的人員可以採取的動作 AWS Marketplace 管理入口網站。也有 AWS Marketplace 建立和管理的政策，結合數個許可。

下列資源提供有關開始使用和使用 IAM 的詳細資訊。

- 建立管理使用者
- IAM 中的安全最佳實務
- 管理 IAM 政策
- 將政策連接至 IAM 使用者群組
- IAM 身分（使用者、群組和角色）
- 使用 政策控制對 AWS 資源的存取

下列主題提供一些高階指導，以建立使用者和群組，並以使用者身分登入。

主題

- 建立 使用者
- 建立或使用群組
- 以使用者身分登入

建立 使用者

若要允許公司中的人員登入 AWS Marketplace 管理入口網站，請為每個需要存取的人員建立使用者。

若要建立使用者

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中的存取管理下，選擇使用者，然後選擇建立使用者。
3. 在編號文字方塊中，輸入您要建立的每個使用者的名稱。
4. 清除為每個使用者產生存取金鑰核取方塊，然後選擇建立。

為您剛建立的每個使用者指派密碼

1. 在使用者清單中，選擇新使用者的名稱。
2. 選擇安全登入資料索引標籤，然後選擇管理密碼。
3. 選擇自動產生的密碼或自訂密碼的選項。或者，若要要求使用者在下次登入時選擇新密碼，請選取要求使用者在下次登入時建立新密碼的方塊。選擇套用。

4. 選擇下載登入資料，將登入憑證和帳戶特定的登入 URL 儲存到電腦上的逗號分隔值 (CSV) 檔案。然後，選擇 Close (關閉)。

> **Note**
>
> 若要使用您剛建立的登入憑證登入，使用者必須導覽至您的帳戶特定的登入 URL。此 URL 位於您剛下載的登入資料檔案中，也可在 IAM 主控台上使用。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者如何登入您的 AWS 帳戶 。

> **Tip**
>
> 即使您是 AWS 帳戶 擁有者，也要為自己建立登入憑證。這是建議每個人以使用者身分使用 AWS Marketplace 的最佳實務，即使是帳戶擁有者也是如此。如需如何為自己建立具有管理許可之使用者的說明，請參閱《IAM 使用者指南》中的建立管理使用者。

建立或使用群組

建立使用者之後，請建立群組、建立存取 中頁面的許可 AWS Marketplace 管理入口網站、將這些許可新增至群組，然後將使用者新增至群組。

當您將許可指派給群組時，您可以允許該群組的任何成員執行特定動作。當您將新使用者新增至群組時，該使用者會自動取得指派給群組的許可。群組可以具有多個動作的許可。我們建議您使用 AWS Marketplace 受管政策，而不是建立自己的政策。

將 的受管政策指派給 AWS Marketplace 群組

1. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在導覽窗格中，選擇群組，然後選擇您要連接政策的群組。
3. 在群組的摘要頁面上，在許可索引標籤下，選擇連接政策。
4. 在附加政策頁面上，篩選條件旁：輸入 awsmarketplace。
5. 選擇您要連接的政策，然後選擇連接政策。

建立具有 AWS Marketplace 管理入口網站 許可的政策

1. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。

2. 在導覽窗格中，選擇 Policies (政策)，然後選擇 Create Policy (建立政策)。
3. 在 Policy Generator (政策產生器) 旁選擇 Select (選取)。
4. 在 Edit Permissions (編輯許可) 頁面上，執行下列動作：

    a. 針對 Effect (效果)，選擇 Allow (允許)。
    b. 針對 AWS Service (服務)，選擇 AWS Marketplace 管理入口網站。
    c. 針對動作，選取要允許的許可。
    d. 選擇新增陳述式。
    e. 選擇 Next Step (後續步驟)。
5. 在檢閱政策頁面上，執行下列動作：

    a. 在政策名稱中，輸入此政策的名稱。請記下政策名稱，因為在後續步驟中需要它。
    b. （選用）針對描述，輸入此政策的描述。
    c. 選擇 Create Policy (建立政策)。

建立具有適當許可的 IAM 群組，並將使用者新增至群組

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中選擇 Groups (群組)，然後選擇 Create New Group (建立新群組)。
3. 針對群組名稱：，輸入群組的名稱。然後選擇 Next Step (下一步)。
4. 在連接政策頁面上，執行下列動作：

    a. 針對篩選條件：，選擇客戶受管政策。
    b. 選取您要連接到此群組的政策名稱旁的核取方塊。這通常是您剛建立的政策。
    c. 選擇 Next Step (後續步驟)。
5. 選擇 Create Group (建立群組)。
6. 在群組清單中尋找您的新群組，然後選取其旁邊的核取方塊。選擇群組動作，然後將使用者新增至群組。
7. 選取每個要新增至群組的使用者旁邊的核取方塊，然後選擇新增使用者。

以使用者身分登入

在 IAM 中建立使用者之後，使用者可以使用自己的登入憑證登入。若要這樣做，他們需要使用與您的相關聯的唯一 URL AWS 帳戶。您可以取得登入 URL 並將其分發給使用者。

取得您帳戶的唯一登入 URL

1. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在導覽窗格中，選擇 Dashboard (儀表板)。
3. 在內容窗格頂端附近，尋找 IAM 使用者登入連結：並記下登入連結，其格式如下：

```
https://AWS_account_ID.signin.aws.amazon.com/console/
```

> **Note**
>
> 如果您希望登入頁面的 URL 包含您的公司名稱（或其他易記識別符），而不是您的 AWS 帳戶 ID，您可以選擇自訂，為您的帳戶建立別名。如需詳細資訊，請參閱《IAM 使用者指南》中的AWS 您的帳戶 ID 及其別名。

4. 將此 URL 分發給您公司中可以與其合作的人員 AWS Marketplace，以及您為每個使用者建立的登入憑證。指示他們使用您帳戶的唯一登入 URL 在存取之前登入 AWS Marketplace。

賣方的 AWS Marketplace 政策和許可

AWS Marketplace 提供一組可搭配 使用的 受管政策 AWS Marketplace 管理入口網站。此外，您可以使用個別許可來建立自己的 AWS Identity and Access Management (IAM) 政策。

您也可以 AWS Marketplace 管理入口網站 為設定、聯絡我們、檔案上傳和 Insights 索引標籤提供對的精細存取。精細存取可讓您執行下列動作：

- 授予其他人許可，以管理和使用 AWS 帳戶 中的資源，而無需共用您的密碼或存取金鑰。
- 為多個人員授予各種資源的精細許可。例如，您可以允許某些使用者檢視 中的設定索引標籤 AWS Marketplace 管理入口網站。對於其他使用者，您可以在設定和聯絡我們索引標籤中允許編輯存取權。

> **Note**
>
> 如需資料產品 AWS Data Exchange 中政策和許可的詳細資訊，請參閱《AWS Data Exchange 使用者指南》中的 AWS Data Exchange 中的 Identity and Access Management。
> 如需 AWS Marketplace 買方政策和許可的詳細資訊，請參閱《 AWS Marketplace 買方指南》中的控制 AWS Marketplace 訂閱的存取。

AWS Marketplace 賣方的政策

您可以使用下列 受管政策，為使用者提供 的受控存取權 AWS Marketplace 管理入口網站：

`AWSMarketplaceSellerFullAccess`

允許完整存取 AWS Marketplace 管理入口網站 和其他 AWS 服務中的所有頁面，例如 Amazon Machine Image (AMI) 管理。

`AWSMarketplaceSellerProductsFullAccess`

允許完整存取 中的產品頁面 AWS Marketplace 管理入口網站。

`AWSMarketplaceSellerProductsReadOnly`

允許唯讀存取 中的產品頁面 AWS Marketplace 管理入口網站。

> **Important**
>
> AWS Marketplace 買方可以使用 受管政策來管理他們購買的訂閱。您以 AWS Marketplace 管理入口網站 開頭的受管政策名稱`AWSMarketplaceSeller`。當您在 IAM 中搜尋政策時，請務必搜尋開頭為 的政策名稱`AWSMarketplaceSeller`。如需這些政策的詳細資訊，請參閱 AWS 受管政策參考。

AWS Marketplace 也針對特定案例提供專門的 受管政策。如需 AWS Marketplace 賣方的 AWS 受管政策完整清單，以及其提供的許可說明，請參閱 AWSAWS Marketplace 賣方的 受管政策。

