卖家指南

AWS Marketplace

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Marketplace: 卖家指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 AWS Marketplace？ 1
AWS Marketplace 作为卖家使用 1
的合同结构 AWS Marketplace 3
中的产品定价 AWS Marketplace 4
以卖方身份开始使用 5
发布免费软件产品的卖方要求 5
付费产品的其他卖方要求 5
适用于付费产品的符合条件的司法管辖区 6
AWS Marketplace 管理门户 7
卖家注册过程 8
创建您的公开资料 9
提供税务信息 9
提供美国银行账户信息 10
完成“客户背景调查”流程 11
完成银行账户验证流程 13
（可选）为“客户背景调查”过程添加辅助用户 14
付款和买家账单 15
已经是卖家？ 16
投诉处理政策 – Amazon Payments Europe 16
上架费 18
公开优惠上架费 18
专属优惠上架费 18
渠道合作伙伴私募报价 (CPPO) 上市费用 18
专业服务上架费 18
卖方工具包 19
AWS Marketplace 商务分析服务 19
AWS Marketplace 现场演示计划 34
更多资源在 AWS Marketplace 管理门户 35
准备您的产品 36
产品交付 36
产品定价 39
定价模式 39
更改定价模式 43
更改价格 43

专属优惠 44
产品退款 44
区域和国家/地区 46
AWS 区域 46
国家/地区 47
标准化合同 47
的标准合同 AWS Marketplace 48
的经销商合同 AWS Marketplace 49
类别和元数据 50
命名和描述您的产品 50
选择类别和关键词 51
AMI和容器产品使用说明 51
..... 50
撰写 CloudFormation 配送说明 53
监控和评估应用程序功能 53
轮换编程系统凭证和加密密钥 54
产品搜索引擎优化 54
搜索引擎优化 54
AWS Marketplace 搜索 55
准备您的私人报价 58
专属优惠的工作原理 58
私人报价注意事项 59
为买家提供私人优惠体验 59
专属优惠报告 60
支持的产品类型 61
AMI 产品的专属优惠 61
容器产品的专属优惠 61
专业服务产品的专属优惠 62
SaaS 产品的专属优惠 62
机器学习产品的专属优惠 62
创建和管理私密优惠 63
开始新的私人报价 64
了解报价状态 64
起草和发布私人报价 65
向买家发送专属优惠 66
保存您的私人报价进度 66

更新私募报价的到期时间 67
取消私人报价 67
渠道合作伙伴专属优惠 68
其他信息 69
创建销售授权作为 ISV 69
分期付款计划 72
创建付款计划 72
报告分期付款计划 73
私人优惠升级、续订和修改 73
管理私人报价协议 73
私募报价修改支持的产品类型 74
升级和续订的提交流程 74
报告升级、续订和修订 75
远期协议 76
创建未来日期的协议 77
使用具有 future 日期协议的分期付款计划 77
接收有关未来日期协议的通知 77
使用未来日期协议转售渠道合作伙伴专属优惠 78
AMI基于产品 79
AMI基于产品的交付方式 79
其他 资源 79
AMI基于理解的产品 80
产品生命周期 80
AMI产品代码 82
更改请求 82
产品加载表单 83
年度协议修订 84
创建单AMI品 84
先决条件 85
了解自助服务体验 85
创建清单 86
其他 资源 88
管理单一AMI产品 88
创建更改请求 89
获取更改请求的状态 91
更新产品信息 92

更新允许列表（预览账户） 93
更新产品可见性 93
添加一个 AWS 区域 94
限制 AWS 区域 94
更新对未来 AWS 区域的支持 95
添加实例 96
限制实例 97
更新版本信息 97
添加新版本 98
限制版本 101
更新定价 102
按国家/地区更新可用性 103
更新你的 EULA 104
更新退款政策 104
允许 AWS Marketplace 访问您的 AMI 105
从 AWS Marketplace中删除产品 106
解决提交更改请求时的常见错误 107
AMI基于交付使用 CloudFormation 108
生成产品列表 108
准备 CloudFormation 模板 109
获取模板基础设施的成本估算 111
架构示意图 111
满足提交要求 111
提交您的产品请求 113
添加无服务器应用程序组件 113
建筑最佳实践 AMIs 121
保护转售权 121
建造一个 AMI 121
为你AMI做好准备和保护 AWS Marketplace 122
扫描您的AMI发布要求 123
验证您的软件是否正在您的计算机上运行 AWS Marketplace AMI 123
AMI 产品定价 124
AMI 定价模式 125
AWS 费用和软件费用 127
AMI 产品的自定义计量定价 128
AMI产品的合同定价 136

AMI产品计费、计量和许可集成 139
使用 AWS Marketplace Metering Service对 AMI 产品进行自定义计量 139
使用 AWS License Manager的 AMI 产品的合同定价 144
亚马逊AMI商品SNS通知 156
Amazon SNS 主题：aws-mp-subscription-notification 156
为亚马逊SQS队列订阅亚马逊主题 SNS 157
AMI 产品核对清单 157
基于 AMI 的产品要求 159
安全策略 159
访问策略 160
客户信息策略 161
产品使用策略 161
架构策略 162
AMI 产品使用说明 163
基于容器的产品 164
获取帮助 164
容器产品入门 165
先决条件 165
创建容器产品 166
产品生命周期 166
更新产品可见性 167
更新允许名单 AWS 账户 IDs 168
添加定价维度 168
更新维度信息 169
更新定价条款 169
按国家/地区更新可用性 169
更新最终用户许可协议 170
更新产品的退款政策 170
为您的容器产品创建产品 ID 和产品代码 171
创建初始列表 171
AWS Marketplace Metering Service 为您的容器产品集成 172
AWS License Manager 为您的容器产品集成 172
添加产品的新版本 172
测试并发布您的产品 179
更新版本信息 181
限制您的 Amazon EKS 附加组件的某个版本 182

为容器产品创建或更新产品信息 182
容器产品扫描是否存在安全问题 183
基于容器的产品要求 183
安全要求 184
访问要求 184
客户信息要求 184
产品使用要求 184
架构要求 185
容器产品使用说明 186
亚马逊 EKS 附加商品的要求 186
容器产品定价 199
容器定价模式 200
容器产品的合同定价 203
容器产品计费、计量和许可集成 206
采用 AWS Marketplace Metering Service按小时计量和自定义计量 206
与的合同定价 AWS License Manager 209
使用 AWS Marketplace 计量服务按小时计 210
容器产品自定义计量 220
集装箱产品的合同定价 AWS License Manager 231
亚马逊关于集装箱商品的SNS通知 261
Amazon SNS 主题：`aws-mp-subscription-notification` 262
为亚马逊SQS队列订阅亚马逊主题 SNS 262
机器学习产品 263
机器学习产品入门 263
SageMaker 模型包 263
SageMaker 算法 263
部署推理模型 264
安全性和知识产权 264
保护知识产权 264
无网络访问 265
客户数据安全 265
机器学习产品 265
基础设施定价 265
软件定价 265
在里面准备好你的产品 SageMaker 267
将代码打包成映像 268

上传映像 289
创建您的亚马逊 SageMaker资源 291
在 AWS Marketplace中发布产品 297
发布过程概述 297
所需权限 297
创建产品列表 298
测试您的产品 305
签署发布 305
更新您的产品 306
创建机器学习产品的要求和最佳实操 307
所需资产 308
机器学习产品的一般最佳实操 308
使用信息要求 309
输入和输出的要求 309
Jupyter 笔记本的要求 310
机器学习产品列表的要求和建议摘要 311
服务限制和限额 313
网络隔离 314
映像大小 314
存储大小 314
实例大小 314
推理的有效负载大小 314
推理的处理时间 314
服务限额 314
异步推理 315
无服务器推理 315
托管的竞价型训练 315
Docker 镜像和 AWS 账户 315
根据内置算法发布模型包或 AWS Marketplace 315
AWS 区域 支持发布 315
故障排除 316
机器学习报告 317
每日业务报告 317
月度收入报告 318
支付报告 318
其他报告和分析 318

基于 SaaS 的产品 319
开始使用 SaaS 产品 319
先决条件 319
SaaS 产品生命周期 320
创建 SaaS 产品 321
创建初始 SaaS 产品页面 323
SaaS 产品设置 324
集成 SaaS 订阅产品 333
集成 SaaS 合同产品 336
将你的 SaaS 合同与 pay-as-you-go 产品集成 338
部署无服务器 SaaS 集成解决方案 342
规划您的 SaaS 产品 343
规划您的定价 344
规划您的账单集成 344
规划您的 Amazon SNS 集成 344
规划客户访问您的产品的方式 344
SaaS 产品准则 345
产品设置准则 345
客户信息要求 346
产品使用准则 346
架构准则 346
SaaS 产品定价 347
SaaS 订阅定价 348
SaaS 合同定价 350
SaaS 免费试用 353
创建 SaaS 免费试用优惠 354
取消 SaaS 免费试用优惠 354
SaaS 客户载入 355
配置您的 SaaS 产品以接受新买家 355
SaaS 产品的亚马逊SNS通知 357
Amazon SNS 主题：`aws-mp-entitlement-notification` 358
Amazon SNS 主题：`aws-mp-subscription-notification` 359
为SQS队列订阅主题 SNS 360
访问 AWS Marketplace 计量和授权服务 APIs 360
用量的计量 361
检查权利 367

SaaS 产品集成核对清单 368
报告 371
SaaS 代码示例 371
ResolveCustomer 代码示例 372
GetEntitlement 代码示例 372
BatchMeterUsage 代码示例 374
带有用量分配标签的 BatchMeterUsage 代码示例（可选） 375
AWS PrivateLink 与一起使用 AWS Marketplace 378
简介 378
配置产品 379
将您的产品提交至 AWS Marketplace 379
买家访问VPC终端节点 380
附录：核对清单 381
专业服务产品 383
获取帮助 383
开始使用专业服务产品 383
先决条件 384
创建专业服务产品 384
创建专属优惠 385
编辑产品信息 387
编辑产品定价 387
编辑产品可见性 388
删除专业服务产品 389
产品详细信息 389
产品描述 390
其他 资源 391
支持信息 391
定价维度 391
产品可见性 392
产品要求 392
产品设置准则 392
客户信息要求 392
产品使用准则 393
架构准则 393
专业服务产品定价 393
数据产品 395

提交您的产品 396
使用“产品”选项卡 397
公司和产品徽标要求 398
提交付费重新打包软件的要求 398
对带有硬件组件的产品的要求 399
AWS CloudFormation-已推出的产品（免费或付费）或基于使用量的付费产品 AMI 400
更新您的产品 403
产品更改和更新 404
时限和预期 404
正在提交AMIs给 AWS Marketplace 405
AMI自助扫描 405
AMI克隆和产品代码分配 406
最终核对清单 406
营销您的产品 408
为期 180 天的学院 GTM 408
宣传您的产品发布 408
AWS Marketplace 消息传递 408
的评论 AWS Marketplace 409
链接到 AWS Marketplace 410
使用 AWS Marketplace 品牌推广 410
通过以下网址直接链接到您的产品 AWS Marketplace 411
新闻稿 411
AWS Marketplace 商标使用指南 412
通知 414
电子邮件通知 414
事件类型 415
管理通知 415
亚马逊 EventBridge 通知 416
AWS Marketplace 为API亚马逊 EventBridge 活动编目 416
Amazon Simple Notification Service 通知 424
卖家报告、数据源和控制面板 425
卖家传输数据源服务 425
数据源的存储和结构 426
访问数据源 428
取消订阅数据源 431
使用数据源 432

数据源表概述 432
数据源查询示例 440
数据源 563
卖家报告 599
访问报告 600
每日业务报告 600
每日客户订阅用户报告 609
支付报告 611
月度账单收入报告 618
销售补偿报告 626
补充报告 628
协议详细信息报告 629
卖家控制面板 630
访问控制面板 600
财务运营控制面板 632
销售运营控制面板 656
用于营销的仪表板 670
AWS Marketplace 供应商见解 681
了解 AWS Marketplace 供应商见解 682
以卖家的身份设置 682
创建安全配置文件 683
上传认证文件 683
上传自我评估 685
启用 AWS Audit Manager 自动评估 686
编辑配置文件 691
以卖家身份查看您的安全配置文件 691
管理快照 692
创建快照 693
查看快照 694
导出快照 694
查看最新发布的快照 694
推迟快照发布 695
更改快照列表的首选项 695
控制访问权限 695
AWS Marketplace 供应商洞察卖家的权限 696
`CreateDataSource` 696

DeleteDataSource ... 697
GetDataSource ... 697
UpdateDataSource ... 697
ListDataSources ... 697
CreateSecurityProfile ... 697
ListSecurityProfiles ... 698
GetSecurityProfile ... 698
AssociateDataSource ... 698
DisassociateDataSource ... 698
UpdateSecurityProfile ... 699
ActivateSecurityProfile ... 699
DeactivateSecurityProfile ... 699
UpdateSecurityProfileSnapshotCreationConfiguration ... 699
UpdateSecurityProfileSnapshotReleaseConfiguration ... 700
ListSecurityProfileSnapshots ... 700
GetSecurityProfileSnapshot ... 700
TagResource ... 700
UntagResource ... 700
ListTagsForResource ... 701
其他 资源 ... 701
安全性 ... 159
IAM对于 AWS Marketplace ... 703
创建用户 ... 704
创建或使用组 ... 705
作为用户登录 ... 706
AWS Marketplace 卖家策略和权限 ... 707
策略 ... 707
权限 ... 708
AWS 托管策略 ... 714
AWSMarketplaceAmiIngestion ... 715
AWSMarketplaceFullAccess ... 716
AWSMarketplaceGetEntitlements ... 718
AWSMarketplaceMeteringFullAccess ... 719
AWSMarketplaceMeteringRegisterUsage ... 719
AWSMarketplaceSellerFullAccess ... 720
AWSMarketplaceSellerProductsFullAccess ... 723

AWSMarketplaceSellerProductsReadOnly 725
AWSVendorInsightsVendorFullAccess 726
AWSVendorInsightsVendorReadOnly 728
策略更新 729
AWS Marketplace 商务分析服务账号权限 732
Amazon SQS 权限 733
AWS Marketplace 计量和授权 API 权限 734
SaaS 产品的 IAM 策略 734
AMI 产品的 IAM 策略 735
容器产品的 IAM 策略 735
使用服务相关角色 736
转售授权角色 736
使用记录 AWS Marketplace API通话 AWS CloudTrail 740
AWS Marketplace 计量API日志文件条目示例 741
文档历史记录 747
AWS 词汇表 762
..... dcclxiii

什么是 AWS Marketplace？

AWS Marketplace 是一个精心策划的数字目录，客户可以使用它来查找、购买、部署和管理第三方软件、数据和服务，以构建解决方案和运营业务。 AWS Marketplace 包括数以千计的软件清单，这些软件来自特定行业（例如医疗保健、金融服务和电信）的安全、商业应用程序、机器学习和数据产品。客户可以快速启动预配置的软件，并选择 Amazon Machine Images (AMIs)、软件即服务 (SaaS) 和其他格式的软件解决方案。还提供专业服务来帮助客户配置、部署和管理第三方软件。有关交付方式的完整列表，请参阅产品交付。

您可以 AWS Marketplace 作为买方（订阅者）、卖方（提供商）或两者兼而有之。任何拥有者 AWS 账户 都可以 AWS Marketplace 用作买家，并且可以注册成为卖家。卖家可以是独立的软件供应商 (ISV)、渠道合作伙伴、托管服务提供商 (MSP)，也可以是可以提供与 AWS 产品和服务配合使用的产品的个人。

> **Note**
>
> 数据产品提供商必须满足 AWS Data Exchange 资格要求。有关更多信息，请参阅《Data Exchange 用户指南》中的在 AWS Data Exchange 上提供AWS数据产品。

符合条件的合作伙伴可以通过编程方式在外部发布 AWS Marketplace 产品。 AWS Marketplace有关成为合格合作伙伴的更多信息，请联系您的 AWS Marketplace 业务发展合作伙伴。

以下视频详细介绍了如何进行销售 AWS Marketplace。

AWSMarketplace 介绍

AWS Marketplace 作为卖家使用

销售软件产品的过程 AWS Marketplace 包括以下七个步骤。

卖家流程

步骤	操作	描述
1	注册	作为卖家，您首先要注册 AWS Marketplace 管理门户。我们建议您实施一个 AWS 账户 可以轻松与现有 AWS 组织关联的新专用。验证 AWS 合作伙伴的税务信息是否符合司法管辖区资格标准。对于

步骤	操作	描述
		没有现有美国银行账户的 AWS 合作伙伴，您可以使用 Hyperwallet 创建账户，无需支付额外费用。
2	确定产品类型	确定您要销售的产品类型。有关在中创建产品类型的更多信息 AWS Marketplace，请参阅以下内容： • AMI基于产品 • 基于容器的产品 • 机器学习产品 • 基于 SaaS 的产品 • 专业服务产品 • 数据产品（有关数据产品的更多信息，请参阅什么是AWS数据交换？ 在 AWSData Exchange 用户指南中。）
3	准备产品	配置您的套餐，设置定价方案，确定上架产品的相关类别，并添加关键词，以便您的产品出现在相关搜索中。为了简化采购流程，您可以对公开上架产品和专属优惠使用标准化许可条款。
4	提交产品	使用产品提交流程在 AWS Marketplace上提供您的产品。商品可以很简单，例如，具有单一价格结构的 Amazon 系统映像 (AMI)。或者，产品可能很复杂，包括多个 AWS CloudFormation 模板AMIs，以及复杂的定价选项和付款时间表。
5	市场产品	通过提高对产品页面的知名度 AWS Marketplace 以及将流量直接吸引到您的产品页面，为产品的成功做出贡献 AWS Marketplace。
6	查看报告和数据源	注册为卖家后，使用 AWS Marketplace 管理门户 访问商品的使用报告。 AWS Marketplace 提供用于收集和分析您的产品销售信息的工具。
7	管理产品	使用 AWS Marketplace 管理门户 来管理您的账户和产品页面。

作为卖家，请转到 AWS Marketplace 管理门户 进行注册。如果您对产品使用进行收费，还必须在注册过程中提供税务和银行信息。在注册时，您可以为公司创建配置文件，或创建可在 AWS Marketplace

中发现的自己的配置文件。您还可以为您的产品使用 AWS Marketplace 管理门户 创建和管理产品页面。

的合同结构 AWS Marketplace

出售的软件、服务和数据产品的使用受买方和卖方之间协议的约束。 AWS Marketplace AWS 不是这些协议的缔约方。

作为卖家，您的协议包括以下内容：

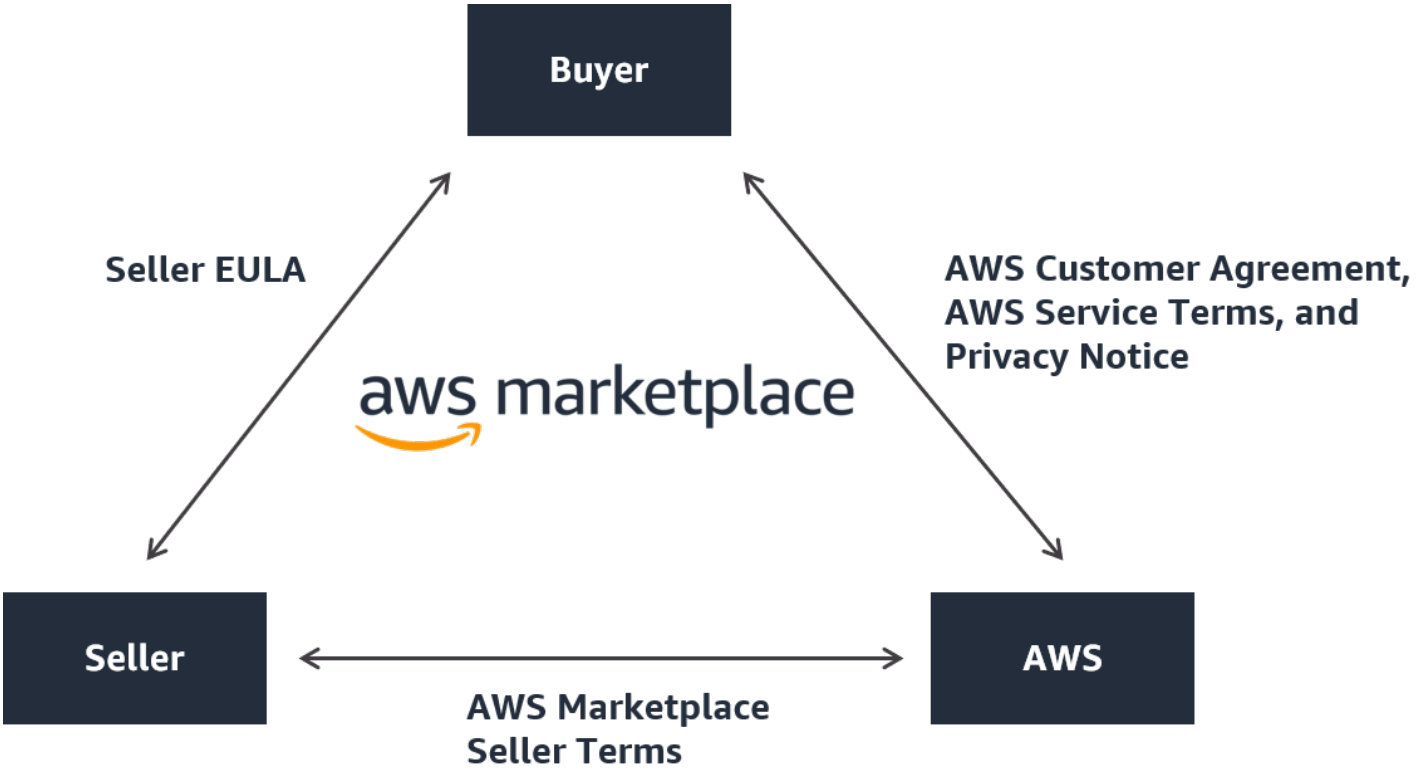
- 您与买家签订的最终用户许可协议 (EULA)，该协议位于中公开软件列表的产品列表页面上 AWS Marketplace。许多卖家使用 AWSMarketplace 标准合同 (SCMP) 作为他们的默认合同EULA。您也可以将SCMP作为私下报价谈判的基础，并使用修改模板进行修改SCMP。专属优惠也可以包括双方协商的定制合同条款。

  > **Note**
  >
  > 有关何时会根据报价类型和定价模型进行更新的信息，请参阅《AWS Marketplace 买家指南》中的EULA更新。EULA

- AWS Marketplace 卖家服务条款，用于管理您在中的活动 AWS Marketplace。

买方的使用 AWS Marketplace 受AWS服务条款、AWS客户协议和隐私声明的约束。

下图显示了的合约结构 AWS Marketplace。



中的产品定价 AWS Marketplace

在中 AWS Marketplace，产品可以免费使用，也可以收取相关费用。费用成为买方 AWS 账单的一部分，在买方付款后，向卖方 AWS 付款。产品可采用多种形式。例如，可以将商品作为亚马逊机器图片 (AMI) 提供，并使用买家的图片进行实例化。 AWS 账户也可以将商品配置为使用 CloudFormation模板向买家配送。产品也可以是来自网络访问控制列表（网络ACL）ISV、一组规则或条件的 SaaS 产品。AWS WAF产品也可以是渠道合作伙伴提供的专业服务，或者MSP。ISV

灵活的定价选项包括免费试用、每小时、每月、每年、多年和自带许可模式 (BYOL)，并且从一个来源计费。 AWS 处理账单和付款，费用显示在客户的 AWS 账单上。

可以使用的标准最终用户许可协议（EULA）按标价购买软件产品。ISV此外，软件产品可以定制定价和EULA通过私人报价提供。此外，还可依据合同的指定时间或用量界限购买产品。订阅产品后，买家可以使用 AWS Service Catalog 来复制产品并管理在买方组织中如何访问和使用该产品。有买家体验的更多信息，请参阅 https://docs.aws.amazon.com/marketplace/latest/buyerguide/service-catalog.html。有关定价的更多信息，请参阅the section called “产品定价”。

以卖方身份开始使用

如果您想销售您的软件 AWS Marketplace，请查看要求，然后按照步骤注册为卖家。根据您所在的位置以及要销售的产品类型，将具有不同的注册要求。要在中注册为卖家 AWS Marketplace，您可以使用现有账户 AWS 账户 或创建新账户。所有 AWS Marketplace 互动都与您选择的帐户相关联。

> **注意**
>
> - 注册为 AWS Marketplace 卖家是在 Data Exchange 上发布AWS数据产品并在上架的先决条件 AWS Marketplace。有关这些要求的更多信息，请参阅《Data Exchange 用户指南》中的在 AWS Data Exchange 上提供AWS数据产品。
> - 有关 AWS Marketplace 卖家所需权限的信息，请参阅AWS Marketplace 卖家策略和权限。
> - 有关产品上架费的更多信息，注册卖家可以在 AWS Marketplace 管理门户中查看 AWS Marketplace 卖家条款。
> - 有关常见问题的答案，请参阅AWS Marketplace 卖家FAQ。

发布免费软件产品的卖方要求

无论您在提供商品时是否收费 AWS Marketplace，都是在销售该产品。客户的成本为 0.00 美元，但您和客户就使用该产品签订一份合同。如果您仅提供免费产品，则无需向 AWS Marketplace提供银行信息。要在中创建和提供免费商品 AWS Marketplace，您必须：

- 销售公开发布且特征齐全的生产就绪软件。
- 具有明确的客户支持流程和支持团队。
- 提供相应的方法以定期更新软件并且没有漏洞。
- 在 AWS Marketplace中推广您的产品时，遵循最佳实践和准则。
- 成为信誉良好的 AWS 买家，并满足 AWS Marketplace 卖家条款和条件中的要求。

付费产品的其他卖方要求

如果您为自己的产品收费或提供自带许可证型号 (BYOL) 产品，则还必须满足以下要求并提供此额外信息：

- 您必须是符合条件的司法管辖区内的永久居民或公民，或者是在其中一个地区内组织或注册的商业实体。
- 您必须提供税务和银行账户信息。对于美国实体，需要提供 W-9 表格和美国银行的银行账户。
- 非美国卖家必须提供 (i) W-8 表格、增值税 (VAT) 或商品和服务税 (GST) 注册号，以及 (ii) 美国银行信息。如果没有美国银行账户，您可以从 Hyperwallet 中注册一个虚拟美国银行账户。
- 要提供数据产品，您还必须通过 “创建案例” 向导申请 AWS Support加入。
- 要通过 Amazon Web Services 向居住在欧洲、中东和非洲 (EMEA)（不包括土耳其和南非）的国家和地区的买家销售商品 EMEASARL，您必须完成 “了解您的客户” 流程。 AWS 账户 此外：
  - 您最多会收到两笔付款（用于通过 AWS Inc. 和亚马逊 Web Services 进行的交易 EMEASARL）。
  - 可能会对某些交易的上架费征税，具体视地点而定。有关税费的更多信息，请参阅 AWSMarketplace 卖家税务帮助页面。如果对您的上市费用征收增值税 (VAT)，则 AWS Marketplace 会提供符合税务要求的发票。
  - 有关亚马逊 Web Services 的更多信息 EMEASARL，请参阅AWS EMEA亚马逊网络服务欧洲 FAQs网站上的商城——卖家。

要向销售商品 AWS GovCloud (US) Region，卖家必须拥有一个AWS GovCloud (US) 账户。有关ITAR 要求的详细信息，请参阅《AWS GovCloud (US) 用户指南》。

有关 AWS Marketplace 卖家要求或注册流程的问题，请联系AWS Marketplace 卖家运营团队。

适用于付费产品的符合条件的司法管辖区

要在以下国家/地区销售付费软件 AWS Marketplace，您必须是以下国家/地区的永久居民或公民 SARs，或者是在该国家/地区组建或注册的商业实体：

- 澳大利亚[1]
- 巴林[1 2]
- 欧盟 (EU) 成员国[1]
- 香港 SAR
- 以色列[1 2]
- 日本[1 2 3]
- 新西兰[1]
- 挪威[1 2]

- 卡塔尔
- 瑞士[1] [2]
- 阿拉伯联合酋长国 (UAE) [1] [2]
- 英国 (UK)[1]
- 美国（US）

[1] 在这些国家/地区购买付费商品的卖家必须提供公司所在国的VAT注册信息。

[2] 如果您作为卖家与买家位于同一国家/地区，则可能需要负责开具税务发票、收款和汇款。请咨询您的税务顾问。

[3] 日本的卖家有义务自行核算上架费用的日本消费税 (JCT)。位于其他司法管辖区的卖家可能有类似的义务。请咨询您的税务顾问。

有关VAT开具发票和作为卖家的纳税义务的更多信息，请参阅亚马逊网络服务的 AWSMarketplace Sellers 税务帮助。

如果企业实体未在其中一个国家/地区注册成立或未在上面SARs列出，请参阅AWS Marketplace 司法管辖区以外的公司资源。

AWS Marketplace 管理门户

AWS Marketplace 管理门户是您用来注册为 AWS Marketplace 卖家的工具。然后，您可以使用该门户来管理您在 AWS Marketplace中销售的产品。您可以在门户上完成以下任务：

- 注册为 AWS Marketplace 卖家。
- 使用产品页面提交新的软件产品和更新现有软件产品。
- 监控您的请求的状态。
- 上传创建和管理新软件产品所需的文件。
- 利用这些 go-to-market 活动，将您的软件产品管理为增量渠道收入。
- 在推出后几小时内评估您的营销活动效果，包括您的活动带来的用量和收入。
- 使客户服务代表可以实时检索客户数据。
- 启动 Amazon 系统映像 (AMI) 自动扫描以检测漏洞。

> **Note**
>
> 数据产品是通过 Data Exchange 控制台发布和管理的。AWSAWSData Exchange 提供商可以使用注册 AWS Marketplace 管理门户 为卖家、申请 AWS Data Exchange 加入、访问卖家报告和提交退款申请。

所有注册卖家都 AWS Marketplace 管理门户 可以使用其用于创建商品的账户的 AWS 凭证访问。在客户订阅您的产品时，您使用的账户定义为记录卖家。如果需要提供帮助以确定作为您的产品记录卖家的特定账户，请与 AWS Marketplace 卖家运营团队联系。

AWS Marketplace 强烈建议使用 AWS Identity and Access Management (IAM) 角色登录， AWS Marketplace 管理门户 而不是使用您的根账户证书。关更多信息，请参阅 AWS Marketplace 安全性。

卖家注册过程

通过注册为的卖家 AWS Marketplace，您可以向其他 AWS Marketplace 买家销售您的产品和服务。

要注册为卖家，需要执行以下步骤：

1. 创建您的公开个人资料 — 您 AWS Marketplace 向买家提供显示的信息，让他们了解您的公司，例如您的公司名称和徽标。完成此过程后，您可以免费销售产品。要销售付费产品，您必须完成第二步和第三步。
2. 提供您的税务信息-要适当评估、报告和（如果适用）预扣已付销售额，您必须提供您的税收和增值税 (VAT) 信息。
3. 提供您的银行信息-您提供美国银行信息， AWS Marketplace 以便支付您的销售费用。

以下部分详细介绍了这些步骤。

完成卖家账户注册后，您可以创建要出售给买家的商品 AWS Marketplace。有关更多信息，请参阅 准备您的产品。

您可以使用 AWS Identity and Access Management (IAM) 将您的主服务器配置 AWS 账户 为允许多个具有不同权限的用户访问 AWS Marketplace 管理门户。有关更多信息，请参阅 the section called “IAM 对于 AWS Marketplace”。

创建您的公开资料

注册的第一步是选择 AWS 账户 要用作您的主要 AWS Marketplace 账户，然后在 AWS Marketplace 控制台中向潜在买家提供显示的信息。该账户将是您的商品的 AWS Marketplace 登记卖家，用于报告、付款和与您沟通。 AWS Marketplace

一旦您使用注册 AWS 账户 为卖家并在上架商品 AWS Marketplace，就无法更改与该商品关联的账户。我们建议您使用新账户注册为 AWS Marketplace 卖家。但是，如果现有账户是在 2017 年 9 月 27 日之后创建的，则可以使用该账户。

要创建您的公开资料，请执行以下操作：

1. 从 AWS Marketplace 管理门户(AMMP) 中选择 “立即注册”，然后登录到您选择的卖家 AWS 账户。
2. 选择添加公开资料以提供您的卖家信息。

完成公开资料后，您可以发布和销售免费产品。要销售付费产品，您必须提供税务和银行信息。

提供税务信息

您必须提供您的税费和增值税 (VAT)（如果适用）的信息，以便 AWS Marketplace 能够准确地报告和预扣商品销售的税款。

要提供您的税务信息，请执行以下操作：

1. 登录 AWS Marketplace 管理门户并选择设置。
2. 在付款信息部分中选择转到税务控制面板。
3. 完成美国税务调查。
   - 要销售专业服务 AWS Marketplace，您必须填写以下内容的税务问卷DAC7。
   - 如果您看到错误消息 “税务调查地点与营业地点不匹配”，请确保 B illing and Cost Management 中提供的银行和税务信息与输入的信息相匹配 AWS Marketplace 管理门户。您的税务调查地点必须与营业地点一致。
4. 填写完税务信息后，返回到 “设置” 页面，然后选择 “完成VAT信息”（如果有）。此选项会重定向到 AWS Billing 控制台上的 “税务设置” 页面。

> **Note**
> VAT信息部分仅在您所在的支持下才可用VAT。 AWS 区域

访问税务文件

您可以从中访问您的税务文件，例如 1099 表格。 AWS Marketplace 管理门户

访问您的税务文件

1. 登录 AWS Marketplace 管理门户并选择设置。
2. 前往 “付款信息” 部分。
3. 选择相关的纳税申报表（1099K 或DAC7）。
4. 如果您的纳税申报表可用，则可以在税务控制面板页面上下载。

提供美国银行账户信息

所有想要在美国销售付费商品的卖家都需要一个美国银行账户 AWS Marketplace。 AWS Marketplace 仅向美国银行账户付款。

> **Note**
> 有关您可以在其中提供付费产品的国家/地区列表 AWS Marketplace，请参阅适用于付费产品的符合条件的司法管辖区。

要提供美国银行信息，请执行以下操作：

1. 登录 AWS Marketplace 管理门户并选择设置。
2. 在付款信息部分中选择完成银行信息。
3. 提供有关您的美国银行账户的必要信息。

> **Note**
> 如果您尚未提供税务信息（以及增值税信息，如适用），则无法提供银行信息。

如果您还没有美国银行账户，则可以通过该账户获取Hyperwallet。 Hyperwallet可以为您提供一个美国账户，您可以将其提供给您的 AWS Marketplace AWS Marketplace 付款。

Hyperwallet 是一个独立的服务提供商，您可以通过该提供商将资金以支持的货币形式转到另一个银行账户。在有限的时间内，您不需要支付与 AWS Marketplace 支付相关的特定 Hyperwallet 服务费。

- 将您的Hyperwallet账户详细信息添加到您的 AWS Marketplace 卖家账户，即表示您同意并确认 AWS Marketplace 将与您共享您的姓名、电子邮件地址和账号，Hyperwallet以确认您的 AWS Marketplace 卖家身份。
- 可能会向您收取使用 Hyperwallet 服务的额外费用 (包括转账费用以及将资金转成当地货币所需的外汇费用) 并使用相应的汇率。Hyperwallet服务费将在有限的时间内免除，且仅适用于将 AWS Marketplace 付费商品的收益存入您的Hyperwallet账户。有关更多信息，请参阅 Hyperwallet 网站的费用部分或联系 Hyperwallet 以获取更多信息并查看适用的费用。有关其服务的更多信息，请参阅 Hyperwallet 支持网站。

要开始在 Hyperwallet 中注册并获取您的美国银行账户信息，请执行以下操作：

1. 登录 AWS Marketplace 管理门户 并选择设置，然后在付款信息部分中选择完成银行信息。
2. 如果您没有 Hyperwallet 账户，但需要一个账户在 AWS Marketplace中使用，请为您是否拥有美国银行账户？和您是否注册了 Hyperwallet？两个问题选择否。您将获得一个个人识别码 (PIN) 和注册链接Hyperwallet。
3. 在激活您的 Hyperwallet 账户后，请按照 Hyperwallet 注册门户中所述的步骤完成注册，并接收您的存款账户信息。
4. 当您从中获取账户后Hyperwallet，请登录将您的Hyperwallet账户信息添加到您的 AWS 账户 账户中AWS Marketplace 管理门户。然后，选择设置，然后在付款信息部分中选择完成银行信息。

完成“客户背景调查”流程

Know Your Customer (KYC) 是金融机构和在线企业用来验证其客户身份的合规要求。这一要求源于管理银行和其他支付机构等金融机构的修订版《支付服务指令》(PSD2) 和欧盟反洗钱指令。

AWS Marketplace 通过亚马逊 Web Services EMEA SARL 进行的交易通过 Amazon Payments Europe, S.C.A. (APE) 处理，这是一家位于卢森堡的持牌电子货币机构，需要验证您的身份才能使用支付服务。

作为卖家，要通过亚马逊 Web Services 进行交易 EMEASARL，您需要完成该KYC流程。此流程包括提供有关贵公司、关键联系人、受益所有权和支持文档的更多信息。

要完成该KYC过程

1. 在 AWS Marketplace 管理门户上，选择设置。
2. 在账户摘要部分，确认显示的国家/地区正确无误。

> **Note**
>
> 选择信息链接，查看如何更改您的国家/地区。

3. 选择 “转至KYC信息” 或选择 “了解您的客户” (KYC) 选项卡，然后选择 “开始KYC合规”，您将被重定向到KYC注册门户。

   有关如何使用和共享信息的更多信息，请参阅 Amazon Payments Europe 隐私声明。 AWS Marketplace

4. 选择 “转至KYC概览”。
5. 在 “了解你的客户 (KYC) 概述” 中，通读所需信息和文档的列表，并收集所需的文档（如果您尚未这样做）。然后选择 “继续KYC合规”。
6. 按照指示输入基本详细信息。查看 Amazon Payments Europe 条款和条件后，选择同意并继续。

   当您继续下一页或KYC流程的下一步时，该操作表示您接受 Amazon Payments Europe 条款和条件。

   如有疑问，请参阅主机右侧的常见问题解答 (FAQ)。

7. 按照指示输入所需的业务信息，然后选择下一步。

> **Note**
>
> 每次您选择下一步以转到下一步时，都会保存您的信息。

8. 按照指示输入所需的联系人信息，然后选择下一步。
9. 选择受益所有人是否与联系人相同，必要时添加受益所有人（最多四个），确认您的添加内容，然后选择下一步。
10. 选择法定代表人是与联系人相同还是受益所有人。如果法定代表人是其他实体，请提供所需信息，保存输入内容，然后选择 “下一步”。
11. 如需其他文件，请上传您的营业执照、身份证件和授权委托书（如果适用）。
12. 在查看并提交中，查看并验证您输入的所有信息。

如有必要，您可以选择编辑返回到之前的任何部分。

13. 选择提交以进行验证。

将对您的KYC合规状态进行审查（通常在 24 小时内）。审核完成后，您将收到一封电子邮件通知。整个KYC过程通常需要大约 2 周。

您可以返回 “设置” 选项卡，在 “账户摘要” 卡上查看您的KYC合规状态。有关您的KYC状态的更多信息，请选择 “账户摘要” 卡下的 “了解您的客户” (KYC) 选项卡。在审核完成之前，它将显示正在审核。

验证完成后，您必须在 “付款信息” 选项卡上提供银行对账单，然后才能通过APE其接收付款。KYC

完成银行账户验证流程

要接收来自 Amazon Payments Europe (APE) 的付款，您必须提供其他信息来验证您的付款银行账户，该账户列在的 “付款信息” 选项卡中。 AWS Marketplace 管理门户

提供其他银行信息

要提供其他银行信息，请执行以下操作：

1. 登录 AWS Marketplace 管理门户，然后选择 “设置”。
2. 在付款信息部分中选择更新银行信息。
3. 选择相应的支付账户。

   验证状态显示为未验证。
4. 选择验证。
5. 您将被重定向到银行账户验证注册门户，在那里您可以上传和提交银行对账单。

   如果您使用 Hyperwallet 虚拟银行账户解决方案，请参阅the section called “从 Hyperwallet 下载您的银行对账单”。
6. 在门户中，选择上传银行单据，然后选择提交。

从 Hyperwallet 下载您的银行对账单

对于使用 Hyperwallet 虚拟银行账户解决方案的卖家，您可以使用以下步骤下载 Hyperwallet 银行对账单。然后，您可以按照完成银行账户验证流程中的说明上传银行单据。

要从 Hyperwallet 下载您的银行对账单，请执行以下操作：

1. 登录您的 Hyperwallet 账户。
2. 前往存款账户信息页面。
3. 下载您的银行账户验证单。

（可选）为“客户背景调查”过程添加辅助用户

> **Note**
>
> 用户需要启用多因素身份验证 (MFA) 才能更新付款信息。有关更多信息MFA，请参阅的多重身份验证 (MFA)。IAM

次要用户是指能够修改KYC信息、控制资金流或退款流以及更改银行账户详细信息等财务信息的个人。

只有KYC经过验证的次要用户才能进行上述更新。这些辅助用户受到与主账号所有者相同的持续筛查控制。

要通过KYC验证，次要用户必须完成中的步骤完成“客户背景调查”流程。

要为“客户背景调查”过程添加辅助用户，请执行以下操作：

1. 要求该用户登录 AWS Marketplace 管理门户。
2. 导航到设置选项卡。
3. 选择 “了解您的客户” (KYC) 选项卡，并查看次要用户信息部分。
4. 选择完成辅助用户信息。