AWS Marketplace 賣方的許可

您可以在 的 IAM 政策中使用下列許可 AWS Marketplace 管理入口網站：

`aws-marketplace-management:PutSellerVerificationDetails`

允許 存取以開始了解您的客戶 (KYC) 程序。

`aws-marketplace-management:GetSellerVerificationDetails`

允許 存取以檢視 中的 KYC 狀態 AWS Marketplace 管理入口網站。

`aws-marketplace-management:PutBankAccountVerificationDetails`

允許存取以啟動銀行帳戶驗證程序。

aws-marketplace-management:GetBankAccountVerificationDetails

允許在 中檢視銀行帳戶驗證狀態的存取權 AWS Marketplace 管理入口網站。

aws-marketplace-management:PutSecondaryUserVerificationDetails

允許在 中新增次要使用者的存取權 AWS Marketplace 管理入口網站。

aws-marketplace-management:GetSecondaryUserVerificationDetails

允許 存取以檢視 中的次要使用者狀態 AWS Marketplace 管理入口網站。

aws-marketplace-management:GetAdditionalSellerNotificationRecipients

允許 存取以檢視 AWS Marketplace 通知的電子郵件聯絡人。

aws-marketplace-management:PutAdditionalSellerNotificationRecipients

允許存取以更新 AWS Marketplace 通知的電子郵件聯絡人。

tax:PutTaxInterview

允許 存取在 中接受稅務面試 AWS Marketplace 管理入口網站。

tax:GetTaxInterview

允許 在 中檢視稅務面試狀態的存取權 AWS Marketplace 管理入口網站。

tax:GetTaxInfoReportingDocument

允許 AWS Marketplace 賣方從稅務儀表板檢視和下載稅務文件 （ 例如 1099-K 表格 ）

payments:CreatePaymentInstrument

允許將銀行帳戶新增至 的存取權 AWS Marketplace 管理入口網站。

payments:GetPaymentInstrument

允許在 中檢視現有銀行帳戶的存取權 AWS Marketplace 管理入口網站。

support:CreateCase

允許 存取，以在 中建立 AWS Marketplace 案例 AWS Marketplace 管理入口網站。

aws-marketplace-management:viewSupport

允許存取 中的客戶支援資格頁面 AWS Marketplace 管理入口網站。

aws-marketplace-management:viewReports

允許存取 中的報告頁面 AWS Marketplace 管理入口網站。

aws-marketplace:ListEntities

允許 存取列出其中的物件 AWS Marketplace 管理入口網站。存取 中的檔案上傳、優惠和合作夥伴頁面時需要 AWS Marketplace 管理入口網站。

> **Note**
>
> 若要允許檢視設定索引標籤的存取權，您可以使用此許可、 `ListEntity`許可和下列 Amazon Resource Name (ARN)：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`。

aws-marketplace:DescribeEntity

允許 存取以檢視 中物件的詳細資訊 AWS Marketplace 管理入口網站。存取 中的檔案上傳、優惠、合作夥伴和協議頁面時需要 AWS Marketplace 管理入口網站。

> **Note**
>
> 若要允許檢視設定索引標籤的存取權，您可以使用此許可、 `DescribeEntity`許可和下列 ARN：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/*`。

aws-marketplace:StartChangeSet

允許在 中建立產品變更的存取權 AWS Marketplace 管理入口網站。在 中的檔案上傳、優惠、合作夥伴和協議頁面進行變更時需要 AWS Marketplace 管理入口網站。

> **Note**
>
> 若要允許在 中註冊為賣方的存取權 AWS Marketplace，您可以使用此許可、`catalog:ChangeType: "CreateSeller"`條件金鑰和下列 ARN：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`。
> 若要允許 存取以更新 中的賣方設定檔 AWS Marketplace，您可以使用此許可、 `catalog:ChangeType: "UpdateInformation"` 條件金鑰和下列

ARN：arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}。
若要允許存取更新 Amazon Web Services 的支出偏好設定，您可以使用此許可、catalog:ChangeType: "UpdateDisbursementPreferences"條件金鑰和下列 ARN：arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}。

aws-marketplace:SearchAgreements

允許在協議頁面上檢視協議的高階清單，以及在合作夥伴頁面上檢視 ISVs和管道合作夥伴之間的機會。

aws-marketplace:DescribeAgreement

允許檢視協議頁面上的高階協議詳細資訊，以及合作夥伴頁面上 ISVs和管道合作夥伴之間的機會。

aws-marketplace:GetAgreementTerms

允許檢視協議頁面上的所有協議條款詳細資訊，以及合作夥伴頁面上 ISVs和管道合作夥伴之間的機會。

aws-marketplace:GetSellerDashboard

允許存取 中 Insights 頁面上的儀表板 AWS Marketplace 管理入口網站。

aws-marketplace:ListAssessments

允許 存取以檢視待定賣方動作的評估清單。

aws-marketplace:DescribeAssessment

允許 存取以檢視待定賣方動作的評估詳細資訊。

> **Note**
>
> 若要讓使用者存取管理產品頁面，您必須使用 AWSMarketplaceSellerProductsFullAccess或 AWSMarketplaceSellerProductsReadOnly受管許可。

您可以將上述許可合併為單一 IAM 政策，以授予您想要的許可。請參閱以下範例。

範例 1：檢視 KYC 狀態的許可

若要授予在 中檢視 GMT 狀態的許可 AWS Marketplace 管理入口網站，請使用類似下列範例的政策。

若要授予在 中檢視 GMT 狀態的許可 AWS Marketplace 管理入口網站，請使用類似下列範例的政策。

JSON

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

範例 2：為私有優惠建立升級和續約的許可

若要授予檢視和使用協議頁面為私有優惠建立升級和續約的許可，請使用類似下列範例的政策。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aws-marketplace:SearchAgreements",
                "aws-marketplace:DescribeAgreement",
                "aws-marketplace:GetAgreementTerms",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:StartChangeSet"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws-marketplace:PartyType": "Proposer"
                },
                "ForAllValues:StringEquals": {
                    "aws-marketplace:AgreementType": [
                        "PurchaseAgreement"
                    ]
                }
            }
        }
    ]
}
```

範例 3：存取優惠頁面並建立新的私有優惠的許可

若要授予檢視和使用優惠頁面來檢視現有私有優惠和建立私有優惠的許可，請使用類似下列範例的政策。

JSON

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

範例 4：存取設定頁面的許可

若要授予檢視和使用設定頁面的許可，請使用類似下列範例的政策。

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

範例 5：存取檔案上傳頁面的許可

若要授予檢視和使用檔案上傳頁面的許可，請使用類似下列範例的政策。

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

使用 IAM 群組

或者，您可以建立個別的 IAM 群組，以授予 中每個個別頁面的存取權 AWS Marketplace 管理入口網站。使用者可以屬於多個群組。因此，如果使用者需要存取多個頁面，您可以將使用者新增至所有適當的群組。例如，建立一個 IAM 群組並授予該群組存取 Insights 頁面的許可、建立另一個群組並授予該群組存取檔案上傳頁面的許可，以此類推。如果使用者需要同時存取 Insights 頁面和檔案上傳頁面的許可，請將使用者新增至這兩個群組。

如需使用者和群組的詳細資訊，請參閱《IAM 使用者指南》中的 IAM 身分（使用者、群組和角色）。

AWSAWS Marketplace 賣方的 受管政策

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住， AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受 AWS 管政策中定義的許可，則更新會影響政策連接的所有委託人身分（使用者、群組和角色）。當新的 AWS 服務 啟動或新的 API 操作可用於現有服務時， AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

本節列出用於管理賣方存取的每個政策 AWS Marketplace。如需買方政策的相關資訊，請參閱AWS Marketplace 《 買方指南》中的AWSAWS Marketplace 買方受管政策。

主題

- AWS 受管政策： AWSMarketplaceAmiIngestion
- AWS 受管政策： AWSMarketplaceFullAccess
- AWS 受管政策： AWSMarketplaceGetEntitlements
- AWS 受管政策： AWSMarketplaceMeteringFullAccess
- AWS 受管政策：AWSMarketplaceMeteringRegisterUsage
- AWS 受管政策： AWSMarketplaceSellerFullAccess
- AWS 受管政策：AWSMarketplaceSellerProductsFullAccess
- AWS 受管政策：AWSMarketplaceSellerProductsReadOnly
- AWS 受管政策：AWSMarketplaceSellerOfferManagement
- AWS 受管政策： AWSVendorInsightsVendorFullAccess
- AWS 受管政策： AWSVendorInsightsVendorReadOnly
- AWS MarketplaceAWS 受管政策的更新

AWS 受管政策： AWSMarketplaceAmiIngestion

您可以使用此政策建立服務角色，然後由 代表您 AWS Marketplace 執行動作。如需有關使用 `AWSMarketplaceAmiIngestion` 的詳細資訊，請參閱 提供 AMI 的 AWS Marketplace 存取權。

此政策授予參與者許可， AWS Marketplace 允許 複製您的 Amazon Machine Image (AMIs) 以列出它們 AWS Marketplace。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceAmiIngestion。

AWS 受管政策： AWSMarketplaceFullAccess

您可將 `AWSMarketplaceFullAccess` 政策連接到 IAM 身分。

此政策授予管理許可，允許以賣方和買方身分完整存取 AWS Marketplace 和相關服務。這些許可包括下列功能：

- 訂閱和取消訂閱 AWS Marketplace 軟體。
- 從 管理 AWS Marketplace 軟體執行個體 AWS Marketplace。
- 在帳戶中建立和管理私有市集。
- 提供 Amazon EC2 AWS CloudFormation和 Amazon EC2 Systems Manager 的存取權。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceFullAccess。

AWS 受管政策： AWSMarketplaceGetEntitlements

您可將 `AWSMarketplaceGetEntitlements` 政策連接到 IAM 身分。

此政策授予唯讀許可，允許軟體即服務 (SaaS) 產品賣方檢查客戶是否已訂閱其 AWS Marketplace SaaS 產品。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceGetEntitlements。

AWS 受管政策： AWSMarketplaceMeteringFullAccess

您可將 `AWSMarketplaceMeteringFullAccess` 政策連接到 IAM 身分。

此政策會授予參與者許可，以允許報告對應至 AMI 和容器產品的計量用量，並在 上使用靈活的耗用量定價 AWS Marketplace。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceMeteringFullAccess。

AWS 受管政策：AWSMarketplaceMeteringRegisterUsage

您可將 `AWSMarketplaceMeteringRegisterUsage` 政策連接到 IAM 身分。

此政策會授予參與者許可，允許報告對應於每小時定價的容器產品的計量用量 AWS Marketplace。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceMeteringRegisterUsage。

AWS 受管政策： AWSMarketplaceSellerFullAccess

您可將 AWSMarketplaceSellerFullAccess 政策連接到 IAM 身分。

此政策授予管理許可 AWS Marketplace，允許完整存取 AMI 產品中使用的所有賣方操作 AWS Marketplace 管理入口網站，包括和管理 Amazon EC2 AMI。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MarketplaceManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace-management:uploadFiles",
                "aws-marketplace-management:viewReports",
                "aws-marketplace-management:viewSupport",
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:StartChangeSet",
                "aws-marketplace:CancelChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:GetSellerDashboard",
                "aws-marketplace:ListAssessments",
                "aws-marketplace:DescribeAssessment",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots",
                "ec2:ModifyImageAttribute",
                "ec2:ModifySnapshotAttribute"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AgreementAccess",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:SearchAgreements",
                "aws-marketplace:DescribeAgreement",
```

```
                "aws-marketplace:GetAgreementTerms"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws-marketplace:PartyType": "Proposer"
                },
                "ForAllValues:StringEquals": {
                    "aws-marketplace:AgreementType": [
                        "PurchaseAgreement"
                    ]
                }
            }
        },
        {
            "Sid": "IAMGetRole",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Sid": "AssetScanning",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "assets.marketplace.amazonaws.com"
                }
            }
        },
        {
            "Sid": "VendorInsights",
            "Effect": "Allow",
            "Action": [
                "vendor-insights:GetDataSource",
                "vendor-insights:ListDataSources",
                "vendor-insights:ListSecurityProfiles",
                "vendor-insights:GetSecurityProfile",
                "vendor-insights:GetSecurityProfileSnapshot",
```

```
                "vendor-insights:ListSecurityProfileSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TagManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:TagResource",
                "aws-marketplace:UntagResource",
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Sid": "SellerSettings",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace-management:GetSellerVerificationDetails",
                "aws-marketplace-management:PutSellerVerificationDetails",
                "aws-marketplace-management:GetBankAccountVerificationDetails",
                "aws-marketplace-management:PutBankAccountVerificationDetails",
                "aws-marketplace-management:GetSecondaryUserVerificationDetails",
                "aws-marketplace-management:PutSecondaryUserVerificationDetails",
                "aws-marketplace-
management:GetAdditionalSellerNotificationRecipients",
                "aws-marketplace-
management:PutAdditionalSellerNotificationRecipients",
                "payments:GetPaymentInstrument",
                "payments:CreatePaymentInstrument",
                "tax:GetTaxInterview",
                "tax:PutTaxInterview",
                "tax:GetTaxInfoReportingDocument",
                "tax:ListSupplementalTaxRegistrations",
                "tax:PutSupplementalTaxRegistration",
                "tax:DeleteSupplementalTaxRegistration",
                "tax:GetTaxRegistration"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Support",
            "Effect": "Allow",
            "Action": [
```

```
                "support:CreateCase"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ResourcePolicyManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:GetResourcePolicy",
                "aws-marketplace:PutResourcePolicy",
                "aws-marketplace:DeleteResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "resale-
authorization.marketplace.amazonaws.com"
                }
            }
        }
    ]
}
```

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceSellerFullAccess。

AWS 受管政策：AWSMarketplaceSellerProductsFullAccess

您可將 AWSMarketplaceSellerProductsFullAccess 政策連接到 IAM 身分。

此政策授予參與者許可，允許完整存取 和 AWS Marketplace 管理入口網站，以及管理 AMI 型產品中使用的 Amazon EC2 AMI。

許可詳細資訊

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:StartChangeSet",
                "aws-marketplace:CancelChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:ListAssessments",
                "aws-marketplace:DescribeAssessment",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots",
                "ec2:ModifyImageAttribute",
                "ec2:ModifySnapshotAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "assets.marketplace.amazonaws.com"
                }
            }
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "vendor-insights:GetDataSource",
                "vendor-insights:ListDataSources",
                "vendor-insights:ListSecurityProfiles",
                "vendor-insights:GetSecurityProfile",
                "vendor-insights:GetSecurityProfileSnapshot",
                "vendor-insights:ListSecurityProfileSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:TagResource",
                "aws-marketplace:UntagResource",
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:GetResourcePolicy",
                "aws-marketplace:PutResourcePolicy",
                "aws-marketplace:DeleteResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        }
    ]
}
```

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceSellerProductsFullAccess。

AWS 受管政策：AWSMarketplaceSellerProductsReadOnly

您可將 AWSMarketplaceSellerProductsReadOnly 政策連接到 IAM 身分。

此政策授予唯讀許可，允許在 上檢視產品 AWS Marketplace 管理入口網站，以及檢視 AMI 型產品中使用的 Amazon EC2 AMI。

許可詳細資訊

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:ListAssessments",
                "aws-marketplace:DescribeAssessment",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:GetResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        }
    ]
}
```

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceSellerProductsReadOnly。

AWS 受管政策：AWSMarketplaceSellerOfferManagement

您可將 AWSMarketplaceSellerOfferManagement 政策連接到 IAM 身分。

此政策授予賣方對優惠和協議管理活動的存取權。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSMarketplaceSellerOfferManagement。

AWS 受管政策： AWSVendorInsightsVendorFullAccess

您可將 AWSVendorInsightsVendorFullAccess 政策連接到 IAM 身分。

此政策授予在 AWS Marketplace Vendor Insights 上建立和管理所有資源的完整存取權。在 AWS Marketplace Vendor Insights 中，評估者等於買方，而基於本指南的目的，廠商等於賣方。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSVendorInsightsVendorFullAccess。

AWS 受管政策： AWSVendorInsightsVendorReadOnly

您可將 AWSVendorInsightsVendorReadOnly 政策連接到 IAM 身分。

此政策授予檢視 AWS Marketplace Vendor Insights 設定檔和相關資源的唯讀存取權。在 AWS Marketplace Vendor Insights 中，評估者等於買方，而基於本指南的目的，廠商等於賣方。

若要檢視此政策的許可，請參閱 AWS 受管政策參考中的 AWSVendorInsightsVendorReadOnly。

AWS MarketplaceAWS 受管政策的更新

檢視自此服務開始追蹤這些變更 AWS Marketplace 以來， AWS 受管政策更新的詳細資訊。如需有關此頁面變更的自動提醒，請訂閱 AWS Marketplace 文件歷史紀錄 頁面的 RSS 摘要。

變更	描述	日期
AWSMarketplaceSellerFullAccess – 已更新政策	AWS Marketplace 為補充稅務設定檔功能新增了四個新SellerSettings 許可：ListSupplementalTaxRegistrations 、PutSupplementalTaxRegistration 、DeleteSup	2024 年 12 月 20 日

變更	描述	日期
	plementalTaxRegistration 、GetTaxRegistration 。	
AWSMarketplaceSellerFullAccess – 已更新政策 AWSMarketplaceSellerProductsFullAccess – 已更新政策 AWSMarketplaceSellerFullAccess – 更新的政策 AWSMarketplaceSellerProductsReadOnly – 更新的政策	AWS Marketplace 已移除 ListTasks 、UpdateTasks 、 DescribeTask 和 CompleteTasks 許可。	2024 年 12 月 10 日
AWSMarketplaceSellerOfferManagement – 新增了政策	AWS Marketplace 已新增政策： AWSMarketplaceSellerOfferManagement	2024 年 11 月 18 日
AWSMarketplaceSellerFullAccess – 更新的政策	AWS Marketplace 新增 UploadFiles 許可。變更可讓賣方使用 中的已棄用頁面 AWS Marketplace 管理入口網站。	2024 年 11 月 6 日
AWSMarketplaceSellerFullAccess – 更新的政策	AWS Marketplace 新增 ListAssessments 和 DescribeAssessments 許可。這些變更可讓 SSLv2 使用者存取評估資料。	2024 年 10 月 22 日

變更	描述	日期
AWSMarketplaceSellerProductsFullAccess – 更新的政策	AWS Marketplace 新增 `ListAssessments` 和 `DescribeAssessments` 許可。這些變更可讓 SSLv2 使用者存取評估資料。	2024 年 10 月 22 日
AWSMarketplaceSellerProductsReadOnly – 更新的政策	AWS Marketplace 新增 `ListAssessments` 和 `DescribeAssessments` 許可。這些變更可讓 SSLv2 使用者存取評估資料。	2024 年 10 月 22 日
AWSMarketplaceSellerFullAccess – 已更新政策	更新`AWSMarketplaceSellerFullAccess` 文件以反映移除下列動作：`aws-marketplace-management:viewMarketing` 、`aws-marketplace-management:viewSettings` 和 `aws-marketplace-management:uploadFiles` 。此更新也包含移除使用精細許可區段。	2024 年 6 月 4 日
AWSMarketplaceGetEntitlements – 已更新政策	AWS Marketplace 已更新`AWSMarketplaceGetEntitlements` 為政策陳述`sid`式新增 。	2024 年 3 月 22 日
AWSMarketplaceSellerFullAccess – 已更新政策	AWS Marketplace 已更新`AWSMarketplaceSellerFullAccess` 以新增建立服務連結角色的許可。	2024 年 3 月 15 日