   您将被重定向到辅助用户注册门户。
5. 在辅助用户注册门户中，填写必填字段，然后选择下一步。
6. 在查看并提交页面上，上传身份证件（上传护照）和地址证明（上传文件）的副本。
7. 选择提交以进行验证。

将对您的KYC合规状态进行审查（通常在 24 小时内）。审核完成后，您将收到一封电子邮件通知。整个KYC过程通常需要大约 2 周。

付款和买家账单

AWS Marketplace 卖家，包括独立软件供应商 (ISVs) 和渠道合作伙伴，可以设置付款偏好以接收其未付余额。卖家可以选择每日或每月支付选项，并可以选择当月的哪一天接收付款。

要设置您的支付首选项，请执行以下操作：

1. 登录 AWS Marketplace 管理门户，然后选择 “设置”。
2. 选择付款信息选项卡，然后选择更新银行信息。
3. 在支付偏好部分，查看您当前的支付选项。要更改为每日支付计划，请选择每日，然后选择提交。您将看到一个百分比栏，显示您更新后的银行信息更新进度，直到完成为止。
4. 要从每日支付更改为每月支付，请选择每月，然后选择一个介于 1-28 之间的数字，您将在当月的当天处理支付。选择提交。

> **Note**
>
> 卖家应获得访问付款偏好选项的精细IAM权限。要加入细粒度IAM权限，请参阅。the section called “AWS Marketplace 卖家策略和权限”

AWS 代表您充当计费机制。买家最常用的两种付款方式是信用卡和发票。

以下是有关 AWS Marketplace 订阅账单的信息：

- 使用预付款的购买将在订阅后立即计费。
- 专属优惠的账单计划由买家和卖家商定。
- 发票付款条款（包括账单到期日期）由买家和 AWS商定。这些条款未向供应商披露。
- 使用灵活付款计划程序的专属优惠必须将发票作为付款选项。
- 您可以使用账单收入控制面板验证发票。此控制面板代表您汇总发票开具 AWS 情况，并提供发票创建日期和发票到期日的可见性。

以下是关于您作为卖家如何获得付款的信息：

- 需要具有有效的付款方式、注册的美国银行账户和提交的 W9 表才能进行支付。

- 付费商品的卖家必须提供 W-8、增值税 (VAT) 或商品和服务税 (GST) 注册号以及美国银行账户。Hyperwallet可以为您提供一个美国银行账户，您可以将其提供给您的 AWS Marketplace AWS Marketplace 付款。
- AWS 通过以下方式支付款项：
  - 每日 – 每日支付在可用时生效。卖家必须有正余额才能收到支付。
  - 每月 - 卖家选择当月的某一天（1-28）来接收支付。收款和支出控制面板显示您的付款日期。
- AWS 买家支付发票后，使用自动清算所 (ACH) 转账支付款项。
- 只有从客户收取资金后，才会支付相应的款项。
- 付款大约需要 1-2 个工作日才能在支付日期之后到达卖家的银行。确切的时间视银行和时区而定。
- 付款控制面板将在付款后的 AWS Marketplace 管理门户 3-5 天内更新。
- 有关已支付款项和未收款项的详细信息可在支付报告中找到，包括任何未结应收账款。

已经是卖家？

利用中提供的 go-to-market 活动，将您的产品管理为增量渠道收入AWS Marketplace 管理门户。活动包括：

- 在数小时内衡量您的营销活动效果，包括活动带来的用量和收入。
- 使客户服务代表可以实时检索客户数据。
- 上传创建和管理您的产品所需的文件，并监控我们的处理进度。

投诉处理政策 – Amazon Payments Europe

如果您对 Amazon Payments Europe (APE) 提供的服务有任何疑问，请告诉我们。您的反馈有助于我们为您和所有买家和卖家创造更好的体验。

> **Note**
>
> 只有具体的投诉才 AWS Marketplace 会通过以下程序处理。Amazon Payments Europe S.C.A. 提供的服务其中包括处理付款交易、验证费用中可能出现的错误以及款项支付。

提交投诉

如果您在 Amazon Payments Europe S.C.A. AWS Marketplace 开设账户，您的投诉将由 Amazon Payments Europe S.C.A. 处理。

要提交投诉，请执行以下操作：

1. 登录您的 AWS Marketplace 卖家账户。
2. 转到联系我们。
3. 选择商业 Marketplace、卖家账户、注册。
4. 提供有关您的投诉的详细信息，然后选择提交。

Amazon Payments Europe 投诉解决时限

Amazon Payments Europe S.C.A. (APE) 将在收到您的投诉之日起 15 个工作日内回复您的投诉的最新情况。在无法控制的特殊情况下APE，自APE首次收到投诉之日起，投诉的解决时间可延长至35个工作日。

投诉升级

如果您对我们的回复不满意，可以选择通过联系以下人员来升级您的投诉：

- Amazon Payments Europe 高级管理层

  通过向高级管理层发送电子邮件 (<ape-management@amazon.lu>) 来提交您的投诉。我们将认真思索您的评论，并在高级管理层收到您的投诉后的 15 个工作日内做出回应。在超出 Amazon Payment Europe 管理范围的特殊情况下，自高级管理层首次收到投诉之日起，投诉解决时限将延长至 35 个工作日。

- Commission de Surveillance du Secteur Financier (CSSF)

  CSSF该机构负责对卢森堡金融业的公司进行审慎监督。你可以致电 110 Route d'Arlon L-2991 卢森堡或使用联系页面，网址为：https://www.cssf.lu/contacts/。CSSF要获取有关他们CSSF以及如何联系他们的更多信息，请参阅CSSF网站上的客户投诉。

- 在线争议解决

  如果您在欧盟在线开设账户，也可以选择使用在线争议解决平台将CSSF您的投诉提交给我们。此选项之所以可用，是因为 Amazon Payments Europe S.C.A. 提供金融服务，并且CSSF是负责其许可的机构。有关更多信息，请访问欧盟委员会网站上的在线争议解决平台。

上架费

AWS Marketplace 为商品提供以下上架费。

> **Note**
>
> 这些上市费用自2024年1月5日午夜UTC起生效。

公开优惠上架费

软件和数据公开优惠的上架费由部署方法决定：

- S oftware-as-a-service (SaaS) — 3%
- 服务器（Amazon 机器映像 (AMI)、容器和机器学习）— 20%
- AWSData Exchange — 3%

专属优惠上架费

专属优惠的上架费取决于合同总价值，以及该专属优惠是从先前的专属优惠还是 AWS Marketplace外的先前协议续订而来的：

- 小于 100 万美元 – 3%
- 100 万至 1 千万美元 – 2%
- 等于或大于 1 千万美元 – 1.5%
- 所有续订 – 1.5%

渠道合作伙伴私募报价 (CPPO) 上市费用

CPPO无论商品类型或部署方式如何，商品的上架费用均可提高 0.5%。假如产品为 SaaS 专属优惠，合同总价值小于 100 万美元，则上架费为 3.5%。

专业服务上架费

所有专业服务产品的私人报价均收取2.5%的上市费。

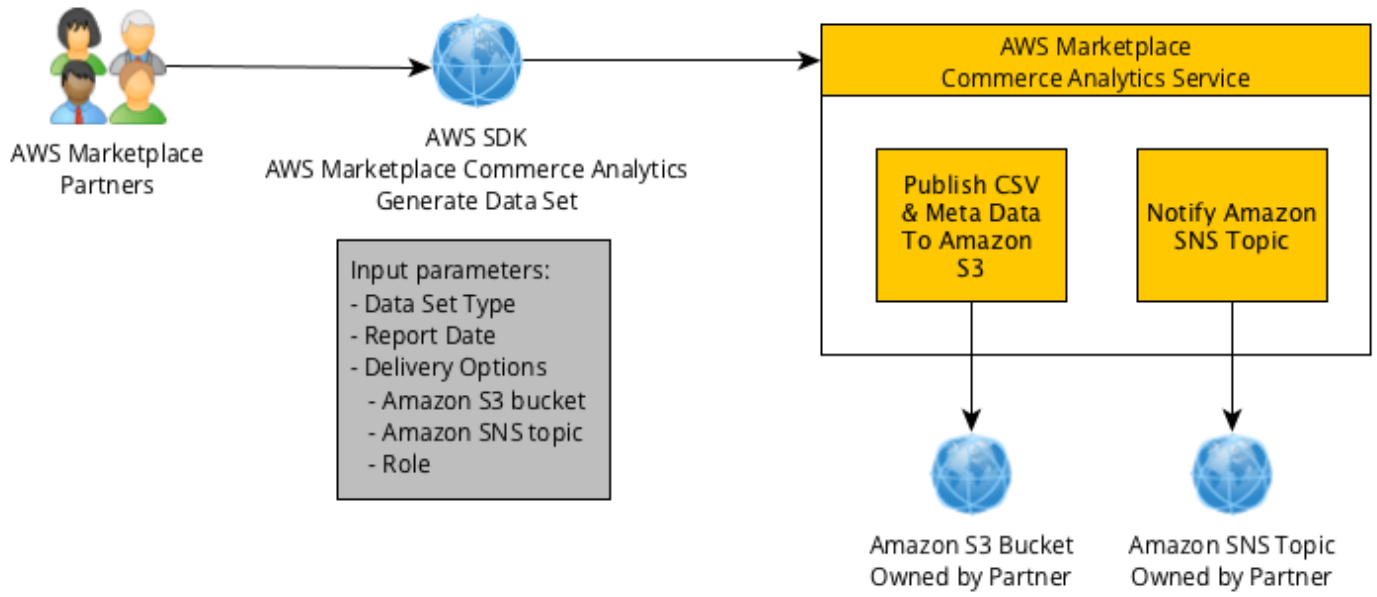
卖方工具包

AWS Marketplace 管理门户是您在上销售商品的主要工具 AWS Marketplace。以下是一些其他工具，可以让您更深入地了解客户群，并帮助您更好地了解销售情况。

- AWS Marketplace 商务分析服务
- AWS Marketplace 现场演示计划
- 卖家报告、数据源和控制面板
- 更多资源在 AWS Marketplace 管理门户

AWS Marketplace 商务分析服务

AWS Marketplace 商务分析服务允许您通过 AWS Marketplace以编程方式访问产品和客户数据。注册该服务后，您可以通过访问使用情况、订阅和账单报告 AWS SDK。



您使用这些SDK工具请求的数据将 AWS 账户 作为数据集传送给您。大多数数据集对应的数据与 AWS Marketplace 管理门户 上提供的基于文本的报告相同。您可以请求特定日期的数据集，该数据将交付到所提供的 Amazon S3 存储桶。数据传送通知由亚马逊简单通知服务 (AmazonSNS) 提供。

条款和条件

这些 AWS Marketplace 商业分析服务条款和条件（这些 “CAS条款”）包含您使用和访问 AWS Marketplace 商务分析服务（“CA 服务”）的特定条款和条件，自您点击这些CAS条款中显示的 “我接

受” 按钮或复选框之日起生效，或者如果更早的话，则在您使用任何 CA 服务产品时生效。这些CAS条款是您与 Amazon Web Services, Inc. (”、“我们” 或 “我们的” AWS) 之间的《AWS Marketplace 卖家条款和条件》 (“卖家条款”) 的附录，其条款特此纳入此处。 AWS Marketplace 如果这些条款与 AWS Marketplace 卖方CAS条款发生冲突，则这些条款的条款和条件适用，但仅限于此类冲突的范围，并且仅适用于您对加州服务的使用。CAS此处使用但未定义的大写术语应具有卖方条款中 AWS Marketplace 规定的含义。

1. CA 服务和CAS数据。要获得加州服务的访问资格，您必须是受现有 AWS Marketplace 卖家条款约束的 AWS Marketplace 卖家。您收到或可以访问的与 CA 服务相关的信息和数据 (“CAS数据”) 构成订阅者信息，并受 AWS Marketplace 卖方条款中规定的限制和义务的约束。您可以在保密的基础上使用CAS数据来改善和定位与您的 AWS Marketplace 内容相关的营销和其他促销活动，前提是您不要 (a) 向任何第三方披露CAS数据；(b) 以任何不符合适用的隐私政策或法律的方式使用任何CAS数据；(c) 联系订阅者以影响他们在之外进行其他购买 AWS Marketplace；(d) 贬低我们、我们的关联公司或他们或我们各自的任何产品；或 (e) 定向基于以下内容的任何形式的通信预期的收件人是 AWS Marketplace 订阅者。
2. CA 服务限制和安全性。您仅通过 CA 服务文档中所述的方法访问 (或尝试访问) CA 服务。在使用 CA 服务时，您不会歪曲或掩盖您的身份或您的客户身份。我们保留权利对您使用 CA 服务设置和实施限制 (由我们自行决定)，包括但不限于在任何时间段内允许访问 CA 服务的连接、调用和服务器数。您同意并且不会尝试绕过这些限制。如果我们认为您可能违反了这些CAS条款或滥用了 CA 服务，我们保留限制、暂停或终止您访问 CA 服务的权利。
3. CA 服务凭证保密性和安全性。CA 服务凭证 (例如密码、密钥和客户端IDs) 旨在供您用来识别您的API客户端。应由您全权负责保证您的凭证的机密性，并采取一切合理的措施以避免泄露、传播或未经授权使用此类凭证，至少包括您为保护自己的类似性质的机密信息而采取的措施。CA 服务凭证不得嵌入在开源项目中。应由您全权负责使用您的凭证对 CA 服务进行的任何和所有访问。
4. 修改。我们可以随时修改这些CAS条款，方法是在AWS网站上发布修订版或根据 AWS Marketplace 卖方条款向您发出通知。修改的条款将在发布后生效，或者如果我们通过电子邮件通知您，则在电子邮件中指定的日期生效。在本CAS条款的任何修改生效日期之后继续使用或访问 CA 服务，即表示您同意受修改后的条款的约束。
5. 终止。无论出于何种原因终止您的 AWS Marketplace 卖家CAS条款，无论是否通知您，这些条款和此处授予CAS的数据使用权都将终止。此外，我们可能还会随时由于任何原因或无任何理由停止提供 CA 服务或终止您对 CA 服务的访问。

信息载入指南

您必须配置您的 AWS 账户 和 AWS 服务 才能使用 AWS Marketplace 商务分析服务。

使用 AWS Marketplace 商务分析服务

1. AWS 账户 使用权限设置你的.
2. 创建目标 Amazon S3 存储桶.
3. 为回复通知配置一个 Amazon SNS 主题.
4. 注册到 Commerce Analytics Service 计划.
5. 验证配置.

AWS 账户 使用权限设置你的

AWS Marketplace 强烈建议使用 AWS Identity and Access Management (IAM) 角色登录， AWS Marketplace 管理门户 而不是使用您的根账户证书。the section called “AWS Marketplace 卖家策略和权限”有关 AWS Marketplace 商务分析服务IAM权限的具体权限，请参阅。在为访问您的账户的人员创建单独的用户时，您可授予每个用户一组独特的安全凭证。您还可向每个用户授予不同的权限。如有必要，您可随时更改或撤销用户的权限。

创建目标 Amazon S3 存储桶

Commerce Analytics Service 将请求的数据传输到您指定的 Amazon S3 存储桶。如果您已有 Amazon S3 存储桶可供使用，请继续到下一个步骤。

如果您没有 Amazon S3 存储桶，或者希望为此数据专门创建新 Amazon S3 存储桶，请参阅如何创建 S3 存储桶。

为回复通知配置一个 Amazon SNS 主题

商务分析服务使用 Amazon 发送回复通知SNS。在您的数据集可用或者出错时，该服务发布消息到此主题来通知您。如果您已经有用于此目的的 Amazon SNS 主题，请继续下一步。

如果您没有为此服务配置 Amazon SNS 主题，请立即配置一个。有关说明，请参阅创建监视器。

记录您创建的主题的主题 Amazon 资源名称 (ARN)，因为调用该服务需要使用。ARN

注册到 Commerce Analytics Service 计划

在您使用存储桶的SNS主题和名称配置服务后，商务分析服务将访问 Amazon S3 存储桶和亚马逊主题。ARN

启用访问

1. 使用 AWS 账户 您用来管理 AWS Marketplace 商品的，登录到。AWS Marketplace 管理门户

2. 确保您拥有注册 AWS Marketplace 商务分析服务的必要IAM权限。
3. 导航到 Commerce Analytics Service 注册页面。
4. 输入 Amazon S3 存储桶名称和亚马逊SNS主题ARN，然后选择注册。
5. 在权限页面上，选择允许。
6. 在 AWS Marketplace 管理门户，在成功消息ARN中记录角色名称。您需要拨ARN打服务电话。

> **Note**
>
> 加入商务分析服务会在您 AWS 账户中IAM扮演一个角色. 该IAM角色 AWS Marketplace 允许写入 Amazon S3 存储桶并向亚马逊SNS主题发布通知。 AWS Marketplace 使用账户 452565589796 来执行与该角色相关的操作。IAM

验证配置

最后一步是验证您的配置是否按预期运行。

测试配置

1. 下载、安装和配置AWS命令行界面 (AWS CLI)。
2. 使用 AWS CLI，运行此命令。

```
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}"
```

- 对于`--data-set-publication-date`，使用 ISO -8601格式替换`{TODAY'S DATE}`为当前日期，其中`YYYY`是四位数的年份`YYYY-MM-DDT00:00:00Z`，是两位数的月份，`MM`是两位数的`DD`日期。
- 对于`--role-name-arn`，请`{YOUR-ROLE-NAME-ARN}`替换为您在中注册过程中获得的角色注册到 Commerce Analytics Service 计划。ARN

- 对于 --destination-s3-bucket-name，将 {amzn-s3-demo-bucket-bucket-name} 替换为您在中创建的 Amazon S3 存储桶的名称。创建目标 Amazon S3 存储桶
- 对于 — sns-topic-arn，将 {YOUR-SNS-TOPIC-ARN} 替换为您在中创建的亚马逊SNS主题为回复通知配置一个 Amazon SNS 主题。

如果您收到来自该服务的回复，包括dataSetRequest身份证回复，则说明您已完成入职流程。成功的响应类似于以下内容：

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

技术实施指南

AWS Marketplace 商务分析服务是通过提供的AWSSDK。本指南演示如何使用 AWS CLI 和 AWS SDK for Java 与服务交互。

IAM商务分析服务政策

要允许您的用户使用 Commerce Analytics Service，需要以下权限。

使用以下IAM权限策略注册 AWS Marketplace 商务分析服务。

```
{
    "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权限策略允许用户向 AWS Marketplace 商务分析服务提出请求。

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

有关更多信息，请参阅《IAM用户指南》中的在IAM控制台中创建策略。

使用提出请求 AWS CLI

首先，请下载 AWS CLI。以下 AWS CLI 示例请求获取 2017 年 10 月 1 日的 “每小时/每月订阅” 数据集。此数据集使用前缀 demo-prefix 发布到演示存储桶 Amazon S3 存储桶，通知消息将发送到演示主题的 Amazon 主题。 SNS

```
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"
```

此请求返回对每个请求唯一的标识符。您可以使用此标识符将请求与发布到您的 Amazon SNS 主题的通知关联起来。以下是此标识符的示例。

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

使用 AWS SDK for Java发出请求

首先，请下载 AWSJava SDK。以下 AWS SDK for Java 示例请求获取 2015 年 10 月 1 日的每小时/每月订阅数据集。此数据集使用前缀 demo-prefix 发布到演示存储桶 Amazon S3 存储桶，通知消息将发送到演示主题的 Amazon 主题。 SNS

```
/*
* Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
*
* Licensed under the Apache License, Version 2.0 (the "License").
* You may not use this file except in compliance with the License.
* A copy of the License is located at
*
* http://aws.amazon.com/apache2.0
*
* or in the "license" file accompanying this file. This file is distributed
* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing
* permissions and limitations under the License.
*/
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import com.amazonaws.AmazonClientException;
import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import
 com.amazonaws.services.marketplacecommerceanalytics.AWSMarketplaceCommerceAnalyticsClient;
import
 com.amazonaws.services.marketplacecommerceanalytics.model.GenerateDataSetRequest;
import com.amazonaws.services.marketplacecommerceanalytics.model.GenerateDataSetResult;
/**
* This sample demonstrates how to make basic requests to the AWS Marketplace Commerce
* Analytics service using the AWS SDK for Java.
* <p>
* <b>Prerequisites:</b> Follow the on-boarding guide: {URL OR SOMETHING}
* <p>
```

```
* Fill in your AWS access credentials in the provided credentials file
* template, and be sure to move the file to the default location
* (~/.aws/credentials) where the sample code will load the credentials from.
* <p>
* <b>WARNING:</b> To avoid accidental leakage of your credentials, DO NOT keep
* the credentials file in your source directory.
* <p>
* http://aws.amazon.com/security-credentials
*/
public class MarketplaceCommerceAnalyticsSample {
public static void main(String[] args) throws ParseException {
/*
* The ProfileCredentialsProvider will return your [default]
* credential profile by reading from the credentials file located at
* (~/.aws/credentials).
*/
AWSCredentials credentials = null;
try {
credentials = new ProfileCredentialsProvider().getCredentials();
} catch (Exception e) {
throw new AmazonClientException("Cannot load the credentials from the credential
 profiles "
+ "file. Make sure that your credentials file is at the correct "
+ "location (~/.aws/credentials), and is in valid
format.", e);
}
AWSMarketplaceCommerceAnalyticsClient client = new
 AWSMarketplaceCommerceAnalyticsClient(credentials);
Region usEast1 = Region.getRegion(Regions.US_EAST_1);
client.setRegion(usEast1);
System.out.println("===============================================================");
System.out.println("Getting Started with AWS Marketplace Commerce Analytics Service");
System.out.println("===============================================================
\n");
// Create a data set request with the desired parameters
GenerateDataSetRequest request = new GenerateDataSetRequest();
request.setDataSetType("customer_subscriber_hourly_monthly_subscriptions");
request.setDataSetPublicationDate(convertIso8601StringToDateUtc("2014-06-09T00:00:00Z"));
request.setRoleNameArn("arn:aws:iam::864545609859:role/
MarketplaceCommerceAnalyticsRole");
request.setDestinationS3BucketName("awsmp-goldmine-seller");
request.setDestinationS3Prefix("java-sdk-test");
request.setSnsTopicArn("arn:aws:sns:us-west-2:864545609859:awsmp-goldmine-seller-
topic");
```

```
System.out.println(
String.format("Creating a request for data set %s for publication date %s.",
request.getDataSetType(), request.getDataSetPublicationDate()));
try {
// Make the request to the service
GenerateDataSetResult result = client.generateDataSet(request);
// The Data Set Request ID is a unique identifier that you can use to correlate the
// request with responses on your Amazon SNS topic
System.out.println("Request successful, unique ID: " + result.getDataSetRequestId());
} catch (AmazonServiceException ase) {
System.out.println("Caught an AmazonServiceException, which means your request made it
 "
+ "to the AWS Marketplace Commerce Analytics service, but was rejected with an "
+ "error response for some reason.");
System.out.println("Error Message: " + ase.getMessage());
System.out.println("HTTP Status Code: " + ase.getStatusCode());
System.out.println("AWS Error Code: " + ase.getErrorCode());
System.out.println("Error Type: " + ase.getErrorType());
System.out.println("Request ID: " + ase.getRequestId());
} catch (AmazonClientException ace) {
System.out.println("Caught an AmazonClientException, which means the client encountered
 "
+ "a serious internal problem while trying to communicate with the AWS Marketplace"
+ "Commerce Analytics service, such as not being able to access the "
+ "network.");
System.out.println("Error Message: " + ace.getMessage());
}
}
private static Date convertIso8601StringToDateUtc(String dateIso8601) throws
 ParseException {
TimeZone utcTimeZone = TimeZone.getTimeZone("UTC");
DateFormat utcDateFormat = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ssX");
utcDateFormat.setTimeZone(utcTimeZone);
return utcDateFormat.parse(dateIso8601);
}
}
```

您应该预期看与此示例类似的结果。

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

技术文档

该服务公开了一个方法 `GenerateDataSet`，您可以使用该方法请求将数据集发布到 Amazon S3 存储桶。下表中列出了 `GenerateDataSet` 的参数。

数据集参数

字段	描述
数据集类型	此数据集将作为请求的结果返回。
数据集发布日期	发布数据集的日期。 对于每日数据集，请为所需日期提供具有天级别粒度的日期。 对于每月数据集，请为所需月份提供具有月级别粒度的日期。忽略天值。
角色名 ARN	具有附加权限策略ARN的角色，该策略为服务提供对您的资源的访问权限。
目标 Amazon S3 存储桶名称	目标 Amazon S3 存储桶的名称（友好名称，不是ARN）。您的数据集将发布到此位置。
目标 Amazon S3 前缀	（可选）已发布数据集的 Amazon S3 前缀，类似于标准文件系统中的目录路径。 例如，如果给定存储桶名称 `mybucket` 以及前缀 `myprefix/mydatasets`，则输出文件将发布到 `s3://amzn-s3-demo-bucket/myprefix/mydatasets/outputfile`。 如果前缀目录结构不存在，则会自动创建。

字段	描述
	如果没有提供前缀，数据集将发布到 Amazon S3 存储桶根。
SNS话题 ARN	ARN适用于 Amazon SNS 主题，当数据集发布或出现错误时会收到通知。

响应

AWS Marketplace 商务分析服务返回两个响应。第一个是同步的，它会立即返回，第二个是异步的，使用 Amazon 返回SNS。同步响应类似于此示例。

数据集参数

字段	描述
数据集请求 ID	表示对服务的特定请求的唯一标识符。此标识符可用于将请求与有关 Amazon SNS 主题的通知关联起来。

异步响应将作为JSON格式的文档发布到您的 Amazon SNS 主题中，与本示例类似。

数据集参数

字段	描述
数据集 S3 位置	已交付数据集的存储桶名称和密钥。
数据集元数据 S3 位置	已交付数据集元数据文件的存储桶名称和密钥。
数据集请求 ID	表示对服务的特定请求的唯一标识符。此标识符可用于将请求与有关 Amazon SNS 主题的通知关联起来。
成功	如果操作成功，则为 True；否则为 False。
消息	（可选）如果出现错误（例如，“Success”为“False”），此消息将包含有关故障的信息。

JSON格式化异步响应示例

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

输出

成功请求之后，请求的数据集将以 .csv 文件的格式交付到您的 Amazon S3 存储桶。JSON格式的元数据文件将发布到与数据集文件相同的位置。元数据文件提供了有关数据集和原始请求参数的有用信息。元数据文件与数据集文件具有相同名称，但以扩展名 .meta.json 结尾。下表列出了 .csv 文件中的元数据字段。

元数据字段

字段	描述
数据集请求 ID	表示对服务的特定请求的唯一标识符。此标识符可用于将请求与有关 Amazon SNS 主题的通知关联起来。
数据集覆盖范围	定义数据覆盖范围的开始日期/时间和结束日期/时间。这些日期采用 ISO 8601 格式。
数据集请求参数	`GenerateDataSet` 方法的原始请求参数。
数据集 S3 位置	已交付数据集的存储桶名称和密钥。
数据集元数据 S3 位置	已交付数据集元数据文件的存储桶名称和密钥。

以下是JSON格式化元数据内容的示例。

```
{
"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"
}
}
```

有关可用数据集的完整列表，包括可用日期，请参阅AWSSDK文档。

故障排除

本节介绍您在使用 AWS Marketplace 商务分析服务时可能遇到的问题的解决方案。

由于允许列表问题，我无法访问服务。

如果您尚未在上注册为卖家 AWS Marketplace，请AWS Marketplace 管理门户访问注册。如果您已经注册为卖家 AWS Marketplace，请联系AWS Marketplace 卖家运营团队。

我无法请求过去某个日期的数据集，尽管SDK文档说该日期应该可以使用。

即使过去特定日期的数据集列出为可用，但我们只有自您加入 AWS Marketplace日期之后的数据。如果您仍认为这是一个错误，请联系 AWS Marketplace 卖家运营团队。

当我调用服务时，我收到错误消息 “无法连接到端点URL： https://marketplacecommerceanalytics.eu-central-1.amazonaws.com/”

AWS Marketplace 商务分析服务仅在美国东部（弗吉尼亚北部）地区提供。您必须将对 Commerce Analytics Service 的所有调用针对 `us-east-1` 端点。

如果您使用的是 AWS CLI，请在每个调用中添加 “`--region`标志并指定 AWS 区域 为`us-east-1`，如以下示例所示。

```
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
```

我想使用与我在完成入职流程时选择的不同的 Amazon S3 存储桶或亚马逊SNS主题。

注册 AWS Marketplace 商业分析服务时，您指定了 Amazon S3 存储桶和亚马逊SNS主题。入职流程将您的IAM权限配置为仅允许服务访问这些特定资源。要使用不同的资源，您需要修改您的IAM策略：

1. 登录 AWS Management Console 并打开IAM控制台，网址为https://console.aws.amazon.com/iam/。
2. 选择IAM控制台左侧的 “角色”。
3. 选择MarketplaceCommerceAnalyticsRole。
4. 如果尚未展开内联角色部分，请展开该部分。
5. 找到名称以 oneClick_MarketplaceCommerceAnalyticsRole 开头的策略，然后选择编辑策略。
6. 在策略文档中，找到指定与所要修改服务相关的操作的部分。例如，要更改 Amazon S3 存储桶，找到包含以 s3: 开头的操作的部分，更改相应的资源部分或者指定新的 Amazon S3 存储桶。

有关IAM策略的更多信息，请参阅以下指南：https://docs.aws.amazon.com/IAM/latest/ / access_policies.html UserGuide

在调用 **GenerateDataSet** 操作时，我收到 **AccessDeniedException** 错误

如果您的用户没有调用 GenerateDataSet 所需的权限，则会出现此情况。以下过程概述了使用IAM控制台创建具有这些权限的IAM策略以及向您的用户、群组或角色添加权限所需的步骤。

使用JSON策略编辑器创建策略

1. 登录 AWS Management Console 并打开IAM控制台，网址为https://console.aws.amazon.com/iam/。
2. 在左侧的导航窗格中，选择策略。

   如果这是您首次选择策略，则会显示欢迎访问托管式策略页面。选择开始使用。
3. 在页面的顶部，选择 Create Policy（创建策略）。
4. 在 “策略编辑器” 部分中，选择JSON选项。
5. 输入以下JSON策略文档：

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

6. 选择下一步。

> **Note**
>
> 您可以随时在 “可视化” 和 “JSON编辑器” 选项之间切换。但是，如果您进行更改或在可视化编辑器中选择 “下一步”，则IAM可能会重构您的策略以针对可视化编辑器对其进行优化。有关更多信息，请参阅《IAM用户指南》中的策略重组。

7. 在查看并创建页面上，为您要创建的策略输入策略名称和描述（可选）。查看此策略中定义的权限以查看策略授予的权限。

8. 选择创建策略可保存新策略。

要提供访问权限，请为您的用户、组或角色添加权限：

- 中的用户和群组 AWS IAM Identity Center：

  创建权限集合。按照《AWS IAM Identity Center 用户指南》中创建权限集的说明进行操作。
- IAM通过身份提供商管理的用户：

  创建适用于身份联合验证的角色。按照IAM用户指南中为第三方身份提供商创建角色（联合）中的说明进行操作。
- IAM用户：
  - 创建您的用户可以担任的角色。按照用户指南中为IAM用户创建角色中的说明IAM进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《用户指南》中向用户（控制台）添加权限中的IAM说明进行操作。

此处未列出我的问题。

联系 AWS Marketplace 卖家运营团队。

AWS Marketplace 现场演示计划

AWS Marketplace 现场演示计划 (FDP) 允许 AWS 现场团队（内部批准的 AWS 员工）免费使用某些产品和解决方案。 AWS Marketplace

获批 AWS 员工的例子可能包括解决方案架构师以及销售和营销专业人员。FDP 允许这些员工演示产品在融入教育和客户工作负载方面的能力。

支持以下产品类型：

- 亚马逊机器映像 (AMI)
- 容器
- 机器学习算法和模型包 (SageMaker)
- 数据集 (AWS Data Exchange)

> **Note**
>
> 对于 AWS Data Exchange 产品，FDP 仅适用于公开优惠为 0 美元（免费）的产品。

> 对于启用了订阅验证的 AWS Data Exchange 产品，提供商需要批准订阅请求。有关订阅验证的更多信息，请参阅《AWS Data Exchange 用户指南》中的针对订阅用户的订阅验证。

当您注册为 AWS Marketplace 卖家时，您将自动注册加入 FDP 计划。要选择退出，请向 AWS Marketplace 卖家运营团队提交支持请求。

要查看有关该计划下产品使用情况的信息，请参阅月度账单收入报告中的 AWS 现场演示使用情况部分。

更多资源在 AWS Marketplace 管理门户

AWS Marketplace 管理门户中还有更多资源可供您使用。如果您打开 AWS Marketplace 管理门户 并登录，则可以在主页的 Marketplace 资源部分看到指向其他资源的链接。

例如，要在产品发布前 90 天内和发布后 90 天内获得营销支持，您可以在 Marketplace Resources 下查看 AWS Marketplace 管理门户 主页上链接的。为期 180 天的学院 GTM

准备您的产品

发布商品的准备工作 AWS Marketplace 包括配置您的套餐、设置定价方案、确定发布商品的相关类别，以及添加关键字以使您的产品出现在相关搜索中。

主题

- 产品交付
- 产品定价
- 您的 AWS Marketplace 产品所在的地区和国家
- 中的标准化合同 AWS Marketplace
- 类别和元数据
- AMI和容器产品使用说明
- 产品搜索引擎优化

产品交付

每种产品交付方式都具有多种包装、定价和交付选项。在您注册支持该方法的计划 AWS Marketplace 之前，作为卖家，您无法使用某些方法。

您可以使用标准标价和最终用户许可协议 (EULA) 创建产品。您还可以使用自定义定价为个人客户创建私人报价，以及EULAs. 如果您需要对合同条款进行额外的更改，可以与 AWS Marketplace 团队合作以创建专属优惠。为了简化采购流程，您可以对公开上架产品和专属优惠使用标准化许可条款。

> **Note**
>
> AWS 为某些卖家提供了提供指导性演示的选项 AWS Marketplace。如果您是符合APN客户互动 (APN) 资格的 AWS 合作伙伴网络 (ACE) 合作伙伴，并且您想向买家提供此选项，请联系您的 AWS 代表以确定进一步的资格。

下表列出了您可以用来交付软件产品的方法以及 AWS Marketplace 买家如何在控制台中找到每种可交付内容。 AWS Marketplace

产品交付方式

产品交付方式	控制台上的“交付方式”筛选条件	描述
单曲 AMI	Amazon 机器映像 (AMI)	您为商品提供一张自定义 Amazon 机器映像 (AMI)。AMI 提供了启动亚马逊弹性计算云 (AmazonEC2) 实例所需的信息。 买家可以使用该单一AMI创建亚马逊EC2实例，您的产品已经安装并可以使用。 有关更多信息，请参阅 AMI基于产品。
AMI使用 AWS CloudFormation 模板交付	CloudFormation 模板	您可以使用 CloudFormation 模板AMI列出配送给 AWS Marketplace 买家的商品。 买家可以购买单一解决方案，使他们有权获得该产品AMIs中的所有内容。 有关以模板AMIs形式交付的更多信息，请参阅使用AMI基于 CloudFormation 模板的交付 AWS CloudFormation。 有关 CloudFormation 模板的更多信息，请参阅《AWS CloudFormation 用户指南》中的AWS CloudFormation 概念。
基于容器的产品或应用程序	容器	您交付包装在容器映像中的产品。容器产品由选项组成，这

产品交付方式	控制台上的“交付方式”筛选条件	描述
		些选项是一起使用的一组容器映像和部署模板。 有关更多信息，请参阅 基于容器的产品。
数据产品	AWS Data Exchange	您可以使用 AWS Data Exchange 来创建数据产品。 有关通过 Data Exchange 发布和管理数据产品和报价的信息，请参阅《AWS数据交换用户指南》中的 “在AWS数据交换上提供AWS数据产品”。
机器学习算法和模型包	SageMaker 模型	您可以使用 Amazon SageMaker 创建算法或模型包，然后将其发布到上 AWS Marketplace。 有关交付机器学习算法和模型包的更多信息，请参阅机器学习产品。 有关的信息 SageMaker，请参阅什么是 SageMaker？ 在《亚马逊 SageMaker 开发者指南》中。
软件即服务 (SaaS)	SaaS	您可以通过基于订阅、基于合同或带有消费产品的合同定价模式来提供 SaaS 产品。 有关更多信息，请参阅 基于 SaaS 的产品。

产品交付方式	控制台上的“交付方式”筛选条件	描述
专业服务	专业服务	您可以提供支持或与其他 AWS Marketplace 产品配合使用的专业服务。

产品定价

本主题提供有关 AWS Marketplace中软件产品的一般定价信息。所有定价均以美元为基准 (USD)。

对于付费产品， AWS Marketplace 向客户收取软件费用。

免费向客户提供的免费或开源软件不会收取服务费。

有关退款的信息，请参阅商品退款在 AWS Marketplace。

主题

- 定价模式
- 更改定价模式
- 更改价格
- 专属优惠
- 商品退款在 AWS Marketplace

定价模式

下表提供了有关 AWS Marketplace中提供的定价模式的一般信息。

主题

- 年度定价
- 用量定价
- 合同定价
- 自带许可定价

有关特定产品交付方式的定价模式的信息，请参阅：

- AMI 产品定价
- 容器产品定价
- 机器学习产品
- SaaS 产品定价
- 专业服务产品定价

年度定价

使用年度定价模式，您可以向可购买 12 个月订阅的客户提供产品。例如，与在这段时间内按小时费率运行相同产品相比，该订阅定价最多可以节省 40%。将在订阅时为客户开具全部合同金额的发票。有关如何向客户提供年度订阅的更多信息，请参阅付费容器产品的AMI订阅或定价模型。

使用年度订阅时，请注意包含以下事项：

- 年度定价是针对每种实例类型定义的。所有亚马逊弹性计算云 (AmazonEC2) 实例类型可能相同，也可能每种实例类型不同。
- 所有年度实例类型还必须定义每小时实例类型。 AWS Marketplace 不会为同一产品提供仅年度定价或没有年度定价的小时定价。对于提供年度定价的任何产品，还需要指定小时定价。
- 允许在特定实例类型上使用 0 美元年度价格，但前提是小时价格也为 0 美元并定义了其他非 0 美元年度实例类型。
- 在年度订阅期结束时，将开始按小时价格对客户进行收费。
- 如果客户购买 X 个年度订阅，但在 Y 个实例上运行 Y 软件，则将按小时软件价格向客户收取 (Y-X) 实例的费用，年度订阅未涵盖这些实例。因此，所有年度定价实例类型必须包含小时费率。
- 使用卖家私享优惠，您可以提供多年（最长3年）或自定义期限，AMI包括预付款，或者灵活的付款时间表。有关多年和自定义期限合同的更多信息，请参阅准备您的私人报价 和the section called “分期付款计划”。

如果您在中提供年度产品 AWS Marketplace，则表示您同意年度产品的特定退款政策，该政策位于文件上传器文档部分中AWS Marketplace 管理门户。

价格更改

您可以每 90 天更改一次年度价格（美元值，例如将 1000 美元/年改为 1200 美元/年）。但是，您必须提前 90 天向现有客户发出年度定价通知。新价格适用于新订阅，但对现有订阅没有影响。

> **Note**
>
> 在 90 天通知期内，您无法更新支持的实例类型。

只有在自动续订日期之前至少 90 天更改价格时，价格更改才对自动续订有效。在包含新价格的自动续订之前，客户将收到一封电子邮件。

最终用户许可协议

根据年度订阅， AWS 客户在 12 个月内对软件的使用将EULA由您在产品详情页面上提供的内容中提供 AWS Marketplace。

用量定价

用量定价模式（也称为即用即付定价模式）使您能够向仅按用量付费的客户提供产品。

作为卖家，您可以选择以下任一用量类别：

- 用户
- 主机
- 带宽
- 数据
- 层
- 单位（用于自定义类别）

您还可以为产品最多定义 24 个维度。当软件调用时API，会测量和报告费用。我们建议卖家将配置API为每小时调用一次，以此作为最佳实践，具体取决于他们的用例。所有使用量均按月计算，并使用与现有 AWS Marketplace 软件相同的机制按月计费。

使用 AWS Marketplace 计量服务，您可以处理几种新的定价方案。

Example 按主机收费

如果您的软件监控主机，您可以对每个监控的主机收费，并根据主机大小设置不同的定价。

Example 按用户收费

如果允许组织中的多个用户使用您的软件，您可以按用户收费。每小时，将针对预配置的总用户数对客户收取一次费用。

> **Note**
>
> 在 “商品负荷表” (PLF) 中，相关列以 “FCP” (灵活消费定价) 开头。例如：FCP类别 (自定义定价类别) 。

对于 AWS Marketplace Metering Service 产品，请注意以下几点：

- 如果您的软件已开启 AWS Marketplace，则需要创建产品以启用其他使用维度。您无法将标准商品转换为使用 AWS Marketplace 计量服务。在发布新产品后，您可以删除旧产品，或者在网站上保留两个产品。
- AWSMarketplace 计量服务要求您的软件每小时报告使用情况，记录该小时的客户使用情况。如果计量服务记录的传输或接收出现故障， AWS 将无法为此类使用开具账单。您有责任确保成功接收计量记录。
- 使用 AWS Marketplace 计量服务的产品不支持 “一键下单”。购买者必须使用具有特定权限的 AWS Identity and Access Management (IAM) 角色启动您的软件，并拥有互联网网关。
- 免费试用和年度定价与 Mar AWS ketplace 计量服务不兼容。
- 不支持更改维度 (用户、主机、带宽和数据) 或维度名称。您需要创建新的产品。

合同定价

使用合同定价模式，您可以向客户提供预付定价，使他们能够购买 1 个月、12 个月、24 个月或 36 个月的许可证。

以下产品可采用合同定价：

- AMI基于单一的产品和 AWS CloudFormation 基于模板AMI的产品。有关更多信息，请参阅 AMI产品的合同定价
- 基于容器的产品 有关更多信息，请参阅 容器产品的合同定价。
- 基于软件即服务 (SaaS) 的产品 有关更多信息，请参阅 SaaS 合同定价。

> **Note**
>
> 基于容器的产品的合同定价仅适用于新产品。AMI
> 如果您有现有产品AMI或基于容器的产品并想使用合同定价，请创建一个新的清单，然后使用产品加载表 (PLF) 应用合同定价模型，添加不同的维度，将基于容器的产品与 AWS License

Manager集成，然后发布该产品或基于容器的产品，然后发布该产品AMI或基于容器的产品。AMI
当客户以合同定价购买产品时，将由 AWS Marketplace 客户创建许可证，您的软件可以使用 Lic AWS 账户 ense Manager 查看该许可证API。客户需要一个IAM角色才能启动基于容器的产品AMI或基于容器的产品的实例。

自带许可定价

自带许可证 (BYOL) 产品不收取任何服务费 AWS Marketplace。

为了兑现我们对客户的选择承诺，我们要求所有BYOL产品都有付费选项。这样当前没有许可证的客户便可以选择购买并使用这些产品。

对于BYOL产品，我们意识到在线购买软件与某些公司的业务方式背道而驰。因此，在推出后的前 90 天，我们放宽了该软件必须附带可在 AWS Marketplace上购买的版本的要求。在此期间， AWS Marketplace 客户管理团队将与您合作应对挑战。该团队可以帮助您确定该软件是否以及如何在 AWS Marketplace上可供购买。

更改定价模式

定价模式的变更必须经过审查和批准， AWS Marketplace 以确保良好的客户体验并降低各方面临的风险。联系 M AWSarketplace 卖家运营团队，讨论您想要更改的定价模式。

处理和审核所有定价模式更改请求可能需要 30–90 天的时间。

更改价格

您可以通过 AWS Marketplace 管理门户更新价格和元数据。

更改价格

1. 登录到 AWS Marketplace 管理门户。
2. 在产品选项卡中，可以找到您创建的当前产品的列表。您可以在此处编辑您的产品列表或请求更改。

> **Note**
>
> 对于新订阅用户，价格变动立即生效。对于现有订阅用户，价格变动在自发送价格变动通知之日起的 90 天期限之后的下一个月的第一天生效。例如，假设您在 3 月 16 日发送了价格变动通

知。3 月 16 日之后的 90 天大约是 6 月 16 日。由于价格变动发生在 90 天期限之后的下一个月的第一天，因此变更的生效日期为 7 月 1 日。

专属优惠

在 AWS Marketplace 卖家私人报价计划中， AWS Marketplace 卖家可以EULAs与个人 AWS Marketplace 客户（买家）协商自定义定价。有关更多信息，请参阅 准备您的私人报价。

商品退款在 AWS Marketplace

所有付费产品 AWS Marketplace，无论定价模式如何，都必须有明确的软件费用退款政策。该退款政策必须包含退款条款以及联系卖家以申请退款的方式。作为卖家，退款政策的详细信息由您自己决定。但是，我们鼓励您为买家提供某种形式的产品使用退款。您必须遵守您发布的退款政策。本主题提供有关 AWS Marketplace 商品退款申请的类型、相关政策和批准流程以及如何为买家提交退款申请的信息。

AWS Marketplace 商品的退款申请类型

买家可以为 AWS Marketplace 商品申请不同类型的退款。对于由销售的 AWS Marketplace 产品 AWS，请参阅退款政策页面，然后使用提交联系支持表格 AWS Support Center Console。如果客户直接向其申请软件退款 AWS，我们会指导他们使用您发布的相关产品的支持联系信息联系卖家。任何 AWS 基础设施费用的退款均由用户自行决定， AWS 与软件退款分开处理。

如果产品由第三方出售，将指导买家查看产品详细信息页面上的退款政策。 AWS Marketplace 订阅的软件费用由产品的卖家支付，并且必须直接向卖家申请退款。每位 AWS Marketplace 卖家都必须在其 AWS Marketplace 页面上包含退款政策。

AWS Marketplace 产品退款政策和批准

以下列表概述了 AWS Marketplace 退款政策以及是否需要您的批准：

- 免费试用

  如果您将软件列为免费试用产品，则 AWS 可以在从免费试用版转换为付费订阅后的七天内代表您发放软件费用的退款。对于与免费试用转换相关的退款发放，您不需要执行任何操作。一经在产品上启用免费试用，即表示您同意该政策。
- 专属优惠

  私人优惠的所有退款都必须先获得您的授权，然后 AWS 才能处理。

- 软件计量退款

  如果您使用计量软件的使用情况 AWS Marketplace Metering Service，则 AWS 可以代表您退还因软件计数错误而产生的软件费用。如果这些错误在多个客户中很常见，则 AWS 保留为每位客户确定适当退款并直接向每位客户申请退款的权利。发放的与之相关的退款 AWS Marketplace Metering Service 必须与卖家确认一次，但不要求卖家确认每笔退款。与产品 AWS Marketplace Metering Service 一起使用，即表示您同意本政策。
- 购买后 48 小时内取消订阅

  如果买家在非专属优惠购买后的 48 小时内取消订阅，则 AWS 将发放全额退款（取消并获得 100% 的退款）。购买后 48 小时内取消即可获得退款，无需您采取任何措施。48 小时后，买家的此类请求由您自行决定。在 AWS Marketplace上架产品即表示您同意本政策。
- 订阅升级

  如果买家用更昂贵的订阅或同等价值的订阅取代现有的非私密优惠订阅，则 AWS 可以代表您为较低级别的订阅发放退款。对于买家来说，这是一个两步流程：购买新的订阅，然后申请取消旧订阅并获得退款。
- 订阅降级

  所有降级订阅退款请求都必须先获得您的授权，然后 AWS 才能处理。

所有 AWS 已授权的退款均自动处理，您无需采取任何行动。

AWS Marketplace 产品退款流程

您可以通过提交退款申请表来启动对产品软件使用的退款。Buyer S AWS Marketplace upport 团队收到后，将在AWS Support 中心控制台中创建相关的支持案例，并在主题行中注明退款状态。直接通过这些案例促进退款相关支持。有关更多信息，请参阅访问 AWS Support。

以下过程概述了如何为外部客户或内部测试账户申请退款。

为客户发起软件退款

1. 从客户处收集以下信息：

   - 与客户关联的电子邮件地址 AWS 账户。
   - 用于订阅您的产品的账户的客户 AWS 账户 号码。提醒您的客户，如果他们是某个组织的付款人，他们需要向您提供订阅您产品的关联账户的 AWS 账户 ID。
   - 客户要申请退款的账单周期。

2. 登录您的， AWS 账户 然后导航到退款申请表。
3. 在表格中输入客户的信息。
4. 输入客户申请退款的产品的产品 ID。您可以在每日客户订阅报告中找到产品 ID。
5. 对于客户申请退款、升级或降级的年度产品，您必须执行以下任务：
   a. 使用您的每日客户订阅报告确认客户购买了年度订阅（可能会出现 24 小时延迟）。
   b. 在注释字段中提供 Subscription Cancellation Date (订阅取消日期)。
   c. 在注释字段中提供您授权的更改（退款、升级或降级）的说明。
6. 提交表单。我们将收到通知，开始处理退款并向客户发放退款。
7. 将在 AWS Support 中心控制台中创建一个出站案例，并在退款申请中包含状态信息。主题行将包含下列项之一：
   - 已完成 – 退款已处理，无需采取进一步措施。
   - 待处理 – 退款将在当前账单周期结束后得到处理。
   - 所需操作 – 无法处理申请，我们需要您提供其他信息。您可以直接对支持案例做出回应，但还需要提交一个新的退款申请表。
8. 一旦成功处理退款，退款将在 24–48 小时内反映到客户的账户中。不过，资金出现在客户的财务账户中可能需要 5 个工作日。

您的 AWS Marketplace 产品所在的地区和国家

在中创建产品时 AWS Marketplace，您可以选择可用的 AWS 区域。您还可以选择买家可以从哪些国家/地区购买您的产品。这两个属性相似，但两者并不相同。例如，买家可能位于美国，并从美国购买产品，但正在 欧洲地区（法兰克福）区域安装您的产品。为了让该买家购买您的产品，您必须在国家/地区列表中同时包括美国，在区域列表中包括欧洲地区（法兰克福）区域。

AWS 区域

在创建或编辑服务器或机器学习产品信息时，您可以将产品限制在用户可以安装和使用该产品的特定 AWS 区域。

对于服务器产品，包括亚马逊系统映像 (AMI)-、基于容器的产品和 AWS CloudFormation基于容器的产品，您可以选择该产品可用的特定区域。您也可以选择在新的美国区域、非美国区域或所有区域推出时自动上架您的产品。

对于机器学习产品，您可以选择特定的区域，也可以选择所有区域，包括未来推出的区域。

有关 AWS 区域的更多信息，请参阅《AWS一般参考》中的AWS 服务终端节点。

国家/地区

默认情况下，所有可售国家/地区的买家均 AWS Marketplace 可购买您的商品。对于新的和现有的服务器和软件即服务 (SaaS) 产品，您可以出于税务、合规、支持或营销目的控制特定国家/地区的产品供应。

此功能有例外：

- 之前的购买 – 使用新的国家/地区列表更新您的产品后，已订阅您产品的买家在订阅有效期间仍可以访问您的产品。
- 专属优惠 – 当您将产品限制在仅对特定国家/地区的买家销售时，它不会限制专属优惠。当您向特定买家创建专属优惠时，该买家可以获得该优惠，即使他们所在的国家/地区未包含在您的指定国家/地区。

> **Note**
>
> 客户资格是在 AWS 关联账户级别确定的。有关更多信息，请参阅如何AWS确定您的账户位置？
>
> 共享其权利的客户只能在您允许的区域激活该权利。有关管理权利的更多信息，请参阅《AWS Marketplace 买家指南》中的在组织中共享订阅。

中的标准化合同 AWS Marketplace

为产品做好准备时，您需要确定约束产品使用的最终用户许可协议 (EULA)。您可以申请自己的最终用户许可协议，也可以使用标准合同 AWS Marketplace （SCMP）。南华早报是一种合同模板， AWS Marketplace 可帮助简化采购工作流程和加快交易速度。还提供了 AWS Marketplace (RCMP) 的经销商合同，这是独立软件供应商可以使用的标准化经销商合同模板。

本部分概述了您可以在 AWS Marketplace 中使用的两份标准化合同。

主题

- 的标准合同 AWS Marketplace
- 的经销商合同 AWS Marketplace

的标准合同 AWS Marketplace

AWS Marketplace 与买方和卖方社区合作制定了 AWS Marketplace （SCMP）的标准合同。SCMP 约束对数字解决方案的使用，并定义了买家和卖家对数字解决方案的义务。数字解决方案的示例包括服务器软件、软件即服务 (SaaS) 以及人工智能和机器学习 (AI/ML) 算法。

SCMP 主动定义了关键合同条款（如使用、保修、赔偿和适用法律）的共同点。卖家可以提供 SCMP 条款作为自助交易的 EULA，买家可以在其中搜索、购买和快速部署解决方案。对于专属优惠，买家可以向卖家请求 SCMP 模板，并且可以修改条款以满足双方商定的自定义交易要求。

您还可以将以下可选附录与 SCMP 一起用于自助服务或专属优惠：

- 增强安全附录 - 支持具有更高数据安全要求的交易。
- HIPAA 商业增订合约 - 支持符合《1996 健康保险责任与保护法》(HIPAA) 合规要求的交易。

> **免责声明**
>
> EULA 由您和买家签订。是否使用 SCMP 作为 EULA 由您自行决定。将 SCMP 应用于您的产品列表，即表示您选择加入 SCMP 计划。根据该计划， AWS 可以定期更新南华早报模板，并可能使用当前版本更新带有条款的产品清单。您可以随时通过将 SCMP 模板替换为自己的 EULA 来退出 SCMP 计划。

开始使用 SCMP

您可以将 EULA 更新为 SCMP，并将其提供给购买全新和现有单一亚马逊机器实例 (AMI) 和软件即服务 (SaaS) 产品的买家。您请求更新的过程取决于产品是否通过自助服务清单 (SSL) 上市。

如果您有任何疑问，请向 AWS Marketplace 标准合同团队发送一封电子邮件，网址为aws-mp-standardcontract@amazon.com。

要将通过 SSL 上架的 AMI 和 SaaS 产品的 EULA 更新为 SCMP，请执行以下操作：

1. 查看 AWS Marketplace 标准合同条款。
2. 登录到 AWS Marketplace 管理门户。
3. 对于产品，选择产品类型并选择要为其更新合同的产品列表。
4. 选择请求更改，然后选择更新区域和价格。
5. 如果尚未选择 EULA，请选择 AWS Marketplace 标准合同。

6. 在备注和通知 中，键入 **`Please update this product to SCMP V2`**。
7. 查看更改，然后选择提交以供审核。

要将并非通过 SSL 上架的 AMI 和 SaaS 产品的 EULA 更新为 SCMP，请执行以下操作：

1. 打开 AWS Marketplace 管理门户上的联系我们页面。
2. 登录您的 AWS Marketplace 卖家账户。
3. 按如下说明完成表单：
   - 对于您的问题主题，请选择商业 Marketplace。
   - 对于类别，选择产品上架。
   - 对于子类别，选择标准合同请求。
   - 在描述中，键入 **`Please update these products to SCMP V2`** 并提供您要使用 SCMP 更新的产品的标题和 ID。

的经销商合同 AWS Marketplace

经销商合同 AWS Marketplace (RCMP) 是一个标准化的经销商合同模板，ISV 在授权渠道合作伙伴向买家转售 ISV 产品时可以使用该模板。 AWS Marketplace 当 ISV 和渠道合作伙伴建立经销商关系和/或对其渠道合作伙伴专属优惠 (CPPO) 使用经销商条款时，该合同有助于减少法律合同审查的冗余性，并缩短上市时间。创建 CPPO 时，ISV 可以将合同上传到商机（转售授权），然后渠道合作伙伴可以查看和接受合同。 AWS Marketplace 买家无法查看加拿大皇家骑警。

> **免责声明**
>
> RCMP 是 ISV 的可选合同。如果 ISV 决定通过渠道合作伙伴转售其产品，则在创造机会时，他们可以附上 RCMP 或他们自己的定制合同条款（现有或预先协商的）。

开始使用 RCMP

本部分介绍如何查看条款和使用 RCMP。

在创造机会的同时使用加拿大皇家骑警 AWS Marketplace

1. 查看 AWS Marketplace 经销商合同条款。
2. 参考 RCMP 指南，为渠道合作伙伴创建转售机会。

类别和元数据

以下是提供产品元数据的最佳做法和信息。 AWS Marketplace 仅出于质量保证和错误更正目的而修改产品元数据。

命名和描述您的产品

买家可以看到您提供的有关产品的信息。确保潜在买家有足够的信息来做出有关购买您产品的明智决策。

创建产品名称

在创建产品名称时，请遵循以下准则：

- 使用标题大小写（每个重要单词的第一个字母大写）
- 确保买家仅按名称即可识别产品
- 使用品牌或制造商名称
- 避免描述性数据或夸张性措辞

产品名称示例：Smart Solution Load Balancer - Premium Edition。

编写产品说明

产品描述列出产品的特征、优势和用法。还可以提供其他相关的、具体的产品信息。说明长度最多为 350 个字符。

在编写产品说明时，请遵循以下准则：

- 避免不必要的大写
- 避免不必要的标点符号
- 不要包含重定向信息
- 检查拼写和语法
- 仅包含关键的有用信息

示例产品解决方案：智能解决方案自动将传入的应用程序流量分配到多个 Amazon EC2 实例。该解决方案可让您实现更好的应用程序容错性能，从而提供响应传入的应用程序流量所需的负载均衡容量。Smart Solution 可以检测出池中运行不正常的实例，并自动将流量重新路由到正常的实例，直到不

正常的实例恢复。您可以在单个AWS可用区或跨多个可用区启用智能解决方案，以确保更稳定的应用程序性能。

编写产品特色

产品信息页面最多显示三条产品特色信息。可使用这些特色信息简要描述产品的主要卖点。

产品特色示例：项目成本：使用 Smart Solution，您只需按实际用量付费。Smart Solution 的运行按小时计费，不足一小时以一小时计。

选择类别和关键词

在列出您的产品时，您最多可以为您的产品选择三个软件类别和相应的子类别。这可以帮助买家在浏览或搜索商品时发现您的商品 AWS Marketplace。仅选择与您的产品相关的分类，大多数情况下，仅一个分类适用。产品加载表单和 Products (产品) 选项卡均包含类别的完整列表。

类别与关键词不同。可用的类别和子类别是预定义的 AWS Marketplace，在产品请求过程中，您可以通过从列表中选择它们来决定哪些类别和子类别适用于您的产品。关键词不是预定义的，而是在相应过程中创建的。您不需要将类别添加为关键词。

创建搜索关键词

在产品请求过程中，您最多可以输入三个关键词（单个单词或短语），以帮助买家通过网站搜索找到您的产品。关键词字段最多可以包含 250 个字符。

下面的提示可以帮助您创建一组相关的搜索关键词：

- 使用相关术语。
- 请勿使用其他卖家已发布的产品名称或使用其他卖家的名称。
- m从买家角度选择关键词，即，买家在考虑您的产品类型时可能使用的单词和短语。
- 根据您的产品中的特定特征创建关键词。
- 请勿使用产品标题作为关键词。产品标题已经在搜索中编制索引。

注意：关键词与软件类别不同。关键词是与您的产品相关的更具体的搜索词。

AMI和容器产品使用说明

为产品创建使用说明时，必须包含以下信息：

- 客户保存的所有敏感信息的位置
- 解释所有数据加密配置
- S tep-by-step 关于轮换编程系统凭证和加密密钥的指令。the section called “基于 AMI 的产品要求”解释了使用凭证和加密密钥的列表的基本要求。
- 如果您的应用程序使用任何加密技术，请提供有关用户如何与您的应用程序交互以解密必要数据的详细说明
- S tep-by-step 说明如何评估和监控应用程序的运行状况和正常功能。例如：
  - 导航到您的 Amazon EC2 控制台并确认您位于正确的区域。
  - 选择实例并选择您启动的实例。
  - 选择服务器以显示您的元数据页面，然后选择页面底部的状态检查选项卡，以查看您的状态检查是通过还是失败。

编写发行说明

每次更新产品时，必须在发行说明中提供更改说明。发行说明应包含特定的信息以帮助用户决定是否安装更新。请在更新中使用清晰的标签，例如，在安全更新中使用“关键”，或者在其他类型的更新中使用“重要”或“可选”。

编写使用说明

提供使用说明，帮助确保买家能够成功配置和运行软件。您提供的使用说明将显示在配置过程中。

要编写有效的使用说明，请遵循以下准则：

- 假设用户感兴趣但不知情。
- 向用户提供启动和使用产品所需的所有信息，包括任何配置设置和特殊步骤。

示例使用说明：

1. 通过一键式操作启动产品。
2. 使用 Web 浏览器访问应用程序，网址为 https: //< EC2 _Instance_Public_ DNS >/index.html。
3. 使用以下凭证登录：
   - 用户名：user
   - 密码：实例的 instance_id

编写升级说明

提供有关买家如何从产品早期版本进行升级的详细信息。包括有关如何在创建另一个实例时保留数据和设置的信息。如果没有升级路径，请编辑此字段以明确说明。

示例升级说明：

1. 执行 ****，然后 ****。
2. 通过执行 *** 检查由项目使用的所有插件是否与版本 *.* 兼容。如果它们不兼容，执行 ***。
3. 通过执行 *** 备份您的数据。

撰写 CloudFormation 配送说明

使用 CloudFormation 配送时，还必须包括以下内容：

- AWS CloudFormation 模板创建的每个 AWS Identity and Access Management (IAM) 角色和IAM策略的用途
- AWS CloudFormation 模板创建的每个密钥的用途和位置
- 涉及多个元素的部署中的网络配置详细信息
- 详细指南，介绍如何启动应用程序，以及在部署包含多个 AWS 资源时如何配置应用程序以进行通信
- 定价明细，包括超出标准限制的运行 AWS 资源的成本。提供有关管理 AWS 服务限制的规范性指导。
- 所有数据加密配置。例如：亚马逊 S3 服务器端加密、亚马逊弹性区块存储 (亚马逊EBS) 加密、Linux 统一密钥设置 (LUKS) 等)

监控和评估应用程序功能

要监控和评估应用程序功能，请执行以下操作：

1. 导航到您的 Amazon EC2 控制台并确认您位于正确的区域。
2. 选择实例并选择您启动的实例。
3. 选择服务器以显示您的元数据页面，然后选择页面底部的状态检查选项卡，以查看您的状态检查是通过还是失败。

> **Note**
>
> 如果有任何数据存储是专有的， step-by-step 请提供配置、备份和恢复说明。

轮换编程系统凭证和加密密钥

the section called “基于 AMI 的产品要求”解释了使用凭证和加密密钥的列表的基本要求。

包括以下用于轮换编程系统凭证和加密密钥的内容：

- 有关管理 AWS 服务配额的规范性指南。有关更多信息，请参阅《AWS通用参考指南》。
- 定价明细，包括在标准配额之外添加的运行 AWS 资源的成本。这可以包含在您的产品使用说明中，也可以链接到包含有关管理和请求增加服务限额的详细信息的文档。

产品搜索引擎优化

搜索是买家旅程中的关键工具，它使客户能够找到满足其独特需求的最佳产品。对于 AWS Marketplace 客户而言，搜索主要发生在两个地方：搜索引擎（例如 Google 或 Bing）和中的搜索功能 AWS Marketplace。本页提供有关如何优化这两个地点的房源的信息。

搜索引擎优化

优化您的商品详情页面以提高相关关键词的排名对于通过搜索引擎吸引更多独立访问者访问您的商品详情页面至关重要。

有三个主要页面元素对于改善任何网页（包括商品详情页面）的自然搜索至关重要：关键词、标题标签和 H1 标题标签。

关键词

关键字是搜索引擎优化的核心要素，因为它们将话题提炼成有针对性的查询，从而推动搜索引擎的结果。确定与您的页面最相关的关键字的过程涉及关键字研究。搜索引擎优化工具可以提供有价值的信息，例如关键字搜索量（Google每月搜索一个关键字的次数）、当前排名、搜索趋势、关键字竞争力和相关关键词。通过这项研究，您可以确定主要和次要关键字。

您的主要搜索引擎优化关键字应该是代表页面主要主题的唯一单词或短语。这个主要关键词应该自然地编织到你的产品标题、简短描述和亮点部分的副本中。次要关键词应该是在剩余页面内容中找到的高度相关的术语。

标题标签

标题标签出现在搜索引擎结果页面中，并作为浏览器窗口或选项卡中页面的标题，它向读者和搜索引擎网络抓取者告知该页面的内容。对于 AWS Marketplace 产品详情页面，产品标题作为标题标签，因此使用搜索引擎优化关键字优化产品标题以提高排名潜力非常重要。为了增加在搜索引擎结果页面中获得高排名的可能性，请在标题标签中加入您的品牌名称、产品名称和相关关键字。

H1 标题标签

H1 标题标签有三个作用：

- 它们可以帮助访问者扫描页面内容以获取他们需要的信息。
- 它们为使用屏幕阅读器来理解页面内容的有视力障碍的访客提供了可访问性。
- 它们提供在页面标题中找到的关键字，如果后面的页面内容支持，这些关键字将获得额外的搜索引擎优化相关性权重。

AWS Marketplace 搜索

该 AWS Marketplace 网站使用类似于整个行业使用的搜索优化技术对搜索查询结果进行排名。通过了解搜索结果的 AWS Marketplace 排名和返回情况，您可以创建针对 AWS Marketplace 搜索引擎优化的产品详情。我们建议您在创建产品详情页面时考虑这一指南。

关键词

在产品创建过程中，您最多可以提交三个关键词（单个单词或短语），以帮助客户通过网站搜索找到您的产品。关键词文本框中最多可包含 250 个字符。

按照以下提示来创建搜索关键词：

- 使用相关的搜索词，以便客户可以轻松找到您的产品。
- 从您的客户角度选择关键词，即，客户在考虑您的产品类型时可能使用的单词和短语。
- 根据您的产品中的特定特征创建关键词。
- 请勿在您提交的关键词中包含产品标题。产品标题已经在搜索中编制索引。

> **Note**
>
> 关键词与软件类别不同。关键词是与您的产品相关的更具体的搜索词。

创建产品后，您可以通过编辑产品的元数据来编辑关键词。对于使用 AWS Marketplace 管理门户中的产品选项卡创建的产品，您也可以使用产品选项卡对其进行更改。有关更多信息，请参阅产品变更和更新。

AWS Marketplace 卖家运营团队帮助重定向使用发音相似的单词或含义相似的单词的查询。例如，当客户搜索 automobile 时，您希望他们搜索 car。

软件类别

在列出您的产品时，您最多可以为您的产品选择三个软件类别和相应的子类别。这可以帮助买家在浏览或搜索商品时发现您的商品 AWS Marketplace。选择与您的产品相关的类别。大多数情况下，仅一个类别适用。产品加载表单和 Products (产品) 页面均包含完整的类别列表。

> **Note**
>
> 类别与关键词不同。可用的类别和子类别是为 AWS Marketplace预定义的。您可以从列表进行选择，确定将哪些类别应用于您的产品。关键词不是预定义的，它们是在相应过程中创建的。

特色部分

产品详情页面最多显示三条产品特色信息。客户可以通过特色搜索产品，因此在创建产品时请包括特色。特色应该以简明扼要的信息性措辞，描述产品的主要卖点。

Example 要点

- 预测成本：使用 AnyCompany 's产品，您只需为实际使用量付费。它的运行按小时计费，不足一小时以一小时计。

简短描述

产品说明列出了产品的特征、优势和使用说明，并提供其他相关和特定的产品信息。在创建产品说明时，请遵循以下准则：

- 避免不必要的加粗和标点符号
- 不要包含重定向信息
- 检查拼写和语法
- 仅包含关键的有用信息

Example 简短描述

AnyCompany的产品会自动将传入的应用程序流量分配到多个 Amazon EC2 实例。它使您能够无缝提供响应传入应用程序流量所需的负载平衡容量，从而提高应用程序的容错能力。 AnyCompany的产品会检测池中的不健康实例，并自动将流量重新路由到运行正常的实例，直到恢复不健康的实例。客户可以在单个 AWS 可用区或跨多个可用区启用它，以实现更稳定的应用程序性能。

准备您的私人报价

私人报价是用于从中购买产品的协商条款 AWS Marketplace。这可能涉及自定义定价计划、最终用户许可协议 (EULA) 或自定义解决方案。买卖双方在承诺与公开报价不同的私募报价之前先进行谈判。

您可以为单个买家创建和提供多个专属优惠。您为其提供专属优惠的买家可以在专属优惠和公开优惠之间进行选择。在任何给定时间买家只能订阅一种优惠。他们不能同时订阅专属和公开优惠。

> **Note**
>
> AWS 为具有独特或企业用例的 AWS Marketplace 买家提供直接从商品详情页面为您的商品申请私人报价。如果您是符合APN客户互动 (APN) 资格的 AWS 合作伙伴网络 (ACE) 合作伙伴，并且您想向买家提供此选项，请联系您的 AWS 代表以确定进一步的资格。

主题

- 专属优惠的工作原理
- 私人优惠支持的产品类型
- 创建和管理私密优惠
- AWS Marketplace 渠道合作伙伴私享优惠
- 分期付款计划
- 私人优惠升级、续订和修改
- 未到期的协议和私人报价

专属优惠的工作原理

您可以从中的优惠页面创建和管理您的私人报价AWS Marketplace 管理门户。您可以为报价指定产品以生成唯一的 ID 和URL。您将为私募报价制定定价计划，添加法律条款和销售文件，并将报价范围扩大到特定的买家 AWS 账户。该优惠仅对您创建出价的账户可见。

在您创建私人报价并通知潜在买家后，他们可以查看和接受报价。要查看报价，买家必须登录收到报价的。 AWS 账户

> **Note**
>
> 除非您将其扩展到他们的关联账户或管理账户，否则买家无法查看报价。您不能在报价中提供服务限制，因此除非产品有限制，否则买家可以按协议价格随心所欲地使用您的商品。

有关创建私有优惠的信息，请参阅创建和管理私有优惠。

卖家报告中会跟踪专属优惠。有关更多信息，请参阅私人报价报告和卖家报告指南。

私人报价注意事项

在处理私人报价时，请考虑以下几点：

- 当您添加对新实例类型的支持或时 AWS 区域，已订阅产品私有优惠的客户将无法自动访问新添加的实例或区域。您必须使用您希望客户访问的实例和地区创建另一个私有报价。接受新优惠后，客户可以访问新添加的实例和区域。在将来的某个日期订阅您的产品的客户也可以访问这些产品，前提是这些产品包含在私人优惠中。有关如何创建新的私有优惠的更多信息，请参阅私有优惠升级、续订和修改。
- 您不能为第二方、每月的 Amazon Machine Image (AMI)、使用 AWS CloudFormation 产品的多AMI 基配送创建私密报价，也不能为限制买家使用创建私密报价。
- 对于分期付款计划的私人报价，可以随着时间的推移将预付款分为多笔付款。欲了解更多信息，请参阅分期付款计划。
- 如果您的专属优惠的买家账户是通过 Private Marketplace 管理的，则您必须在优惠中同时包括买家账户和包含其 Private Marketplace 管理员的账户。
- 私人优惠不支持 “自带许可证” (BYOL) 模式。
- 在私募报价中创建具有独特协议合同条款的私募报价时，请使用 “自定义” EULA 选项。您最多可以附加五份文档。
- 对于软件即服务 (SaaS) 合同和带消费产品的 SaaS 合同，您可以为买家接受专属优惠时所订立的协议提供升级和续订。例如，您可以这样做来授予新的权利、提供定价折扣、调整付款时间表或更改最终用户许可协议 (EULA) 以使用标准化许可条款。有关更多信息，请参阅私人优惠升级、续订和修改。

为买家提供私人优惠体验

在买家导航到您的产品的订阅页面时，将显示一个横幅以指示具有可用的专属优惠。买家接受报价后，将使用与所有 AWS Marketplace 交易相同的门户网站工具为他们开具购买发票。接受的报价变成协

议。买家可以在的 “管理订阅” 部分中找到协议详情 AWS Management Console，卖家可以在的 “协议” 选项卡中找到详细信息 AWS Marketplace 管理门户。

AWS Marketplace 买家可以获得第三方融资以获得私人报价。有关更多信息，请参阅中现已推出客户融资 AWS Marketplace。

> **Note**
>
> 优惠只能在到期日之前接受。如果报价过期，则会移至 “已接受和已过期报价” 选项卡。

要查看和接受专属优惠，请执行以下操作：	买家可以
从控制 AWS Marketplace 台	在 AWS Marketplace 控制台中导航到私人优惠，然后从可用优惠选项卡中选择优惠 ID。 有关专属优惠的买家体验的更多信息，请参阅《AWS Marketplace 买家指南》中的专属优惠。
使用卖家提供的链接	点击卖家发送的链接直接访问专属优惠。 有关更多信息，请参阅向买家发送专属优惠。
在您的产品页面上	导航到该产品的产品页面，然后在横幅中选择链接以查看专属优惠。 有关专属优惠的买家体验的更多信息，请参阅《AWS Marketplace 买家指南》中的专属优惠。

专属优惠报告

专属优惠显示在现有卖家报告以及与优惠相关的报告中。月度账单收入报告每月生成一次，并具有优惠可见性和优惠 ID 信息。在为买家生成发票时，它将显示在涵盖相应计费周期的报告中。有关更多信息，请参阅卖家控制面板。

优惠 ID 字段包含为专属优惠生成的唯一优惠 ID。除非报告条目适用于专属优惠，否则，该字段是空白的。优惠可见性字段指示报告条目是普通优惠还是专属优惠。对于所有专属优惠，该条目标记为专属。

私人优惠支持的产品类型

私有产品支持亚马逊机器映像 (AMI)、容器、专业服务和软件即服务 (SaaS) 产品。

AMI 产品的专属优惠

您可以为 AMI 产品供专属优惠定价。

该优惠可以是以下任意自定义期限：

- AMI 按小时或 AMI 按小时 - 年度合同专属优惠：最长 3 年（1095 天）。只有 AMI 按小时 - 年度专属优惠或 AMI 合同支持灵活付款计划。
- AMI 合同专属优惠：最长 5 年（60 个月）

  对于 AMI 合同，专属优惠不监控用量。

  买家可以随时手动升级到新的合同等级。但是，独立软件供应商 (ISV) 负责定义合同等级，强制执行服务限制，并建议买家使用更多单元手动升级合同。目前只有基于非分层定价的合同支持升级。专属优惠的合同期限可以与公开产品列表相匹配，也可以是以月为单位的自定义期限（最多 60）。

许可证授权从买家接受专属优惠开始计算。

对于具有灵活付款计划的 AMI 专属优惠，您可以在合同中设置合同期限内的约定年度实例类型数。

> **Note**
>
> 专属优惠不适用于按月计费合同。

容器产品的专属优惠

您可以为基于容器的产品合同提供专属优惠定价。

该优惠可以是以下任意自定义期限：

- 容器按小时或容器按小时 - 长期专属优惠 - 最长 3 年（1095 天）。只容器有按小时 - 长期专属优惠或容器合同支持灵活付款计划。
- 容器合同专属优惠 - 最长 5 年（60 个月）

对于容器合同，专属优惠不监控用量。只有在使用非分层定价时，才能升级容器合同。

买家可以随时手动升级到新的合同等级。但是，独立软件供应商 (ISV) 定义合同等级，强制执行服务限制，并建议买家使用更多单元手动升级合同。目前只有基于非分层定价的合同支持升级。专属优惠的合同期限可以与公开产品列表相匹配，也可以是以月为单位的自定义期限（最多 60 个月）。

许可证授权从买家接受专属优惠开始计算。对于具有灵活付款计划的容器专属优惠，您可以在合同中设置合同期限内的约定单位数。如果买家使用更多单位，您也可以为同一批单位定义自定义的每小时价格。

> **Note**
>
> 专属优惠不适用于按月计费合同。

专业服务产品的专属优惠

所有专业服务产品均通过专属优惠提供。有关更多信息，请参阅 创建专属优惠。

SaaS 产品的专属优惠

软件即服务 (SaaS) 专属优惠产品不能根据时间更改给定定价层的定价级别。例如，优惠不能在前三个月内按每小时 0.80 美元收费，然后又将相同定价层的定价更改为每小时 0.60 美元。对于 SaaS 合同，专属优惠不监控用量。

买家可以随时手动升级到新的合同等级。但是，独立软件供应商 (ISV) 定义合同等级，强制执行服务限制，并建议买家在需要时手动升级到更高的合同等级。专属优惠的合同期限可以与公开产品列表相匹配，也可以是以月为单位的自定义期限（最多 60 个月）。

机器学习产品的专属优惠

机器学习 (ML) 专属优惠产品为特定买家提供与公开显示价格不同的价格。您和买家之间在专属优惠中达成的条款和协议可能与公开优惠或其他专属优惠中的条款和协议不同。

专属优惠采用以下任一工作原理：

- 按小时 - 专属优惠可以是与公开显示的小时费率不同的小时费率。这种小时费率是永久性的，因为机器学习产品的私人优惠不会过期。如果将来需要更改价格，则买方必须切换到新的私募报价。产品的

现有运行实例或终端节点将自动按新接受的报价中设定的小时费率计费。在私人报价中的任何合同部分到期后，请务必将其设置为产品的小时费率。将此小时费率设置为 0 美元，买家可以无限期使用产品，无需支付软件费用。

- 按推理 - 如果您已在将产品部署为端点时配置了推理定价，则专属优惠的推理费率可能与公开显示的推理费率不同。
- 合同 - 专属优惠可以是一份针对指定天数收取固定预付费用的合同。允许买家在整个合同期限内使用无限数量的实例。合同到期时，任何继续运行的实例均按您在专属优惠中设定的小时费率计费。例如，您可以创建一个具有固定预付费用的合同，365 天无限制使用。您还可以为专属优惠设置小时费率。当买家接受此专属优惠时，他们需要支付预付费用。合同到期后，任何仍在运行的实例都将按该小时费率计费。如果您提供免费的专属试用，请确保在免费试用期结束后设置正确的小时费率，以免提供免费的永久许可。

您可以为单个买家创建和提供多个专属优惠。您为其提供专属优惠的买家可以在专属优惠和公开优惠之间进行选择。在任何给定时间买家只能订阅一种优惠。他们不能同时订阅专属和公开优惠。

要为特定买家创建 SageMaker 产品私下报价，请联系 AWS Marketplace 卖家运营团队。

申请创建私有报价时，您必须提供以下信息：ProductId、目标买家 AWS 账户、日期和时间（指明何时必须接受报价）、自定义 EULA 文件（可选）、退款政策、许可期限（可选）、许可价格（可选）、灵活的付款计划（可选）以及每种实例类型的价格。优惠创建后，买家必须在指定的日期和时间内接受优惠。

创建和管理私密优惠

以下各节介绍如何创建和管理私密优惠。

主题

- 开始新的私人报价
- 了解报价状态
- 起草和发布私人报价
- 向买家发送专属优惠
- 保存您的私人报价进度