變更	描述	日期
AWSMarketplaceSellerFullAccess – 已更新政策	AWS Marketplace 已更新AWSMarketplaceSellerFullAccess 以新增存取稅務資訊的許可。	2024 年 2 月 8 日
AWSVendorInsightsVendorFullAccess - 已更新政策	AWS Marketplace 已更新AWSVendorInsightsVendorFullAccess 以新增更新資料來源的許可。	2023 年 10 月 18 日
AWSMarketplaceSellerFullAccess – 已更新政策	AWS Marketplace 已更新AWSMarketplaceSellerFullAccess 以新增共用實體的許可。	2023 年 6 月 1 日
AWSMarketplaceSellerFullAccess – 已更新政策	AWS Marketplace 已更新AWSMarketplaceSellerFullAccess 以新增與帳戶驗證、銀行帳戶驗證、案例管理和賣方通知詳細資訊相關的許可。	2023 年 6 月 1 日
AWSMarketplaceSellerFullAccess – 已更新政策	AWS Marketplace 已更新AWSMarketplaceSellerFullAccess 以新增存取賣方儀表板的許可。	2022 年 12 月 23 日
AWSMarketplaceSellerFullAccess、AWSMarketplaceSellerProductsFullAccess、 AWSMarketplaceSellerProductsReadOnly – 更新現有政策	AWS Marketplace 更新了新標籤型授權功能的政策。	2022 年 12 月 9 日

變更	描述	日期
AWS Marketplace 已更新 AWSVendorInsightsVendorFullAccess	AWS Marketplace 已更新AWSMarketplaceSellerProductsFullAccess 以新增協議搜尋、更新設定檔快照、廠商標記，並允許唯讀存取 AWS Artifact 第三方報告 （預覽）。	2022 年 11 月 30 日
AWS Marketplace 已更新 AWSVendorInsightsVendorReadOnly	AWS Marketplace 已更新AWSVendorInsightsVendorReadOnly 以新增列出標籤的許可，並允許唯讀 AWS Artifact 存取第三方報告（預覽）。	2022 年 11 月 30 日
AWSVendorInsightsVendorFullAccess 和 AWSVendorInsightsVendorReadOnly – 新增了 政策	AWS Marketplace 新增了新功能 AWS Marketplace Vendor Insights 的政策： AWSMarketplaceSellerProductsFullAccess 和 AWSVendorInsightsVendorReadOnly 。	2022 年 7 月 26 日
AWSMarketplaceSellerProductsFullAccess和 AWSMarketplaceSellerFullAccess– 更新的政策	AWS Marketplace 更新了新功能 AWS Marketplace Vendor Insights： AWSMarketplaceSellerProductsFullAccess 和 的政策AWSMarketplaceSellerFullAccess 。	2022 年 7 月 26 日
AWSMarketplaceSellerFullAccess 和 AWSMarketplaceSellerProductsFullAccess – 更新現有政策	AWS Marketplace 已更新政策，使iam:PassedToService 條件僅套用至 iam:PassRole 。	2021 年 11 月 22 日

變更	描述	日期
AWSMarketplaceFullAccess – 更新現有政策	AWS Marketplace 已從`AWSMarketplaceFullAccess` 政策移除重複的`ec2:DescribeAccountAttributes` 許可。	2021 年 7 月 20 日
AWS Marketplace 開始追蹤變更	AWS Marketplace 已開始追蹤其 AWS 受管政策的變更。	2021 年 4 月 20 日

AWS Marketplace Commerce Analytics Service 帳戶許可

使用下列 IAM 許可政策註冊 AWS Marketplace Commerce Analytics Service。

如需如何註冊的說明，請遵循 入門指南。

JSON

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

使用以下 IAM 許可政策，允許使用者向 AWS Marketplace Commerce Analytics Service 提出請求。

JSON

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

如需使用此功能的詳細資訊，請參閱「使用 AWS Marketplace Commerce Analytics Service 存取產品和客戶資料」。

Amazon SQS 許可

作為 SaaS 產品發佈程序的一部分， AWS Marketplace 為您提供一個 Amazon SNS 主題，您可以在客戶的訂閱或權利狀態變更時用來接收通知。您可以將一或多個 Amazon SQS 佇列設定為主題，以便佇列可以對通知採取動作。例如，如果客戶將更多儲存空間新增至 SaaS 產品中的訂閱，Amazon SNS 主題可以傳送訊息至 Amazon SQS 佇列，該佇列會啟動程序以自動增加該客戶的可用儲存容量。

當您訂閱 Amazon Simple Queue Service (Amazon SQS) 佇列至提供的 Amazon SNS 主題時，會自動新增許可，以允許主題將訊息發佈至佇列。不過，您仍然需要 IAM 政策來授予 AWS Marketplace 計量和權利服務 API 使用者對佇列的存取權。如果服務使用相同的登入資料執行，這可以套用到相同的使用者。使用下列內容建立政策，並將其連接至您的使用者或角色。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:ReceiveMessage", "sqs:DeleteMessage",
 "sqs:GetQueueAttributes", "sqs:GetQueueUrl"
            ],
            "Effect": "Allow",
```

```
            "Resource": "arn:aws:sqs:REGION_HERE:XXXXXXXXXXXX:NAME_HERE"
        }
    ]
}
```

> **Note**
>
> `Resource` 欄位是 Amazon SQS 佇列的 Amazon Resource Name (ARN)。

如需 SaaS 產品的訊息通知和佇列的詳細資訊，請參閱 the section called “訂閱 SQS 佇列至 SNS 主題”和 the section called “存取 AWS Marketplace 計量和權利服務 APIs”。

AWS Marketplace 計量和權利 API 許可

軟體即服務 (SaaS) 產品、Amazon Machine Image (AMI) 產品和容器產品可以使用 AWS Marketplace Metering Service 和 API。 AWS Marketplace Entitlement Service APIs 每種類型都需要不同的 AWS Identity and Access Management (IAM) 許可。針對您的產品或產品，您會針對所有用量進行計量，而 AWS 會根據您提供的計量記錄向客戶收費。若要啟用提供 AWS Marketplace 計量記錄所需的整合，整合正在使用的服務帳戶需要受限制的 IAM 政策才能啟用存取。將您要為 傳送計量資訊之產品類型的政策連接至您用於整合的使用者或角色。

主題

- SaaS 產品的 IAM 政策
- AMI 產品的 IAM 政策
- 容器產品的 IAM 政策

SaaS 產品的 IAM 政策

在下列政策中，所有 SaaS 整合都需要第一個許可 `aws-marketplace:ResolveCustomer`。API 需要 AWS Marketplace Metering Service 第二個許可 `aws-marketplace:BatchMeterUsage`。 AWS Marketplace Entitlement Service API 需要第三個許可 `aws-marketplace:GetEntitlements`。

JSON

```
{
```

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

如需 SaaS 產品的詳細資訊，請參閱 中的 SaaS 型產品 AWS Marketplace。

AMI 產品的 IAM 政策

針對 AMI 產品使用以下 IAM 政策。

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

如需 AMI 產品的詳細資訊，請參閱 中的 AMI 型產品 AWS Marketplace。

容器產品的 IAM 政策

針對容器產品使用以下 IAM 政策。

JSON

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

如需容器產品的詳細資訊，請參閱 上的容器型產品 AWS Marketplace。

如需建立使用者的詳細資訊，請參閱《IAM 使用者指南》中的在 中建立 AWS 帳戶使用者。 如需建立和指派政策的詳細資訊，請參閱變更 IAM 使用者的許可。

此政策會授予您連接政策之 IAM 角色或使用者的 APIs 存取權。如需如何為這些 API 呼叫啟用另一個帳戶的角色假設的詳細資訊，請參閱 AWS Partner Network (APN) 部落格中的如何跨多個 最佳架構 Your AWS Marketplace SaaS 訂閱 AWS 帳戶。

搭配 使用服務連結角色進行轉售授權 AWS Marketplace

AWS Marketplace use AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 的唯一 IAM 角色類型 AWS Marketplace。服務連結角色由 預先定義， AWS Marketplace 並包含服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更 AWS Marketplace 輕鬆地設定，因為您不必手動新增必要的許可。 AWS Marketplace 會定義其服務連結角色的許可，除非另有定義，否則只能 AWS Marketplace 擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。這可保護您的 AWS Marketplace 資源，因為您不會不小心移除存取資源的許可。

如需有關支援服務連結角色的其他 服務的資訊，請參閱服務連結角色欄中與 AWS IAM 搭配使用的 服務，並尋找具有是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

主題

- 的服務連結角色許可 AWS Marketplace
- 為 建立服務連結角色 AWS Marketplace
- 編輯 的服務連結角色 AWS Marketplace
- 刪除 的服務連結角色 AWS Marketplace
- AWS Marketplace 服務連結角色支援的 區域

的服務連結角色許可 AWS Marketplace

AWS Marketplace 使用名為 AWSServiceRoleForMarketplaceResaleAuthorization 的服務連結角色，這可讓您存取 AWS Marketplace 為重新銷售授權而使用或管理 AWS 的服務和資源。

AWSServiceRoleForMarketplaceResaleAuthorization 服務連結角色信任下列服務擔任該角色：

- `resale-authorization.marketplace.amazonaws.com`

名為 AWSMarketplaceResaleAuthorizationServiceRolePolicy 的角色許可政策允許 對指定的資源 AWS Marketplace 完成下列動作。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "AllowResaleAuthorizationShareActionsRAMCreate",
            "Effect": "Allow",
            "Action": [
                "ram:CreateResourceShare"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ],
            "Condition": {
                "StringEquals": {
                    "ram:RequestedResourceType": "aws-marketplace:Entity"
                },
                "ArnLike": {
```

```
                    "ram:ResourceArn": "arn:aws:aws-
marketplace:*:*:AWSMarketplace/ResaleAuthorization/*"
                },
                "Null": {
                    "ram:Principal": "true"
                }
            }
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsRAMAssociate",
            "Effect": "Allow",
            "Action": [
                "ram:AssociateResourceShare"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ],
            "Condition": {
                "Null": {
                    "ram:Principal": "false"
                },
                "StringEquals": {
                    "ram:ResourceShareName": "AWSMarketplaceResaleAuthorization"
                }
            }
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsRAMAccept",
            "Effect": "Allow",
            "Action": [
                "ram:AcceptResourceShareInvitation"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ],
            "Condition": {
                "StringEquals": {
                    "ram:ResourceShareName": "AWSMarketplaceResaleAuthorization"
                }
            }
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsRAMGet",
            "Effect": "Allow",
```

```
            "Action": [
                "ram:GetResourceShareInvitations",
                "ram:GetResourceShareAssociations"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ]
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsMarketplace",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:PutResourcePolicy",
                "aws-marketplace:GetResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/
ResaleAuthorization/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["ram.amazonaws.com"]
                }
            }
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsMarketplaceDescribe",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:DescribeEntity"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/
ResaleAuthorization/*"
        }
    ]
}
```

您必須設定許可，以允許您的使用者、群組或角色建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

為 建立服務連結角色 AWS Marketplace

您不需要手動建立一個服務連結角色。當您在 中選擇服務連結角色時 AWS Marketplace 管理入口網站， 會為您 AWS Marketplace 建立服務連結角色。

建立服務連結角色

1. 在 中AWS Marketplace 管理入口網站，登入管理帳戶，然後選擇設定。
2. 在設定區段中，選取服務連結角色索引標籤。
3. 在服務連結角色頁面上，選取用於轉售授權或轉售授權整合的服務連結角色，然後選擇建立服務連結角色或設定整合。
4. 在重新銷售授權或建立重新銷售授權整合的服務連結角色頁面上，檢閱資訊，然後選擇建立服務連結角色或建立整合進行確認。

   服務連結角色頁面上會出現一則訊息，指出已成功建立轉售授權服務連結角色。

如果您刪除服務連結角色，您可以依照下列步驟重新建立該角色。

編輯 的服務連結角色 AWS Marketplace

AWS Marketplace 不允許您編輯 AWSServiceRoleForMarketplaceResaleAuthorization 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱「IAM 使用者指南」的編輯服務連結角色。

刪除 的服務連結角色 AWS Marketplace

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。

> **Note**
>
> 如果獨立軟體廠商 (ISVs) 沒有 角色， AWS Resource Access Manager 則不會自動與目標通路合作夥伴共用新的轉售授權。如果頻道合作夥伴沒有 角色， AWS Resource Access Manager 則不會自動接受目標為他們的轉售授權。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForMarketplaceResaleAuthorization 服務連結角色。如需詳細資訊，請參閱「IAM 使用者指南」中的刪除服務連結角色。

AWS Marketplace 服務連結角色支援的 區域

AWS Marketplace 支援在所有提供服務的區域中使用服務連結角色。如需詳細資訊，請參閱AWS 區域與端點。

使用 記錄 AWS Marketplace 計量 API 呼叫 AWS CloudTrail

AWS Marketplace 已與 整合 AWS CloudTrail，此服務提供由使用者、角色或 AWS 服務 所採取動作的記錄 AWS Marketplace。CloudTrail 會將 的 API 呼叫擷取 AWS Marketplace 為事件。擷取的呼叫包括從 AWS Marketplace 主控台的呼叫，以及對 AWS Marketplace API 操作的程式碼呼叫。

建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。在 中發生支援的事件活動時 AWS Marketplace，該活動會與事件歷史記錄中的其他 AWS 服務 事件一起記錄在 CloudTrail 事件中。 您可以檢視、搜尋和下載 帳戶的最新事件。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 請求是使用根登入資料還是 AWS Identity and Access Management 使用者登入資料提出。
- 請求是使用角色或聯合身分使用者的臨時安全登入資料提出。
- 該請求是否由另一項 AWS 服務服務提出。

AWS Marketplace 支援將`BatchMeterUsage`操作記錄為 CloudTrail 日誌檔案中的事件。

AWS Marketplace 計量 API 日誌檔案項目範例

範例：`BatchMeterUsage`

下列範例顯示示範 `BatchMeterUsage`動作的 CloudTrail 日誌項目 AWS Marketplace Metering Service。當賣方傳送計量記錄以報告客戶在 中列出的軟體即服務 (SaaS) 產品的用量時 AWS Marketplace，此 CloudTrail 日誌項目會記錄在賣方的 中 AWS 帳戶。 SaaS

```
{
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/*****",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "*****"
```

```
            },
            "eventTime": "2018-04-19T16:32:51Z",
            "eventSource": "metering-marketplace.amazonaws.com",
            "eventName": "BatchMeterUsage",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "************",
            "userAgent": "Coral/Netty14",
            "requestParameters": {
                "usageRecords": [
                    {
                        "dimension": "Dimension1",
                        "timestamp": "Apr 19, 2018 4:32:50 PM",
                        "customerIdentifier": "customer1",
                        "customerAWSAccountID": "987654321098",
                        "quantity": 1
                    }
                ],
                "productCode": "EXAMPLE_proCode"
            },
            "responseElements": {
                "results": [
                    {
                        "usageRecord": {
                            "dimension": "Dimension1",
                            "timestamp": "Apr 19, 2018 4:32:50 PM",
                            "customerIdentifier": "customer1",
                            "customerAWSAccountID": "987654321098",
                            "quantity": 1
                        },
                        "meteringRecordId": "bEXAMPLE-98f0-4e90-8bd2-bf0EXAMPLE1e",
                        "status": "Success"
                    }
                ],
                "unprocessedRecords": [ ]
            },
            "requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
            "eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
            "readOnly": false,
            "eventType": "AwsApiCall",
            "recipientAccountId": "123456789012"
        }
    ]
}
```

範例：RegisterUsage適用於容器的

下列範例顯示示範 RegisterUsage動作的 CloudTrail 日誌項目 AWS Marketplace Metering Service。當買方的 中 AWS Marketplace 部署每小時定價的容器產品時 AWS 帳戶，容器中的軟體會在買方的 RegisterUsage內呼叫 AWS 帳戶 ，以啟動該 Amazon Elastic Container Service (Amazon ECS) 任務或 Amazon Elastic Kubernetes Service (Amazon EKS) Pod 的每小時計量。此 CloudTrail 日誌項目會記錄在買方的 中 AWS 帳戶。


```
{
    "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": "eyJhbGciOiJQUzI1Ni..."
    },
    "requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
    "eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

範例：**MeterUsage**適用於 Amazon EKS 上的容器

下列範例顯示 CloudTrail 日誌項目，示範 Amazon EKS 上容器 AWS Marketplace Metering Service 的 MeterUsage動作。當具有自訂計量的容器產品 AWS Marketplace 部署在買方的 中時 AWS 帳戶，容器內的軟體會呼叫買方的 MeterUsage ，每小時 AWS 帳戶 回報一次。此 CloudTrail 日誌項目會記錄在買方的 中 AWS 帳戶。

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

```
                "mfaAuthenticated": "false",
                "creationDate": "2020-07-23T01:03:26Z"
            }
        }
    },
    "eventTime": "2020-07-23T01:38:13Z",
    "eventSource": "metering-marketplace.amazonaws.com",
    "eventName": "MeterUsage",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "1.2.3.4",
    "userAgent": "aws-cli/1.18.103 Python/3.8.2 Linux/4.14.181-142.260.amzn2.x86_64
 botocore/1.17.26",
    "requestParameters": {
        "timestamp": "Jul 23, 2020 1:35:44 AM",
        "usageQuantity": 1,
        "usageDimension": "Dimension1",
        "productCode": "EXAMPLE_proCode"
    },
    "responseElements": {
        "meteringRecordId": "bEXAMPLE-98f0-4e90-8bd2-bf0EXAMPLE1e"
    },
    "requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
    "eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

範例：在 AMIs**MeterUsage**上

下列範例顯示 CloudTrail 日誌項目，示範 Amazon Machine Image (AMIs) AWS Marketplace Metering Service 的 MeterUsage動作。當具有自訂計量來源的 AMI 產品 AWS Marketplace 部署在買方的 中時 AWS 帳戶，來自 AMI 的軟體會在買方的 MeterUsage內呼叫 AWS 帳戶 ，以每小時報告用量。此 CloudTrail 日誌項目會記錄在買方的 中 AWS 帳戶。

```
{
    "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"
}
```

故障診斷

常見的賣方註冊錯誤訊息

下表說明常見的賣方註冊錯誤訊息及其解決方案。

錯誤訊息	解決方案
您的合規要求尚未完成	對於所有日本賣方，我們需要收集其他資訊以用於法規目的，然後才能發佈付費產品。如果您是日本賣家，並有興趣列出 AWS Marketplace，請聯絡日本賣家加入團隊，電子郵件地址為 <aws-jp-marketplace@amazon.co.jp> ，以取得其他詳細資訊。如需詳細資訊，請參閱 JP SOR 常見問答集。
AWS Marketplace 由於內部問題，目前無法處理您的請求。請稍後再試。如果問題仍然存在，請聯絡我們尋求協助。	清除瀏覽器的快取，然後再試一次。如果問題仍然存在，請聯絡AWS Marketplace 賣方營運團隊。
您必須先將公有設定檔新增至賣方帳戶，才能將產品更新為公有。	在將產品更新為公有之前，將公有設定檔新增至您的賣方帳戶。在AWS Marketplace 管理入口網站中，前往設定，然後是公有設定檔。
更新您的付款設定，使其與 CurrencyCode 相容。	更新您的付款設定，使其與貨幣代碼相容。請務必新增 USD，因為使用定價不支援當地貨幣。如需詳細資訊，請參閱可用貨幣。
需要 VAT/GST 資訊才能維持付費賣方。	您有兩種選擇： 1. 更新主要帳戶中的稅務設定。檢閱稅務繼承文件，以了解這會對所有連結帳戶產生的影響。 2. 從組織中移除此帳戶，並將其作為獨立帳戶。這表示您將需要在帳單的稅務設定頁面中手動新增 VAT/GST 設定。
稅務面試位置與業務位置不相符	若要檢查目前的位置設定： 1. 稅務面試位置：