- 更新私募报价的到期时间
- 取消私人报价

开始新的私人报价

使用以下流程创建报价并使用 createOffer 变更请求生成报价 ID。API它会在草稿状态下创建空白报价。

开始创建新的私人报价

1. 登录 AWS Marketplace 管理门户，然后选择 “优惠”。
2. 在报价页面上，选择创建报价。
3. 在 “创建报价” 页面上，选择产品类型和您想要创建私人报价的产品。处理最多需要 30 秒。在此过程中，请勿关闭或刷新页面。

> **Note**
>
> 创建报价后，您将无法更改商品类型和商品。有关每种产品类型的私人优惠的更多具体信息，请参阅支持的产品类型。
> 如果您是 AWS Marketplace 渠道合作伙伴，请选择是为自己的产品创建报价，还是通过转售授权创建 AWS Marketplace 渠道合作伙伴私有报价 (CPPO)。如果是CPPO，请选择独立软件供应商 (ISV)、产品和授权。

4. 选择 “继续” 以提供详细信息。 step-by-step 体验将开放，因此您可以继续创建私人优惠。

了解报价状态

根据生命周期，优惠具有以下三种状态之一：

- 草稿 — 报价不完整，您仍在准备中。必须填写并提交所有必填的详细信息才能发布报价并将其扩展到您的买家。
- 在售-报价已发布并提供给买家。该优惠尚未过期，因此买家可以订阅该报价。
- 已过期-报价已发布并提供给买家。该优惠已过期，因此买家无法订阅该报价。可以更新到期日期，让您的买家有更多时间接受报价。要更新报价到期时间，请参阅更新私募报价的到期时间。

> **Note**
>
> 报价被接受后，它将作为协议显示在 “协议” 选项卡中。优惠状态不会改变。

起草和发布私人报价

使用以下流程起草和发布您的私人报价。

起草和发布您的私人报价

1. 在提供优惠信息页面上，提供优惠名称、优惠详情、续订类型和优惠到期日期。如果这是续订优惠，则必须选择 “现有客户启 AWS Marketplace用” 以续订在上创建的现有协议 AWS Marketplace，或者选择 “现有客户移至 AWS Marketplace” 以续订现有客户，以便将现有客户迁移到。 AWS Marketplace

   > **Note**
   >
   > 报价到期日是报价失效的日期。在此日期 23:59:59 UTC 之后，买家将无法查看和接受此私下报价。

2. 选择下一步。
3. 在配置优惠定价和期限页面上，选择定价模式、合同或使用期限、定价和付款时间表。有关具有分期付款计划的定价模式，请参阅分期付款计划。
4. 在添加买家页面上，为要向其提供私密报价的每位 AWS Marketplace 买家提供一个 AWS 账户 ID。选择 “添加其他买家” 按钮以添加另一个 AWS 账户 ID。您最多可以为每份私人报价添加24位买家。
5. 选择下一步。
6. 在配置法律条款和报价文件页面上，选择以下选项之一：
   - 公开发售最终用户许可协议 (EULA) — 使用公开报价中的。EULA
   - AWS Marketplace (SCMP) 的标准合同-使用提供的标准合约 AWS Marketplace。
   - 自定义法律条款 — 最多上传五个与您的私人报价相关的文件，包括法律条款、工作说明书、物料清单、定价表或其他附录。创建选件时，这些文件将合并为一个文档。
7. 在 “查看并创建” 页面上，查看您的私人报价的详细信息。查看并确认后，选择创建报价以发布报价并将其扩展到您选择的买家。报价发布包括对 AWS Marketplace 目录的请求API，因此验证和处理报价最多可能需要一个小时。可以在请求页面上查看此请求。

> **Note**
>
> 只有在申请成功后，才会发布和延长报价。如果请求失败，则不会将其扩展到客户。失败意味着存在系统错误或错误，您需要在重新提交之前更正。

以下指南提供了有关为特定商品创建私密报价的更多信息。

- AWS Marketplace — 创建 SaaS PAYG 私人优惠
- AWS Marketplace — 创建AMI每小时/私人优惠 PAYG
- AWS Marketplace — 使用年度私人优惠按AMI小时创建
- AWS Marketplace — 创建 SaaS 合同私人报价

以下视频详细介绍了如何创建SaaS合同私募报价。

向买家发送专属优惠

私募报价发布后，买家可以通过导航到 “私人报价” 页面上的 “可用私人报价” 选项卡进行查看。 AWS Marketplace 管理门户在 “可用的私人报价” 选项卡上，买家可以在记录在案的卖家栏中看到 AWS Marketplace 渠道合作伙伴提供的报价。独立软件供应商 (ISV) 将显示在 “出版商” 栏中。买家可以通过在优惠列表中选择相应的 优惠 ID 来导航到专属优惠。

买家可以在已接受或IDs已过期的报价选项卡上查看已接受或已过期的报价。

私下报价发布后，您可以将买家发送URL到报价的配送页面。

向您的买家发送私密报价

1. 登录 AWS Marketplace 管理门户，然后选择 “优惠”。
2. 选择报价旁边的单选按钮。
3. 选择 “活动”，然后选择 “复制报价” URL。
4. 将其发送URL给您的买家。

保存您的私人报价进度

使用以下流程保存您的进度并稍后继续。

保存并恢复您的工作

1. 在任何已完成的步骤中，选择 “保存” 并退出。在对话框中，确认您正在将内容保存为草稿状态并查看所有验证错误。如果存在任何验证错误或缺少细节，则可以选择 “修复”，进入步骤并解决问题。准备就绪后，选择 “保存” 并退出以保存更改。

   保存并退出后，请求正在审核中。完成处理可能需要几分钟或几小时。在请求成功之前，您无法继续执行这些步骤或进行更改。请求成功后，您就完成了保存。如果请求失败，则说明存在系统错误或错误，您需要在重新提交之前进行更正。
2. 要继续处理您的报价，请打开选件页面，选择您的报价，然后选择恢复选件创建。
3. 完成后，您可以选择 “保存并退出” 以保存进度，也可以选择 “创建报价” 来发布私密报价并将其扩展到选定的买家。

更新私募报价的到期时间

使用以下流程更新私人报价的到期日期。

更新私募报价的到期日期

1. 登录 AWS Marketplace 管理门户，然后选择 “优惠”。
2. 在优惠页面上，选择您要更新的报价。
3. 选择编辑。
4. 提供新的报价到期日期。
5. 选择提交。

   更新完成后，报价将变为 “在售” 状态，您的买家可以接受报价。

取消私人报价

使用以下流程取消私人报价。

1. 登录 AWS Marketplace 管理门户，然后选择 “优惠”。
2. 在优惠页面上，选择您要更新的报价。

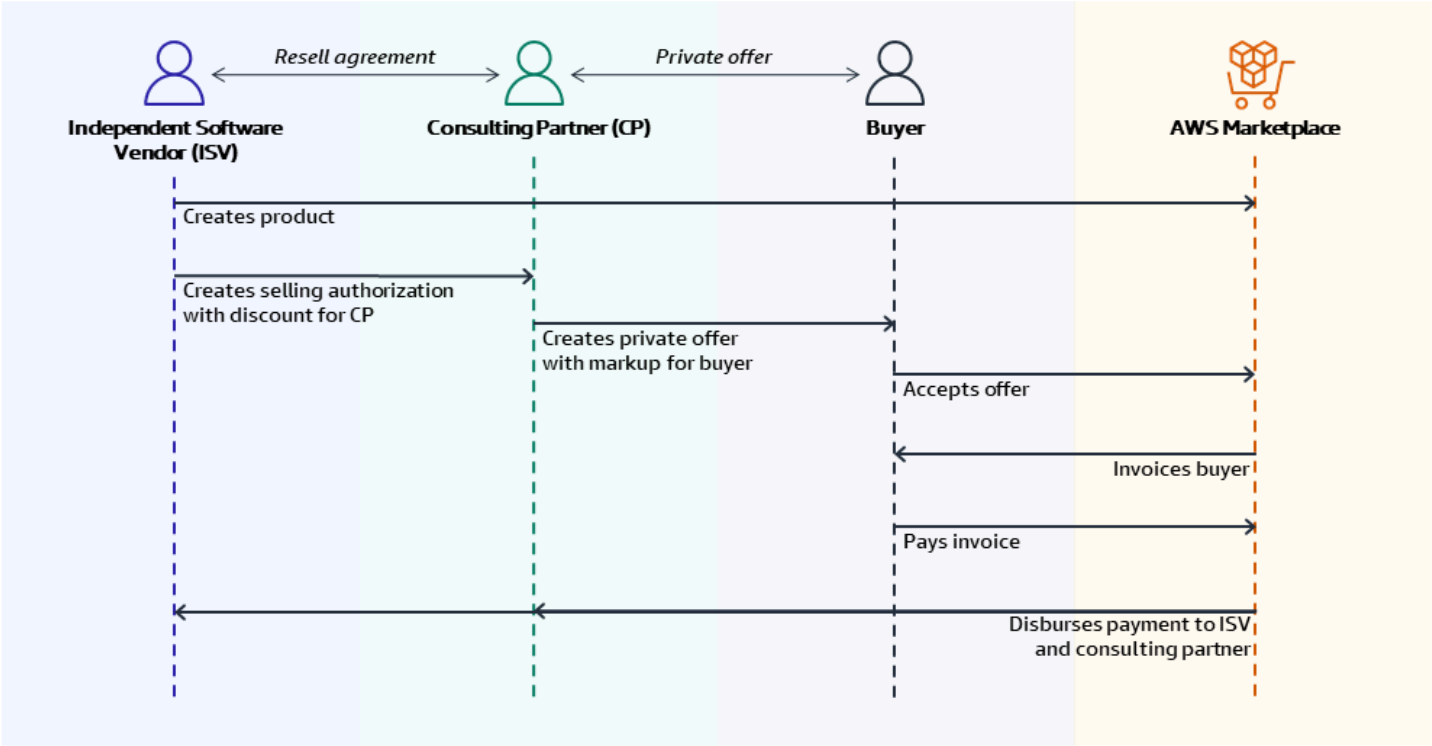
> **Note**
>
> 取消报价将修改报价到期日期，因此对于延长此报价的买家，该报价将显示为已过期。

3. 选择 “操作”，然后选择 “取消报价”。

AWS Marketplace 渠道合作伙伴私享优惠

AWS Marketplace 渠道合作伙伴私有优惠允许渠道合作伙伴通过以下方式转售独立软件供应商 (ISVs) 的产品。 AWS Marketplace AWS Marketplace 渠道合作伙伴签订协议，转售其中的ISV一个或多个产品，然后他们向买家提供该产品的私下报价。ISV

下图显示了渠道合作伙伴和买家之间的这种关系。ISV



> **Note**
>
> 有关为渠道合作伙伴创建销售授权的更多信息ISV，请参阅为 AWS Marketplace 渠道合作伙伴创建销售授权 ISV。

每个 AWS Marketplace 渠道合作伙伴的私人报价仅对单个买家可见，并提供定制的定价和独特的商业条款，以满足该买家的需求。创建私人报价时，您需要从设定的批发成本开始ISV。然后，您可以在该价格的基础上加价以创建买家优惠价格。采用以下两种方法之一确定批发成本：

- 经常性折扣 — ISV 授权 AWS Marketplace 渠道合作伙伴通过定期销售授权，以标价的商定折扣转售其或多款产品。此折扣（discount）允许 AWS Marketplace 渠道合作伙伴继续转售产品，而无需与渠道合作伙伴进一步协商价格。ISV此折扣可以设置为持续到指定日期，也可以无限期地持续到或渠道合作伙伴终止为止。ISV
- 非经常性折扣 — 授ISV予 AWS Marketplace 渠道合作伙伴的销售授权是一次性折扣，仅适用于特定的买家。

在这两种情况下，在买方支付私募报价之后，都 AWS Marketplace 使用标准流程将资金分配给 AWS Marketplace 渠道合作伙伴，并ISV根据商定的定价向渠道合作伙伴分配。

> **Tip**
>
> 作为ISV或渠道合作伙伴，您可以从的合作伙伴菜单中查看您已授予或收到的AWS Marketplace 管理门户销售授权。

有关创建专属优惠的详细说明，请参阅 AWS Marketplace 渠道合作伙伴创建。

有关专属优惠的第三方融资的信息，请参阅 AWS Marketplace中现已提供客户融资。

其他信息

如需更多信息和问题，我们鼓励ISVs渠道合作伙伴与 AWS Marketplace 渠道团队联系。如果您不知道具体联系谁，请发送电子邮件至 aws-mp-channel@amazon .com，团队成员将在一个工作日内回复您。

为 AWS Marketplace 渠道合作伙伴创建销售授权 ISV

作为一个ISV，您可以通过为 AWS Marketplace 渠道合作伙伴创建销售授权来授权该合作伙伴转售您的产品。您可以为每个产品维度指定折扣百分比或自定义价格，为 AWS Marketplace 渠道合作伙伴创建批发价格。合作伙伴可以在为买家创建 AWS Marketplace 渠道合作伙伴私人报价时加大批发价格。支持的产品类型包括：

- AMI基于产品
- 基于容器的产品

- 基于 SaaS 的产品
- 专业服务产品

以下程序概述了ISVs如何为 AWS Marketplace 渠道合作伙伴创建销售授权。要使用此特征，您必须具有使用 AWS Marketplace 管理门户中的合作伙伴选项卡的权限。有关更多信息，请参阅 AWS Marketplace 卖家政策。

为 AWS Marketplace 渠道合作伙伴创建销售授权 ISV

1. 使用您的 AWS Marketplace 卖家账户登录。AWS Marketplace 管理门户

   > **Tip**
   > 在使用 AWS Marketplace 卖家账户登录 AWS 账户 之前，请确保您已从其他账户注销。

2. 选择 “合作伙伴” 选项卡，然后选择 “创建销售授权”。
3. 在销售授权详情页面上，输入销售授权名称和销售授权描述。

   > **Note**
   > 渠道合作伙伴将在其卖家报告中看到您在销售授权名称和销售授权描述中输入的信息。

4. 对于经销商，请从下拉列表中选择要授权的 AWS Marketplace 渠道合作伙伴（经销商）。您可以按名称或账户 ID 选择经销商。
5. 选择您的一款商品作为此销售授权的一部分。
6. 选择要应用的折扣。

   折扣类型可以通过多种方式发放：

   - 折扣百分比 - 对所有选定产品应用同一折扣率（百分比）。
   - 个人定价 - 对特定产品应用特定折扣。
   - 灵活的付款时间表-为 AWS Marketplace 渠道合作伙伴的销售授权设置灵活的付款时间表。

   > **Note**
   > 专业服务卖家仅支持个人定价和灵活的付款时间表。

7. 选择销售授权的期限。

> **Note**
> 经销商的开始日期必须早于制造商在转售授权中列出的日期。

销售授权的时间长度可以通过多种方式发放：

- 单次使用 — 适用于一次销售授权，在 AWS Marketplace 渠道合作伙伴创建私人报价后不再适用。
- 特定持续时间 —持续一段特定的持续时间，该持续时间在选择的日期之后不再适用ISV.
- 未设定持续时间 - 持续到其中一方结束为止。

8. （可选）对于 SaaS 合同产品，添加或删除自定义产品维度，并修改额外使用费以自定义您的销售授权。
9. （可选）设置一个或多个买家账户 IDs，指定销售授权仅适用于这些买家。
10. （可选）选择最终用户许可协议 (EULA) 版本或上传EULA要包含在销售授权中的版本。

> **Note**
> 专业服务卖家EULAs仅支持定制。

11. （可选）选择 AWSMarketplace 的经销商合同（RCMP）或上传要包含在销售授权中的自定义合同。
12. 选择 “查看销售授权”，并确保信息正确无误。
13. 选择创建销售授权以完成销售授权并向渠道合作伙伴授权。

    销售授权创建表已更新，显示相关的销售授权详细信息，包括销售授权名称、产品名称、经销商名称、折扣、创建日期和状态。

创建销售授权后，您无法延长其日期。但是，您可以随时撤销销售授权并重新创建。当您撤销销售授权时，新报价不能使用该折扣。任何现有报价均不受影响，并保留其销售授权折扣（discount）。

您也可以通过选择销售授权并选择 “克隆” 来克隆销售授权。这会预先填充所有内容，然后您可以编辑字段

分期付款计划

分期付款计划（也称为灵活的付款时间表）使您可以通过自定义的付款时间表来延长私人优惠。时间表可以分散在已接受的合同期限内，客户可以定期分期付款。订阅后，您的客户可以在时间表和 AWS 发票上看到所有付款，从而帮助他们跟踪支出。分期付款计划适用于某些产品和定价类型的私人优惠。有关更多信息，请参阅符合专属优惠条件的产品类型。

创建付款计划

使用分期付款计划创建自定义付款时间表的过程是创建私人报价过程的一部分。要为您的私人报价启用分期付款计划选项，请在创建私人报价时，在 “买家付款选项” 部分中选择 “带分期付款计划的合同定价”。选择买家付款选项后，选择此优惠的合同期限并指定报价详情。有关更多信息，请参阅私人优惠。

> **Note**
>
> 对于分期付款计划的私有优惠、多年期和自定义期限的 Amazon Machine Image (AMI) 产品，请为优惠中包含的每种实例类型设置实例数量以及任何其他已启动实例的小时定价。客户在启动指定数量的实例后，启动任何其他实例都按专属优惠中指定的小时费率收费。

在 “买家分期付款计划” 下，提供合同总额、付款次数和第一张发票日期，然后选择生成分期付款计划。您最多可以添加 60 笔付款。您还可以选择对每个付款项目进行调整。每次调整付款项目时，买家应付的总金额都会更新。

> **Note**
>
> 第一期分期付款的发票日期是为您的专属优惠向客户开发票的日期。收到客户的付款后，您将 AWS Marketplace 收到每张发票的付款。

灵活付款计划程序特征验证发票日期是否在合同期限内。如果任何发票日期晚于合同期限，则您会收到一条错误消息。

验证所有发票金额和日期后，确认买家应付的总金额与您希望客户在私募报价期间支付的总价格相符。要完成私密报价的创建，请完成其余步骤。

> **Note**
>
> 在您向客户提供优惠接受日期之前，只能有一个发票日期。

您的客户将根据您定义的计划开具发票，并在他们接受优惠后开始开具发票。如果第一张发票日期安排在报价被接受之前，则会在报价被接受后立即进行处理。

> **Note**
>
> 对于已向买家提供并由买家订阅的专属报价，您无法修改付款计划。要进行更改，您必须创建一个新的优惠。

报告分期付款计划

具有灵活付款计划的专属优惠的报告位于月度账单收入报告的 第 4 部分：具有灵活付款计划的合同中。

私人优惠升级、续订和修改

管理私人报价协议

对于软件即服务 (SaaS) 合同和消费类产品的 SaaS 合同，您可以提供升级、续订和修改，以取代最初在买方接受您的公开报价或私募报价时创建的有效协议。例如，您可以这样做来授予新的权利、提供定价折扣、调整付款时间表或将最终用户许可协议 (EULA) 更改为使用标准化许可条款。您还可以更改件数和付款计划，并添加自定义的结束日期。

优惠 和协议 之间的区别在于买家是否接受了其条款：

- 优惠是买家使用产品的一组条款。优惠可以是普通的或专属的。
- 协议是买家接受的优惠。协议包括卖家通过公开或私人报价提供的已购买和免费产品。

此功能适用于所有 AWS Marketplace 卖家，包括独立软件供应商 (ISVs) 和渠道合作伙伴。您不能修改协议以指定与原始协议的记录卖家不同的记录卖家。

要使用此特征，您必须具有使用 AWS Marketplace 管理门户中的协议选项卡的权限。有关信息，请参阅AWS Marketplace 卖家权限。

私募报价修改支持的产品类型

您可以在 “协议” 页面上查看以下产品类型：

- SaaS 合同
- 具有用量的 SaaS 合同
- 基于 SaaS 使用情况的产品
- AMI基于的产品
- 基于容器的产品
- 服务器合同
- 专业服务产品

但是，只有以下产品类型支持私募报价修改：

- SaaS 合同
- 具有用量的 SaaS 合同

升级和续订的提交流程

您可以使用以下步骤从中创建私有优惠升级、续订和修改。 AWS Marketplace 管理门户

创建私有优惠升级、续订和修改

1. 登录到 AWS Marketplace 管理门户 并选择协议。
2. 在 “协议” 页面上，选中协议旁边的复选框，然后选择 “查看详细信息”。
3. 在查看协议页面上，选择创建基于协议的报价。
4. 在修改协议详情页面上，卖家必须指明私募报价是否用于续订。当系统询问私人报价是否需要续订时，选择 “是” 或 “否”。

   > **Note**
   >
   > 您还可以在修改协议详情页面上更改服务日期、产品尺寸、付款时间表、续订状态和优惠到期日期。

5. 完成后，选择 “创建报价”，然后选择 “提交”。

> **Tip**
>
> 输入描述性的自定义优惠名称可帮助您在优惠页面上区分有效优惠。买家也可以看到自定义优惠名称。
>
> AWS 建议您指定包含任何其他识别详细信息的自定义报价名称，例如您自己的编号IDs和采购订单号。还建议使用概要描述，如 **`upgrade`** 或 **`renewal`** 和自定义公司名称。请勿使用任何个人身份数据（例如，名字或姓氏、电话号码或地址）。您最多可以为此字段输入 150 个字符。
>
> 编辑任何日期、维度、付款时间表以及EULAs您想要更改的信息，然后选择 “下一步”。
>
> 在查看并创建页面上，查看信息。准备就绪后，选择创建基于协议的报价。

新的专属优惠将在管理专属优惠页面上显示大约 45 分钟的时间。要查看优惠，请登录 AWS Marketplace 管理门户 并选择优惠以打开 “管理私人优惠” 页面。

从那里，买家可以选择接受它或继续按照原始协议运营：

- 如果买家接受专属优惠升级或续订，则新协议将立即生效，并且该协议将列在 AWS Marketplace 管理门户的协议页面上。将取消以前协议中的任何剩余计划付款。

  买家接受基于协议的专属优惠与他们接受专属优惠的方式相同。有关专属优惠的买家体验的更多信息，请参阅《AWS Marketplace 买家指南》中的专属优惠。
- 如果买家在专属优惠升级或续订到期前未接受它，则原始协议仍然有效，不作任何更改。

> **Note**
>
> 对于 SaaS 产品的亚马逊SNS通知，当买家接受基于协议的报价`offer-identifier`时，会随新产品一起发送`subscribe-success`一条消息。

报告升级、续订和修订

升级和续订专属优惠显示在现有卖家报告以及与优惠相关的报告中。每天都会生成每日客户订阅用户报告报告和每日业务报告报告。每月生成一次月度账单收入报告报告。

在每日客户订阅用户报告中，订阅意图字段指示报告条目是否为新的专属优惠。以前的优惠 ID字段指示新优惠之前的优惠 ID（如果存在）。对于所有私人报价，该条目都标记为私人。

> **Important**
>
> 基于协议的报价取代了买家当前的订阅。现有发票保持不变。但是，基于协议的报价中的付款时间表取代了先前订阅的待处理发票。

未到期的协议和私人报价

在未来日期协议 (FDA) 中 AWS Marketplace，买方在预先确定的未来日期获得产品许可证或权利。在典型的 AWS Marketplace 交易中，买家将在优惠接受或协议创建后立即获得产品许可证或授权。

如果您是使用 FDA 的卖家，可以根据买家选择的时间，而不是在买家想要开始使用产品时与他们完成交易。FDA 帮助卖家在以下交易中独立执行以下操作 AWS Marketplace ：

- 根据销售需求预订（买家接受优惠）交易。
- 根据您的财务或会计需求向买家收费。
- 根据买家的需求为买家提供产品的访问权限，例如激活许可证或授权。

可以使用 FDA，为与买家之间的现有交易设置续约。

对于消费定价合同（CCP）中的软件即服务（SaaS）产品，无论是否采用灵活付款计划，都支持使用 FDA。

使用未来日期协议时，请记住以下日期：

协议签署日期

买家接受优惠并创建协议的日期。

协议开始日期

买家对产品的许可证或权利被激活且买家可以开始使用产品的日期。

协议结束日期

协议的结束日期。协议和买家的许可证或权利将在这一天到期。

主题

- 创建未来日期的协议
- 使用具有 future 日期协议的分期付款计划

- 接收有关未来日期协议的通知
- 使用未来日期协议转售渠道合作伙伴专属优惠

创建未来日期的协议

记录的卖家在生成具有未来开始日期的专属优惠时设置协议开始日期。买家无法更改开始日期，但可以在 AWS Marketplace中接受专属优惠前查看开始日期。

创建具有未来开始日期的专属优惠

1. 创建私募报价时，在 “合同期限” 下选择 “在将来的某个日期开始”。
2. 在服务日期部分，输入服务开始日期和服务结束日期。您在此处选择的服务开始日期将是买家接受优惠时，您的未来日期协议的开始日期。

> **Note**
>
> 要使用 FDA 进行续订，请将服务开始日期设定为您要续订的协议的结束日期。
> 卖家最长可以选择未来 3 年内的日期作为服务开始日期。

使用具有 future 日期协议的分期付款计划

使用FDA的分期付款计划，您可以设置在协议签署日期和协议结束日期之间的任何时间进行购买的付款。付款日期可以早于协议开始日期，也可以晚于协议开始日期。

记录的卖家选择专属优惠的付款日期和金额。有关设置分期付款计划的更多详细信息，请参阅the section called “ 创建付款计划”。

接收有关未来日期协议的通知

您的指定根账户会收到电子邮件通知，告知您的未来日期协议发生了以下事项：

- 优惠接受/协议创建（协议签署日期）
- 许可证或授权激活（协议开始日期）
- 提前 30 天、60 天或 90 天发出协议到期提醒
- 协议到期（协议结束日期）
- 协议发生修改或替换

> **Note**
>
> SaaS 的所有现有 Amazon Simple Notification Service（Amazon SNS）通知也适用于 FDA。对于 FDA，两个 Amazon SNS 主题都是在协议开始日期（而不是协议签署日期）启动的。有关更多信息，请参阅 the section called “SaaS 产品的亚马逊SNS通知”。

使用未来日期协议转售渠道合作伙伴专属优惠

制造商和经销商可以使用 future 过时的协议购买 AWS Marketplace 渠道合作伙伴的私人报价。

作为制造商：

- 与标准 AWS Marketplace 渠道合作伙伴私密优惠 (CPPO) 类似，制造商必须通过向 AWS Marketplace 渠道合作伙伴提供转售授权，授权渠道合作伙伴创建具有未来开始日期的CPPO。

  要了解如何创建转售授权，请执行the section called “创建销售授权作为 ISV”页面上的步骤。
- 在创建转售授权时，制造商可以选择指定允许的最晚服务开始日期。这将是渠道合作伙伴在创建相应的 AWS Marketplace AWS Marketplace 渠道合作伙伴私人优惠时可以指定的最长服务开始日期。

> **Note**
>
> 如果制造商未指定最长日期，则 AWS Marketplace 渠道合作伙伴可以指定未来不超过 3 年的任何服务日期。

作为经销商：

- 对经销商和渠道合作伙伴来说，创建未来日期的渠道合作伙伴专属优惠和普通的未来日期专属优惠的步骤是相同的，但有一点不同。经销商可以指定的协议开始日期，必须早于制造商在转售授权中指定的最晚允许服务开始日期。
- 要了解如何创建渠道合作伙伴专属优惠，请参阅the section called “渠道合作伙伴专属优惠”。

AMI基于产品

向买家配送商品的一种方式是使用 Amazon 机器图片 (AMIs)。AMI提供启动亚马逊弹性计算云 (AmazonEC2) 实例所需的信息。您AMI为产品创建自定义项，买家可以使用它来创建已安装并可以使用的产品EC2实例。

当买家使用您提供的时，他们会根据AMI您为产品创建的定价和计量选项为他们创建的实例付费。买家可以像使用其他AMIs产品一样使用您的商品 AMI AWS，包括制作新的自定义版本AMI。EC2根据产品代码AMI，从中创建的实例仍按您的AMI产品计费。

AMI基于产品的交付方式

您可以通过以下方式之一交付AMI基于商品的商品：

- 单个 AMI — 买家选择并使用AMI作为EC2实例的模板。买家可以使用亚马逊机器映像交付方式筛选器来查找这些产品。有关更多信息，请参阅 创建单AMI品。
- AWS CloudFormation 模板 — 您创建的模板允许买家将具有不同角色的多个实例作为一个单元安装一个系统。买家可以使用CloudFormation配送方式筛选器找到这些商品。有关更多信息，请参阅 AMI基于交付使用 AWS CloudFormation。

其他 资源

有关更多信息，请参阅以下主题。

AWSMarketp

- 产品定价
- 使用 AWS Marketplace Metering Service对 AMI 产品进行自定义计量

AMI基于产品

- AMI产品代码
- AMI基于理解的产品

AMI基于理解的产品

本节介绍使用AMI基于产品的关键概念。

主题

- 产品生命周期
- AMI产品代码
- 更改请求
- 产品加载表单
- 年度协议修订

产品生命周期

AMI基于的产品包括一组软件的一个或多个版本，以及有关整个产品的元数据。创建产品时，您可以配置其属性， AWS Marketplace 包括产品的名称、描述和定价。您还可以为产品确定合适的类别并添加关键词，以便您的产品出现在相关搜索中。

您还可以创建软件的第一个版本。根据您交付软件的方式，这可能是单个AMI、一组或多个，其中AMIs包含 AWS CloudFormation 模板或软件包，供买家在创建自己的软件时使用AMIs。有关更多信息，请参阅 AMI基于产品的交付方式。

对于付费产品，买家需要按已安装的实例数量付费。要根据软件跟踪的不同维度（例如产品用户数量）进行计量，请将您的产品与 AWS Marketplace 计量服务集成。有关更多信息，请参阅 使用 AWS Marketplace Metering Service对 AMI 产品进行自定义计量。

当你创建产品和软件的第一个版本时，它最初是在有限的范围内发布的，因此只有您的账户可以访问。准备就绪后，您可以将其发布到 AWS Marketplace 目录中，以允许买家订阅和购买您的产品。

您可以使用 S erver 产品页面查看您的产品列表。产品将处于以下状态之一：

- 暂存 – 未完成的产品，您仍在为其添加信息。在第一次保存并退出创建自助服务体验时，成功的变更请求会使用您提交的完整步骤中的信息创建未发布的产品。在此状态下，您可以继续向产品添加信息或通过变更请求更改已提交的详细信息。
- 受限 – 产品在提交给系统并通过系统中的所有验证后即已完成。然后，产品将发布到受限状态。此时，该产品有一个详细信息页面，只有您的账户和您列入允许列表的人才能访问该页面。您可以通过详细信息页面测试您的产品。如需更多信息和帮助，请根据需要联系AWS Marketplace 卖家运营团队。

- 公开 – 当您准备发布产品以便买家可以查看和订阅产品时，您可以使用更新可见性更改请求。这将启动 AWS Marketplace 卖家运营团队根据我们的政策审查和审核您的商品的工作流程。产品获得批准并处理变更请求后，产品将从受限状态更改为公开。
- 受限 – 果您想阻止新用户订阅您的产品，则可以使用更新可见性更改请求来限制该产品。受限状态表示现有用户可以继续使用该产品。但是，该产品将不再对公众可见，也不会再向新用户提供。

发布第一个版本后，AMI基于产品的生命周期 AWS Marketplace 不会结束。您应该使用软件的新版本和基本操作系统的安全补丁来更新您的产品。

举一个AMI基于完整的产品生命周期的示例，想象一下卖家想要销售他们的AMI产品 AWS Marketplace。以下是卖家如何随着时间的推移创建和维护产品：

1. 创建产品-卖家创建产品，并将版本 1.0.0 发布到。 AWS Marketplace买家可以创建版本 1.0.0 的实例并使用它。
2. 添加新版本 – 之后，卖家向产品添加一项新特征，并添加包含该特征的新版本 1.1.0。买家仍然可以使用原始版本 1.0.0，也可以选择新版本 1.1.0。

> **Note**
>
> 与新产品不同，新版本已全面公开发布。只有当整个产品处于限量发行状态时，您才能在买家看不到 AWS Marketplace 的情况下对其进行测试。

3. 更新产品信息 – 在版本 1.1.0 可用时，卖家通过使用描述该特征的新亮点文本更新产品信息，让买家了解新特征。
4. 添加次要版本 – 当卖家修复版本 1.1.0 中的错误时，他们会通过添加新版本 1.1.1 来发布修复。买家现在可以选择使用 1.0.0、1.1.0 或 1.1.1 版本。
5. 限制版本 – 卖家认为错误非常严重，以至于他们不希望买家能够使用 1.1.0 版本，因此他们限制了该版本。届时新客户将不可以购买 1.1.0（他们只能选择 1.0.0 或 1.1.1），尽管现有买家仍然可以使用 1.1.0。
6. 更新版本信息 – 为了帮助这些现有买家，卖家更新了 1.1.0 的版本信息，并建议升级到版本 1.1.1。
7. 监控使用情况 — 当买家购买和使用产品时，卖家会使用监控销售、使用情况和其他指标 AWS Marketplace 卖家报告、数据源和控制面板。
8. 移除产品 — 当不再需要该产品时，卖家会将其从中移除 AWS Marketplace。

在此示例中，卖家在产品AMI中创建了三个不同的版本，但只有两个版本可供新买家使用（在移除商品之前）。

要修改版本或产品信息，请在中创建更改请求 AWS Marketplace 管理门户。

有关创建和管理AMI基于您的产品的步骤的详细说明，请参阅创建单AMI品。

AMI产品代码

当您在 AWS Marketplace中创建产品时，系统会为其分配一个唯一的产品代码。该产品代码与您的产品相关联，用于跟踪产品的使用情况。AMIs产品代码随买家使用软件而自动传播。例如，客户订阅并启动一个AMI，对其进行配置，然后生成一个新的。AMI新的AMI仍然包含原始产品代码，因此正确的使用情况跟踪和权限仍然有效。

> **Note**
>
> 产品代码 与您产品的产品 ID 不同。中的每个产品都 AWS Marketplace 被分配了一个唯一的产品 ID。产品 ID 用于在 AWS Marketplace 目录、买家账单和卖家报告中标识您的产品。产品代码将附加到根据您的AMI作为实例元数据创建的实例上。当使用AMI带有该产品代码的创建实例时，客户将收到一张显示相关产品 ID 的账单。创建产品后，在产品 AWS Marketplace 管理门户 页面中找到产品代码和产品编号。

作为卖家，您的软件可以在运行时从实例元数据中获取正在运行的亚马逊弹性计算云 (AmazonEC2) 实例的产品代码。您可以使用产品代码来增强安全性，如在产品启动时验证产品代码。在产品发布到受限状态进行测试之前，您无法API调用该产品的代码。AMI有关验证产品代码的更多信息，请参阅验证您的软件是否正在您的计算机上运行 AWS Marketplace AMI。

更改请求

要对中的产品或版本进行更改 AWS Marketplace，请通过提交变更请求 AWS Marketplace 管理门户。变更请求会添加到队列中，可能需要几分钟到几天才能解决，具体取决于请求的类型。您可以在 AWS Marketplace 管理门户中查看请求的状态。

> **Note**
>
> 除此之外 AWS Marketplace 管理门户，您还可以使用AWS Marketplace 目录创建变更请求 API。

您可以请求对AMI基于产品的更改类型包括：

- 更新向买家显示的产品信息。

- 更新向买家显示的版本信息。
- 添加产品的新版本。
- 限制版本，使新买家无法再访问该版本。
- 更新产品 AWS 区域 在中可用的信息。
- 更新产品的定价和实例类型。
- 从中移除产品 AWS Marketplace。

> **Note**
> 某些变更请求要求您使用产品加载表单来创建请求。有关更多信息，请参阅 产品加载表单。

更新更改请求

以更新开头的变更请求将加载当前的详细信息。然后，您进行更新，这些更新将覆盖现有细节。

添加或限制变更请求

添加和限制请求对专门用于在每个请求成功后预置的步骤和更新。在自助服务体验中选择保存并退出和提交操作后，请求成功。

例如，如果AMI资产在添加后配置到实例和区域，则只能对其进行限制，而不能完全移除。也就是说，现有订阅用户和用户可以继续使用该产品，直到他们的订阅或合同到期。但是，不能向处于受限状态的产品添加新的订阅用户。

产品加载表单

通常，当你创建或编辑产品时，你需要在 AWS Marketplace 管理门户 用户界面中进行所需的更改。但是，有一些操作会引导您使用产品装货表 (PLF)。

A PLF 是一个包含产品所有信息的电子表格。有几种方法可以获得PLF：

- 您可以从中PLF产品的详细信息页面下载现有产品的 AWS Marketplace 管理门户。
- 当你为需要它的操作选择菜单项PLF时，系统会提示你下载。例如，如果您选择创建新的按月计费服务器产品，则系统将提示您下载相应的PLF产品。

  如果操作是编辑现有产品，则系统会预先填充该产品的信息，因此您只需要更改要更新的详细信息即可。PLF

- 如果您需要一个新的空白PLF，则根据您要创建的商品类型PLFs，可以在 AWS Marketplace 管理门户 文件上传页面上找到指向的链接。

完成后PLF，将其上传到 AWS Marketplace 管理门户 文件上传页面。其PLF本身在 “说明” 选项卡中有更详细的说明。

年度协议修订

每小时的年度（年度）计划修改允许您和您的买家对现有计划进行以下更改：

- 在 Amazon EC2 实例类型系列之间切换
- 在 Amazon EC2 实例类型大小之间切换
- 添加新的实例类型
- 增加协议中现有实例类型的数量

只要按比例分配的变更费用大于零（他们不能降低订阅价值），买家就可以进行更改。新增的 Amazon EC2 实例的按比例分摊费用基于根据协议剩余期限调整的实例类型的年度成本。切换实例类型时，将从新添加的 Amazon EC2 实例类型的按比例分摊成本中扣除已移除的 Ama EC2 zon 实例类型的按比例分摊成本。

无需采取其他措施即可对AMI年度产品进行修订。所有由公开报价达成的协议和不使用分期付款计划的私人报价达成的协议均支持修订。

您可以在以下仪表板上查看买家所做的修改：

- 协议和续订仪表板-修订后的协议清单。
- 已计费收入控制面板-向客户收费。
- 收款和付款控制面板-支出。

创建单AMI品

本主题介绍如何使用AMI自助服务体验为单张 Amazon 机器图片 (AMI) 创建产品清单。客户使用已安装和配置您的产品AMIs来创建 Amazon EC2 实例。

AMI自助服务体验将指导您创建产品清单和提出更改请求。通过使用自助服务体验，您可以直接更新商品清单，减少 AWS Marketplace 卖家运营团队处理所需的时间。自助服务体验的许多步骤都与中的目

录系统一致 AWS Marketplace，这便于直接验证，而不必等待 AWS Marketplace 卖家运营团队的处理和验证。

主题

- 先决条件
- 了解自助服务体验
- 创建清单
- 其他 资源

先决条件

在创建AMI商品信息之前，您必须满足以下先决条件：

1. 拥有 AWS Marketplace 管理门户的访问权限。这是您用来注册为卖家并管理您销售的商品的工具 AWS Marketplace。要了解有关获取访问权限的更多信息 AWS Marketplace 管理门户，请参阅AWS Marketplace 卖家策略和权限。
2. 注册为卖家，如果您想对产品收费，请提交您的税务和银行信息。要了解有关成为卖家的更多信息，请参阅以卖方身份开始使用。
3. 有一款您想销售的产品。对于AMI基于产品的产品，这通常意味着您已经创建或修改了服务器软件，并创建了AMI供客户使用的软件。要了解有关准备在中使用的AMI更多信息 AWS Marketplace，请参阅建筑最佳实践 AMIs。

了解自助服务体验

自助服务体验将引导您完成产品创建过程 AWS Marketplace。在继续执行这些步骤时，您需要指定产品信息和AMI部署设置 AWS 区域，例如实例类型和AMI详细信息。您还可以配置交易详情，包括定价、国家/地区可用性和退款政策。EULA作为一种选择，您可以指定允许名单， AWS 账户 IDs以便在产品处于受限状态时访问和测试该产品。

在开始之前，请查看自助服务体验的以下关键方面：

- 只有在完成当前步骤中的必填字段后，才能进入下一步。此要求是因为每个步骤的末尾都要进行页面级验证。您无法保存或提交未完成的步骤。
- 如果您需要在完成流程中的所有步骤之前结束会话，则可以选择保存并退出，将已完成的步骤提交到暂存区域。
- 未完成且未通过验证的步骤不会提交给系统。部分完成的步骤无效且无法保存。

- 选择保存并退出时，保存并退出对话框将显示通过验证检查的步骤。您可以查看并选择保存上次完成和已验证的步骤。如果存在验证错误或缺少详细信息，则可以选择修复它以返回该步骤。
- 保存并退出后，请求进入处理和审核阶段。完成处理可能需要几分钟或几小时。在请求成功之前，您无法继续执行这些步骤或进行更改。对于第一个保存并退出，请求将与您已完成的步骤并行创建产品。
  - 请求成功后，您就完成了保存。要在产品概览页面上恢复更改，请选择恢复产品创建，或者使用请求更改来更新您之前在上次会话中提交的详细信息。恢复时，请注意，您已完成的步骤标有绿色的成功标签。要更新之前提交的步骤，请使用请求更改。必须先完成之前的保存并退出请求，然后才能继续此步骤。
- 完成所有步骤后，您可以选择下一步进行查看。选择提交，请求系统执行最终验证。在您收到成功响应后，产品将变为受限状态。您可以在详细信息页面上看到，该产品现在可供允许列表上的所有人使用。如果请求失败，产品将保持暂存状态，需要您在重新提交之前进行更正。

创建清单

本节中的步骤说明了如何为单一AMI商品创建清单。

> **Note**
>
> 只有在完成当前步骤中的必填字段时，才能进入下一步。您无法保存或提交未完成的步骤。如果您需要在完成流程中的所有步骤之前结束会话，请选择保存并退出，将已完成的步骤提交到暂存区域。有关更多信息，请参阅 了解自助服务体验。

要创建单一AMI商品

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 从产品菜单中选择服务器。或者，您可以直接转到服务器产品页面。
3. 从 “服务器产品” 选项卡中，选择 “创建服务器产品”，选择 “Amazon 系统映像” (AMI)，然后为单一AMI产品选择一种许可类型：
   - 自带许可证 (BYOL) — 用户从您之外获得许可证的产品 AWS Marketplace。它可以是付费许可证，也可以是免费许可证。
   - 免费 – 可供订阅用户免费使用的产品。（他们仍将为任何关联的亚马逊弹性计算云 (AmazonEC2) 实例或其他 AWS 资源支付费用。）

- 按小时或按小时按年支付 — 买方按小时或按年合同按小时支付的产品。 AWS 根据上的产品代码进行计量AMI。
- 按月付费 – AWS按月向买家收费产品。
- 付费用量 – 根据您提供的值以及以下 4 个用量类别之一直接针对软件收费：用户、数据、带宽或主机。您可以为产品最多定义 24 个维度。所有费用仍是客户产生的。
- AMI使用合同定价 — 单一AMI产品或单一产品，其 AWS CloudFormation 堆栈由买方支付预付费用。AMI

4. 自助服务体验将引导您完成创建 AWS Marketplace 列表的步骤。您必须输入产品信息（元数据）、产品部署详细信息（AWS 区域、实例和AMI详细信息）和公开报价详情（价格、EULA、按国家/地区划分的可用性EULA、退款）。作为一种选择，您可以将账户添加到允许列表中以测试产品。完成每个步骤，进入该流程的下一步。

> **Note**
>
> 如果您需要在完成流程中的所有步骤之前结束会话，则可以选择保存并退出功能将已完成的步骤提交到暂存区域。这会创建对您提供的信息进行验证的请求。在处理请求期间，您无法编辑产品。请求成功后，您可以通过选择恢复产品创建来继续创建产品。
> 请求失败意味着由于验证错误而未对产品进行更新。这将显示在您的产品的请求日志中。您可以选择查看错误的请求，使用操作下的复制到新的更正错误，然后重新提交请求。恢复这些步骤后，可以从上次会话中保存的步骤之后的步骤继续操作。要更新之前的步骤，请前往产品概述页面并提交更改请求以更新您之前提交的步骤。

5. 输入所有更改请求步骤所需的信息后，选择提交。如果验证通过，此提交会向 AWS Marketplace 目录系统发出请求，要求其验证信息并将产品发布为受限状态。在处理请求期间，您无法继续继续编辑产品。请求成功后，产品将移至受限状态。

   - 当您的商品最初发布时，只有您 AWS 账户 （您用来创建商品的账户）和 AWS Marketplace 卖家运营团队的测试账户才能访问该商品。如果您从服务器产品页面查看商品，则可以选择查看开启 AWS Marketplace以查看商品详情，因为它们将显示在 AWS Marketplace 买家面前。其他 AWS Marketplace 用户看不到此详情清单。
   - 此功能允许您在公开发布产品之前对其进行测试（甚至发布多个版本进行测试）。

6. 在 “受限” 状态下测试您的产品，并确保其符合 AWS Marketplace AMI基于产品的要求和产品清单。然后，要请求将您的产品发布为公开状态，请选择更新可见性。 AWS Marketplace 卖家运营团队必须先审核您的商品，然后才能批准其上市。

> **Note**
>
> 商品验证和发布是一个手动流程，由 AWS Marketplace 卖家运营团队处理。如果没有错误，则可能需要 7-10 个工作日才能发布您的初始产品版本。有关时限的更多详细信息，请参阅时限和预期。

有关准备和提交单件AMI商品和商品信息的更多信息，请参阅其他 资源。

其他 资源

有关准备产品信息并提交产品信息以供发布的更多信息，请参阅以下资源：

- 准备您的产品
- 提交要发布的产品

有关准备提交给的单件AMI商品的更多信息 AWS Marketplace，请参阅以下资源：

- 建筑最佳实践 AMIs
- AMI 产品核对清单
- 基于 AMI 的产品要求

管理单一AMI产品

本主题中的章节现在解释了如何管理单一AMI商品。

主题

- 创建更改请求
- 获取更改请求的状态
- 更新产品信息
- 更新允许列表（预览账户）
- 更新产品可见性
- 添加一个 AWS 区域
- 限制 AWS 区域

- 更新对未来 AWS 区域的支持
- 添加实例
- 限制实例
- 更新版本信息
- 添加新版本
- 限制版本
- 更新定价
- 按国家/地区更新可用性
- 更新你的 EULA
- 更新退款政策
- 允许 AWS Marketplace 访问您的 AMI
- 从 AWS Marketplace中删除产品
- 解决提交更改请求时的常见错误

创建更改请求

本节提供了可用于为中的单一AMI产品创建变更请求的程序 AWS Marketplace，包括使用自助服务体验的选项。对于以下情况，您可以创建更改请求：

- 在使用自助服务体验创建单一AMI产品清单时，您保存了正在进行的步骤，但没有完成整个流程。要完成其余步骤，您可以创建更改请求。
- 您想修改处于受限状态或公开状态的产品的产品信息。要更新信息，您可以创建更改请求。有关您可以请求对AMI基于产品的更改类型的更多信息，请参阅创建更改请求。

> **Note**
>
> 除此之外 AWS Marketplace 管理门户，您还可以使用AWS Marketplace 目录创建变更请求 API。

主题

- 创建更改请求
- 使用自助服务创建更改请求

- 其他 资源

创建更改请求

> **Important**
>
> 2023年6月15日， AWS Marketplace 我将停止以下程序。2023 年 6 月 15 日之后，将使用 the section called “使用自助服务创建更改请求” 步骤。

要修改版本或产品信息，请在 AWS Marketplace 管理门户中创建更改请求。

要创建更改请求，请执行以下步骤：

1. 打开网 AWS Marketplace 管理门户 https://aws.amazon.com/marketplace/management/tour/址并登录您的卖家账户，然后前往服务器产品页面。
2. 在服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中选择一个选项。

   对于大多数更改请求，您只需在用户界面中填写表单并提交即可。但是，对于某些更改，您必须下载、填写并上传商品装货表 (PLF)。这是一个电子表格，其中包含一个表格，供您填写所需信息。当您选择其中一个变更请求时，系统会提示您下载与您尝试创建PLF的请求相对应的正确变更请求。PLF已预先填充了您现有商品详情中的信息。您可以将已完成的内容上传PLF到 AWS Marketplace 管理门户 文件上传页面。

   > **Note**
   >
   > 我们强烈建议您下载并使用最新的PLF。该表单会定期更新新信息，包括实例类型 AWS 区域 和可用信息。您可以从服务器产品页面找到产品的最新PLF信息，方法是选择产品，然后选择下载产品加载表。

有关更改请求的状态的信息，请参阅the section called “获取更改请求的状态”。要深入了解更改请求的潜在问题，请参阅解决提交更改请求时的常见错误。

使用自助服务创建更改请求

要修改版本或产品信息，请在 AWS Marketplace 管理门户中创建更改请求。更改请求是您用来对产品进行更改的自助列表的组成部分。每次选择步骤的保存并退出或选择提交进行任何更新时，都是在提出更改请求。您可以在 “请求” 选项卡上找到您的 AWS Marketplace 管理门户 请求。

要使用自助服务创建更改请求，请执行以下操作：

1. 打开网 AWS Marketplace 管理门户 https://aws.amazon.com/marketplace/management/tour/址并登录您的卖家账户，然后前往服务器产品页面。
2. 在服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中选择一个选项。
4. 在您提出更改请求后，系统需要等待一段时间才能处理您的请求，请求将处于正在审核状态。请求完成后，会成功或失败。
   - 提交请求后，它会通过以下状态开始处理：正在审核、准备更改和正在应用更改。
   - 成功表示请求的更改已处理完毕，并且更改已反映在系统中。
   - 失败表示请求出现问题，因此更改未得到处理。如果状态为失败，您可以选择请求以查找错误代码，这些错误代码提供了有关如何更正错误的建议。此时，您可以对错误进行故障排除并创建新的更改请求。为了加快处理速度，您可以选择复制到新请求来复制失败请求的详细信息。然后，您可以进行调整并重新提交请求。

其他 资源

有关针对特定更新类型的更改请求的更多详细信息，请参阅以下资源：

- 更新产品信息
- 更新版本信息
- 添加新版本
- 限制版本

获取更改请求的状态

> **Important**
>
> 2023年6月15日， AWS Marketplace 我将停止以下程序。自助服务体验不再需要此步骤。

提交更改请求后，您可以从 AWS Marketplace 管理门户中服务器产品页面的请求选项卡中查看请求的状态。该状态可以是以下任一值：

- 正在审核表示您的请求正在审核中。有些请求需要 AWS Marketplace 团队进行人工审查，但大多数请求都是在系统中自动审核的。
- 成功表示您的请求已完成。您的产品或版本已按您的要求更新。
- 需要执行的操作标识您需要更新请求以修复问题或回答有关该请求的问题。选择请求以查看详细信息，包括所有问题。
- 失败表示请求出错，您应该使用相同的数据创建新的更改请求。

更新产品信息

创建产品后，您可能需要在 AWS Marketplace中更改与其关联的一些信息。例如，如果新版本修改了产品的描述或要点，则您可以使用新数据编辑产品信息。

更新产品信息

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新产品信息。
4. 更新以下任何您需要更改的字段：
    - 产品标题
    - SKU
    - 简短描述
    - 详细描述
    - 产品徽标图片 URL
    - 要点
    - 产品类别
    - 关键词
    - 产品视频 URL
    - 资源
    - 支持信息

> **Note**
>
> 有关徽标格式的详细信息，请参阅公司和产品徽标要求。

5. 选择提交。
6. 验证请求是否显示在请求选项卡上，状态为正在审核。您可能需要刷新页面才能在列表中看到请求。

您可以随时从服务器产品页面的请求选项卡中查看请求的状态。有关更多信息，请参阅 获取更改请求的状态。

更新允许列表（预览账户）

要更改 AWS 账户 IDs可以看到您的产品处于受限状态的列表，请使用更新许可名单。

要更新允许列表，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新允许列表。当前列表中提供了当前已添加至允许列表的账户列表。
4. 添加为了提高可见性而首选的，并IDs用逗号分隔。 AWS 账户 IDs
5. 选择提交更改请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

更新产品可见性

要更改哪些买家可以查看您的商品 AWS Marketplace，您可以使用 “更新可见性”。

要更新可见性，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。

3. 从请求更改下拉列表中，选择更新可见性。

> **Note**
>
> 您可以使用此更改请求请求将产品从受限态更改为公开状态。但是，变更请求必须经过 AWS Marketplace 卖家运营团队的批准才能移至公开。

4. 选择提交更改请求，提交您的请求以供审核。
5. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

添加一个 AWS 区域

您可以添加买家可以在其中使用您的产品的区域。

要添加区域，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择添加区域。
4. 从可用区域列表中选择要添加的区域。
5. 选择提交请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

> **Note**
>
> 当您添加对新产品的支持时 AWS 区域，已订阅产品私人优惠的客户将无法自动访问新添加的区域。您必须使用您希望客户访问的地区创建另一个私有报价。接受新优惠后，客户可以访问新添加的区域。在将来的某个日期订阅您的产品的客户也可以访问该地区，前提是该地区包含在私人优惠中。有关如何创建新的私有优惠的更多信息，请参阅私有优惠升级、续订和修改。

限制 AWS 区域

为了防止新买家在特定地区使用您的商品 AWS 区域，您可以限制该区域。您可以稍后重新添加该区域。只要订阅了该地区的产品，该地区的现有订阅用户就可以继续使用该地区的产品。

要限制区域，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择限制区域。
4. 选择下拉菜单以查看您的产品当前可用的区域列表。
5. 选择要限制的区域。
6. 您选择的区域显示为令牌。查看您要限制的区域列表，对于不想限制的区域，请输入 X。
7. 选择提交更改请求，提交您的请求以供审核。
8. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

如果您的请求成功，您的现有用户将收到以下电子邮件，通知他们该区域将受到限制。只要他们保持订阅状态，他们就可以继续使用您的产品，但是如果他们取消订阅，他们就无法重新订阅。