錯誤訊息	解決方案
	• 在 AWS Marketplace 管理入口網站中，前往設定，然後稅務資訊。 • 選擇更新稅務資訊，後面接著重新取得。 • 選擇更新稅務設定檔後，向下捲動並檢查您提供的國家/地區。 2. 營業據點： • 在AWS Marketplace 管理入口網站中，檢查設定索引標籤上帳戶摘要中提供的業務位置。 如果位置不相符： • 更新帳單中的稅務資訊：如需詳細資訊，請參閱帳戶位置 • 在 AMMP 設定中更新稅務面試：從稅務資訊重新進行稅務面試 確保這兩個位置都相同，以避免付款處理問題。
若要開始使用 AWS Marketplace，請完成 AWS 帳戶註冊	完成AWS 您的帳戶註冊。 如需賣方註冊的詳細資訊，請參閱 註冊為 AWS Marketplace 賣方。
正在檢閱您的公有設定檔。在驗證公有設定檔之前，您無法新增支出方法。在 2 個工作天內再次檢查。您目前只能發佈免費產品。	正在檢閱您的公有設定檔。在驗證公有設定檔之前，您無法新增支出方法並完成 LDAP。核准需要兩到三個工作天。您目前只能發佈免費產品。一旦設定檔獲得核准，您會收到一封電子郵件至您的根電子郵件地址。 如果您仍然無法在三個工作天後新增付款資訊，請聯絡AWS Marketplace 賣方營運團隊。
您的營業據點位於不支援的司法管轄區。您不符合發佈付費產品的資格。	您的營業據點位於不支援的司法管轄區。您不符合發佈付費產品的資格。如需合格司法管轄區的詳細資訊，請參閱 付費產品的合格司法管轄區。
AWS 帳戶未註冊為 中的賣方 AWS Marketplace。	您未註冊為 中的賣方 AWS Marketplace。從 AWS Marketplace 管理入口網站完成註冊。

錯誤訊息	解決方案
賣方必須擁有公開設定檔，才能成為付費賣方。	您必須完成並核准您的賣方公開設定檔。完成它，並在 AWS Marketplace Management Portal 中檢查您的帳戶狀態。

文件歷史紀錄

下表說明此版本 AWS Marketplace 賣方指南的文件。

如需有關本文件更新的通知，您可以訂閱 RSS 摘要。

變更	描述	日期
新增將支出偏好設定設定為賣方註冊的步驟	如需賣方註冊步驟https://docs.aws.amazon.com/marketplace/latest/userguide/seller-registration-process.html，請參閱 。	2025 年 5 月 21 日
聯邦增補合約已新增至標準 Marketplace 合約	附錄可協助簡化美國政府的採購。如需詳細資訊，請參閱在 中使用標準化合約 AWS Marketplace。	2025 年 4 月 29 日
SaaS 準則更新	新增 "Deployed on AWS" 指定的相關資訊。請參閱 https://docs.aws.amazon.com/marketplace/latest/userguide/saas-guidelines.html，了解詳細資訊。	2025 年 4 月 8 日
授權書	新增letter-of-authorization範本，用於授予某人代表公司開立 AWS 帳戶的許可。如需詳細資訊，請參閱以 AWS Marketplace 賣方身分註冊。	2025 年 4 月 8 日
已將自助式清單資訊新增至 ML	如需自助式清單資訊https://docs.aws.amazon.com/marketplace/latest/userguide/machine-learning-products，請參閱 。	2025 年 3 月 26 日

更新 受管政策	下列許可已新增至AWSMarketplaceSellerFullAccess政策：ListSupplementalTaxRegistrations 、DeleteSupplementalTaxRegistratio 、 PutSupplementalTaxRegistration 和 GetTaxRegistration 。	2024 年 12 月 20 日
使用促銷媒體增強您的 AWS Marketplace 產品	新增如何使用促銷媒體的指南。	2024 年 12 月 17 日
更新 受管政策	ListTasks 、UpdateTasks 、 DescribeTask 和 CompleteTasks 許可已從下列 AWS 受管政策中移除： • AWSMarketplaceSellerFullAccess • AWSMarketplaceSellerProductsFullAccess • AWSMarketplaceSellerProductsReadOnly	2024 年 12 月 10 日
新增使用 AWS 功能購買的文件	使用 購買 AWS 可讓 AWS Marketplace 賣方在自己的網站上顯示和銷售 AWS Marketplace 產品。 AWS Marketplace 賣方可以與 AWS Marketplace call-to-action(CTA) 按鈕和 APIs 整合，以在賣方擁有的網站上建置 AWS採用 技術的軟體探索和採購體驗。	2024 年 12 月 4 日

EC2 Image Builder 元件型產品	新增 EC2 Image Builder 元件型產品做為產品交付方法。	2024 年 12 月 1 日
新的 AWS 合作夥伴助理	AWS Partner Assistant 是 AWS Partners 的生成式 AI 支援聊天機器人。它可以從 Partner Central 和 AWS Marketplace Portal (AMMP) 存取。	2024 年 11 月 25 日
更新 受管政策	下列政策現在具有 `uploadFiles` 許可： • awsmarketplacefullaccess • AWSMarketplaceSellerFullAccess	2024 年 11 月 18 日
新增了新的 受管政策	新增`AWSMarketplaceSellerOfferManagement` 文件。	2024 年 11 月 18 日
更新 受管政策	下列 AWS 受管政策現在具有 `DescribeAssessment` 和 `ListAssessments` 許可： • AWSMarketplaceSellerFullAccess • AWSMarketplaceSellerProductsFullAccess • AWSMarketplaceSellerProductsReadOnly	2024 年 10 月 9 日
已移除已棄用的許可	更新`AWSMarketplaceSellerFullAccess` 文件以移除三個動作。	2024 年 6 月 4 日

已更新 Amazon Machine Image (AMI) 年度協議修訂選項	賣方現在可以新增或切換現有訂閱的執行個體類型。	2024 年 5 月 30 日
更新了解您的客戶 (KYC) 程序	為賣方新增了額外的步驟至 銷售商的 GMT 程序。	2024 年 5 月 21 日
已更新 AWS Marketplace 賣方的私有優惠體驗	新增建立和管理私有優惠的增強體驗內容。	2024 年 5 月 20 日
已更新 Amazon EKS 附加元件產品的需求	更新「將容器產品準備為 AWS Marketplace 附加元件」一節，並新增「附加元件組態需求和附加元件提供者的最佳實務」。	2024 年 5 月 8 日
已更新 AWS Marketplace 賣方的許可	更新許可範例以修正語法錯誤。	2024 年 4 月 2 日
更新 SaaS 合約定價	更新了與使用pay-as-go定價模型報告 SaaS 合約超額相關的內容。	2024 年 4 月 2 日
上的新示範和私有優惠選項 AWS Marketplace	AWS Marketplace 現在支援特定賣家產品詳細資訊頁面上的示範和私有優惠請求選項。	2024 年 4 月 1 日
用於稅務的新儀表板	AWS Marketplace 現在支援稅務儀表板，可為 中的交易提供美國和國際稅務的視覺化和精細資料 AWS Marketplace。	2024 年 3 月 29 日
更新了 AWSMarketplaceGetEntitlements	已sid新增至 AWSMarketplaceGetEntitlements 受管政策。	2024 年 3 月 22 日
已更新專業服務程序	更新編輯產品可見性和移除專業服務產品程序。	2024 年 3 月 19 日

更新 AMI 存取政策	更新章節以釐清 Linux 特定和類似 Unix 的 AMI 存取政策。	2024 年 3 月 19 日
新增 的搜尋引擎最佳化資訊 AWS Marketplace	新增與 搜尋引擎最佳化相關的內容 AWS Marketplace。	2024 年 3 月 19 日
更新 中賣方的受管政策 AWS Marketplace	更新`AWSMarketplaceSellerFullAccess` 以新增與建立服務連結角色相關的許可。	2024 年 3 月 15 日
中產品的新服務連結角色 AWS Marketplace	AWS Marketplace 現在提供服務連結角色，可讓您存取 AWS Marketplace 為轉售授權而使用或管理 AWS 的服務和資源。	2024 年 3 月 15 日
更新 中賣方的受管政策 AWS Marketplace	更新`AWSMarketplaceSellerFullAccess` 以新增與存取稅務資訊相關的許可。	2024 年 2 月 8 日
安全性報告的 Amazon EventBridge 事件	AWS Marketplace 當安全性漏洞報告可供賣方的產品使用時， 現在支援 Amazon EventBridge 事件，先前稱為 Amazon CloudWatch Events。	2024 年 1 月 31 日
支援 Amazon EKS 附加元件	新增從 AWS Marketplace 容器型產品發佈至 Amazon EKS 附加元件的相關內容和程序。	2024 年 1 月 29 日
新增對專業服務銷售商轉售產品的支援	賣方現在可以為管道合作夥伴建立作為獨立軟體供應商 (ISVs 轉售機會。	2024 年 1 月 18 日
中未來日期協議的一般可用性 AWS Marketplace	所有 ISVs和 AWS Marketplace 頻道合作夥伴都可以指定未來的開始日期，做為以預付定價發佈私有優惠的一部分。	2024 年 1 月 16 日

新的儀表板以供使用	AWS Marketplace 現在支援用量儀表板，可為使用 SaaS 和伺服器用量型產品的客戶提供視覺化和精細資料。	2024 年 1 月 10 日
適用於賣方的新 Quick Launch 部署選項	新增與軟體即服務 (SaaS) 產品新 Quick Launch 部署選項相關的內容和程序。	2023 年 11 月 29 日
彈性付款排程適用於私有優惠	私有優惠的彈性付款排程 (FPS) 現在可供 中的所有客戶使用 AWS Marketplace。	2023 年 11 月 17 日
新增容器產品的自助式內容	新增與容器產品的自助式動作相關的內容和程序。	2023 年 11 月 3 日
適用於協議和續約的新儀表板	AWS Marketplace 現在為賣方提供協議和續約儀表板。	2023 年 10 月 31 日
變更集的 Amazon EventBridge 事件	AWS Marketplace 當變更集完成且狀態為成功、失敗或取消時，現在支援 Amazon EventBridge 事件，先前稱為 Amazon CloudWatch Events。	2023 年 10 月 31 日
更新 AWS Marketplace Vendor Insights 中賣方的受管政策	AWS Marketplace 已更新`AWSVendorInsightsVendorFullAccess` 以新增更新資料來源的許可。	2023 年 10 月 18 日
結束 AWS Marketplace 對 桌面應用程式的支援 (AMDA)	AWS Marketplace 已於 2023 年 10 月 2 日終止對 AMDA 的支援。與 AMDA 相關的所有內容和程序已從指南中移除。	2023 年 10 月 2 日
新增 SaaS 產品的自助式內容	新增軟體即服務 (SaaS) 產品自助式動作的相關內容和程序。	2023 年 9 月 12 日

AWS Marketplace 賣方現在可使用每日支出	中的賣方 AWS Marketplace 管理入口網站 現在可選擇每天或每月接收付款。	2023 年 9 月 7 日
支援 Amazon EventBridge	AWS Marketplace 現在支援賣方收到新優惠時的 EventBridge 事件。	2023 年 9 月 6 日
AWS Marketplace Vendor Insights 的新自助式加入體驗	AWS Marketplace Vendor Insights 現在支援自助式加入體驗。	2023 年 8 月 17 日
更新 中賣方的受管政策 AWS Marketplace	更新AWSMarketplaceSellerFullAccess 以新增與共用實體相關的許可。	2023 年 6 月 1 日
更新 中賣方的受管政策 AWS Marketplace	更新AWSMarketplaceSellerFullAccess 以新增與帳戶驗證、銀行帳戶驗證、案例管理和賣方通知詳細資訊相關的許可。	2023 年 6 月 1 日
新增有關 精細存取的內容 AWS Marketplace	新增 中精細存取的概觀資訊、許可和程序 AWS Marketplace 管理入口網站。	2023 年 6 月 1 日
新增 AMI 產品的程序	新增 AMI 產品的自助式程序。	2023 年 5 月 12 日
在 上結束對 Amazon Tax Exemption Program 和美國稅務計算服務的支援 AWS Marketplace	Amazon Tax Exemption Program 和美國稅務計算服務已從指南中移除，因為此服務的支援已於 2023 年 3 月 6 日終止。	2023 年 3 月 6 日
新增容器產品的程序	新增了對容器產品設定進行變更的新程序。	2023 年 2 月 13 日

結束對 AWS Marketplace 產品支援連線的支援	AWS Marketplace 自 2022 年 11 月 30 日起，不再支援使用 Commerce Analytics Service 連線和共用客戶聯絡詳細資訊。已從指南 AWS Marketplace 中移除 AWS Marketplace 產品支援連線內容。	2023 年 1 月 27 日
諮詢合作夥伴名稱變更	AWS Marketplace 現在將諮詢合作夥伴稱為管道合作夥伴。本指南已更新，僅反映名稱變更。	2023 年 1 月 26 日
客戶服務團隊名稱變更	Managed Catalog Operations (MCO) 團隊將其名稱變更為 AWS Marketplace 賣方營運團隊。本指南已更新，僅反映名稱變更。	2023 年 1 月 24 日
私有優惠頁面	已驗證的買方現在可以 AWS 帳戶 在 AWS Marketplace 私有優惠 頁面上檢視延伸到其的私有優惠。	2023 年 1 月 19 日
新增有關 SaaS 產品設定的主題	新增了新主題，其中包含變更 SaaS 產品設定的程序。	2023 年 1 月 6 日
更新 中賣方的受管政策 AWS Marketplace	AWS Marketplace 已更新`AWSMarketplaceSellerFullAccess` 以新增存取賣方儀表板的許可。	2022 年 12 月 23 日
已更新賣方的電子郵件通知	發佈私有優惠時，賣方現在會收到通知。	2022 年 12 月 22 日
上的賣方現在可以使用訂閱的 SaaS 免費試用 AWS Marketplace	賣方現在可以為訂閱產品建立免費試用。	2022 年 12 月 16 日

在 上更新 AMI 自助服務清單（第 2 版）的賣方體驗 AWS Marketplace	上的賣方 AWS Marketplace 現在可以建立單一 AMI 產品自助式清單。賣方可以自行進行更新，無需額外的處理等待時間。	2022 年 12 月 14 日
已更新標籤型授權的三個政策	已更新 AWS Marketplace 標籤型授權功能的三個政策 `AWSMarketplaceSellerFullAccess` (`AWSMarketplaceSellerProductsFullAccess` 、 和 `AWSMarketplaceSellerProductsReadOnly`)。	2022 年 12 月 9 日
更新 AWS Marketplace Vendor Insights 中賣方的政策	更新 AWS Marketplace Vendor Insights 賣方`AWSVendorInsightsAssessorFullAccess` `AWSVendorInsightsVendorReadOnly` 的 受管政策和 。	2022 年 11 月 30 日
在 AWS Marketplace Vendor Insights 中控制賣方的存取權	新增 AWS Marketplace 了 Vendor Insights 主題，以描述可供賣方使用的動作和許可。	2022 年 11 月 30 日
已更新 AWS Marketplace 廠商洞見的四個受管政策	已更新 AWS Marketplace Vendor Insights 的 `AWSVendorInsightsVendorFullAccess` 和 `AWSVendorInsightsVendorReadOnly` 受管政策。	2022 年 11 月 28 日

賣方可以在 Amazon EKS 中發佈附加元件產品	AWS Marketplace 和 Amazon EKS 的整合可讓賣方在 Amazon EKS 主控台中向買方展示其產品。	2022 年 11 月 28 日
更新 AWS Marketplace Vendor Insights 設定	更新 AWS Marketplace Vendor Insights 的設定程序。	2022 年 11 月 18 日
已更新 AWS Marketplace 兩個 Vendor Insights 政策	針對 AWS Marketplace Vendor Insights 更新了兩個政策 `AWSMarketplaceSellerFullAccess` `AWSMarketplaceSellerProductsFullAccess` 和 。	2022 年 7 月 26 日
為 AWS Marketplace Vendor Insights 新增了兩項政策，這項功能提供軟體風險評估。	`AWSVendorInsightsVendorFullAccess` `AWSVendorInsightsVendorReadOnly` 為 AWS Marketplace Vendor Insights 新增了兩個政策和一項功能，提供軟體風險評估。	2022 年 7 月 26 日
AWS Marketplace Vendor Insights 是新增至 的新功能 AWS Marketplace	AWS Marketplace Vendor Insights 是一種提供軟體風險評估的功能。 AWS Marketplace Vendor Insights Vendor Insights 是一種提供軟體風險評估的功能。	2022 年 7 月 26 日
AWS Marketplace Commerce Analytics Service 許可更新	Commerce Analytics Service AWS Marketplace 具有其他 IAM 許可。	2022 年 7 月 21 日
已新增賣方交付資料饋送服務區段	僅文件更新，以新增賣方交付資料饋送服務區段，並重新組織與資料饋送相關的區段。	2022 年 6 月 15 日

新增補充報告區段	針對為最近功能啟動 AWS Marketplace 提供的補充報告，新增了新章節。	2022 年 6 月 14 日
SaaS 合約免費試用現在可供上的賣方使用 AWS Marketplace	賣方現在可以建立免費試用，無需額外的開發工作，方法是定義免費試用長度、試用期間可用的維度，以及客戶獲得的免費用量容量。	2022 年 5 月 31 日
新增至買方和賣方交易的電子郵件通知	新功能可向買方和賣方傳送電子郵件通知，以驗證 中的優惠和協議 AWS Marketplace。	2022 年 5 月 23 日
新增至建立機器學習產品清單的範例	機器學習區段的僅文件更新，包含範例，以在建立機器學習產品時顯示賣方檢視和買方檢視的比較。	2022 年 4 月 22 日
機器學習區段的更新	機器學習區段已進行僅限文件的更新，以釐清程序。	2022 年 4 月 15 日
以色列現在是符合資格的司法管轄區	以色列的居民現在有資格成為賣方 AWS Marketplace。	2022 年 4 月 13 日
針對優惠資料饋送中的新項目更新	製造商現在會收到其管道合作夥伴所建立優惠的優惠資訊。這包括有關帳戶是記錄賣方的優惠，以及優惠的製造商。	2022 年 3 月 29 日
經銷商機會通知	賣方現在能夠接收經銷商機會的通知。	2022 年 3 月 28 日
已將影片新增至專業服務產品	將專業服務產品頁面更新為影片，其中包含如何管理服務產品的詳細資訊。	2022 年 2 月 24 日