```
Greetings from AWS Marketplace,

This message is a notification detailing a recent change for <ProductName>.
{{{sellerName}}} has opted to restrict the <ProductType> product in <Restricted
 Region(s)> beginning <DateOfChange>.

This impacts you in the following ways:

1. As long as you're subscribed to the product, you can continue using the software
 product in the restricted Region.
2. You can't begin new instances of the software product in the restricted Region.
3. You can continue using the software product in all available AWS Regions.

Regards,
The AWS Marketplace Team

Amazon Web Services, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com (http://
amazon.com/) is a registered
trademark of Amazon.com, Inc. This message was produced and distributed by Amazon Web
Services Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.
```

更新对未来 AWS 区域的支持

如果你想让你的产品加入到新发布的版本中 AWS 区域，你可以使用 Update f uture 区域支持。

要更新未来区域支持，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新未来区域支持。
4. 您可以选择激活 future 区域支持， AWS Marketplace 以允许代表您将您的产品加入到新推 AWS 区域 出的产品中。
5. 激活该特征后，您可以在所有未来区域之间进行选择，也可以仅限于美国区域。
6. 选择提交更改请求，提交您的请求以供审核。
7. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

添加实例

您可以添加一个新实例，买家可以使用单个-AMI。

要添加实例，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择添加实例。
4. 从可用实例列表中选择要添加的实例。
5. 选择提交请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

> **Note**
>
> 如果您当前的定价模式不是免费的，或者使用的是自带许可证 (BYOL) 模式，则还有一个额外的步骤来增加价格。
> 如果您创建了包含新实例价格的添加实例请求或提高价格的更新定价请求，则在进行更改之日起的 90 天内，您将无法使用自助服务添加实例。要进行这些更改，请联系 AWS Marketplace 卖家运营团队。
> 当您添加对新实例类型的支持时，已经为您的产品订阅了私有优惠的客户将无法自动启动新添加的实例。您必须使用您希望客户访问的实例创建另一个私有报价。接受新优惠后，

客户可以启动新添加的实例。在将来的某个日期订阅您的产品的客户也可以启动该实例，前提是该实例包含在私有优惠中。有关如何创建新的私有优惠的更多信息，请参阅私有优惠升级、续订和修改。

限制实例

如果您想阻止新买家在特定实例中使用您的单一AMI产品，则可以限制该实例。如果需要，您可以稍后重新添加该实例。受限实例AMI上的单一用户只要订阅了该地区的产品，他们就可以继续使用该地区的产品。

要限制实例，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择限制实例。
4. 选择要限制的实例，然后选择限制。
5. 选择提交更改请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

> **Note**
>
> 如果该复选框为阴影，则表示该实例与一到多个版本关联作为推荐实例类型。要限制此类实例，请使用更新版本选择不同的推荐实例类型。更改请求完成且您要限制的实例不再是推荐的实例类型后，您可以返回限制实例来限制您选择的实例。

更新版本信息

创建版本后，可通过修改与版本关联的信息，向买家提供更新的信息。例如，如果您计划在版本 1.1 发布后限制版本 1.0，则可以更新版本 1.0 的描述（注明该版本的受限日期），将买家引导到版本 1.1。您可以从 AWS Marketplace 管理门户中更新版本信息。

要更新版本信息，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到当前服务器产品页面，然后在服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新版本信息。
4. 在更新版本页面上，选择要更新的版本。
5. 更新以下任何需要修改的信息：
   - 发布说明
   - 使用说明
   - 64 位 (x86) Amazon 系统映像 (AMI) — 有关使用情况和安全组的详细信息
6. 选择提交。
7. 验证请求是否显示在请求选项卡上，状态为正在审核。

> **Note**
> 您不能使用此过程更新版本标题或与版本AMI关联的标题。相反，创建新版本并限制之前的版本。

您可以随时从服务器产品页面的请求选项卡中查看请求的状态。有关更多信息，请参阅 获取更改请求的状态。

添加新版本

当您对产品、基本图片进行更改或需要修改产品的任何其他时间时，您可以添加产品的新版本。AMI从中添加产品的新版本 AWS Marketplace 管理门户。

> **Note**
> 有关创建 for 的信息 AWS Marketplace，AMI请参见建筑最佳实践 AMIs。

要添加新版本，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择添加新版本。将出现添加新版本，其中预先填充了最新版本的信息。
4. 在公司信息部分中，提供以下信息：
   - 版本标题-输入有效的字符串（例如 *1.1* 或者 *Version 2.0*)。 它在整个产品中必须是唯一的。
   - 发布说明 – 输入文本以描述有关此版本的详细信息。
5. 在 “新建 Amazon 系统映像 (AMI)” 部分中，提供以下信息：
   - Amazon 机器映像 AMI ID — 输入您AMI要用于此版本的 ID。您可以从控制台的列表AMIs中找到 AMI ID。AMI必须存在于美国东部（弗吉尼亚北部）地区和您的 AWS Marketplace 卖家账户中。与此相关的快照AMI无法加密。
   - IAM访问角色 ARN — 输入允许 AWS Marketplace 访问您的 AWS Identity and Access Management (ARNIAM) 角色的 Amazon 资源名称 () AMI。有关如何创建IAM角色的说明，请参阅允许 AWS Marketplace 访问您的 AMI。使用标准格式表示 IAMARN，例如：*arn:aws:iam::123456789012:role/RoleName*。 ARN必须存在于您的 AWS Marketplace 卖家账户中。
   - 操作系统用户名-对于 Linux基于AMIs，输入可用于登录实例的用户名。我们建议使用 ec2-user。
   - 扫描端口-输入可用于登录操作系统的SSH端口号：的端口 Linux AMI或者是 a 的RDP端口 Windows AMI.
6. 如果还没有，请展开配置设置以发布AMI到 AWS Marketplace 客户网站部分，然后提供以下信息：
   - 使用说明-输入使用说明AMI或链接以获取更多有关使用的信息AMI。例如：*To get started with the product, navigate to https://example.com/usage.htm.*
   - 终端节点 URL-提供有关购买者在创建实例后如何访问软件的信息。输入协议（https 或 http）、相对协议URL（例如 */index.html*) 和端口（例如，*443*），买家可以用它来访问您的商品。（主机名取决于EC2实例，因此您只需要提供相对路径即可）。
   - 操作系统 (OS)-输入所使用的操作系统的名称AMI（例如 *Amazon Linux*).
   - 操作系统版本-在中输入操作系统的特定版本AMI。

- 推荐的实例类型 – 选择买家默认获得的实例类型。
- 安全组建议 — 输入一项或多项建议的信息，包括协议（TCP或 UDP）、允许的端口范围以及列表（以 xxx.xxx.xxx.xxx .xxx/nn 的IPv4CIDRIPs形式），例如 *192.0.2.0/24*).

7. 选择提交以提交添加新版本的请求。
8. 验证请求是否显示在请求选项卡上，状态为正在审核。如果有错误需要修复，则页面会在页面顶部的表格中显示错误，需要更新的特定字段将显示为红色。

您可以随时从服务器产品页面的请求选项卡中查看请求的状态。将对新版本进行审核，如果成功，将作为产品的新公开版本发布。如果存在问题，则状态可能为需要操作。选择请求以查看详细信息，包括所有问题。

如果您的请求成功，您的现有用户将收到以下电子邮件。该消息会通知他们新版本已上市，链接到该版本的发布说明，并建议他们升级到最新版本。作为 AWS 账户 root 用户，您还会在与您关联的电子邮件帐户中收到电子邮件的副本 AWS 账户。

```
Greetings from AWS Marketplace,

Thank you for subscribing to <product-title>

We are writing to inform you that <seller-name> has added a new version to <product-
title> on AWS Marketplace.
As an existing customer, your subscription to the product, any running instances and
 access to previous versions
are unaffected. However, <seller-name> does recommend you to update to the latest
 version, <product-title>/<version-title>
by visiting <product-detail-page-of-new-listing>.

For additional questions or upgrade information, please contact <seller-name> directly.
 Click here <link of seller page on MP>
to visit the seller’s profile page on AWS Marketplace.

Release notes for <product-title>/<version-title>:

<release-notes>

Thank you,
The AWS Marketplace Team
https://aws.amazon.com/marketplace
```

```
Amazon Web Services, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a
 registered trademark of Amazon.com, Inc.
This message was produced and distributed by Amazon Web Services Inc., 410 Terry Ave.
 North, Seattle, WA 98109-5210
```

限制版本

如果您想阻止买家访问您的公开产品的特定版本，则可以限制该版本。

> **Note**
>
> 无论限制状态如何，所有订阅用户都可以使用当前版本。 AWS Marketplace 指南要求您在限制版本后的 90 天内继续向现有买家提供支持。版本受限后，您的版本AMI将被标记为已弃用。有关更多信息，请参阅《亚马逊弹性计算云用户指南》AMI中的 “弃用” Windows 实例。

要限制版本，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择限制版本。
4. 在限制版本页面上，选择要限制的一个或多个版本。
5. 选择提交请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态为成功。

> **Note**
>
> 您不能限制产品的所有版本。如果您尝试限制产品的最后剩余公共版本，会收到错误。要完全删除产品，请参阅the section called “从 AWS Marketplace中删除产品”。

您可以随时从服务器产品页面的请求选项卡中查看请求的状态。有关更多信息，请参阅 获取更改请求的状态。

> **Note**
>
> 限制版本可能最多需要 3 天才能完成。

如果您的请求成功，您的现有用户将收到以下电子邮件，通知他们版本限制并建议他们使用最新的可用版本。作为 AWS 账户 root 用户，您还会在与您关联的电子邮件帐户中收到电子邮件的副本 AWS 账户。

```
Greetings from AWS Marketplace,

Thank you for subscribing to <product-title>.

We are writing to inform you that, as of <Version-Restriction-Date>, <Seller Name>
 will no longer offer version(s) "<version-title>" to new subscribers. Your use and
 subscription is unaffected for this version(s), however it is recommended that users
 upgrade to the latest version on AWS Marketplace.

For additional questions or upgrade information, please contact <seller-name> directly.
 Click here<link of seller page on MP> to visit the seller’s profile page on AWS
 Marketplace.

Thank you,
The AWS Marketplace Team
https://aws.amazon.com/marketplace

Amazon Web Services, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a
 registered trademark of Amazon.com, Inc. This message was produced and distributed by
 Amazon Web Services Inc., 410 Terry Ave. North, Seattle, WA 98109-5210
```

更新定价

如果您想更改单一AMI产品的每种实例类型的定价，则可以更新定价。

> **Note**
>
> 如果您创建了包含新实例价格的添加实例更改请求或提高价格（而不是降低价格）的更新定价请求，则在进行更改之日起的 90 天内，您将无法使用自助服务进行价格更新。如需其他支持，请联系 AWS Marketplace 卖家运营团队进行这些更改。

要更新产品定价，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新定价。
4. 文本框中预先填充了当前的定价信息。您可以删除当前价格，然后输入新的价格。我们建议您查看您请求的价格，以验证定价是否正确。

> **Note**
>
> 任何实例的定价提高都将导致添加实例和更新定价在接下来的 90 天内被锁定。

5. 选择提交更改请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

> **Note**
>
> 如果您创建了包含新实例价格的添加实例请求或提高价格（而不是降低价格）的更新定价请求，则在进行更改之日起的 90 天内，您将无法使用自助服务进行价格更新。如需其他支持，请联系 AWS Marketplace 卖家运营团队进行这些更改。

按国家/地区更新可用性

如果您想更改可以订阅和销售产品的国家/地区，可以使用更新可用性。

要按国家/地区更新可用性，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新可用性。
4. 请选择以下选项之一：
   1. 所有国家/地区 – 适用于所有受支持的国家/地区。
   2. 所有国家/地区（部分除外） – 在所有受支持的国家/地区均可用，部分国家/地区除外。

   3. 自定义列表 – 提供产品的国家/地区的具体列表。
5. 选择提交更改请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

更新你的 EULA

如果要更改最终用户许可协议 (EULA)，则可以使用 “更新” EULA。此操作会EULA针对订阅您的产品的新用户和产品续订进行更新。

更新EULA

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 选择服务器产品选项卡，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新最终用户许可协议。
4. 您可以选择 AWSMarketplace 的标准合同 (SCMP) 或提交自己的自定义合同EULA。对于定制EULA，您必须从 Amazon S3 存储桶URL为自定义合同提供。

   > **Note**
   >
   > 必须在 Amazon S3 存储桶上启用公开可见性。

5. 选择提交更改请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

更新退款政策

如果您想更改产品的退款政策，可以使用更新退款政策。

要更新退款政策，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，然后在当前服务器产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新退款政策。
4. 文本框中提供了当前退款政策的详细信息供您编辑。提交请求会改写当前的退款政策。

5. 选择提交更改请求，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

允许 AWS Marketplace 访问您的 AMI

当您创建包括向添加新AMI请求的请求时 AWS Marketplace，AMI必须将其复制到 AWS Marketplace 系统中，然后扫描是否存在安全问题。您必须AMI通过创建具有对您执行操作权限的 AWS Identity and Access Management (IAM) 角色AMI和允许代 AWS Marketplace 入该角色的信任策略 AWS Marketplace 来授予访问权限。您只需要创建一次IAM角色。

为 AWS Marketplace AMI资产摄取创建角色

1. 登录 AWS Management Console，打开IAM控制台，然后转到角色页面。
2. 选择创建角色。
3. 在创建角色页面上，进行以下选择：
   - 选择可信实体的类型-选择AWS服务。
   - 选择一个用例 — 选择 AWSMarketplace。
   - 选择您的用例 — 选择 Marketplace — Ass AMI ets Ingestion。
   - 选择下一步：权限，以转到下一个页面。
4. 选择 AWSMarketplaceAmiIngestion政策。根据需要添加权限边界，然后选择下一步：标签继续。

   > **Note**
   > 您可以使用权限边界来限制您通过此角色授予 AWS Marketplace 的访问权限。有关更多信息，请参阅《AWS Identity and Access Management 用户指南》中的IAM实体的权限边界。

5. 要继续，请选择下一步：查看。
6. 提供角色的名称，然后选择创建角色。
7. 你应该看到 “角色 *rolename* 已创建” 位于页面顶部，该角色应出现在角色列表中。

在此页面上，当你选择刚刚创建的角色时，你可以看到它的形式为 arn: aws:: iam:: 12 ARN 3456789012: role/。exampleRole在ARN创建更改请求ARN时（例如，向产品添加新版本时）使用IAM访问角色。

从 AWS Marketplace中删除产品

产品发布后，您可以将其从中删除（也称为日落） AWS Marketplace。要删除产品，请标识该产品，并提交删除请求，在其中说明删除原因和您的联系电子邮件地址。如果您要将当前产品替换为新产品，也可以提供一个替换产品 ID。请求删除产品后，新客户将无法再订阅该产品。您需要为任何现有客户提供至少 90 天的支持。我们会根据以下条件 AWS Marketplace 处理下架商品的请求：

- 该产品已从 AWS Marketplace 搜索、浏览和其他发现工具中删除。任何订阅按钮或功能都被禁用，并在页面上发布消息，明确指出产品不再可用。仍可使用访问商品详情页面，URL并可能在公共搜索引擎中编制索引。
- 必须指定删除原因（例如，终止支持、终止产品更新或替换产品）。有关继续支持已移除商品的要求，请参阅 AWSMarketplace 卖家条款和条件。
- AWS Marketplace 通过电子邮件联系当前买家，告知他们商品下架、移除原因，并提供卖家联系信息。
- 当前已订阅的买家仍可 继续正常使用软件，除非他们取消订阅。产品删除不会对他们产生任何影响。

要移除使用创建的产品 AWS Marketplace 管理门户

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 选择产品选项卡，然后选择服务器。
3. 在产品页面上的服务器产品下，找到您想要删除的产品。从 “请求更改” 下拉列表中，选择 “更新商品曝光度”。
4. 在更新产品知名度页面上，选择受限。
5. （可选）如果有其他产品可以替代您要删除的产品，请提供替换产品 ID。
6. 检查信息的准确性，然后选择 “提交”。

提交产品删除请求后，将显示下一步信息页面。 AWS Marketplace 卖家运营部门会审查并处理您的请求。通过查看请求来查看您的提交状态。

删除您的产品后，产品会显示在 AWS Marketplace 管理门户中的当前产品列表中。在当前产品中，您唯一可执行的操作是下载产品的电子表格。您无法编辑或提交另一个下架请求。

如果您对产品删除有任何疑问，请联系 AWS Marketplace 卖家运营团队。

解决提交更改请求时的常见错误

当您更改产品信息时，可能会遇到错误。以下是一些常见问题以及如何解决这些问题的建议：

- 扫描你的 AMI — 扫描你的：时可能会出现几个问题AMI：
  - 您尚未授予扫描您的 AWS Marketplace 权限AMI。授予 AWS Marketplace 访问权限。或者您已授予权限，但权限边界过于严格。有关更多信息，请参阅 允许 AWS Marketplace 访问您的 AMI。
  - 如果扫描发现您的镜像中存在安全问题或常见漏洞和漏洞 (CVEs)AMI，请确保使用的是针对映像中操作系统的最新补丁。有关更多信息，请参阅 基于 AMI 的产品要求。

  有关构建的一般指南AMI，请参阅建筑最佳实践 AMIs。
- AWS Marketplace 管理门户 字段-中的某些字段 AWS Marketplace 管理门户 需要非常具体的信息：
  - 如果您不确定该字段在请求什么，请尝试在控制台中查看详细信息。大多数字段的字段上方都有文字描述，字段下方有格式要求。
  - 如果您尝试提交包含一个或多个无效字段的表单，则会显示问题列表。提供了建议的操作来帮助您解决问题。
  - 如果要求你提供ARN，你通常会在控制台的其他地方找到它。例如，在ARNIAM控制台的 “IAM角色” 页面上可以找到您AMI为授予 AWS Marketplace 访问权限而创建的角色的。ARNs都具有相似的格式。例如，IAM角色的形式为 arn: aws: ARN iam:: 123456789012: role/。exampleRole
  - 您的徽标和视频必须URL直接提供给内容。有关徽标格式的更多信息，请参阅公司和产品徽标要求。

  有关提交产品和版本更改请求的更多信息，请参阅提交要发布的产品。
- 商品加载表 (PLF) 问题-PLFs 包含电子表格中包含的说明。说明表中提供了总体说明。每个字段都提供关于如何填写的说明 – 选择该字段以显示说明。
- 请求正在进行中 – 有些请求不能并行发生。您一次只能提一个请求来更新某一产品的特定信息。您可以在 AWS Marketplace 管理门户中 服务器产品页面的请求选项卡上查看所有正在审核中的请求。如果您有未处理的请求，但您不打算这样做，则可以将其取消，然后提交包含您想要更改的新请求。
  - 当版本更新（添加或限制）正在进行时，您无法更新版本信息。
  - 如果 AWS Marketplace 卖家运营团队有待处理的请求，则无法提交任何新的更改。
- 无法解释的错误 – 如果您的提交失败且没有任何解释，请重试。有时，服务器负载会导致提交失败。

如果您仍然遇到更改请求的问题，请联系 AWS Marketplace 卖家运营团队。

AMI基于交付使用 AWS CloudFormation

AWS Marketplace 卖家可以使用AMI AWS CloudFormation 模板列出配送给 AWS Marketplace 买家的商品。您可以使用模板为产品定义集群或分布式架构，或者选择不同的AMI组合或产品配置。可以将 CloudFormation 模板配置为提供单个或多个 Amazon 系统映像 (AMI) AMIs 以及相关的配置文件和 Lambda 函数。买家可以浏览所选解决方案 AWS Marketplace，一键购买，然后使用您提供的CloudFormation模板进行部署。

单个AMI解决方案最多可以包含三个 CloudFormation 模板。

多AMI解决方案最多可以包含 20 个模板AMIs和最多 20 个 CloudFormation 模板。每个CloudFormation 模板都可以引用解决方案中AMIs包含的任意组合或子集。买方购买单一解决方案，使他们有权获得该产品AMIs中的所有内容。当产品有多个产品时AMIs，每个产品AMI都有自己唯一的产品代码，可以单独定价和计量。不过，在产品上下文之外，您无法搜索或购买解决方案的单独组件。

如果您已有单AMI件商品，则无法将其迁移或合并到新的多件AMI商品中。但是，您的新解决方案可以采用与现有产品相同的软件或副AMIs本。上创建的每个清单 AWS Marketplace 都是带有新商品编码的清单。

您还可以在无服务器应用程序中包含 Lambda 函数，AMI以便买家可以通过部署它们。CloudFormation有关如何在中包含 Lambda 函数和无服务器应用程序的说明AMI，请参阅本指南添加无服务器应用程序组件中的。

生成产品列表

要提交您的产品，您需要准备和验证您的产品AMIs，创建 AWS CloudFormation 模板，创建架构图，填写产品加载表，然后将材料提交给 AWS Marketplace。我们建议您首先创建和验证您的AMIs模板，然后完成并验证这些 CloudFormation 模板。完成这些步骤后，您应创建架构示意图并估算软件和基础设施价格。 AWS Marketplace 将验证您提交的材料并与您合作，以公开发布您的产品。使用 AWS Pricing Calculator 可帮助估算您模板的基础设施成本。提供指 AWS Marketplace 向您保存的计算器配置的链接。以下是多AMI解决方案产品的局限性：

- 不支持将现有 AWS Marketplace 产品从独立AMI产品更新为多产品。要在多产品中提供AMI产品，请复制，AMI然后将其作为组件提交到新的多AMI产品中。由AMI此产生的唯一产品代码与先前产品的代码不同。
- 在亚马逊弹性计算云 (AmazonEC2) 控制台的 “启动实例” 页面的AWS Marketplace选项卡上看不到多AMI解决方案。
- 不得在多AMI解决方案中列出的 CloudFormation 模板AMIs之外启动模板。

- CloudFormation 模板必须以公众的形式提交URL。模板中URLs包含的所有嵌套模板也必须可供公众访问。

准备 CloudFormation 模板

要构建 CloudFormation 模板，必须满足模板的先决条件并提供所需的输入和安全参数。提交 CloudFormation 模板时，请使用以下各节中的指南。

模板先决条件

- 验证模板是否已通过 AWS CloudFormation 控制台成功启动， AWS 区域 且已为您的产品启用所有功能。您可以使用该TaskCat 工具来测试您的模板。
- 如果您要创建单一AMI商品，则模板必须仅包含一个商品AMI。
- AMIs必须位于每个区域的映射表中。 AWS Marketplace 团队会在他们被克隆AMIIDs后对其进行更新。您的来源AMI必须位于`us-east-1`，其他地区可以使用占位符。请参阅以下 YAML 示例。

```
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
```

- AMIs您的 CloudFormation 模板中必须包含您要发布AMI的商品或AWS托管产品，AMI例如最新的 Amazon Linux 2。请勿包含任何由您AMI或任何其他第三方AMI拥有和共享的社区。要使用AWS-managedAMI，请使用 S AWSystems Manager 参数存储中的公共参数而不是硬AMIIDs编码。例如，在您指定 AMI ID 的 CloudFormation 模板中，您可以使用动态引用`ImageId: '{{resolve:ssm:/aws/service/ecs/optimized-ami/amazon-linux-2/recommended/image_id}}'`。
- 生成模板，使其不必依赖于使用特定可用区 (AZ)。并非所有客户都能访问所有账户AZs，AZs而且不同账户的映射方式也不同。
- 您可以随身携带 Lambda 函数、配置文件和脚本等依赖项。AMI有关更多信息，请参阅 创建无服务器应用程序。

- 如果您在生成使用自动扩缩组的集群解决方案，我们建议您考虑到扩展事件。新节点应自动加入正在运行的集群。
- 对于单节点产品，我们建议使用自动扩缩组。
- 在解决方案涉及含多个实例的集群时，如果您希望在集群之间降低网络延迟时和/或提升网络吞吐量，请考虑使用置放群组。
- 如果您的解决方案涉及 Docker 容器，则必须将 Docker 镜像合并到。AMI
- 为了便于 AWS Marketplace 团队审核并向买家透明，我们建议您在UserData栏目中添加评论。

模板输入参数

- 模板的输入参数不得包含 AWS Marketplace 客户的 AWS 凭证（例如密码、公钥、私钥或证书）。
- 对于密码等敏感输入参数，请选择 `NoEcho` 属性并启用更严格的正则表达式。对于其他输入参数，设置最常见的输入以及相应的帮助文本。
- 如果可用，请使用 AWS CloudFormation 参数类型进行输入。
- 使用 `AWS::CloudFormation::Interface` 来分组和排序输入参数。
- 请勿为以下输入参数设置任何默认值：

> **Note**
>
> 客户必须提供这些内容作为输入参数。

  - 允许从公共互联网进入远程访问端口的默认CIDR范围
  - 允许从公共互联网进入数据库连接端口的默认CIDR范围
  - 用户或数据库的默认密码

网络和安全参数

- 确保默认SSH端口 (22) 或端RDP口 (3389) 未向 0.0.0.0 开放。
- 我们建议您使用适当的访问控制列表 (VPC) 和安全组来构建，而不是使用默认的虚拟私VPC有云 (ACLs)。
- 通过使用 AWS Identity and Access Management (IAM) 角色AssumeRole从中调用来启用对客户 AWS 环境的访问权限 AWS Security Token Service。

- 将IAM角色和策略设置为授予最低权限，并仅在绝对必要时才启用写入权限。例如，如果您的应用程序只需要 `S3:GET`、`PUT` 和 `DELETE` 操作，则仅指定这些操作。在这种情况下，我们不建议使用 `S3:*`。

收到您的模板后，将 AWS Marketplace 验证产品配置和信息，并就所有必需的修订提供反馈。

获取模板基础设施的成本估算

向客户显示的各个模板的基础设施成本估算基于您使用 AWS Pricing Calculator 提供的估算值。该估算应包括将作为模板的一部分部署的服务列表，以及典型部署的默认值。

计算模板的每月估计费用后，请 AWS Marketplace 提供美国东部（弗吉尼亚北部）区域的 “保存并共享” 链接。这是提交过程的一部分。

架构示意图

您必须为每个模板提供架构示意图。要了解有关绘制图表的更多信息，请参阅什么是架构绘图？

示意图必须符合以下标准：

- 在上演示标准部署 AWS。
- 从逻辑上描述资源的部署位置。例如，像 Amazon EC2 实例这样的资源位于正确的子网中。
- 对通过 AWS CloudFormation 模板 AWS 服务 部署的每个 AWS 产品使用最新的产品图标。要下载当前的架构图标集，请参阅AWS 架构图标。
- 包括 AWS CloudFormation 模板部署的所有服务的元数据。
- 包括 AWS CloudFormation 模板部署的所有网络和子网。VPCs
- 显示集成点，包括第三方资产APIs和本地混合资产。
- 图表的大小必须为 1100 x 700 像素。保持原始图表比例，无需拉伸或裁剪。

满足提交要求

要提交使用 AWS CloudFormation 模板配送的商品，您必须提供以下资源：

- CloudFormation 一个或多个模板
  - 单个AMI产品可以有一到三个 CloudFormation 模板
  - 一个多AMI产品最多可以有 20 个 CloudFormation 模板
- 各个模板的默认配置的估算基础设施价格

- 架构示意图和架构元数据
- 已完成产品表单（在 AWS Marketplace 管理门户 中提供）
  - 对于单一AMI产品，请使用商业产品表格
  - 对于多AMI产品，请使用多AMI产品表单

产品表包含提交示例供您参考。

对于每种产品，所需的大部分产品数据和元数据都与传统的单一AMI产品相同。因此，使用CloudFormation模板交付的每AMI件商品都必须继续符合所述的标准和要求 AWS Marketplace。

对于每个 CloudFormation 模板，您还必须提供以下信息。

字段	描述	限制
Title	架构的标题。此项显示在详情页面和交付页面上，以及显示架构详细信息的弹出窗口中。	50 个字符
简短描述	此项显示在详情页面和产品交付页面上。	200 个字符
详细描述	此项显示在架构详细信息弹出窗口中。	2000 个字符

对于多AMI件商品，必须填写以下字段：

- 解决方案标题
- 解决方案简短描述
- 解决方案详细描述
- 对于 CloudFormation 模板（每个解决方案最多 20 个）
  - 部署标题（每个模板）
  - 简短描述（每个模板）
  - 详细描述（每个模板）
  - 架构图（每个模板）
  - 基础设施定价估算（每个模板）

- 此 CloudFormation 模板中包含的产品/组件清单
- 此 CloudFormation 模板支持的区域列表

提交您的产品请求

使用 AWS Marketplace 管理门户 提交您的产品。在资产选项卡上，选择文件上传。上传要提交的任何文件并输入简要描述。同时支持YAML和JSON格式。请求处理时间为 3 到 5 个星期，包括：

- 查看 CloudFormation 模板以及AMI和 CloudFormation模板的AMI元数据
- 将您的 CloudFormation 模板发布到 AWS Marketplace 产品

添加无服务器应用程序组件

您可以创建包含一个或多个 Amazon 系统映像 (AMIs) 的产品，这些映像使用一个或多个 AWS CloudFormation 模板交付，产品中包含无服务器组件。例如，创建一个产品，其中一个AMI配置为控制器服务器，另一个AMI配置为工作服务器，作为 AWS CloudFormation 堆栈交付。用于创建堆栈的 AWS CloudFormation 模板可以包括用于设置由其中一台服务器中的事件触发的 AWS Lambda 函数的定义。

使用这种方法来设计产品时，可以简化体系结构并使买家更容易启动产品。此方法还可以让您更轻松地更新您的产品。

有关AMIs为您的产品创建的信息，请参阅AMI基于产品。有关完成商品 AWS CloudFormation 模板的信息，请参阅AMI基于交付使用 AWS CloudFormation。

定义无服务器应用程序时，使用存储在中的 AWS Serverless Application Model AWS Serverless Application Repository(AWS SAM) 模板。 AWS SAM 是一个用于构建无服务器应用程序的开源框架。在部署期间， AWS SAM 将语法转换为语法并将其扩展为 AWS Serverless Application Model AWS CloudFormation 语法。 AWS Serverless Application Repository 是无服务器应用程序的托管存储库。它使您可以存储和共享可重用的应用程序，以便买家可以组装和部署无服务器架构。要创建和提供这种类型的产品，请完成以下步骤：

步骤

- 创建无服务器应用程序
- 将应用程序发布到存储库
- 创建 CloudFormation模板
- 提交您的 CloudFormation 模板和配置文件

- 更新您的 AWS Serverless Application Repository 应用程序权限
- 分享你的 AMI
- 使用AMI无服务器应用程序提交您的 CloudFormation 产品

AWS Marketplace 在创建商品信息之前，请对您的商品进行审核和验证。如果在列出产品之前必须解决一些问题，我们将向您发送电子邮件。

作为完成订阅的一部分，我们会将无服务器应用程序和模板复制到每个应用程序和 AWS CloudFormation 模板的 AWS Marketplace自有存储库中。AMIs AWS 区域当买家订阅您的产品时，我们将向买家提供访问权限，还会在您更新软件时通知他们。

创建无服务器应用程序

第一步是打包用于创建无服务器应用程序的 AWS Lambda 函数。您的应用程序是 Lambda 函数、事件源以及共同执行任务的其他资源的组合。无服务器应用程序可以像一个 Lambda 函数一样简单，也可以包含具有其他资源（APIs例如数据库和事件源映射）的多个函数。

使用为您的无服务器应用程序定义模型。 AWS SAM 有关属性名称和类型的描述，请参阅上的:: ServerlessAWS:: Appless:: Application in Lab s。 AWS GitHub以下是具有单个 Lambda 函数和 AWS Identity and Access Management (IAM) 角色的 AWS SAM 模板示例。

```
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"
```

将应用程序发布到存储库

要发布应用程序，首先需要上传应用程序代码。将您的代码构件（例如，Lambda 函数、脚本、配置文件）存储在您的账户拥有的 Amazon S3 存储桶中。当你上传应用程序时，它最初设置为私有，这意味着只有创建应用程序的人才能使用它。 AWS 账户 您必须创建一个IAM策略，授予访问您上传的构件的 AWS Serverless Application Repository 权限。

将无服务器应用程序发布到无服务器应用程序存储库

1. 打开 Amazon S3 控制台，网址为https://console.aws.amazon.com/s3/。
2. 选择用于打包您的应用程序的 Amazon S3 存储桶。
3. 选择权限选项卡。
4. 选择存储桶策略。
5. 复制粘贴下面的示例策略语句。

> **Note**
>
> 在以下步骤中更新 `aws:SourceAccount` 和 `Resource` 的值之前，示例策略语句将产生错误。

```
{
    "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. 将`Resource`属性值中的 amzn-s3-demo-bucket 替换为存储桶的存储桶名称。

b. Replace（替换）*123456789012* 在带有您的 AWS 账户 ID 的`Condition`元素中。该`Condition`元素可确保 AWS Serverless Application Repository 只有用户有权从指定的应用程序访问应用程序 AWS 账户。

6. 选择保存。
7. 打开 AWS Serverless Application Repository 控制台，网址为https://console.aws.amazon.com/serverlessrepo。
8. 在我的应用程序页面上，选择发布应用程序。
9. 根据需要，填写必填字段和任何可选字段。必填字段为：
   - 应用程序名称
   - 作者
   - 描述
   - 源代码 URL
   - SAM模板
10. 选择发布应用程序。

发布应用程序的后续版本

1. 打开 AWS Serverless Application Repository 控制台，网址为https://console.aws.amazon.com/serverlessrepo。
2. 在导航窗格中，从我的应用程序中选择应用程序。
3. 选择发布新版本。

有关更多信息，请参阅使用发布无服务器应用程序。 AWS SAM CLI

创建 CloudFormation模板

要构建 CloudFormation 模板，您必须满足模板的先决条件并提供所需的输入和安全参数。有关更多信息，请参阅《AWS CloudFormation 用户指南》 中的模板剖析。

在 CloudFormation 模板中，您可以引用您的无服务器应用程序和. AMI 您还可以在根 CloudFormation 模板和嵌套模板中使用嵌套模板并引用无服务器应用程序。要引用无服务器应用程序，您可以使用 AWS SAM 模板。您可以从中自动生成应用程序的 AWS SAM 模板 AWS Serverless Application Repository。以下是一个示例模板。

```
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: {}
```

该 AWS SAM 模板包含以下元素：

- `ApplicationID`— 您的应用程序的 Amazon 资源名称 (ARN)。此信息位于 AWS Serverless Application Repository的我的应用程序部分。
- `SemanticVersion` – 无服务器应用程序的版本。您可以从 AWS Serverless Application Repository 的我的应用程序部分中找到此内容。
- `Parameter`（可选）– 应用程序参数。

> **Note**
>
> 对于 `ApplicationID` 和 `SemanticVersion`，不支持内置函数。您必须对这些字符串进行硬编码。克隆后会更新。`ApplicationID` AWS Marketplace

如果您计划在 CloudFormation模板中引用配置和脚本文件，请使用以下格式。对于嵌套模板 (`AWS::Cloudformation::Stack`)，仅支持不带内置函数的 `TemplateURLs`。请注意模板中的 `Parameters` 内容。

```
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: 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
```

提交您的 CloudFormation 模板和配置文件

要提交您的 CloudFormation 模板、配置和脚本文件，请授予读取存储这些文件的 Amazon S3 存储桶的 AWS Marketplace 权限。要执行此操作，请更新您的存储桶策略以包含以下权限。

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

```
}
```

更新您的 AWS Serverless Application Repository 应用程序权限

要向提交 AWS Serverless Application Repository 申请 AWS Marketplace，您必须授予阅读申请的 AWS Marketplace 权限。为此，将权限添加到与您的无服务器应用程序关联的策略中。可通过两种方式更新您的应用程序策略:

- 前往 AWSServerless Application Repository。从列表中选择您的无服务器应用程序。选择共享选项卡，然后选择创建语句。在语句配置页面上，在账户 ID 字段中输入以下服务主体：**`assets.marketplace.amazonaws.com`**。然后选择保存。
- 使用以下 AWS CLI 命令更新您的应用程序策略。

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

分享你的 AMI

所有AMIs构建和提交给的产品都 AWS Marketplace 必须遵守所有产品政策。中提供了自助AMI扫描功能 AWS Marketplace 管理门户。使用此功能，您可以启动对您的扫描AMIs。您可以快速接收扫描结果（通常不到 1 小时），并从单个位置获得清晰的反馈。成功扫描您的AMI商品后，上传您的商品加载表，提交AMI给 AWS Marketplace 卖家运营团队处理。

使用AMI无服务器应用程序提交您的 CloudFormation 产品

提交您的产品之前，请注意以下几点：

- 您必须为每个模板提供架构示意图。图中必须使用通过 CloudFormation 模板部署的每项 AWS 服务的 AWS 产品图标。此外，该示意图必须包含服务的元数据。要下载我们的官方 AWS 架构图标，请参阅AWS架构图标。
- 向买家显示的各个模板的基础设施成本估算基于您使用 AWS 定价计算器提供的估算值。在估算中，包括将作为模板的一部分部署的服务列表，以及典型部署的默认值。
- 填写产品加载表单。您可以从 AWS Marketplace 管理门户中找到产品加载表单。单个产品和多个AMI产品需要不同的AMI产品装载表。在产品加载表中，您将为 CloudFormation模板提供公开URL信息。 CloudFormation 模板必须以公众的形式提交URL。

- 使用 AWS Marketplace 管理门户 提交您的清单。在资产中，选择文件上传，附加文件，然后选择上传。在我们收到您的模板和元数据后， AWS 开始处理您的请求。

提交清单后，请 AWS Marketplace 查看并验证商品加载表。此外，还可以对应用程序 AWS Marketplace 进行区域AMIs化和无服务器化，并代表您更新 AWS CloudFormation 模板的区域映射。如果出现任何问题， AWS Marketplace 卖家运营团队将通过电子邮件与您联系。

建筑最佳实践 AMIs

本主题提供了一些最佳实践和参考，可帮助您构建用于的 Amazon 系统映像 (AMIs) AWS Marketplace。AMIs构建并提交至 AWS Marketplace 必须遵守所有 AWS Marketplace 产品政策。

保护转售权

您有责任确保非免费 Linux 发行版的转售权，但 AWS提供的亚马逊 Linux RHEL、SUSE、和 Windows 除外。AMIs

建造一个 AMI

使用以下指南进行构建AMIs：

- 确保您AMI符合所有 AWS Marketplace 政策，包括禁用 root 登录。
- 在美国东部（弗吉尼亚北部）地区创建您AMI的。
- 使用由亚马逊弹性区块商店 (AmazonEBS) AMIs 支持的现有产品创建产品，并由值得信赖的信誉良好的来源提供明确定义的生命周期，例如 AWS Marketplace。
- AMIs使用最多的 up-to-date 操作系统、软件包和软件进行构建。
- 确保所有操作都AMIs必须从使用硬件虚拟机 (HVM) 虚拟化和 64 位架构的公共AMI环境开始。
- 为构建、更新和重新AMIs发布制定可重复的流程。
- 跨所有版本和产品使用一致的操作系统 (OS) 用户名。我们建议您使用 ec2-user。
- 配置一个运行中的实例，从你的最终用户体验AMI到你想要的最终用户体验，并在提交到之前测试所有安装方法、功能和性能 AWS Marketplace。
- 按如下方式检查端口设置：
  - 基于 Linux AMIs — 确保打开了有效的SSH端口。默认SSH端口为 22。
  - 基于 Windows AMIs — 确保RDP端口已打开。默认RDP端口为 3389。此外，WinRM 端口（默认为 5985）必须打开到 10.0.0.0/16 和 10.2.0.0/16。

有关创建的更多信息AMI，请参阅以下资源：

在 Amazon EC2 用户指南AMI中@@ 创建自己的作品

AMI在《亚马逊EC2用户指南》中@@ 创建自定义窗口

如何从EBS支持的实例创建 Amazon 系统映像 (AMI)？

亚马逊 Linux AMI

Amazon EC2 实例类型和实例类型

为你AMI做好准备和保护 AWS Marketplace

我们建议您遵循以下准则来创建安全AMIs：

- 使用亚马逊EC2用户指南AMIs中的共享 Linux 指南
- AMI将您的架构设计为最低限度的安装以减少攻击面。禁用或删除不必要的服务和程序。
- 只要有可能，就对网络流量使用 end-to-end 加密。例如，使用安全套接字层 (SSL) 来保护您与买家之间的HTTP会话。确保您的服务仅使用有效的和 up-to-date 证书。
- 向您的AMI产品添加新版本时，请配置安全组以控制对您的实例的入站流量访问。确保您的安全组配置为仅允许访问为服务提供必需功能所需的最少一组端口。允许对所需的最少一组端口和源 IP 地址范围进行管理访问。有关如何向AMI产品添加新版本的更多信息，请参阅添加新版本。
- 可以考虑定期对您的 AWS 计算环境进行渗透测试，或者考虑聘请第三方代表您进行此类测试。有关更多信息，包括渗透测试申请表单，请参阅 AWS 渗透测试。
- 注意 Web 应用程序的十大漏洞并相应地构建您的应用程序。要了解更多信息，请参阅 Open Web 应用程序安全项目 (OWASP)-十大 Web 应用程序安全风险。当发现新的互联网漏洞时，请立即更新您随附的所有 Web 应用程序AMI。包含此信息的资源示例包括SecurityFocus和NIST国家漏洞数据库。

有关安全性的更多信息，请参阅以下相关资源：

- AWS 云安全性
- 互联网安全中心 (CIS)：安全基准
- Open Web 应用程序安全项目 (OWASP)：安全编码实践-快速参考指南
- OWASP十大 Web 应用程序安全风险
- SANS（SysAdmin、审计、网络和安全）常见弱点枚举 (CWE) 前 25 个最危险的软件错误
- 安全焦点
- NIST国家漏洞数据库

扫描您的AMI发布要求

为了在将其作为新产品或版本提交AMI之前进行验证，您可以使用自助扫描。自助扫描器将检查未修补的常见漏洞和漏洞（CVEs），并验证是否遵循了安全最佳实践。有关更多信息，请参阅 the section called “为你AMI做好准备和保护 AWS Marketplace”

从中 AWS Marketplace 管理门户，从 “资产” 菜单中选择 Amazon 系统映像。选择 “AMI添加” 开始扫描过程。您可以返回此页面，查看AMIs的扫描状态。

> **Note**
>
> 要了解如何授予对您的 AWS Marketplace 访问权限AMI，请参阅允许 AWS Marketplace 访问您的 AMI。

验证您的软件是否正在您的计算机上运行 AWS Marketplace AMI

您可能希望让您的软件在运行时验证它是否在使用您的AMI产品创建的 Amazon EC2 实例上运行。

要验证亚马逊EC2实例是基于您的AMI产品创建的，请使用亚马逊内置的实例元数据服务EC2。以下步骤将引导您完成此验证。有关使用元数据服务的更多信息，请参阅《Amazon Elastic Compute Cloud 用户指南》中的实例元数据和用户数据。

1. 获取实例身份证件

   每个正在运行的实例都有一个可从该实例访问的身份证件，该文件提供有关实例本身的数据。以下示例演示如何使用实例中的 curl 检索实例身份文档。

```
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" : [ "0vg00000000000000000000" ],
   "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. 验证实例身份文档

   您可以使用签名验证实例身份是否正确。有关此过程的详细信息，请参阅《Amazon Elastic Compute Cloud 用户指南》中的实例身份文档。

3. 验证产品代码

   当您最初提交AMI商品以供发布时，您的商品会被分配一个商品编码 AWS Marketplace。您可以通过检查实例身份文档中的 `marketplaceProductCodes` 字段来验证产品代码，也可以直接从元数据服务获取产品代码：

   ```
   curl http://169.254.169.254/latest/meta-data/product-codes
   0vg0000000000000000000000
   ```

   如果产品代码与您的AMI产品代码相匹配，则该实例是根据您的产品创建的。

您可能还希望验证实例身份证件中的其他信息，例如 `instanceId` 和实例 `privateIp`。

AMI 产品定价

AWS Marketplace 亚马逊系统映像 (AMI) 产品有多种定价模式。对于卖家专属优惠，这些选项可用于多年和自定义期限合同。

有关多年和自定义期限合同的更多信息，请参阅准备您的私人报价 和分期付款计划。下表提供了有关定价模式的一般信息。

> **Note**
>
> 您必须能够提供 W-9 纳税申报表（美国实体）或 W-8 表（欧盟实体），如卖家注册过程中所述。

AMI 定价模式

下表提供了有关基于 AMI 的产品的定价模式的一般信息。

定价模式	描述
免费	客户可以运行 Amazon Elastic Compute Cloud (Amazon EC2) 支持的任意数量的实例，而不会产生任何额外的软件费用。 **Note** 免费试用和年度定价无法与月度定价结合使用。
自带许可 (BYOL)	AWS Marketplace 不向客户收取软件使用费，但客户必须提供许可证密钥才能激活产品。此密钥是在外部购买的 AWS Marketplace。权利和许可实施以及所有定价和计费都由您处理。
按小时付费或采用小时 - 年度模式	小时 – 软件按小时进行收费。可以按不同的方式对每种实例类型进行定价 (但并非必须这样做)，并将用量向上舍入到最接近的整小时数。 每小时 – 免费试用 – 限制客户只能免费运行一个软件实例。您可以定义期限，介于 5 到 30 天之间。免费试用适用于运行的最昂贵的实例类型，在 1 个实例以外的任何并发使用将按小时费率计费。注意：此模式与 Amazon EC2 使用 AWS 免费套餐不同，后者每月为客户提供 750 小时的免费使用时间。 月度 - 小时 - 小时和月度费用单独应用。月度费用是按月收取的，与用量无关；小时费用仅根据小时用量应用。 小时 – 年度 - 客户可以选择为一个某种实例类型的 Amazon EC2 实例提前购买一年的用量。您为每种实例类型设置定价，可能比小时价格节省一些资金。如果客户用量超过购买的年度订阅数，则按您为该实例类型设置的小时费率进行计费。

定价模式	描述
	小时 – 多年和自定义持续时间 – 此类型优惠仅通过卖家专属优惠提供。使用卖家专属优惠，您可以指定自定义合同期限，最长为 3 年。您可以指定预付款，也可以包含灵活付款计划程序。您设置每个实例类型的定价。如果产品优惠包含灵活付款计划程序，还要为优惠包含的每个实例类型设置发票日期、付款金额和实例数量。对于使用灵活付款计划程序的活动专属优惠，客户在启动指定数量的实例后，启动任何其他实例都按专属优惠中指定的小时费率收费。有关多年和自定义期限合同的更多信息，请参阅准备您的私人报价 和the section called “分期付款计划”。 小时 - 免费试用和年度 – 它与具有年度选项的小时模型基本相同，所不同的是它包括免费试用，允许客户在您规定的天数内免费运行 1 个任意类型的实例。可以随时购买年度订阅，它们将与免费试用订阅合并在一起。 小时 – 年度 – 与年度定价模式的小时费用相同。客户可以选择为一个某种实例类型的 Amazon EC2 实例提前购买一年的用量。您为每种实例类型设置定价，可能比小时价格节省一些资金（但这不是必需的）。如果客户用量超过购买的年度订阅数，则按您为该实例类型设置的小时费率进行计费。 小时 – 多年和自定义持续时间 – 这只通过 准备您的私人报价 提供。使用卖家专属优惠，您可以指定长达三年的自定义期限合同。您可以要求客户预付费，也可以提供灵活付款计划程序。您可以为合同期限内的每个实例类型设置定价，并为推出的其他实例设置小时价格。如果您提供灵活付款计划程序，还要为优惠包含的每个实例类型设置发票日期、付款金额和实例数量。对于使用灵活付款计划程序的活动专属优惠，启动指定数量的实例后，客户启动任何其他实例都按专属优惠中指定的小时费率收费。有关多年和自定义期限合同的更多信息，请参阅准备您的私人报价 和the section called “分期付款计划”。

定价模式	描述
	Note 免费试用和年度定价无法与月度定价结合使用。
按月付费	月度 – 每月支付固定的软件费用，而与客户运行的实例数无关。月度费用是在注册时和取消后按比例收取的。示例：订阅 1 天的客户按一个月的 1/30 收取费用。 月度 - 小时 – 小时和月度费用单独应用。月度费用是按月收取的，与用量无关；小时费用仅根据小时用量应用。 Note 免费试用和年度定价无法与月度定价结合使用。
付费用量定价	根据您提供的值以及以下 4 个用量类别之一直接针对软件收费：用户、数据、带宽或主机。您可以为产品最多定义 24 个维度。所有费用仍是客户每小时产生的。 所有使用量均按月计算，并使用与现有 AWS Marketplace 软件相同的机制按月计费。用量定价也称为 AWS Marketplace Metering Service。
合同定价模式	含合同定价的 AM I — 买方支付预付费用的单 AMI 产品或带 AWS CloudFormation 堆栈的单 AMI。

AWS 费用和软件费用

基于亚马逊机器映像 (AMI) 的产品费用分为两类：

- 基础设施定价详情 — 所有基于 AMI 的产品都会产生相关的 AWS 基础设施费用，具体取决于所使用的服务和基础架构。这些费率和费用由定义和控制 AWS，并且可能在两者之间有所不同 AWS 区域。有关更多信息，请参阅 Amazon EC2 定价。
- 软件定价详情 – 对于付费产品，由卖家定义使用软件的费用。

这两个商品费用类别分别显示在 AWS Marketplace 详情页面上，以帮助买家了解使用这些商品的潜在成本。

AMI 按小时计费产品的免费试用

AMI 按小时计费产品有资格参加可选的免费试用计划。在免费试用中，客户可以订阅产品并最多在 31 天内使用单个实例，而无需为产品支付软件费用。适用的 AWS 基础设施费用仍然适用。免费试用到期后会自动转换为付费订阅。对于超出免费单位的额外用量，将向客户收费。要提供按小时计费产品的免费试用，请定义试用期的持续时间并通知 AWS Marketplace 卖家运营团队。试用期可能为 5-31 天。

客户在订阅免费试用产品时将收到一封欢迎电子邮件，其中包含免费试用期限、计算的到期日期以及有关取消订阅的详细信息。在到期日期的前 3 天，将发送提醒电子邮件。

如果您在中提供免费试用产品 AWS Marketplace，则表示您同意退款政策中描述的特定退款政策。

AMI 产品的自定义计量定价

Metering Service 概念

AWS Marketplace Metering Service 使得软件卖家可以修改其软件，通过向端点发送计量记录来捕获使用情况。卖家可以选择使用类别并为该类别最多定义 24 个维度。这些维度每小时计量一次，并根据卖家定义的价格计划收费。作为卖家，您首先必须确定要使用的维度。在 AMI 发布之后，您无法更改它。重要的服务概念包括以下内容：

- 使用类别 – 任何通过使用 Metering Service 定价的软件产品都根据一个使用类别进行分类，该类别决定了向客户收费的相应方式。使用类别包括但不限于：
  - 用户 – 一组与单个标识符关联的定义的权限。此类别适合客户的用户直接连接到软件的软件（例如，客户关系管理或商业智能报告）。
  - 主机 – 任何服务器、节点、实例、端点或计算系统的其他部分。此类别适合用于监控或扫描客户拥有的许多实例的软件（例如，性能或安全监控）。
  - 数据 – 存储空间或信息，度量单位为 MB、GB 或 TB。此类别适合批量管理存储的数据或处理数据的软件。
  - 带宽 – 以 Mbps 或 Gbps 为单位。此类别适用于允许客户指定要配置的带宽量的软件。
  - 单位 - 度量单位；请参阅下面介绍的示例。
- 用量单位 –对应于选定使用类别的软件产品特定用量单位。用量单位描述了软件的收费单位。示例包括：
  - NodesHrs（对应于主机类别）

  - UserHrs（对应于 “用户” 类别）
  - GBStored（对应数据类别）
- 用量 - 通过使用 Metering Service 定价的软件产品将按照以下三种方法之一对用量收费：
  - 预置 – 软件允许客户配置特定数量的资源（例如，用户数或固定的带宽量）供使用。每个小时，客户按照预置的功能付费。
  - 并发 – 软件允许任意数量的不同主机或用户连接到软件。每个小时，客户按照访问软件的主机数或用户数付费。
  - 累计 – 软件允许客户使用任意数量的数据，无论是处理还是存储的数据。每个小时，客户为累积的用量付费。
- 定价 – 使用 Metering Service 进行定价的软件产品必须指定单价或者最多定义 24 个维度，每个维度具有自己的价格。有关定价选项的详细信息包括：
  - 单一维度 – 这是最简单的定价选项。客户按照每资源单位每小时付费，而不考虑大小或量 (例如，每个用户每小时 $0.014，或每台主机每小时 $0.070)。
  - 多个维度 – 在所选使用类别沿不同轴向而异时，此定价选项适用。例如，对于主机监控，可根据主机大小设置不同的价格。或者，对于基于用户的定价，可根据用户的类型（例如，管理员、高级用户和只读用户）设置不同的价格。
- 计量 – 所有用量记录为计量事件，每小时记录一次。您的软件必须配置为发送相应的维度和用量到 AWS Marketplace Metering Service。
  - 分配 – 或者，您可以按您跟踪的属性将用量分发到分配中。这些分配以标签形式呈现给买家。这些标签允许买家按标签查看用量成本。例如，如果您按用户收费，并且用户具有“部门”属性，则可以使用键为“部门”的标签创建使用分配，每个值一个分配。这种方法不会更改您报告的价格、维度或总用量。但是，它允许您的客户按您产品的相应类别查看成本。

为软件定价

在使用 AWS Marketplace Metering Service 为您的软件定价时，您必须先确定使用类别和使用方式。该服务支持 6 种不同的定价方案。只能为您的产品选择其中的一种方案：

- 预置的用户 (每小时)
- 并发用户 (每小时)
- 预置的主机 (每小时)
- 并发主机 (每小时)
- 预置的带宽 (每小时)
- 累计的数据 (每小时)

接下来，您必须决定如何为选定的使用类别定价：

- 单一价格
- 多个维度 (最多 24 个)

将您的产品添加到 AWS Marketplace 介绍如何提供维度和定价的客户友好的描述。

示例：具有非线性定价的预配置带宽

假设您提供网络设备软件。您选择按预置的带宽计费。对于您的使用类别，请选择带宽。除了按带宽收费以外，您还希望随着买家用量的增加按不同的价格收费。您可以在带宽类别中定义多个维度。您可以为 25 Mbps、100 Mbps 和 1 Gbps 定义不同的价格。

示例：具有多个维度的并发主机

假设您提供用于监控其他 Amazon EC2 实例的软件。您选择按监控的主机数计费。对于您的使用类别，请选择主机。除了按主机收费以外，您还希望对监控较大的主机收取额外的费用。您可以在主机类别中使用多个维度。您可以为微型、小型、中型、大型、XL、2XL、4XL 和 8XL 实例定义不同的价格。您的软件负责将每个特定的主机映射到某个定义的维度。您的软件负责为每个使用类别维度发送单独的计量记录 (如果适用)。

将您的产品添加到 AWS Marketplace

要利用计量服务，您必须创建要发布的新 AWS Marketplace 商品。如果您的产品已上线 AWS Marketplace，则需要决定是否在当前产品之外推出新的 AWS Marketplace Metering Service 产品，或者它是否将取代您当前的产品作为唯一可供新用户使用的版本。如果您选择换货，则现有商品将从中移除， AWS Marketplace 因此新买家将无法再购买该商品。现有客户可以继续访问旧产品和实例，但他们可以在方便的时候迁移到新产品。新产品必须使用 AWS Marketplace Metering Service 来计量用量，如修改您的软件以使用 Metering Service中所述。

在具有 AMI 后，请使用自助服务工具按照标准流程共享和扫描您的 AMI。此外，还要使用在管理门户上提供的模板填写产品加载表，并上传该表格以启动提取过程。

使用以下定义来完成 AWS Marketplace Metering Service 的产品加载表单的字段。在产品加载表中，这些字段标有灵活使用定价 (FCP) 与将其与小时和月度定价产品区别开来。

- 标题 — 如果您已经有商品 AWS Marketplace 并且要添加相同的商品 AWS Marketplace Metering Service，请在括号中加上 FCP 类别和维度以区分它们（例如，“产品标题（数据）”）。
- 定价模式 – 从下拉列表中选择用量。

- FCP 类别 – 针对具有用量定价组件的付费产品向客户收费时使用的类别。从下拉列表中，选择用户数、主机数、数据或带宽。
- FCP 单位 – 具有用量定价组件的付费产品向客户收费时使用的测量单位。根据选定的 FCP 类别，将在下拉列表中显示相应的选项。下表列出了每个类别的有效单位。

类别	有效单位
用户	UserHrs
主机	HostHrs
数据	MB、GB、TB
带宽	Mbps、Gbps

- FCP 维度名称 – 在调用 `MeterUsage` API 以发送计量记录时使用的名称。此名称会显示在账单报告中。但是，由于名称不是面向外部的，因此名称不必考虑用户理解。名称不能超过 15 个字符，并且只能包含字母数字字符和下划线字符。设置名称并公开产品后，无法对其进行更改。更改名称需要使用新的 AMI。
- FCP 维度说明 – 面向客户的声明，用于描述产品的维度。描述不能超过 70 个字符，并且应便于用户理解。描述示例包括：管理员每小时和预置的每 Mbps 带宽。在发布产品后，您无法更改该描述。
- FCP 费率 – 该产品的单位软件费用。该字段支持三个小数位数。

注意:

- 您不需要填写小时和年度定价字段。
- 免费试用和年度定价不兼容。
- 使用多个 AMI 以及集群和 AWS 资源功能的产品不能使用 AWS Marketplace 计量服务。
- 价格、实例类型或 AWS 区域 变更将遵循与其他 AWS Marketplace 产品相同的流程。
- 使用 AWS Marketplace Metering Service 的产品无法转换为其他定价模式，如小时、月度或自带许可 (BYOL)。
- AWS Marketplace 建议在您的使用说明或文档中添加 IAM 政策信息。

- 您最多可包含 24 个FCP 维度。创建并发布后，您无法修改现有维度，但可以添加新维度（最多 24 个）。

如果您有任何疑问，请联系 AWS Marketplace 卖家运营团队。

修改您的软件以使用 Metering Service

您需要修改您的软件以记录客户用量，将小时用量报告发送到 Metering Service 并处理新的故障模式。该软件的运行方式与定价无关，但该软件需要了解使用类别、使用方式以及任何维度。

测量用量

您的软件必须确定选定使用类别的用量以及客户使用的维度。该值将发送到 Metering Service（每小时一次）。在所有情况下，假定您的软件能够测量、记录和读取资源用量，以便每小时将其发送到 Metering Service 一次。

对于预置使用，通常从软件配置中将其作为采样值读取，但也可能是每小时记录一次的最大配置值。对于并发使用，这可能是定期样本或每小时记录一次的最大值。对于累计使用，这是每小时累计的值。

对于多个维度的定价，必须测量多个值并将其发送到 Metering Service (每个维度一个)。在提供 AMI 时，这要求为您的软件编写或配置一组已知的维度。在创建产品后，无法更改这组维度。

对于每个定价方案，下表描述了每小时测量用量的建议方法。

场景	如何测量
预配置的用户	当前的预置用户数 (采样)。 - 或者 - 最大预置用户数 (该小时的测量结果)。
并发用户	当前的并发用户数 (采样)。 - 或者 - 最大并发用户数 (该小时的测量结果)。 - 或者 - 不同用户的总数 (该小时的测量结果)。

场景	如何测量
预配置的主机	当前的预置主机数 (采样)。 - 或者 - 最大预置主机数 (该小时的测量结果)。
并发主机	当前的并发主机数 (采样)。 - 或者 - 最大并发主机数 (该小时的测量结果)。 - 或者 - 不同主机的总数 (该小时的测量结果)。
预配置的带宽	当前的预置带宽设置 (采样)。 - 或者 - 最大预置带宽 (该小时的测量结果)。
累计的数据	当前存储的数据量 (GB) (采样)。 - 或者 - 存储的最大数据量 (GB) (该小时的测量结果)。 - 或者 - 在该小时添加或处理的总数据量 (GB)。 - 或者 - 在该小时处理的总数据量 (GB)。

供应商计量标记（可选）

供应商计量标记可帮助独立软件供应商 (ISV) 让买家更精细地了解其软件使用情况并帮助他们进行成本分配。

要标记买家的软件使用情况，您需要确定如何分配成本。首先询问您的买家他们希望在成本分配中看到什么。然后，您可以在为买家帐户跟踪的属性之间分配使用情况。属性的示例包括 Account ID、Business Unit、Cost Centers，以及您的产品的其他相关元数据。这些属性作为标签展示给买家。使用标签，买家可以在 AWS 账单控制台 (https://console.aws.amazon.com/billing/) 中查看按标签值划分为使用量的费用。供应商计量标记不会更改您报告的价格、维度或总用量。它允许您的客户按您产品的相应类别查看成本。

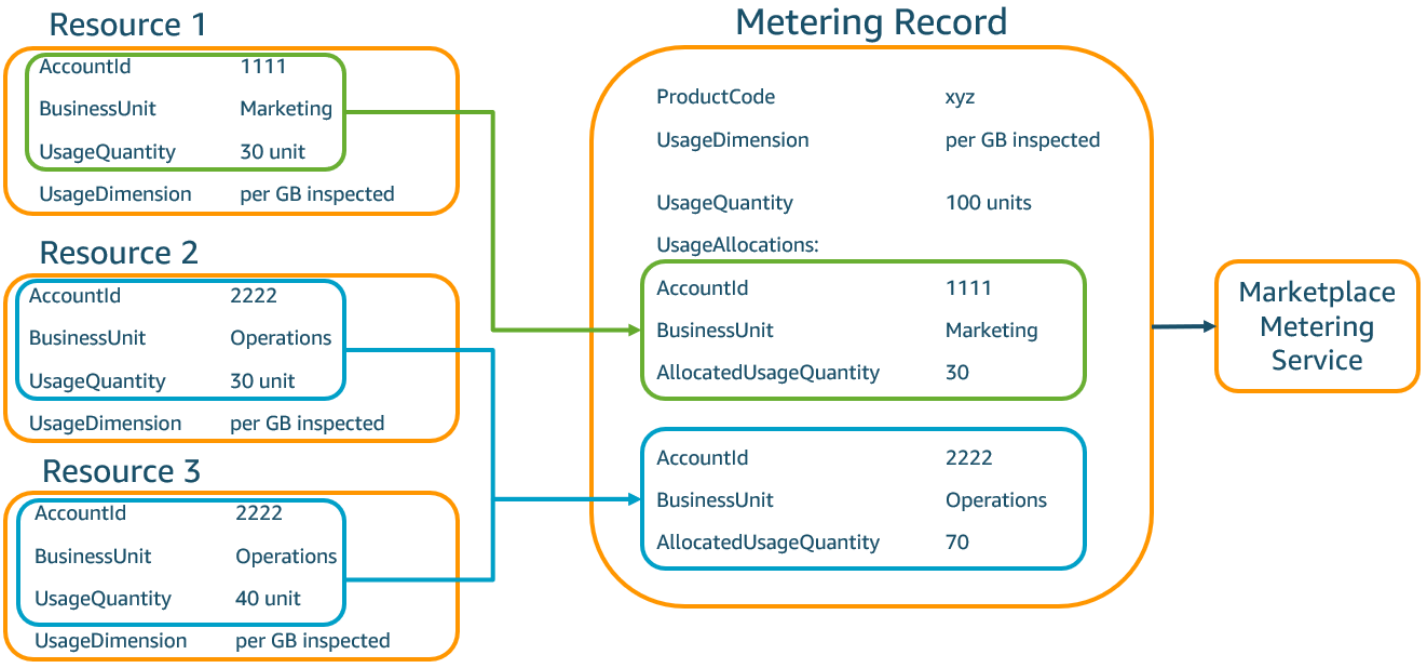
在常见情形中，买家会通过一个 AWS 账户订阅您的产品。买家还有许多与同一产品订阅相关的用户。您可以使用具有 Account ID 键的标签创建用量分配，然后将用量分配给每个用户。在这种情况下，买家可以在其账单与成本管理控制台中激活 Account ID 标签并分析个人用户的使用情况。

卖家体验

卖家可以汇总具有相同标签集的资源的计量记录，而不是汇总所有资源的使用情况。例如，卖家可以构造包含不同的 UsageAllocations 存储桶的计量记录。每个存储桶代表一组标签的 UsageQuantity，例如 AccountId 和 BusinessUnit。

在下图中，资源 1 具有一组唯一的 AccountId 和 BusinessUnit 标签，并作为单个条目出现在计量记录中。

资源 2 和资源 3 都具有相同的 AccountId 标签 2222 和相同的 BusinessUnit 标签 Operations。因此，它们在计量记录中合并为一个 UsageAllocations 条目。



卖家还可以将没有标签的资源组合成一个包含分配用量的 UsageAllocation，然后将其作为 UsageAllocations 中的一个条目发送。

限制包括：

- 标签数 – 5
- `UsageAllocations`（基数）的大小 – 2500

验证包括：

- 标签键和值允许的字符 – a-zA-Z0-9+ -=._:\/@
- `UsageAllocation` 列表中的最大标签数 – 5
- 两个 `UsageAllocations` 的标签不能相同（也就是说，标签键和值的组合相同）。如果是这样的话，他们必须使用相同的 `UsageAllocation`。
- `UsageAllocation` 的 `AllocatedUsageQuantity` 之和必须等于 `UsageQuantity`，即聚合用量。

买家体验

下表显示了买家激活 `AccountId` 和 `BusinessUnit` 供应商标签后的买家体验示例。

在此示例中，买家可以在其成本使用报告中看到分配的用量。供应商计量标签使用前缀 “`aws:marketplace:isv`”。在账单与成本管理中，买家可以再成本分配标签下的 AWS生成的成本分配标签中激活它们。

成本使用报告的第一行和最后一行与卖家向 Metering Service 发送的内容相关（如 卖家体验 示例所示）。

成本使用报告（简化）

ProductCode	买家	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	网络：每 (GB) 检查一次	70	2222	操作

ProductCode	买家	UsageDimension	UsageQuantity	aws:marketplace:isv:AccountId	aws:marketplace:isv:BusinessUnit
xyz	111122223333	网络：每 (GB) 检查一次	30	3333	财务
xyz	111122223333	网络：每 (GB) 检查一次	20	4444	IT
xyz	111122223333	网络：每 (GB) 检查一次	20	5555	市场营销
xyz	111122223333	网络：每 (GB) 检查一次	30	1111	市场营销

有关代码示例，请参阅 带有用量分配标签的 MeterUsage（可选）。

AMI产品的合同定价

基于Amazon Machine Image (AMI) 的产品的合同定价意味着买家为单AMI件商品或AMI带 AWS CloudFormation 堆栈的单件商品支付预付费用。对于采用合同定价AMI的产品，可以预先向客户开具 AWS Marketplace 账单，或者根据您与客户之间的合同按您定义的付款时间表向客户开具账单。之后，客户将有权使用那些资源。

要设置您的定价，请选择您向客户提供的一个或多个合同期限。您可以为每个合同期限输入不同的价格。您可以选择 1 个月、12 个月、24 个月和 36 个月的合同期限。对于专属优惠，您可以指定以月为单位的自定义期限（最多 60 个月）。

选择最能描述您的产品定价的类别。定价类别会显示在 AWS Marketplace 网站上。您可以从带宽（GB/s、MB/s）、数据（GB、MB、TB）、主机、请求、套餐或用户中进行选择。如果所有预定义的类别均无法满足您的需求，您可以选择更通用的单位类别。

该优惠允许向其中添加多达 24 个维度。每个维度都需要以下数据：

- 合同类别 – 对于不含基于消费的定价的合同产品，您可以选择与合同中的维度类别最相似的类别，或者如果没有与合同中该维度的单位相似的值，则可以选择单位
- 合同单位 – 根据所选“类别”，为与您的维度非常匹配的单位选择一个可用值。
- 合同维度允许多次购买 – 此字段用于指明优惠是分层定价优惠还是非分层优惠：

  分层优惠 – 允许买家仅订阅优惠中的一个可用维度。分层优惠中的维度没有数量概念。签订具有特定维度的合同本质上表明买家已经选择了该维度所指示的特定特征。

  非分层优惠 – 允许客户在合同中购买多个维度，并允许他们购买每个此类维度的多个单位。

  将此字段的值设置为 true 表示该优惠是非分层优惠。将此字段的值设置为 false 表示该优惠是分层优惠。

使用产品加载表 (PLF) 为AMI产品创建合同时，您必须为定价维度定义以下字段：

- 合同 DimensionX API 名称 — 应出现在买方账户中生成的许可证中的名称。 AWS License Manager 此名称也用作`CheckoutlicenseAPI`调用`NameEntitlement`中的 in 的值。
- 合同 DimensionX 显示名称 – 将显示在 AWS Marketplace 网站的产品详细信息和采购页面上的面向客户的维度名称。创建一个用户友好的名称。最大名称长度为 24 个字符。在列表公开之后，将无法更改 `Name` 的值。
- 合同 DimensionX 描述 – 维度的面向客户的描述，提供有关产品的维度的其他信息，例如特定维度提供的功能。描述的最大长度为 70 个字符。
- 合同 DimensionX 数量 – 用于计算产品协议修改时的比例。对于所有合同优惠，此字段的值应设置为 1。不应对其进行编辑。
- 合同 DimensionX 1 个月费率 – 针对此维度的 1 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。
- 合同 DimensionX 12 个月费率 – 针对此维度的 12 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。
- 合同 DimensionX 24 个月费率 – 针对此维度的 24 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。
- 合同 DimensionX 36 个月费率 – 针对此维度的 36 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。

示例：数据存储应用程序

	1 个月价格	12 个月价格	24 个月价格	36 个月价格
未加密的数据 (GB)	1.50 美元/GB	16.00 美元/GB	30.00 美元/GB	60.00 美元/GB
加密的数据 (GB)	1.55 美元/GB	16.60 美元/GB	31.20 美元/GB	61.20 美元/GB

示例：日志监控产品

	1 个月价格	12 个月价格	24 个月价格	36 个月价格
基本（监控 10 台主机，监控 5 个容器）	100 USD	1000 美元	2000 美元	4000 美元
标准（监控 20 台主机，监控 10 个容器）	$200	2000 美元	4000 美元	8000 美元
专业（监控 40 台主机，监控 20 个容器）	400 美元	4000 美元	8000 美元	16,000 美元
额外主机的每小时监控成本	10 美元	100 USD	$200	400 美元
额外容器的每小时监控成本	10 美元	100 USD	$200	400 美元

> **Note**
>
> 价格可以采用以下期限：1 个月、12 个月、24 个月或 36 个月。可以选择为产品提供这些选项中的一个或多个选项。期限对于每个维度都必须相同。

Example

例如，在您有`ReadOnlyUsers`和`AdminUsers`维度的情况下，如果您为其提供年度价格 ReadOnlyUsers，则还必须为`AdminUsers`其提供年度价格。

自动续订

当客户 AWS Marketplace 通过AMI合同购买您的产品时，他们可以同意自动续订合同条款。客户继续按每月或按 1 年、2 年或 3 年期限支付权利费用。

客户可以随时修改续订设置。有关更多信息，请参阅《AWS Marketplace 买家指南》中的修改现有合同。

AMI产品计费、计量和许可集成

以下主题提供有关AMI基于产品的计费、计量和许可集成的信息。

主题

- 使用 AWS Marketplace Metering Service对 AMI 产品进行自定义计量
- 使用 AWS License Manager的 AMI 产品的合同定价

使用 AWS Marketplace Metering Service对 AMI 产品进行自定义计量

AWS Marketplace Metering Service 是一项定价和计量特征，卖家可以利用它根据使用类别直接为其软件收费。有五种使用类别：用户、数据、带宽、主机或单元。您可以将 Metering Service 与基于亚马逊机器映像 (AMI)、基于容器和基于软件即服务 (SaaS) 的产品配合使用。有关更多信息，请参阅《AWS Marketplace Metering Service API 参考》。

使用 Metering Service 的所有基于 AMI 的软件必须满足以下要求：

- 您的软件必须 AWS Marketplace 通过亚马逊系统映像 (AMI) 启动。
- 如果您有现有产品 AWS Marketplace，则必须提交新的 AMI 并创建新产品才能启用此功能。
- 所有软件都必须使用 AWS Identity and Access Management (IAM) 角色进行配置。最终客户必须将 IAM 角色添加到用户正在预置软件的 Amazon Elastic Compute Cloud (Amazon EC2) 实例。通过部署软件时，使用 IAM 角色是可选的 AWS Marketplace。部署 AWS Marketplace Metering Service 软件时，这是必需的。

- 您的软件必须能够以某种方式确定使用情况。

使用 Metering Service 的产品必须按照单个使用类别向客户收费，但是可为单个类别最多定义 24 个维度。根据类别，软件可以按照预置资源、并发资源或累积资源使用来定价。所有费用仍是客户每小时产生的。所有使用量均使用与现有 AWS Marketplace 软件相同的机制按月计算和计费。

使用 AWS Marketplace Metering Service 可以启用多种新方案。例如，如果您的软件监控主机，则您可以对监控的每个主机收费。您可以根据主机大小制定不同价格，并根据每小时监控的并发主机数量收费。与此类似，如果软件允许组织中的多个用户登录，您可以按用户数收费。每小时，将针对预配置的总用户数对客户收取一次费用。

调用 AWS Marketplace Metering Service

您的软件必须每小时调用一次 Metering Service，并记录该小时的用量值。

当你的软件启动 minute-of-the-hour 时，它应该记录它的启动时间。这称为启动时刻。在每小时的启动时刻，您的软件必须确定该小时的用量值并调用 Metering Service。有关如何获取此值的信息，请参阅修改软件以使用 Metering Service。

要在每小时的启动时刻唤醒，您的软件必须使用以下方法之一：

- 您的软件中的线程。
- 随实例或软件启动的进程守护程序。
- 在应用程序启动期间配置的 cron 作业。

> **Note**
>
> 您的软件必须使用在客户的实例上配置的 IAM 角色调用 AWS Marketplace Metering Service，并指定使用维度和数量。

您的软件可以使用软件 AWS 开发工具包调用 AWS Marketplace 计量服务，类似于以下示例实现：

1. 使用实例配置文件创建服务客户端。这需要使用为 EC2 实例配置的角色。该开发工具包自动刷新角色凭证。
2. 每小时读取一次您的软件配置和状态，以确定该小时的用量值。这可能包括收集 value-per-dimension.

3. 在开发工具包客户端，使用以下参数调用 `meterUsage` 方法（为每个具有用量的维度额外调用一次）：
   - `timestamp` – 记录的小时时间戳（UTC）。
   - `productCode` – 分配给软件的产品代码。
   - `dimension` – 分配给软件的维度。
   - `quantity` – 该小时的用量值。
   - `allocations` –（可选）您可以为所跟踪的属性的用量提供分配。这些分配必须与记录中的总消耗量相加。对于买家来说，这些标签在他们的计费工具（例如 AWS Billing and Cost Management 控制台）中显示为潜在的成本分配标签。买家必须激活其账户中的标签才能使用这些标签追踪成本。

此外，您的软件还必须调用区域中的 AWS Marketplace Metering Service 端点。您的产品必须设置了正确的区域端点，以便 `us-east-1` 将记录发送到 `us-east-1` 端点，`us-west-2` 将记录发送到 `us-west-2` 端点。进行区域中调用为买家提供更稳定的体验，并防止存在不相关的区域而可能影响在其他区域中运行的软件。

在将计量记录发送到该服务时，您必须连接到您的区域中的 AWS Marketplace Metering Service。使用 `getCurrentRegion()` 帮助程序方法确定 EC2 实例运行的区域，然后将此区域信息传递给 `MeteringServiceClient` 构造函数。如果您未在 SDK 构造函数 AWS 区域 中指定，则使用默认`us-east-1`区域。如果您的应用程序尝试对该服务进行跨区域调用，将拒绝调用。有关更多信息，请参阅确定应用程序的当前区域和 getCurrentRegion()。

故障处理

您的产品必须将计量记录发送到该服务 (公有 Internet 端点)，以便可以捕获该用量和进行计费。由于客户可能使用某种方式修改网络设置而导致无法传送您的计量记录，您的产品应选择一个故障模式以处理这种情况。

> **Note**
>
> 某些计量失败可能是连接时出现的暂时性问题 AWS Marketplace Metering Service。 AWS Marketplace 强烈建议实施最长 30 分钟的重试，并以指数级退缩，以避免短期中断或网络问题。

通常，软件可能在发生故障时打开 (提供一条警告消息，但保留完整功能) 或在发生故障时关闭 (在应用程序中禁用所有功能，直到重新建立连接)。您可以选择在发生故障时打开、在发生故障时关闭或您的应用程序特定的模式。在发生计量故障两小时后内，强烈建议您禁止在发生故障时关闭。

例如，如果在发生故障时部分打开，您可以继续允许访问软件，但不允许买家修改软件设置。或者，买家仍然可以访问软件，但无法创建额外的用户。您的软件负责定义和实施该故障模式。在提交您的 AMI 时，必须包含您的软件的故障模式，以后无法对其进行更改。

限制

在设计和提交支持 Metering Service 的软件时，请务必记住以下限制：

- 客户的 IAM 角色和互联网网关要求 - 您的客户必须拥有互联网网关，并且必须使用带特定权限的 IAM 角色启动软件。有关更多信息，请参阅 AWS Marketplace 计量和授权 API 权限。如果未满足这两个条件，则您的软件无法连接到 Metering Service。
- 无法将新的使用类别添加到现有 Metering Service 产品，也无法更改这些类别 - 一旦客户订阅您的软件产品，即表示他们同意条款和条件。使用 Metering Service 更改产品中的使用类别需要新产品和新订阅。
- 无法将新的维度添加到现有 Metering Service 产品，也无法更改这些纬度 - 一旦客户订阅您的软件产品，即表示他们同意条款和条件。使用 Metering Service 更改产品中的维度需要新产品和新订阅。您可以为现有产品添加新维度，上限为 24 个。
- 缺少免费试用和年度订阅 - Metering Service 产品在启动时不支持免费试用和年度订阅。
- 多实例或基于集群的部署注意事项 - 一些软件将作为多实例部署的一部分部署。在设计您的软件时，请考虑测量用量的方式和位置以及发出计量记录的位置。

代码示例

提供了以下代码示例来帮助您将 AMI 产品与发布和维护产品所需的 AWS Marketplace API 集成。

带有用量分配标签的 **MeterUsage**（可选）

以下代码示例与具有消费定价模式的 AMI 产品相关。Python 示例将带有相应使用量分配标签的计量记录发送给您的客户 AWS Marketplace ，以向您的客户收取 pay-as-you-go 费用。