部署無伺服器 SaaS 整合解決方案的新主題	新增整合無伺服器 SaaS 部署的新資訊，包括 AWS Quick Start 的連結，以取得部署步驟的參考。	2022 年 2 月 15 日
容器型需求和 AMI 區段的最低更新	容器型需求中政策的最少更新，並移除 AMI 定價合約的不正確資訊	2022 年 2 月 14 日
容器版本控制更新	僅文件更新，以釐清如何將容器映像和其他成品推送至儲存庫。	2022 年 2 月 10 日
更新 SaaS 產品的`ResolveCustomer` 程式碼範例	SaaS 產品的`ResolveCustomer` 程式碼範例已更新為包含 `CustomerAWSAccountID` 。	2022 年 2 月 3 日
新增與 AWS Marketplace for Containers Anywhere 產品 AWS License Manager 整合的文件	僅文件更新，透過與 License Manager 整合，將合約定價新增至 AWS Marketplace Containers Anywhere 產品的詳細指引。	2022 年 2 月 1 日
SaaS SNS 通知更新	僅文件更新，以釐清 SaaS 通知訊息。	2022 年 1 月 25 日
賣方能夠透過 Amazon Web Services EMEA SARL 與 EMEA 型買方交易	符合資格的 AWS Marketplace 賣方現在可以透過 Amazon Web Services EMEA SARL 與位於歐洲、中東和非洲 (EMEA) 國家和地區的客戶 AWS 帳戶 交易。	2022 年 1 月 7 日

新增使用 Helm Chart 交付方法為容器型產品建立交付選項的文件	賣方現在可以使用 Helm Chart 交付方法提供交付選項。買方可以使用這些交付選項，在其啟動環境中安裝賣方提供的 Helm Chart，以啟動容器型應用程式。提供 Helm Chart 交付方法時，賣方可以為買方啟用快速啟動。快速啟動是一項功能，買方可用來 AWS CloudFormation 快速建立新的 Amazon EKS 叢集，並在其上啟動容器型應用程式。	2021 年 11 月 29 日
現有政策的更新	已更新 AWS Marketplace 賣方的安全政策。	2021 年 11 月 22 日
AMI 和容器型產品的合約定價	獨立軟體廠商 ISVs) 現在可列出新的 AMI 型產品或容器型產品，並為買方提供預先合約定價。	2021 年 11 月 17 日
供應商計量標記	僅文件更新廠商計量標記，包括程式碼範例。	2021 年 11 月 11 日
Amazon Simple Notification Service for AMI 或容器產品	當客戶透過 Amazon Simple Notification Service 訂閱或取消訂閱 AMI 或容器產品時，獨立軟體廠商 (ISVs) 可以接收通知。	2021 年 11 月 10 日
新的賣方許可	AWS Marketplace 新增存取中優惠和合作夥伴索引標籤的新許可 AWS Marketplace 管理入口網站。	2021 年 11 月 9 日

能夠部署為機器學習產品非同步推論設定的端點	對於預期承載推論高於最大值的機器學習軟體，或需要超過每次調用處理時間上限的處理時間，買方可以部署針對 Amazon SageMaker AI 非同步推論設定的端點。	2021 年 11 月 8 日
退款政策和核准	僅文件更新，以釐清退款政策，並將所有退款資訊移至 AWS Marketplace 賣方指南中的一個集中位置。	2021 年 8 月 20 日
選取或上傳諮詢合作夥伴私有優惠的 EULA	為諮詢合作夥伴建立轉售機會時，獨立軟體供應商現在可以選取或上傳最終使用者授權合約 (EULA)。	2021 年 8 月 17 日
SaaS 合約產品的自訂產品維度	獨立軟體廠商 (ISVs) 現在可在為諮詢合作夥伴建立轉售機會時自訂 SaaS 合約產品維度。	2021 年 8 月 17 日
AWS Marketplace 欄位示範計劃	僅文件更新，以釐清 AWS Marketplace 欄位示範計畫的 AWS Data Exchange 資料集產品需求。	2021 年 8 月 3 日
SaaS 產品指導方針更新	SaaS 產品的產品準則已更新。	2021 年 7 月 29 日
容器型產品需求更新	容器型產品需求已更新。	2021 年 7 月 29 日
AMI 安全政策更新	AMI 產品的安全政策已更新。	2021 年 7 月 29 日
更符合資格的司法管轄區	下列項目現在有資格成為 上的賣方 AWS Marketplace：香港特別行政區和卡達。	2021 年 6 月 23 日
資料饋送概觀	僅文件更新，提供賣方可用資料摘要結構的概觀。	2021 年 6 月 23 日

更新機器學習章節	僅文件更新有關建立和維護機器學習產品的資訊。	2021 年 5 月 27 日
容器產品的自助式更新	賣方現在可以更簡單、更快速地透過 更新其容器型產品 AWS Marketplace 管理入口網站。	2020 年 12 月 17 日
專業服務	賣方現在可以為 AWS Marketplace 買方提供專業服務。已將專業服務章節新增至文件。	2020 年 12 月 3 日
AMI 產品的自助式更新	賣方現在可以更簡單、更快速地透過 更新以 Amazon Machine Image (AMI) 為基礎的產品 AWS Marketplace 管理入口網站。	2020 年 11 月 23 日
更符合資格的司法管轄區	下列 現有資格成為 上的賣方 AWS Marketplace：巴林、挪威、瑞士和阿拉伯聯合大公國（阿拉伯聯合大公國）。	2020 年 6 月 17 日
您可以針對接受的私有優惠提供升級和續約	對於使用消費產品的 SaaS 合約和 SaaS 合約，您可以使用先前接受的私有優惠提供升級和續約。	2020 年 5 月 28 日
如需詳細資訊，請參閱 資料摘要	來自報告的詳細資訊會細分為較小的資料饋送，以簡化尋找和分析資料。	2020 年 5 月 21 日
標準化授權條款現已推出	您可以提供標準化授權條款來取代自訂 EULAs，以簡化簽約程序。	2020 年 4 月 28 日

澳洲和紐西蘭是符合資格的司法管轄區	下列項目現在有資格成為 上的賣方 AWS Marketplace：(i) 澳洲 (AU) 紐西蘭 (NZ) 的永久居民和公民，或 (ii) 在這些區域組織或註冊的商業實體。	2020 年 4 月 2 日
容器產品現在支援自訂計量和定價增強功能	如果您想要定義自己的定價單位和計量，以向我們計費， 會與 AWS Marketplace 計量服務`meterUsage` 的動作整合。	2019 年 11 月 14 日
AWS Marketplace 透過 AWS Data Exchange 支援資料產品	您現在可以在 中提供資料產品 AWS Marketplace。	2019 年 11 月 13 日
AWS Marketplace Catalog API 服務簡介	Catalog AWS Marketplace API 服務為核准的賣方提供 API 界面，以程式設計方式管理其產品。	2019 年 11 月 12 日
AWS Marketplace 支援付費的每小時容器	AWS Marketplace 現在支援在 Amazon Elastic Kubernetes Service (Amazon EKS) 上執行的付費每小時容器。	2019 年 9 月 25 日
更新 AMI 產品功能	您現在可以使用 部署 AMIs 和 Lambda 函數 AWS CloudFormation。	2019 年 9 月 11 日
新增安全性章節	在新的安全性區段下合併安全性內容。	2019 年 5 月 7 日
已更新 AMI 安全政策	更新 AMI 產品的安全政策	2019 年 4 月 11 日
已將版本控制資訊新增至 Machine Learning 產品區段	新增描述機器學習產品的產品版本功能的內容。	2019 年 3 月 21 日
新增Machine Learning產品區段	新增發佈機器學習產品的內容	2018 年 11 月 28 日

新增容器型產品區段	新增發佈容器型產品的內容。	2018 年 11 月 27 日
更新提交賣方說明請求的連結	將電子郵件地址變更為 webform 地址。	2018 年 10 月 22 日
新增使用內容的 SaaS 合約	重組 SaaS 內容並新增內容，以支援發行具有取用功能的 SaaS 合約。	2018 年 10 月 18 日
新增有關私有優惠彈性付款排程的內容	新增內容以支援發行適用於私有優惠內容的彈性付款排程器。	2018 年 10 月 15 日
已更新 IAM 許可內容	新增內容以支援 AMMP 唯讀存取的新 IAM 許可。	2018 年 10 月 9 日
新增有關諮詢合作夥伴私有優惠的內容	新增內容以支援諮詢合作夥伴私人優惠功能版本。	2018 年 10 月 9 日
新增私有映像組建的相關內容	新增內容以支援發行 Private Image Build for AMIs功能。	2018 年 8 月 13 日
已新增賣方的搜尋引擎最佳化指引。	為想要將其產品最佳化以進行搜尋的賣方新增了指導。	2018 年 7 月 3 日
更新 AWS Marketplace 標誌的連結	已更新指向新 AWS Marketplace 標誌的連結。	2018 年 6 月 12 日
新增賣方指南	將所有 PDF 賣方指南轉換為線上內容。	2018 年 5 月 9 日

AWS 詞彙表

如需最新的 AWS 術語，請參閱 AWS 詞彙表 參考中的AWS 詞彙表。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。