```
# 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
)
```

有关更多信息`MeterUsage`，请参阅《AWS Marketplace 计量服务 API 参考》MeterUsage中的。

响应示例

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

使用 AWS License Manager的 AMI 产品的合同定价

对于采用合同定价的基于亚马逊系统映像 (AMI) 的产品，您可以使用 AWS License Manager 将许可证与您的产品关联起来。

AWS License Manager 是一种许可证管理工具，可让您的应用程序跟踪和更新客户购买的许可证（也称为授权）。本部分提供有关如何将您的产品与 AWS License Manager集成的信息。集成完成后，您可以在 AWS Marketplace上发布您的产品清单。

有关的更多信息 AWS License Manager，请参阅《AWS License Manager 用户指南》和《AWS CLI命令参考》一AWS License Manager节。

> **Note**
>
> - 合同到期后，客户无法启动新的 AMI 实例。但是，在合同有效期内，他们可以启动任意数量的实例。这些许可证没有节点锁定或绑定到特定实例。
> - 专属优惠创建 – 卖家可以使用 AWS Marketplace 管理门户中的专属优惠创建工具为产品生成专属优惠。
> - 报告 – 您可以通过在 AWS Marketplace 管理门户中的报告部分设置 Amazon S3 存储桶来设置数据源。有关更多信息，请参阅卖家报告、数据源和控制面板。

许可模式

AWS Marketplace 与集成 AWS License Manager 支持两种许可模式：

- 可配置许可模式
- 分层许可模式

可配置许可模式

可配置许可证模型（也称为可量化许可证模型）在买家获得许可证后授权买家获得特定数量的资源。

您可以设置定价维度和每单位价格。然后，买家可以选择他们想要购买的资源数量。

Example 定价维度和每单位价格

您可以设置定价维度（例如数据备份）和每单位价格（例如每单位 30 美元）

买家可以选择购买 5、10 或 20 个单位。

您的产品会跟踪和计量使用情况，以衡量消耗的资源数量。

在配置模式下，权利将通过以下两种方式之一进行计数：

- 消耗许可证
- 浮动许可证

消耗许可证

在使用时，许可证是从允许数量的许可证池中消耗。该权利已永久签出，无法返回到许可证池。

Example 处理有限数量的数据示例

用户有权处理 500 GB 的数据。当他们继续处理数据时，系统会从 500 GB 的池中消耗数量，直到所有 500 GB 的许可证都用完为止。

对于扣除许可证，您可以使用 `CheckoutLicense` API 操作来查看已消耗的许可证单位。

Example 每年备份到 S3 的多个单位示例

您拥有的存储产品允许将多达 1024 个单位的数据备份到 Amazon Simple Storage Service (Amazon S3) 中以存储一年的数据。您的应用程序可以通过使用多个 Amazon EC2 实例启动。您的应用程序具有跟踪和聚合数据的机制。您的软件在每次备份时或以固定的时间间隔使用产品 ID 调用 `CheckoutLicense` API 操作来更新消耗的数量。

在此示例中，您的软件调用 `CheckoutLicense` 以签出 10 个单位的数据。当总容量达到客户购买的备份限制时，API 调用失败。

请求

```
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"
```

响应

```
{
```

```
  "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"
}
```

浮动许可证

使用后，许可证将返回到允许数量的许可证池中。

Example 固定上限的用户数示例

某用户被授予应用程序 500 个并发用户的权利。当用户登录和注销时，用户会被消耗并返回到 500 个用户的池中。但是，应用程序无法从池中消耗超过 500 个用户，因为 500 个并发用户是固定上限。

对于浮动许可证，您可以使用 `CheckInLicense` API 操作将许可证单位返回到权利池。

Example 一年的并发用户数示例

您的产品根据并发用户数量定价。客户为 10 个用户购买了一份为期一年的许可证。客户通过提供 AWS Identity and Access Management (IAM) 权限来启动软件。当用户登录时，您的应用程序会调用 `CheckoutLicense` API 操作将数量减少 1。当用户注销时，应用程序会通过调用 `CheckInLicense` API 操作将该许可证返回到池中。如果您不调用 `CheckInLicense`，则许可证单位将在 1 小时后自动签入。

> **Note**
>
> 在以下请求中，`key-fingerprint` 不是占位符值，而是发布所有许可证时使用的指纹的实际值。

请求

```
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"
```

响应

```
{
  "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"
}
```

分层许可模式

分层许可模式在买家获得许可后，使买家有权使用特定级别或层的应用程序特征。

您可以为产品创建层，例如基本、中级和高级。然后，买家选择一个预定义的层。

应用程序无需跟踪或计量应用程序的使用情况。

在分层许可模式下，权利不计算在内，而是表示客户购买的服务层。

如果您想同时提供捆绑特征，我们建议您使用分层许可模式。

Example 基本、中级和高级层

客户可以签署软件三个可能层之一的合同：基础、中级或高级。每个层都有自己的定价。您的软件可以通过调用 `CheckoutLicense` API 操作并在请求中指定所有可能的层来识别客户已注册的层。

请求响应包含与客户购买的层相对应的权利。基于这些信息，该软件可以预置适当的客户体验。

请求

```
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"
```

响应

```
{
  "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"
}
```

集成工作流

以下步骤显示了将 AMI 产品与 AWS License Manager集成的工作流：

1. 卖家创建具有 AWS License Manager 集成功能的产品。
2. 卖家在上架商品 AWS Marketplace。
3. 买家找到商品 AWS Marketplace 并购买。
4. 许可证通过买家的 AWS 账户发送给他们。
5. 买家通过启动 Amazon Elastic Compute Cloud (Amazon EC2) 实例、Amazon Elastic Container Service (Amazon ECS) 任务或 Amazon Elastic Kubernetes Service (Amazon EKS) 容器组 (pod) 软件使用软件，使用 IAM 角色进行部署。
6. 软件读取买方 AWS License Manager 账户中的许可证，发现购买的权利并相应地配置功能。

> **Note**
>
> License Manager 不进行任何跟踪或更新；这些是通过卖家的应用程序完成的。

License Manager 集成先决条件

发布产品之前，您必须首先完成以下操作：

1. 在中创建新的 AMI 产品 AWS Marketplace 管理门户，并记下其产品代码。
2. 填写产品加载表单 (PLF) 以及必要的价格信息，并将其返回给我们进行处理。
3. 将用于运行应用程序的任务或容器组 (pod) 的IAM 角色与调用 `CheckoutLicense`、`ExtendLicenseConsumption` 和 `CheckInLicense` 所需的 IAM 权限搭配使用。

   以下 IAM 策略中详细介绍了所需的 IAM 权限。

```
{
   "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. 使用您定义的所有定价维度的记录对 `RegisterUsage` API 操作进行测试调用。

将基于 AMI 的产品与 AWS License Manager

您可以使用 AWS License ManagerAPI 将基于 AMI 的产品与 License Manager 集成。使用 AWS Marketplace 基于 AMI 的产品启动 Amazon EC2 实例。

> **Note**
>
> 在执行以下步骤之前，请确保您已完成 the section called “License Manager 集成先决条件”。

要将基于 AMI 的产品与 License Manager 集成，请执行以下操作：

1. 完成the section called “在 License Manager 中创建测试许可证”中的过程。您必须在 License Manager 中创建测试许可证才能测试您的集成。
2. 使用您在步骤 1 中获得的许可亚马逊资源名称 (ARN) 运行 GetLicenseAPI 操作。记下 `GetLicense` 响应的 `KeyFingerprint` 属性的值以备后用。
3. 下载最新的公共 AWS SDK 并将其包含在您的应用程序中。
4. 要验证买家是否有权使用您的应用程序的许可证，请运行 CheckoutLicenseAPI 操作。使用您在步骤 1 中获得的测试许可证的权利详细信息和密钥指纹。

   如果找不到许可证的授权，或者超过了最大权利计数，则 `CheckoutLicense` API 操作将返回 `NoEntitlementsAllowedException`。如果权利有效或可供使用，则 `CheckoutLicense` 操作将返回包含请求的权利及其值的成功响应。
5. （仅适用于浮动授权）使用`CheckoutLicense`响应中收到`LicenseConsumptionToken`的 CheckinLicenseAPI 运行 API 操作。此操作会将先前已签出的权利释放回可用权利池中。
6. 成功使用您在步骤 1 中创建的测试许可证验证 License Manager 集成后，将代码中的密钥指纹更新为 `aws:294406891311:AWS/Marketplace:issuer-fingerprint`。现在，您可以继续使用由 AWS Marketplace颁发的许可证。

遵循为 AMI 产品构建应用程序的发布流程，然后 AWS Marketplace 按照产品发布流程提交产品。

在 License Manager 中创建测试许可证

您可以使用版本 2 的 AWS Command Line Interface (AWS CLI) 在中创建测试许可证 AWS License Manager。此测试许可证仅用于验证和测试集 AWS License Manager 成。测试完成后，您可以删除测试许可证。实际许可证由 AWS Marketplace 使用不同的密钥指纹生成。

AWS Marketplace 中支持两种类型的权利 AWS License Manager。但是，一个产品只能启用一种类型。在创建许可证（包括测试许可证）时，必须指定以下任一权利：

分层权利 – 分层许可模式使客户有权使用某些应用程序特征。买家无法定义他们想要购买的产品数量。但是，他们可以选择一个预定义的套餐或层。客户可以稍后修改合同以订阅其他层。

可配置的权利 – 客户购买许可证后，可配置许可证模型会向其授予一定数量的资源权利。客户在订阅过程中选择他们想要购买的单位数量，并将根据单价进行计费。客户还可以订阅多个维度。

使用 `CheckoutLicense` API 操作需要以下参数：

- `CheckoutType` – 有效值为 `Perpetual` 或 `Provisional`。
  - `Perpetual` – 当已签出的权利池中的权利数量将用尽时使用。示例：买家有权处理 500 GB 的数据。当他们继续处理数据时，会消耗 500 GB 池的数量并将其耗尽。获取已购买许可证的状态，了解许可证是已过期还是即将过期，以便向客户发送通知。
  - `Provisional` – 用于浮动许可证权利，其中权利从池中签出，并在使用后返回。示例：某用户被授予应用程序 500 个并发用户的权利。当用户登录和注销时，用户会被消耗并返回到 500 个用户的池中。有关浮动许可证权利的更多信息，请参阅 AWS License Manager 中卖家颁发的许可证。
- `ClientToken` – 区分大小写的唯一标识符，可确保结果准确无误，并且无论尝试多少次都相同。我们建议您为每个请求使用随机通用唯一标识符 (UUID)。
- `Entitlements` – 待签出的权利列表。
  - 对于分层权利，请按以下方式提供 `Name` 和 `Unit` 属性：

```
{
"Name": "<Entitlement_Name>",
"Unit": "None"
}
```

  - 对于可配置权利，请按如下方式提供 `Name`、`Unit` 和 `Value` 属性：

```
{
"Name": "<Entitlement_Name>",
"Unit": "<Entitlement_Unit>",
"Value": <Desired_Count>{
```

```
}
```

- `KeyFingerprint` – 使用此密钥指纹验证许可证是否由 AWS Marketplace颁发。由颁发的许可证的密钥指纹 AWS Marketplace 如下：

  `aws:294406891311:AWS/Marketplace:issuer-fingerprint`
- `Product SKU`— 采用与产品关联的全球唯一标识符 (GUID) 格式 AWS Marketplace 的产品ID。

Example 可配置权利示例

以下是使用 `CheckoutLicense` API 操作签出名为 `PowerUsers` 的可配置权利的请求示例。

```
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 分层权利示例

以下是使用 `CheckoutLicense` API 操作签出名为 `EnterpriseEdition` 的特征权利的请求示例。

```
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"
```

为基于 AMI 的产品创建测试许可证

1. 在安装了 AWS CLI v2 的本地环境中，运行以下脚本。该脚本创建测试许可证并配置相应的产品详细信息。

   > **Note**
   >
   > 使用与部署和测试 AWS 账户 软件时 AWS 账户 不同的测试。无法在同一 AWS 账户中创建、授予和签出许可证。

```
#!/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 控制台授予许可证。有关更多信息，请参阅《License Manager 用户指南》中的分发权利。
3. 登录充当买家账户 AWS 账户 的，您将在其中部署和测试您的软件。这必须与创建和授予许可证的 AWS 账户 不同。 AWS 账户
4. 前往 AWS License Manager 控制台接受并激活授予的许可证。有关更多信息，请参阅《License Manager 用户指南》中的 管理授予的许可证。
5. 在您的环境中运行以下命令。

```
# 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)
```

前面的命令使用 PROVISIONAL 作为 CheckoutType 参数的值。如果权利使用消耗许可证，则该值使用 PERPETUAL。

License Manager API 调用

要管理存储在客户 License Manager 帐户中的许可证，您的软件可以使用以下 API 调用：

- GetLicense – 获取已购买许可证的状态，了解许可证是已过期还是即将过期，以便向客户发送通知。
- CheckoutLicense – 发现用户已购买的许可证。当用户使用了一定数量的许可证时，您还可以使用它来更新许可证数量。使用 CheckoutLicense，您可以继续签出客户使用的许可证数量。当客户耗尽所有许可证时，此调用将返回错误。有关建议运行 CheckoutLicense 的节奏的信息，请参阅the section called “许可证续订和升级”。
- ExtendLicenseConsumption – 如果是浮动维度，当软件签出许可证时，它会在 60 分钟后自动将许可证返回到池中。如果您想延长许可证的签出时间，您的软件可以调用 ExtendLicenseConsumption 将许可证再延长 60 分钟。
- CheckInLicense – 如果是浮动维度，则要将许可证返回到权利池时，请使用 CheckInLicense。
- ListReceivedLicenses – 列出买家购买的许可证。

许可证续订和升级

客户可以在 AWS Marketplace 管理门户上续订或升级许可证。在他们进行额外购买后， AWS Marketplace 会生成反映新权利的新版本的许可证。您的软件使用相同的 API 调用读取新的权利。您不必在 License Manager 集成方面执行任何不同的操作即可处理续订和升级。

由于许可证续订、升级、取消等，我们建议在使用产品时该产品按固定节奏执行 `CheckoutLicense` API 调用。通过定期使用 `CheckoutLicense` API 操作，产品可以检测权利的变化，例如升级和到期。

我们建议您每 15 分钟执行一次 `CheckoutLicense` API 调用。

亚马逊AMI商品SNS通知

要接收通知，您需要订阅商品创建期间提供给您的 AWS Marketplace 亚马逊简单通知服务 (AmazonSNS) 主题。这些主题提供有关客户产品订阅发生更改的通知。例如，您可以知道客户何时接受专属优惠。

> **Note**
>
> 在商品创建过程中，会为您的商品创建一个 Amazon SNS 主题。要订阅通知，您需要亚马逊SNS主题的亚马逊资源名称 (ARN)（例如`arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`）。服务器产品的卖家门户中不可用。ARN请联系 AWSMarketplace 运营团队申请ARN。

以下 Amazon SNS 主题适用于AMI商品：

- Amazon SNS 主题：`aws-mp-subscription-notification` – 当买家订阅或取消订阅产品时，此主题会向您发送通知。此通知适用于按小时定价模式，包括按小时定价模式和按年计费模式。

Amazon SNS 主题：`aws-mp-subscription-notification`

`subscribe-success` 和 `subscribe-fail` 操作的 `aws-mp-subscription-notification` 主题中的每条消息都采用以下格式。

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123"
}
```

这些区域有：*`<action-name>`* 将因通知而异。可能的操作包括：

- `subscribe-success`
- `subscribe-fail`
- `unsubscribe-pending`
- `unsubscribe-success`

`offer-identifier` 仅当优惠为专属优惠 时，通知中才会显示。

为亚马逊SQS队列订阅亚马逊主题 SNS

我们建议通过 Amazon SQS 队列订阅所提供的SNS主题。有关创建SQS队列和为队列订阅主题的详细说明，请参阅《亚马逊简单通知服务开发者指南》中的 “为亚马逊SQS队列订阅亚马逊SNS主题”。

> **Note**
>
> 您只能订阅 AWS 账户 用于销售产品的用户 AWS Marketplace SNS的主题。但是，您可以将消息转发到其他帐户。有关更多信息，请参阅《亚马逊简单通知服务开发者指南》中的向其他账户中的亚马逊SQS队列发送亚马逊SNS消息。

轮询SQS队列以获取通知

在您为SQS队列订阅SNS主题后，消息将存储在中SQS。您需要定义一项服务，以不断轮询该队列，查找消息并相应地处理它们。

AMI 产品核对清单

在向提交您的亚马逊系统映像 (AMI) 产品请求之前 AWS Marketplace，请查看此清单。验证此信息有助于确保您的提交顺利通过发布流程。

产品使用：

- 您的 AMI 必须已准备好用于生产。
- 您的 AMI 不能通过时间或任何其他方法来限制产品使用。
- 您的 AMI 必须与一键式交付体验兼容。
- 软件中包含使用产品所需的所有内容，包括客户端应用程序。需要外部依赖关系的产品（例如软件包或客户端应用程序）必须遵守包括适当披露在内的 the section called “产品使用策略”。

- 默认用户使用随机化密码，或者创建初始用户要求使用对实例唯一的值（如实例 ID），验证买家已获授权可以使用此实例。

对于免费或付费产品：

- 使用产品不需要其他许可证。
- 买家不必提供个人身份信息（如电子邮件地址）即可使用产品。

AMI 准备：

- 您的产品名称和描述必须与您提供的 AMI 产品的描述字段相匹配。
- 利用硬件虚拟机 (HVM) 虚拟化和 64 位架构。
- 不包含任何已知的漏洞、恶意软件或病毒。
- 买家对 AMI 拥有操作系统级管理访问权限。
- 通过 AMI 自助服务扫描运行您的 AMI。

对于 Windows AMI：

- 使用 Windows Server 2012 及更高版本的操作系统时，请使用最新版本的 EC2Launch v2。
- 如果你的 AMI 是在 Windows Server 2012 和 2012 R2 的基础上使用 EC2Config 构建的，请使用最新版本的 EC2Config。
- 如果您使用的是 EC2Config v2，请确保完成以下操作：
  - 在 Amazon EC2Launch 设置中，在设置管理员账户下选择随机。
  - 在 Amazon EC2Launch 设置中，在启动 SSM 服务下选择复选框。
  - 添加UserData到 EC2 v2 任务配置中。
- 如果你使用的是 EC2Config，请在你的 AMI 中启用以下设置文件：Ec2 SetPassword、Ec2 和 Ec2 WindowsActivate。HandleUserData
- 确保不存在来宾账户或远程桌面用户。

对于 Linux AMI：

- 根登录已锁定或禁用。
- 不包含已授权密钥、默认密码或其他凭证。

产品加载表单或产品选项卡：

- 所有必填字段均已填写。
- 所有值都在指定字符限制之内。
- 所有 URL 加载均无错误。
- 产品图像至少为 110 像素宽，且长宽比介于 1:1 与 2:1 之间。
- 为所有已启用的实例类型指定定价（对于每小时、基于小时的每月定价和基于小时的每年定价模式）。
- 指定月度定价（对于基于小时的每月定价和每月定价模式）。

基于 AMI 的产品要求

AWS Marketplace 对所有亚马逊系统映像 (AMI) 产品和产品实行以下政策 AWS Marketplace。这些策略为我们的客户提供了一个安全、可靠和值得信赖的平台。

所有产品及其相关元数据在提交时都会经过审核，以确保它们符合或超过当前 AWS Marketplace 政策。这些政策经过审查和调整，以满足不断变化的安全准则。 AWS Marketplace 持续扫描您的产品，以验证它们是否符合安全指南的更改。如果产品不合规， AWS Marketplace 将与您联系以更新您的 AMI 产品以符合新标准。同样，如果找到新发现的漏洞而影响 AMI，我们会要求您提供更新的 AMI 并部署了相关的更新。在提交 AMI 之前，您必须使用自助 AMI 扫描工具。这有助于确保 AMI 符合 AWS Marketplace 策略。

安全策略

所有 AMI 必须符合以下安全策略：

- AMI 不得包含自助服务 AMI 扫描工具或 AWS 安全部门检测到的任何已知漏洞、恶意软件或病毒。
- AMI 必须使用当前支持的操作系统和其他软件包。任何带有即将停用 (EoL) 操作系统或其他软件包的 AMI 版本都将从 AWS Marketplace中除名。您可以使用更新的程序包构建新 AMI，并将其作为新版本发布到 AWS Marketplace。
- 所有实例身份验证都必须使用密钥对访问权限，而非基于密码的身份验证，即使用户在启动时生成、重置或定义了密码也是如此。AMI 不得出于任何原因包含密码、身份验证密钥、密钥对、安全密钥或其他凭证。
- AMI 不得向用户请求或使用访问密钥或私有密钥以访问 AWS 资源。如果您的 AMI 应用程序需要用户访问权限，则必须通过实例化的 AWS Identity and Access Management (IAM) 角色来实现 AWS CloudFormation，该角色将创建实例并关联相应的角色。当为采用 AWS CloudFormation 交付方式

的产品启用单 AMI 启动时，相应的使用说明必须包括创建权限最低的 IAM 角色的明确指南。有关更多信息，请参阅 the section called “AMI基于交付使用 CloudFormation”。

- 基于 Linux 的 AMI 不得允许 SSH 密码身份验证。可以通过 `sshd_config` 文件将 `PasswordAuthentication` 设置为 `NO` 以禁用密码身份验证。

访问策略

访问策略分为三类：一般策略、Linux 特定策略和 Windows 特定策略。

一般访问策略

所有 AMI 必须符合以下访问策略：

- AMI 必须允许操作系统 (OS) 级别管理功能，以满足合规性要求，执行漏洞更新以及访问日志文件。基于 Linux 的 AMI 使用 SSH，而基于 Windows 的 AMI 使用 RDP。
- AMI 不得包含授权的密码或授权的密钥。
- AMI 不得使用固定的密码进行管理访问。AMI 必须使用随机的密码。一种替代的实施是检索实例元数据并将 `instance_id` 作为密码。在允许管理员设置或更改自己的凭证之前，必须提示管理员输入该随机密码。有关检索实例元数据的信息，请参阅 Amazon EC2 用户指南中的实例元数据和用户数据。
- 您不得具有访问客户的运行实例的权限。客户必须显式启用任何外部访问权限，默认情况下，AMI 内置的可访问性必须关闭。

特定于 Linux（或类似 Unix）的访问策略

基于 Linux 或类似 UNIX 的 AMI 必须遵守以下访问策略以及一般访问策略：

- AMI 必须禁用基于密码的远程登录。
- AMI 必须禁用 root 用户远程登录。
- AMI 必须允许用户获得管理员控制权才能执行 root 功能。例如，允许`sudo`访问基于 Linux 的操作系统。对于其他系统，请允许完全权限级别的访问权限。
- AMI 必须记录根活动以进行审计跟踪。
- AMI 不得包含操作系统用户的授权密码。
- AMI 不得包含授权密钥。
- AMI 的根密码不得为空或为空。

特定于 Windows 的访问策略

基于 Windows 的 AMI 必须遵守以下访问策略以及一般访问策略：

- 对于 Windows Server 2016 和更高版本，请使用 `EC2Launch`。
- 对于 Windows Server 2012 R2 和更早版本，请使用最新版本的 `Ec2ConfigService` 并启用 `Ec2SetPassword`、`Ec2WindowsActivate` 和 `Ec2HandleUserData`。
- 不允许删除来宾账户和远程桌面用户。

客户信息策略

所有 AMI 必须遵守以下客户信息策略：

- 除非 BYOL（自带许可）另行要求，否则软件不得在客户不知晓且未明确同意的情况下收集或导出客户数据。收集或导出客户数据的应用程序必须遵循以下准则：
  - 客户数据收集必须是自助服务、自动化和安全的。买家无需等待卖家批准即可部署软件。
  - 必须在列表的描述或使用说明中明确说明对买家数据的要求。这包括收集的内容、客户数据的存储位置以及如何使用这些数据。例如，此产品会收集您的姓名和电子邮件地址。此信息由 <company name> 发送并由其存储。此信息将仅用于就 <product name> 问题联系买家。
  - 不得收集付款信息。

产品使用策略

所有 AMI 必须符合以下产品使用策略：

- 产品不得按时间、用户数或其他条件限制对产品或产品功能的访问。不支持测试版和预发布产品，也不支持唯一目的是提供试用或评估功能的产品。支持开发人员、社区和 BYOL 版本的商业软件，但前提是在 AWS Marketplace上还提供了等效的付费版本。
- 所有 AMI 必须与通过网站启动体验或通过 AWS CloudFormation的基于 AMI 的交付兼容。对于从网站启动，AMI 不能在创建实例时要求提供客户或用户数据以正常运行。
- AMI 及其软件必须可以自助方式部署，并且不得要求额外的付款方式或费用。部署时需要外部依赖的应用程序必须遵循以下准则：
  - 必须在列表的描述或使用说明中披露该要求。例如，此产品需要互联网连接才能正确部署。部署时会下载以下软件包：<list of package>。
  - 卖家需对所有外部依赖的使用负责，并确保其可用性和安全性。

  - 如果外部依赖关系不再可用，则还必须从中 AWS Marketplace 删除该产品。
  - 外部依赖项不得要求额外的付款方式或费用。
- 需要持续连接非买家直接控制的外部资源（例如外部 API 或由卖家或第三方 AWS 服务 管理）的 AMI 必须遵循以下指南：
  - 必须在列表的描述或使用说明中披露该要求。例如，此产品需要持续的互联网连接。需要以下持续的外部服务才能正常运行：<list of resources>。
  - 卖家需对所有外部资源的使用负责，并确保其可用性和安全性。
  - 如果外部资源不再可用，则还必须从中 AWS Marketplace 移除该产品。
  - 外部资源不得要求额外的付款方式或费用，并且必须自动设置连接。
- 产品软件和元数据不得包含将用户重定向到 AWS Marketplace中未提供的其他云平台、其他产品或追加销售服务的语言。
- 如果您的产品是其他产品或其他 ISV 产品的附加组件，则您的产品描述必须表明它扩展了其他产品的功能，如果没有它，产品的应用将受到限制。例如，本产品扩展了 <product name> 的功能，如果没有它，则该产品的应用将受到限制。请注意，<product name> 可能需要自己的许可才能使用此列表的全部功能。

架构策略

所有 AMI 必须符合以下架构策略：

- 的来源 AMI AWS Marketplace 必须在美国东部（弗吉尼亚北部）地区提供。
- AMI 必须使用 HVM 虚拟化。
- AMI 必须使用 64 位或 64 位 ARM 架构。
- AMI 必须是由 Amazon Elastic Block Store (Amazon EBS) 支持的 AMI。我们不支持由 Amazon Simple Storage Service (Amazon S3) 支持的 AMI。
- AMI 不得使用加密的 EBS 快照。
- AMI 不得使用加密的文件系统。
- 必须构建 AMI，使其可以全部运行， AWS 区域 并且不受区域限制。不允许为不同区域构建不同的 AMI。

AMI 产品使用说明

为您的 AMI 产品创建使用说明时，请按照the section called “AMI和容器产品使用说明”中的步骤和指南进行操作。

基于容器的产品

AWS Marketplace 支持使用 Docker 容器的软件产品。容器产品由交付选项组成，这些交付选项是一起使用的一组容器映像和部署模板。您可以为产品提交至少一个，最多为四个交付选项。对于每个交付选项，您都需要提供一组容器映像、使用说明和部署模板链接，以便客户启动该交付选项。

AWS Marketplace 买家可以在已发布的商品详情页面上看到可供他们使用的配送选项。订阅产品并选择首选交付选项后，买家将看到有关启动和使用该产品的信息和说明。有关容器映像交付选项，买家可以看到指向可用部署模板和容器镜像的链接URLs。他们还会收到有关如何拉取单个容器映像的说明。对于Helm图表交付选项，买家将看到使用Helm进行发布的 step-by-step 说明。

有关购买体验的演练，您可以参考此视频：在 Amazon ECS 集群上部署 AWS Marketplace 容器 (3:34)。

你可以在任何环境中的任何 Kubernetes 集群 AWS Marketplace 上查找、订阅和部署第三方 Kubernetes 应用程序。您可以在亚马逊弹性容器服务 (亚马逊)、亚马逊弹性 Kubernetes 服务 (亚马逊ECS) 上部署第三方 Kubernetes 应用程序，也可以使用 Amazon AnywhereEKS（Anywhere AWS Fargate）在本地部署第三方 Kubernetes 应用程序。EKS EKS您也可以在本地或亚马逊弹性计算云 (Amazon) 中将它们部署到自行管理的 Kubernetes 集群上。EC2

你可以在任何兼容 Docker 的运行时上运行 Free 和 Bring Your Own License model (BYOL) 容器产品。

主题

- 获取帮助
- 容器产品入门
- 基于容器的产品要求
- 容器产品定价
- 容器产品计费、计量和许可集成
- 亚马逊关于集装箱商品的SNS通知

获取帮助

如需容器产品的帮助，请联系您的 AWS Marketplace 业务开发合作伙伴或 AWS Marketplace 卖家运营团队。

容器产品入门

本主题介绍了与创建、测试和发布容器产品相关的所有步骤 AWS Marketplace。

主题

- 先决条件
- 创建容器产品
- 产品生命周期
- 更新产品可见性
- 更新允许名单 AWS 账户 IDs
- 添加定价维度
- 更新维度信息
- 更新定价条款
- 按国家/地区更新可用性
- 更新最终用户许可协议
- 更新产品的退款政策
- 为您的容器产品创建产品 ID 和产品代码
- 创建初始列表
- AWS Marketplace Metering Service 为您的容器产品集成
- AWS License Manager 为您的容器产品集成
- 添加产品的新版本
- 测试并发布您的产品
- 更新版本信息
- 限制您的 Amazon EKS 附加组件的某个版本
- 为容器产品创建或更新产品信息
- 容器产品扫描是否存在安全问题

先决条件

在开始之前，您必须满足以下先决条件：

1. 访问和使用 AWS Marketplace 管理门户。这是您用来注册为卖家并管理您销售的商品的工具 AWS Marketplace。有关更多信息，请参阅 AWS Marketplace 管理门户。
2. 注册为卖家并提交您的税务和银行信息。有关更多信息，请参阅 卖家注册过程。
3. 在亚马逊弹性容器服务（亚马逊）、亚马逊 Elastic Kubernetes Service（EKS亚马逊ECS）或中创建至少一个容器。 AWS Fargate请确保您有相关映像的链接。
4. 规划如何在 AWS Marketplace中创建和集成容器产品。

   我们建议您在公开发布产品之前尽早规划您的定价、授权和计量策略。

   - 有关基于容器的产品要求的信息，请参阅基于容器的产品要求。
   - 有关为产品设置定价的信息，请参阅容器产品定价。
   - 有关付费版基于容器的产品的自定义计量的信息，请参阅采用 AWS Marketplace Metering Service按小时计量和自定义计量。

创建容器产品

创建容器产品涉及以下步骤：

1. 创建产品 ID 和代码。
2. the section called “创建初始列表”.
3. 添加产品的初始版本。
4. 对于付费产品，请将计量功能集成到您的产品中。
5. the section called “更新产品可见性”.

> **Note**
>
> 有关产品生命周期的信息，请参阅the section called “产品生命周期”。

产品生命周期

当您在中创建产品时 AWS Marketplace，该产品最初发布时可见性有限，因此允许列表中的账户（包括创建该产品的账户）可以看到该产品。准备就绪后，您可以将其发布到 AWS Marketplace 目录中，以允许买家订阅和购买您的产品。

在服务器产品页面上，您可以查看您的产品列表。根据所处的阶段，产品将具有以下任一状态。

- 暂存 – 未完成的产品，您仍在为其添加信息。在第一次保存并退出自助服务体验时，成功的变更请求会使用您提交的完成步骤中的信息创建未发布的产品。在此状态下，您可以继续向产品添加信息或通过变更请求更改已提交的详细信息。
- 受限 – 产品在提交给系统并通过系统中的所有验证后即已完成。然后，产品将发布到受限状态。此时，该产品有一个详细信息页面，只有您的账户和您加入允许列表的人才能访问该页面。您可以通过详细信息页面测试您的产品。如需更多信息或帮助，请联系 AWSMarketplace 卖家运营团队。
- 公开 – 当您准备发布产品以便买家可以查看和订阅产品时，您可以使用更新可见性更改请求。此请求启动了 AWS Marketplace 卖家运营团队根据 AWS 政策审查和审核您的商品的工作流程。产品获得批准并处理变更请求后，产品将从受限状态更改为公开。有关 AWS 指导方针的信息，请参阅基于容器的产品要求。
- 受限 – 果您想阻止新用户订阅您的产品，则可以使用更新可见性更改请求来限制该产品。受限状态表示现有允许列表中的用户可以继续使用该产品。但是，该产品将不再对公众可见，也不会再向新用户提供。

更新产品可见性

要更改哪些买家可以查看您的商品 AWS Marketplace，您可以使用 “更新可见性”。

要更新可见性，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 athttps://aws.amazon.com/marketplace/management/tour/，然后登录您的卖家账户。
2. 转到服务器产品页面，在当前服务器产品选项卡上，选择要修改的基于容器的产品。
3. 从请求更改下拉列表中，选择更新可见性。

> **Note**
>
> 您可以使用此更改请求请求将产品从受限态更改为公开状态。但是，变更请求必须经过 AWS Marketplace 卖家运营团队的批准才能移至公开。

4. 选择提交，提交您的请求以供审核。
5. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

更新允许名单 AWS 账户 IDs

您可以更改可以在 AWS 账户 IDs受限状态下查看您的产品的列表。允许列表中的账户在产品详细信息页面上的产品版本旁边会显示一个受限徽章。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/ 并登录您的卖家账户。
2. 在服务器产品页面上，选择要修改的容器产品。
3. 从请求更改下拉列表中，选择更新允许列表。将显示当前被列入允许列表的账户列表。
4. 在 “允许名单 AWS 账户” 字段中，输入 AWS 账户 IDs并使用逗号分隔它们。
5. 选择提交，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

添加定价维度

您可以向用于产品计费的定价模式中添加定价维度。有关定价模式的更多信息，请参阅容器定价模式。

> **Note**
>
> 上不提供为包含消费定价的合同添加定 pay-as-you-go 价维度（例如，额外用量的定价） AWS Marketplace 管理门户。
> 您不能在合同、用量和带消费定价的合同之间更改定价模式。联系 AWS Marketplace 卖家运营团队获取帮助。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/ 并登录您的卖家账户。
2. 在服务器产品页面上，选择要修改的容器产品。
3. 从请求更改下拉列表中，选择更新定价维度，然后选择添加定价维度。
4. 根据定价模型，您可以通过提供API标识符、显示名称和描述信息来添加合同维度或使用维度。
5. 选择下一步，然后输入您的合同维度定价。
6. 选择提交，提交您的请求以供审核。
7. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

更新维度信息

您可以更改产品的维度信息。有关定价模式的更多信息，请参阅容器定价模式。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/ 并登录您的卖家账户。
2. 在服务器产品页面上，选择要修改的容器产品。
3. 从请求更改下拉列表中，选择更新定价维度，然后选择更新维度信息。
4. 您可以通过为要更新的维度的显示名称和描述提供信息来添加维度信息。
5. 选择提交，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

更新定价条款

您可以更改产品的定价条款。如果您更改销售产品的国家/地区，则可能需要更新定价条款。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/ 并登录您的卖家账户。
2. 在服务器产品页面上，选择要修改的容器产品。
3. 从 “请求更改” 列表中，选择 “更新公开报价”，然后选择 “更新定价条款”。
4. 当前定价已预先填充供您编辑。您可以删除当前定价并添加新价格。我们建议您在提交请求以供审核之前，先检查您请求的价格。
5. 选择提交，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

> **Note**
>
> 如果您使用更新定价条款提高某个维度的价格，则在 90 天内将无法更新定价。从您发起变更之日起，价格上涨将锁定 90 天。这种价格锁定仅对价格提高而非价格降低有效。

按国家/地区更新可用性

您可以更改销售和订阅产品的国家/地区。有关更多信息，请参阅 国家/地区。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/ 并登录您的卖家账户。
2. 在服务器产品页面上，选择要修改的容器产品。
3. 从请求更改下拉列表中，选择更新公开优惠，然后选择按国家/地区更新可用性。
4. 选择下列选项之一：
   - 所有国家/地区 – 适用于所有受支持的国家/地区。
   - 所有国家/地区（部分除外）– 在所有受支持的国家/地区均可用，部分国家/地区除外。
   - 仅允许列表中的国家/地区 – 买家只能在您指定的国家/地区购买。
5. 选择提交，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

更新最终用户许可协议

您可以更新最终用户许可协议 (EULA) 以使用的标准合同 AWS Marketplace或自定义EULA。所做的更新对订阅您的产品和产品续订的新用户生效。EULA

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/ 并登录您的卖家账户。
2. 在服务器产品页面上，选择要修改的容器产品。
3. 从 “请求更改” 下拉列表中，选择 “更新公开报价”，然后选择 “更新” EULA。
4. 选择 AWSMarketplace 的标准合同或提交您的自定义合同EULA。对于定制EULA，您必须提供来自 Amazon S3 存储桶的合同。
5. 选择提交，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

更新产品的退款政策

您可以更新产品的退款政策。退款政策的更新对所有用户生效。有关更多信息，请参阅 商品退款在 AWS Marketplace。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/ 并登录您的卖家账户。
2. 在服务器产品页面上，选择要修改的容器产品。

3. 从请求更改下拉列表中，选择更新公开优惠，然后选择更新退款政策。
4. 当前的退款政策已预先填充供您编辑。您可以删除当前的退款政策并添加新的退款政策。我们建议您在提交请求以供审核之前，先检查您请求的退款政策。提交请求会改写当前的退款政策。
5. 选择提交，提交您的请求以供审核。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

为您的容器产品创建产品 ID 和产品代码

要开始使用容器产品，您必须在中创建产品编码和产品代码记录 AWS Marketplace。产品 ID 用于在产品的整个生命周期中对其进行跟踪。

使用以下步骤在中创建新的容器产品并生成产品 ID。 AWS Marketplace 管理门户

> **Note**
>
> 此过程还会为您的容器创建与您的产品配对的公钥。

要创建容器产品 ID，请执行以下操作：

1. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
2. 从菜单栏中选择产品，然后选择服务器。
3. 选择创建服务器产品，然后选择容器。
4. 生成容器产品 ID 和代码。

   > **Note**
   >
   > （可选）您可以为产品添加标签以获得基于标签的授权。有关更多信息，请参阅为AWS资源添加标签。

5. 选择继续以继续创建产品。

创建初始列表

生成产品 ID、产品代码和公钥后，您将使用向导创建初始列表。

1. 为您的产品列表提供产品信息。

2. 确定产品的定价模式。

> **Note**
> 有关更多信息，请参阅容器产品定价。

> **Note**
> 对于付费产品，起价为 0.01 美元，这样您和 AWS Marketplace 卖家运营团队测试产品时就不会产生高昂的成本。产品公开上架时请提供实际价格。

3. 提供其他优惠信息，包括退款政策以及EULA报价供应情况。
4. 为您的容器产品添加初始存储库。
5. 在最后一步选择提交，将产品移至“受限可见性”。

> **Note**
> 您的容器产品最初是使用占位符版本创建的。当产品的可见性受限时，您将添加最终版本。

AWS Marketplace Metering Service 为您的容器产品集成

对于按用量定价的基于容器的产品，您可以使用 AWS Marketplace Metering Service 来检查使用产品的权利以及对用量进行计量以进行计费。您必须根据在设置定价信息时创建的定价模式进行计量。有关更多信息，请参阅 采用 AWS Marketplace Metering Service按小时计量和自定义计量。

AWS License Manager 为您的容器产品集成

对于采用合同定价的基于容器的产品，您可以使用 AWS License Manager 将许可证与您的产品相关联。

有关与集成的更多信息 AWS License Manager，请参阅集装箱产品的合同定价 AWS License Manager。

添加产品的新版本

您的产品在其生命周期内可能有多个版本。每个版本都有一组特定于该版本的容器映像。

> **Note**
>
> 只有为产品创建了产品 ID 和定价，才能向产品添加版本。有关这些步骤的更多信息，请参阅为您的容器产品创建产品 ID 和产品代码。

创建产品版本涉及以下步骤：

主题

- 步骤 1：添加存储库
- 第 2 步：将容器映像和构件上传到存储库
- 步骤 3：向容器产品添加新版本

您的产品容器映像和其他构件存储在 AWS Marketplace中的存储库中。通常，您可以为所需的每个构件创建一个存储库，但该存储库可以存储该构件的多个版本（使用不同的标签）。

> **Note**
>
> 产品部署中的所有映像都必须使用 AWS Marketplace 存储库中的映像。

步骤 1：添加存储库

以下过程介绍如何在中添加任何所需的存储库 AWS Marketplace。

要添加存储库，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 从产品菜单中选择服务器。
3. 在服务器产品选项卡上，选择要修改的产品，然后从请求更改下拉列表中选择添加存储库。
4. 输入您要创建的存储库的名称。如果要创建多个新存储库，请为每个额外的存储库选择添加新存储库，并为其指定一个唯一的名称。

   > **Note**
   >
   > 存储库将具有以下结构：`<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>`. 当您

将项目添加到存储库时（在以下过程中），它们将获得一个标签并具有以下结构：`<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>:<tag>`。`repositoryID`是的内部 ID AWS Marketplace。`sellerName` 是基于您为卖家账户创建的名称。您将在此步骤中定义 `respositoryName`。`tag` 是在将构件上传到存储库时设置的。

5. 选择提交。

> **Note**
>
> 对于每个产品，您最多可以有 50 个存储库。

新请求已创建并显示在请求选项卡上。完成后，您可以在几分钟之内开始向已创建的存储库中添加容器映像和其他构件。

第 2 步：将容器映像和构件上传到存储库

要将容器映像和构件上传到存储库，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 从产品菜单中选择服务器。
3. 在服务器产品选项卡上，选择要修改的产品。
4. 从请求更改下拉列表中，选择添加存储库。
5. 选择查看现有存储库。
6. 选择要上传到的存储库。
7. 选择 “查看推送命令” 以打开指令列表，包括可用于推送的命令 Docker 容器镜像和 Helm 该存储库的图表。

   有关如何将容器映像和其他构件推送到存储库的一般信息，请参阅《Amazon Elastic Container Registry 用户指南》中的推送映像。

   > **Note**
   >
   > 在调用`docker pull`或时，您可以使用以下亚马逊弹性容器注册表 (AmazonECR) API 操作`docker push`：

- `DescribeImages` – 用于查看存储库中有关映像的元数据。
- `GetAuthorizationToken` – 用于在将构件上传到存储库之前进行身份验证，然后使用 `docker pull` 或 `docker push` 命令。
- `ListImages` – 用于查看您推送的映像列表。

8. 使用列出的命令将任何需要的工件从本地存储库推送到产品的 AWS Marketplace 存储库。

> **Note**
>
> 您在 `push` 命令中提供的 **`tag`** 用于区分要上传到存储库的构件的版本。使用对构件所属版本有意义的标签。

9. 对版本中需要的每个容器映像或构件重复此操作。

> **Note**
>
> 您的版本在每个交付选项中最多可以包含 50 个容器映像或构件。有关交付选项的更多信息，请参阅以下过程。

上传构件后，您就可以创建产品版本了。

> **Note**
>
> 系统会自动扫描您的容器映像，以查看它们是否符合基于容器的产品要求。有关更多信息，请参阅容器产品扫描是否存在安全问题。

添加新的交付选项

您的集装箱商品的每个版本都需要一个配送选项。配送选项指定了可供买方使用的部署选项。根据以下交付选项之一，您需要将相应的构件上传到存储库中。

- 对于容器镜像传送选项，请将安装产品所需的所有容器镜像上传到 AWS Marketplace 控制台中创建的亚马逊弹性容器注册表 (AmazonECR) 存储库中。
- 对于 Helm 图表交付选项，上传 Helm 图表和容器图像存入 AWS Marketplace 控制台中创建的 Amazon ECR 存储库。

- 要获取 A mazon EKS 控制台附加组件配送选项，请上传 Helm 图表和容器图像存入 AWS Marketplace 控制台中创建的 Amazon ECR 存储库。

步骤 3：向容器产品添加新版本

> **Note**
>
> 如果您在向容器添加新版本时收到任何错误，请参阅AWS Marketplace Catalog API 参考资料中的添加新版本异步错误表。

要向容器产品添加新版本，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 从产品菜单中选择服务器。
3. 在服务器产品选项卡上，选择要向其添加版本的产品。然后，从请求更改下拉列表中选择添加新版本。
4. 在添加新版本页面上，输入版本标题和发布说明。
5. 输入版本详细信息后，下一步是添加交付选项。交付选项是一组说明和信息，买家可以使用这些说明和信息从您的产品版本启动软件。交付选项被称为买家的交付选项。

   > **Note**
   >
   > 您的产品可以支持具有不同容器镜像的多个平台（例如 Kubernetes 以及 Ubuntu 部署）。您可以为客户设置产品的每种方式创建一个配送选项，每个产品版本最多可以创建四个配送选项。

   a. 如果产品在其他版本中已有交付选项，则可以使用现有选项作为模板，向新版本添加交付选项。在交付选项中，从列表中选择要添加的交付选项。您可以按照以下步骤中的说明编辑该选项。

   b. 要添加新的交付选项，请选择新交付选项。添加选项后，按照以下步骤中的说明，对其进行配置。

6. 为交付选项选择一种交付方式。交付方式决定了买家将如何启动您的软件。

- 对于容器镜像传送选项，请在 AWS Marketplace 控制台中创建的亚马逊弹性容器注册表 (AmazonECR) 存储库中提供容器镜像的路径。买家使用容器映像路径通过将映像直接拉入其环境来启动软件。
- 对于 Helm 图表交付选项，提供路径 Helm 在 AWS Marketplace 控制台中创建的 Amazon ECR 存储库中的图表。买家安装 Helm 在其部署环境中使用图表来启动软件。
- 要获取 A mazon EKS 控制台附加组件交付选项，请提供以下路径 Helm 在 AWS Marketplace 控制台中创建的 Amazon ECR 存储库中的图表。买家使用亚马逊EKS控制台或亚马逊原生EKS插件安装容器APIs来启动软件。有关更多信息，请参阅亚马逊提供的亚马逊EKSEKS插件。

a. 要添加容器映像交付选项，请执行以下步骤：

   i. 在容器镜像中，将 Amazon ECR URL 添加到包含产品版本软件的容器镜像中。

   ii. 在交付选项标题和部署选项描述中，输入此交付选项的标题和描述。

   iii. 在使用说明中，输入详细信息，以帮助买家在启动软件后使用该软件。

   iv. 在支持的服务中，选择买家可以在其中启动软件的环境。

   v. 在部署模板中，添加买家可以用来启动软件的资源。为每个模板输入标题和资源。URL

b. 要添加 Helm 图表传送选项，请执行以下步骤：

   i. 中Helm 图表，将 Amazon 添加ECRURL到 Helm 购买者将在部署环境中安装的图表，以启动您的软件。

   ii. 在容器镜像中，将 Amazon ECR URL 添加到包含产品版本软件的容器镜像中。

   iii. 在交付选项标题和部署选项描述中，输入此交付选项的标题和描述。

   iv. 在使用说明中，输入详细信息，以帮助买家在启动软件后使用该软件。

   v. 在支持的服务中，选择买家可以在其中启动软件的环境。

   vi. 可选-输入 Helm 版本名称，输入版本的名称 Kubernetes 命名空间其中 Helm 图表将被安装。

   vii. 可选-输入 Helm 安装命名空间，输入名称 Helm `helm install`命令将使用的版本。

   viii. 可选-输入 Kubernetes 服务帐户名称，输入名称 Kubernetes 将用于连接到 AWS Identity and Access Management (IAM) 的服务帐户。这些区域有：Kubernetes 服务帐户调用诸如许可或计量之类的 AWS 服务。

   ix. 选择在此产品版本QuickLaunch上启用。 QuickLaunch 是中的一项功能 AWS Marketplace。买家可以使用 QuickLaunch 快速创建 Amazon EKS 集群，然后使用在

集群上启动您的软件 AWS CloudFormation。有关更多信息，请参阅QuickLaunch 中的 AWS Marketplace。

x. 在覆盖参数中，输入将用于 Helm CLI启动软件的命令。这些参数允许买家覆盖提供的默认值。如果已启用 QuickLaunch，还要为 CloudFormation表单输入参数名称和描述。使用 AWS Marketplace 管理控制台时，参数限制为 15 个，但使用时没有限制 AWS Marketplace Catalog API。有关更多信息，请参阅向基于容器的产品添加新版本。

> **Note**
>
> 一些覆盖参数是必需的。Amazon EKS Anywhere 产品要求许可证密钥`DefaultValue`的覆盖参数为`"${AWSMP_LICENSE_SECRET}"`。对于付费产品，您必须为 `"${AWSMP_SERVICE_ACCOUNT}"` 设置为 `DefaultValue` 的服务账号配置覆盖参数。

xi. 选择 “隐藏密码和机密” 以屏蔽控制台、命令行工具和中的敏感信息APIs。有关更多信息，请参阅《AWS CloudFormation 用户指南》中参数中的 `NoEcho` 参数文档。

c. 要添加 A mazon EKS 控制台附加组件交付选项，请确保构件符合亚马逊 EKS 附加商品的要求，然后执行以下步骤：

> **Note**
>
> 每个版本仅支持一个 Amazon EKS 附加组件配送选项。在亚马逊EKS控制台上发布您正在使用的当前版本之前，您无法添加新版本。

i. 中Helm 图表，将 Amazon 添加ECRURL到 Helm 购买者将在部署环境中安装的图表，以启动您的软件。

ii. 在容器镜像中，将 Amazon ECR URL 添加到包含产品版本软件的容器镜像中。确保所有图像都在 Helm 图表已列出。

iii. 在交付选项标题和部署选项描述中，输入此交付选项的标题和描述。

iv. 在 “可见性” 中，保持 “受限” 的默认值处于选中状态。

v. 在插件名称中，输入此插件的唯一名称。在亚马逊EKS控制台中显示时，您输入的附加组件名称将与卖家名称一起显示。

vi. 在附加组件版本中，输入安装或升级此插件时将显示的插件版本。遵循格式`major.minor.patch`。

vii. 在 “插件类型” 中，从下拉列表中为您的插件选择一个类别。

viii. 中Kubernetes 版本，选择所有 Kubernetes 您的附加组件将支持的版本。

ix. 在架构中，选择您的插件支持的平台架构。选项为AMD64和ARM64。我们建议同时支持这两种架构，以最大限度地提高兼容性。如果您的附加组件不支持ARM64设备，则必须指定添加支持的计划日期，然后才能在所有广告中发布您的产品 AWS 区域。

x. 在命名空间中，输入唯一的 Kubernetes 您的插件将安装到的命名空间。安装第三方插件时不支持`defaultkube-system`、和`kube-public`命名空间。

xi. 在环境覆盖参数中，您最多可以从 Amazon EKS 插件框架中选择 2 个环境参数。您可以将参数名称从 values.yaml 映射到这些环境变量，即和。`${AWS_REGION}` `${AWS_EKS_CLUSTER_NAME}`

7. 要添加其他配送选项，请选择 “新建配送选项”，然后重复前面步骤中的说明进行配置。

8. 选择提交。

测试并发布您的产品

本节提供有关发布容器产品版本后的后续步骤的指导。它概述了测试和向公众发布产品所需的具体步骤和流程。

容器镜像和 Helm 图表交付选项

本节提供有关发布容器镜像的指导以及 Helm 图表。

您对新版本的请求已创建并应在几分钟内完成。您可以从服务器产品页面的请求选项卡中跟踪请求。如果您在测试或发布插件时遇到任何错误，请参阅参考资料中添加新版本中的 Aynchronous 错误表。AWS Marketplace Catalog API

> **Note**
>
> 如果您的产品当前设置为限量供应，则只有该产品可供购买的买家才能访问该产品版本。如果您的产品目前已设置为公开发售，则所有 AWS Marketplace 买家都可以访问该产品版本。

如果这是您的第一个版本集，那么您的产品现在可以发布了。

Amazon EKS 附加配送选项

本节提供有关测试和发布您的 Amazon EKS 附加组件的指导。

测试您的附加组件

- 提交附加组件后， AWS Marketplace 处理您的请求并在受限状态下发布您的附加组件，以便您在亚马逊EKS附加组件目录中进行验证。您可以从中服务器产品页面的请求选项卡中跟踪请求 AWS Marketplace 管理门户。摄取时间将从 5-10 个工作日不等，具体取决于我们正在处理的请求量。

  当您的请求处于 “正在审核” 状态时， AWS 团队仍在将该附加组件发布 AWS Marketplace 到亚马逊EKS附加组件目录中。插件发布到受限状态后，请求状态更改为 “成功”。之后，你可以开始测试你的附加组件。
- 您的附加组件上市后，您可以在亚太地区（首尔）地区找到它进行测试。 AWS Marketplace 依靠您的专业知识来验证软件的功能。要测试您的附加组件，您必须在亚太地区（首尔）地区的卖家账户中创建一个亚马逊EKS集群，您的插件已列入许可名单。要测试您的附加组件，请按照以下详细说明进行操作。请务必在每个上进行测试 Kubernetes 您的软件支持的版本。
- 如果您要提供付费产品，请向以下内部人员创建私人报价 AWS 账户。这些账户有助于将您的软件集成到所有商业版的 Amazon EKS 控制台中 AWS 区域。

```
288092140294, 288092140294, 408202761791
```

- 在您的插件版本获得 AWS Marketplace 批准并将您的插件版本移至公开之前，请保持您的测试集群处于活动状态。

> **Note**
>
> AWS Marketplace 将不承担在 Amazon 集EKS群上测试您的容器产品期间产生的 AWS 基础设施成本。在我们验证测试结果的同时，您可以遵循正确的规模调整机制，将节点降至最低的运营成本。

向公众发布你的附加组件

在您通过 Amazon EKS 集群验证您的软件作为附加组件后，您可以使用AWS Marketplace 管理门户或提交向公众发布您的亚马逊EKS附加组件版本的请求 AWS Marketplace Catalog API。

有关更多信息，请参阅AWS Marketplace Catalog API 参考资料中的更新亚马逊EKS附加组件的可见性。

您可以从中服务器产品页面的请求选项卡中跟踪请求 AWS Marketplace 管理门户。摄取时间会有所不同。

更新版本信息

创建版本后，可通过修改与版本关联的信息，向买家提供更新的信息。例如，如果您计划在版本 1.1 发布后限制版本 1.0，则可以更新版本 1.0 的描述以引导买家使用版本 1.1。提供版本 1.0 的受限日期。您可以从 AWS Marketplace 管理门户中更新版本信息。

要更新版本信息，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 从产品菜单中选择服务器。
3. 在服务器产品选项卡上，选择要修改的产品。
4. 从请求更改下拉列表中，选择更新版本信息。
5. 在更新版本页面上，选择要更新的版本。
6. 对所选版本进行更新。可供更新的字段取决于产品版本或交付选项的状态。
   a. 对于所有版本，您可以更新发布说明。
   b. 对于尚未公开发布的版本，您可以更新版本标题。
   c. 对于未受限制的交付选项，您可以更新以下字段：
      - 描述
      - 使用说明
      - 支持的服务
   d. 对于尚未公开发布的版本中的交付选项，您可以更新以下字段：
      - 交付选项标题
      - Helm 图表（适用于 Helm 仅限图表交付选项）
      - 容器映像
      - 部署资源
      - AddOn 姓名
      - AddOn 版本
      - AddOn 类型
      - Helm 图表 URI
      - CompatibleKubernetesVersions
      - SupportedArchitectures

- 命名空间
- EnvironmentOverrideParameters

e. 对于公开版本中的配送选项，您可以更新SupportedArchitectures。

7. 选择提交。
8. 验证请求是否显示在请求选项卡上，状态为正在审核。

您可以随时从服务器产品页面的请求选项卡中查看请求的状态。

限制您的 Amazon EKS 附加组件的某个版本

要限制作为附加组件发布的集装箱产品版本，请使用底部的 “联系我们” 表单与 AWS Marketplace 运营团队联系AWS Marketplace 管理门户。

为容器产品创建或更新产品信息

创建产品 ID 并设置定价后，您可以编辑您的产品信息，包括买家将在 AWS Marketplace中看到的有关您的容器产品的内容。以下步骤概述了如何为您的产品创建产品详细信息。

要为您的容器产品创建或更新产品详细信息，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 从产品菜单中选择服务器。
3. 在服务器产品选项卡上，选择要修改的产品。
4. 从请求更改下拉列表中，选择更新产品信息。
5. 更新以下任何您想要更改的字段：

   - 产品标题
   - SKU
   - 简短描述
   - 详细描述
   - 产品徽标图片 URL
   - 要点
   - 产品类别
   - 关键词

- 产品视频 URL
- 资源
- 支持信息

> **Note**
>
> 图片URLs必须位于可公开访问的 Amazon S3 存储桶中。有关徽标格式的更多详细信息，请参阅公司和产品徽标要求。

6. 选择提交。
7. 验证请求是否显示在请求选项卡上，状态为正在审核。您可能需要刷新页面才能在列表中看到请求。

您可以随时从服务器产品页面的请求选项卡中查看请求的状态。

容器产品扫描是否存在安全问题

当您创建更改请求以向容器产品添加新版本时，我们会扫描该新版本中包含的容器映像并检查是否存在安全漏洞。为此，我们对图像执行 layer-by-layer 静态扫描。如果我们发现具有可远程利用风险的关键漏洞，我们将提供已发现的问题列表。我们强烈建议您使用容器图像扫描器进行自己的安全分析，例如 Clair, Twistlock, Aqua Security，或 Trend Micro 以避免摄取和发布过程的延迟。

您为构建容器映像所选择的基本映像可对最终映像的安全配置文件产生重大影响。如果您所选取的基本映像已具有已知关键漏洞，则会因基本层而标记它们，即使应用程序软件层是干净的也是如此。我们建议您在构建镜像并将其提交给之前，先确认自己从没有漏洞的基础容器开始 AWS Marketplace。

基于容器的产品要求

AWS Marketplace 对所有基于容器的产品和产品保持以下要求。 AWS Marketplace这些要求有助于为我们的客户推广安全、可靠和值得信赖的目录。我们还鼓励卖家审查其他控制措施和协议的实施情况，以满足其特定产品的需求。

所有产品及其相关元数据在提交时都要经过审核，以确保它们符合或超过当前 AWS Marketplace 要求。我们会审查并调整这些政策，以满足我们不断变化的安全和其他使用要求。 AWS Marketplace 持续验证现有产品是否继续满足这些要求的任何更改。如果产品不合规， AWS Marketplace 将与您联系以更新您的产品。在某些情况下，您的产品可能暂时无法向新订阅用户提供，直到问题得到解决。

安全要求

所有基于容器的产品必须满足以下安全要求：

- Docker 容器映像必须不含任何已知的恶意软件、病毒或漏洞。当您向容器产品添加新版本时，将扫描该版本中包含的容器映像。
- 如果您的基于容器的产品需要访问权限来管理 AWS 资源，则必须通过服务账户的 IAM 角色（如果通过亚马逊 Elastic Kubernetes Service (Amazon EKS) 运行）或任务的 IAM 角色（如果通过亚马逊弹性容器服务 (Amazon ECS) 运行）来实现访问权限，而不是向用户请求访问密钥。
- 基于容器的产品只需要最低权限即可运行。有关更多信息，请参阅 ECS 安全和 EKS 安全。
- 默认情况下，应将容器映像配置为以非根权限运行。

访问要求

所有基于容器的产品必须满足以下访问要求：

- 基于容器的产品必须使用初始随机密码。基于容器的产品不得使用初始的固定密码或空白密码进行外部管理访问（例如，通过 Web 界面登录应用程序）。在允许买家设置或更改自己的凭证之前，必须提示买家输入该随机密码。
- 对应用程序的任何外部访问都必须得到客户的明确同意和支持。

客户信息要求

所有基于容器的产品必须满足以下客户信息要求：

- 除非 BYOL（自带许可）另行要求，否则软件不得在客户不知晓且未明确同意的情况下收集或导出客户数据。收集或导出客户数据的应用程序必须遵循以下准则：
  - 客户数据收集必须是自助服务、自动化和安全的。买家无需等待卖家批准即可部署软件。
  - 必须在列表的描述或使用说明中明确说明对买家数据的要求。这包括收集的内容、客户数据的存储位置以及如何使用这些数据。例如，此产品会收集您的姓名和电子邮件地址。此信息由 <company name> 发送并由其存储。此信息将仅用于就 <product name> 问题联系买家。
  - 不得收集付款信息。

产品使用要求

所有基于容器的产品必须满足以下产品使用要求：

- 卖家只能发布功能齐全的产品。不允许使用用于试用或评估目的的测试版或预发行产品。如果卖家在提供免费版本后的 90 天 AWS Marketplace 内提供同等的付费版本，则支持商业软件的开发者、社区和 BYOL 版本。
- 基于容器的产品的所有使用说明都必须包括部署基于容器的产品的所有步骤。使用说明必须提供指向AWS Marketplace上相应容器映像的命令和部署资源。
- 基于容器的产品必须包含订阅用户使用该软件所需的所有容器映像。此外，基于容器的产品不得要求用户使用外部的任何映像 AWS Marketplace （例如，来自第三方存储库的容器映像）启动产品。
- 容器及其软件必须可以自助方式部署，并且不得要求额外的付款方式或费用。部署时需要外部依赖的应用程序必须遵循以下准则：
  - 必须在列表的描述或使用说明中披露该要求。例如，此产品需要互联网连接才能正确部署。部署时会下载以下软件包：<list of package>。
  - 卖家需对所有外部依赖的使用负责，并确保其可用性和安全性。
  - 如果外部依赖关系不再可用，则还必须从中 AWS Marketplace 删除该产品。
  - 外部依赖项不得要求额外的付款方式或费用。
- 需要持续连接到不受买家直接控制的外部资源（例如外部 API 或由卖家或第三方管理的 AWS 服务 ）的容器必须遵循以下准则：
  - 必须在列表的描述或使用说明中披露该要求。例如，此产品需要持续的互联网连接。需要以下持续的外部服务才能正常运行：<list of resources>。
  - 卖家需对所有外部资源的使用负责，并确保其可用性和安全性。
  - 如果外部资源不再可用，则还必须从中 AWS Marketplace 移除该产品。
  - 外部资源不得要求额外的付款方式或费用，并且必须自动设置连接。
- 产品软件和元数据不得包含将用户重定向到 AWS Marketplace中未提供的其他云平台、其他产品或追加销售服务的语言。
- 如果您的产品是其他产品或其他 ISV 产品的附加组件，则您的产品描述必须表明它扩展了其他产品的功能，如果没有它，产品的应用将受到限制。例如，本产品扩展了 <product name> 的功能，如果没有它，则该产品的应用将受到限制。请注意，<product name> 可能需要自己的许可才能使用此列表的全部功能。

架构要求

所有基于容器的产品必须满足以下架构要求：

- 的源容器镜像 AWS Marketplace 必须推送到所拥有的亚马逊弹性容器注册表 (Amazon ECR) Container Registry 存储库。 AWS Marketplace您可以在 AWS Marketplace 管理门户 中的服务器产品下为每个容器产品列表创建这些存储库。
- 容器映像必须基于 Linux。
- 付费版基于容器的产品必须能够部署在 Amazon ECS、Amazon EKS 或 AWS Fargate 中。
- 具有合同定价和集成的基于容器的付费产品 AWS License Manager 应部署在亚马逊 EKS、Amazon ECS、Amazon EKS Anywhere AWS Fargate、Amazon ECS Anywhere、Amazon ECS Anywhere、红帽 OpenShift 服务 AWS (ROSA)、本地自行管理的 Kubernetes 集群或亚马逊弹性计算云上。

容器产品使用说明

在为容器产品创建使用说明时，请按照the section called “AMI和容器产品使用说明”中的步骤和指导进行操作。

亚马逊 EKS 附加商品的要求

Amazon EKS 附加组件是一种为Kubernetes应用程序提供操作功能但不是特定于应用程序的软件。例如，Amazon EKS 附加组件包括可观察性代理或Kubernetes驱动程序，允许集群与用于联网、计算和存储的底层 AWS 资源进行交互。

作为集装箱产品的卖家，您可以从多种部署选项中进行选择，包括 Amazon EKS。您可以将产品版本作为 AWS Marketplace 附加组件发布到 Amazon EKS 附加组件目录中。您的附加组件显示在 Amazon EKS 控制台中，旁边是其他供应商维护 AWS 的插件。您的买家可以像使用其他插件一样轻松地将您的软件部署为附加组件。

有关更多信息，请参阅《Amazon EKS 用户指南》中的 Amazon EKS 附加组件。

为将容器产品作为 AWS Marketplace 附加组件做好准备

要将您的容器产品作为 AWS Marketplace 插件发布，它必须满足以下要求：

- 您的容器商品必须发布在 AWS Marketplace。
- 您的容器产品必须兼容 AMD64 和 ARM64 架构。
- 您的容器产品不得使用自带许可 (BYOL) 定价模式。

> **Note**
>
> 亚马逊 EKS 附加组件配送不支持 BYOL。

- 您必须遵守所有基于容器的产品要求，包括将所有容器图像和Helm图表推送到托管的 AWS Marketplace Amazon ECR 存储库。例如，此要求包括开源图像`nginx`。图像和图表不能托管在其他外部存储库中，包括但不限于 Amazon ECR 公开映像浏览馆、Docker Hub 和 Quay。
- Helm图表-准备好要通过Helm图表部署的软件。Amazon EKS 附加框架可将Helm图表转换为清单。Amazon EKS 系统不支持某些Helm功能。以下列表描述了入职前必须满足的要求。在此列表中，所有Helm命令都使用Helm版本 3.8.1：
  - 支持所有`Capabilities`对象，但有例外`.APIVersions`。 `.APIVersions`不支持 non-built-in 自定义 Kubernetes API。
  - 仅支持`Release.Name`和`Release.Namespace`对象。
  - Helm不支持钩子和`lookup`函数。
  - 所有依赖图表都必须位于主Helm图表中（使用存储库路径文件指定：//...）。
  - Helm图表必须成功通过 Helm Lint 和 T Helm emplate 且没有错误。命令如下：
    - Helm棉绒 — `helm lint helm-chart`

      常见问题包括父图表元数据中未声明的图表。例如，`chart metadata is missing these dependencies: chart-base Error: 1 chart(s) linted, 1 chart(s) failed`
    - Helm模板 — `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`

      使用该标志传递任何被覆盖的`–f`配置。
  - 将所有容器二进制文件存储在 AWS Marketplace Amazon ECR 存储库中。要创建清单，请使用前面显示的Helm模板命令。在清单中搜索任何外部图像引用，例如`busybox`或`gcr`图像。使用请求下拉列表中的 “添加存储库” 选项创建的 AWS Marketplace Amazon ECR 存储库将所有容器映像和依赖项上传到创建的 Amazon ECR 存储库中。
- 自定义配置-您可以在部署期间添加自定义变量。有关如何识别最终用户体验、命名软件以及如何使用Helm图表打包成包装的信息`aws_mp_configuration_schema.json`，请参阅 Amazon EKS 附加组件：高级配置。

  根据 “$schema” 关键字，`$schema`必须是指向有效`application/schema+json`资源的 URI。

此文件不得接受任何敏感信息，例如密码、许可证密钥和证书。

要处理密钥和证书安装，您可以向最终用户提供插件安装后或预安装步骤。该产品不应依赖任何外部许可证。该产品应基于 AWS Marketplace 权利运行。

有关限制的更多信息`aws_mp_configuration_schema.json`，请参阅附加组件提供商的插件配置要求和最佳实践。

- 确定并创建要部署软件的命名空间 — 在产品的第一个版本中，必须通过添加模板化命名空间来标识要部署软件的命名空间。
- **`serviceAccount`**如果适用，请创建 — 如果该软件是付费软件 AWS Marketplace 或者必须与其他软件连接 AWS 服务，请确保Helm图表是`serviceAccount`默认创建的。如果`serviceAccount`创建是由`values.yaml`文件中的参数处理的，请将参数值设置为`true`。例如，`serviceAccount.create = true`。这是必需的，因为客户可能会选择通过继承已具有所需权限的底层节点实例的权限来安装插件。如果 Helm 图表未创建`serviceAccount`，则无法将权限绑定到`serviceAccount`。
- 可追踪部署或守护程序集 — 确保你的 Helm 图表有守护程序集或部署。Amazon EKS 插件框架使用它们跟踪您的 Amazon EKS 资源的部署。如果没有可追踪的部署或守护程序集，您的插件将面临部署错误。如果您的插件没有部署或守护程序集，例如，如果您的插件部署了一堆无法追踪的自定义资源或 Kubernetes 作业，请添加虚拟部署或守护程序集对象。
- 支持 AMD 和 ARM 架构 — 如今，许多 Amazon EKS 客户使用 ARM64 来使用 AWS Graviton 实例。第三方软件必须同时支持这两种架构。
- 与来自的许可或计量 API 集成 AWS Marketplace — AWS Marketplace 支持多种计费模式。有关更多信息，请参阅 容器产品计费、计量和许可集成。如果您想通过即付即付机制销售产品，请参阅使用 AWS Marketplace 计量服务为集装箱产品定制计量。如果您想通过预付模式或合同模式销售产品，请参阅集装箱产品的合同定价 AWS License Manager。
- 上传软件以及所有工件和依赖项 — Helm 图表必须是独立的，并且不得要求外部来源的依赖关系，例如，GitHub。如果软件需要外部依赖项，则必须将依赖项推送到同一 AWS Marketplace 列表下的 AWS Marketplace 私有 Amazon ECR 存储库。
- 在您的网站上提供部署说明 — 我们要求您为客户提供部署指南，以确定如何通过 create-add on 命令部署软件。
- IAM 角色 — 列出您的软件运行或与其他角色连接所需的所有 AWS Identity and Access Management (IAM) 策略 AWS 服务。
- 版本更新 — Amazon EKS 会在上游版本发布几周后发布新的 Kubernetes 版本。随着新的 Amazon EKS 集群版本正式上市，供应商有 45 天的时间来认证或更新其软件，使其与新的 Amazon EKS 集

群版本兼容。如果您当前版本的插件支持新的 Kubernetes 版本，请对其进行验证和认证，以便我们更新版本兼容性矩阵。如果需要新的插件版本来支持新的 Kubernetes 版本发布，请提交新版本进行入门。

- 合作伙伴的软件必须属于以下类型之一，或者是能够增强 Kubernetes 或 Amazon EKS 的操作软件：Gitops | 监控 | 日志 | 证书管理 | 策略管理 | 成本管理 | 自动扩展 | 存储 | kubernetes 管理 | 服务网格 | etcd-backup | | 负载均衡器 | 本地注册表| 网络 | 安全 | 备份 | 入口控制器 | 可观察性 ingress-service-type
- 软件不能是容器网络接口 (CNI)。
- 软件必须通过付费产品的许可 AWS Marketplace 和计量 API 进行销售并与之集成。不接受 BYOL 产品。

附加组件提供商的插件配置要求和最佳实践

Amazon EKS 需要插件提供商将其配置为 Helm JSON 架构字符串。需要必需配置或允许可选配置的插件必须包含已提交 Helm Chart 的`aws_mp_configuration_schema.json`文件。 AWS Marketplace Amazon EKS 将使用此架构来验证客户的配置输入，并拒绝输入值不符合架构的 API 调用。附加配置通常分为两类：

- 配置 Kubernetes 的常规 Kubernetes 属性，例如标签、容差、NodeSelector 等。
- 特定于附加组件的配置，例如许可证密钥、功能启用、URL 等。

本节重点介绍与常规 Kubernetes 属性相关的第一类。

Amazon EKS 建议遵循有关亚马逊 EKS 插件配置的最佳实践。

- 架构要求
- 允许配置的常用参数
- 不允许配置的常用参数

架构要求

定义 json 架构时，请确保您使用的是 Amazon EKS 插件支持的 jsonschema 版本。

支持的架构列表：

- 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 架构版本都与 Amazon EKS 插件不兼容，并且会导致该插件在修复之前无法发布。

Helm 架构文件示例

```
{
"$schema": "http://json-schema.org/schema#",
  "type": "object",
  "properties": {
"podAnnotations": {
"description": "Pod Annotations"
"type": "object"
    },
    "podLabels": {
"description": "Pod Labels"
"type": "string"
    },
    "resources": {
"type": "object"
"description": "Resources"
    },
    "logLevel": {
"description": "Logging Level"
"type": "string",
      "enum": [
        "info",
        "debug"
      ]
    },
    "config": {
"description": "Custom Configuration"
"type": "object"
    }
  }
}
```

camelCase

配置参数必须是 camelCase，如果不符合此格式，则会被拒绝。

描述为必填项

务必包括对架构属性的有意义的描述。此描述将用于在 Amazon EKS 控制台中为每个配置参数呈现标签名称。

RBAC 的定义

插件提供商需要使用最小权限原则定义和提供成功安装插件所需的 RBAC 权限。如果需要针对新版本的附加组件或任何修复 CVE 而更改 RBAC 权限，则插件提供商需要将此更改告知 Amazon EKS 团队。每个 Kubernetes 资源所需的权限应仅限于对象的资源名称。

```
apiGroups: ["apps"]
resources: ["daemonsets"]
resourceNames: ["ebs-csi-node"]
verbs: ["create", "delete", "get", "list", "patch", "update", "watch"]
```

密钥管理

本节仅适用于需要客户配置秘密信息（如应用程序密钥、API 密钥、密码等）的插件。目前，由于存在安全隐患，Amazon EKS API 不支持以纯文本形式传递机密信息。但是，客户可以使用配置来传入包含插件所需密钥的 Kubernetes 密钥的名称。客户需要创建包含密钥的 Kubernetes Secret 对象，其命名空间与先决条件步骤相同，然后在创建插件时使用配置 blob 传入密钥的名称。我们建议插件提供商为架构属性命名，这样客户就不会意外将其误认为是实际密钥。例如：appSecretName， connectionSecretName 等等。

总而言之，插件提供商可以利用该架构来允许客户传入密钥的名称，但不允许传递实际保存机密本身的密钥。

配置值示例

您可以在架构中包含配置示例，以帮助客户配置插件。以下示例来自 AWS Distro 的 OpenTelemetry 插件架构。

```
"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": []
      }
    ]
```

允许配置的常用参数

以下是面向客户的 Helm 架构文件中的推荐参数。

参数	描述	应该有默认值吗？
其他标签	为插件管理的所有 Kubernetes 对象添加 Kubernetes 标签。	否
其他注释	为插件管理的所有 Kubernetes 对象添加 Kubernetes 注解。	否
播客标签	为插件管理的容器添加 Kubernetes 标签。	否
podanNotations	向插件管理的容器添加 Kubernetes 注释。	否
logLevel	插件管理的组件的日志级别。	是
节点选择器	推荐的最简单的节点选择约束形式。你可以将 nodeSelector 字段添加到你的 Pod 规格中，并指定你希望目标节点拥有的节点标签。	例如，可能仅限 Linux 节点

参数	描述	应该有默认值吗？
容忍度	容忍度适用于吊舱。容忍度允许调度器调度带有匹配污点的 Pod。容忍度允许调度，但不能保证安排时间。	也许，守护进程集更常见
亲和力	亲和性功能由两种类型的亲和性组成：节点亲和性函数，比如 NodeSelector 字段，但更具表现力，允许你指定软规则；Pod 间亲和性/反亲和性允许你限制 Pod 对抗其他 Pod 上的标签。	也许
拓扑 SpreadConstraints	您可以使用拓扑分布约束来控制 Pod 在集群中的故障域（例如区域、区域、节点和其他用户定义的拓扑域）之间的分布方式。这有助于实现高可用性以及高效的资源利用率。	也许
资源请求/限制	指定每个容器需要多少 CPU/内存。强烈建议设置请求。限制是可选的。	是
副本	插件管理的 Pod 的副本数量。不适用于守护程序集。	是

> **Note**
>
> 对于工作负载调度配置参数，您可能需要在必要时将架构中的顶级组件分开。例如，Amazon EBS CSI 驱动程序包含两个主要组件，即控制器和节点代理，客户对每个组件需要不同的节点选择器/容忍度。

> **Note**
>
> JSON 架构中定义的默认值仅用于用户文档目的，并不能取代在`values.yaml`文件中使用合法默认值的必要性。如果使用默认属性，请确保架构中的默认值与架构

中的默认值values.yaml相匹配，并且每当对 Helm Chart 进行更改时，两个工件（values.schema.json和values.yaml）都保持同步。

```
"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"
            ]
          }
```

不允许配置的常用参数

各种插件（例如 `clusterName` `region` `vpcIdaccountId`，Elastic Load Balancing Controller）可能需要集群元数据参数，例如、、、和其他参数。Amazon EKS 服务已知的任何与这些参数相似的参数都将由 Amazon EKS 插件自动注入，用户无需负责指定为配置选项。这些参数包括：

- AWS 区域
- 亚马逊 EKS 集群名称
- 集群的 VPC ID
- 容器注册表，专门用于 build-prod 账户，供网络插件使用
- DNS 集群 IP，专门用于 coreDNS 附加组件
- 亚马逊 EKS 集群 API 终端节点
- 在集群上启用了 IPv4
- 在集群上启用了 IPv6
- 在集群上启用 IPv6 的前缀委派

插件提供商需要确保您已为此类适用参数定义模板。上述每个参数都将有一个由 Amazon EKS 定义的预定义`parameterType`属性。版本元数据将指定`parameterType`与模板中参数的名称/路径之间的映射。这样，Amazon EKS 就可以动态传入这些值，而无需客户通过配置指定这些值，还可以让附加组件提供商灵活地定义自己的模板名称/路径。应从架构文件中排除 Amazon EKS 需要动态注入的上述参数。

来自发行版元数据的映射示例

```
"defaultConfiguration": [
```

```
        {
             "key": "image.containerRegistry",
             "parameterType": "CONTAINER_REGISTRY"
        }
]
```

不建议在面向客户的 Helm 架构文件中配置以下参数。要么参数应该具有不可修改的默认值，要么根本不包含在插件模板中。

参数	描述	应该有默认值吗？
映像	将部署在 Kubernetes 集群上的容器镜像。	否，通过插件定义进行管理
图片 PullSecrets	将 pod 配置为使用密钥从私有注册表中提取。	不适用
LivenessProbe	Kubelet 进程使用存活探针来知道何时重启容器。例如，liveness 探测器可能会陷入死锁，即应用程序正在运行，但无法取得进展。尽管存在错误，但在这种状态下重启容器有助于提高应用程序的可用性。	是
ReadinessProbe	对容器进行准备情况探测非常重要。这样，在你的数据平面上运行的 Kubelet 进程就会知道容器何时准备好为流量提供服务。当 Pod 的所有容器都准备就绪时，Pod 即被视为准备就绪。此信号的一个用途是控制哪些 Pod 用作服务的后端。当 Pod 未准备就绪时，它会从服务负载均衡器中移除。	是
StartupProbe	kubelet 使用启动探针来了解容器应用程序何时启动。如果配置了这样的探测器，它将禁用存活和就绪检查，直到成功为止，从而确保这些探测器不	可选

参数	描述	应该有默认值吗？
	会干扰应用程序的启动。这可以用来对启动缓慢的容器进行活性检查，避免它们在启动并运行之前被 kubelet 杀死。	
吊舱 DisruptionBudget	定义 Pod Discruption 预算 (PDB)，确保在自愿中断期间，保持最低数量的 POD 继续运行。PDB 限制了因自愿中断而同时关闭的复制应用程序的 Pod 数量。例如，基于法定人数的应用程序希望确保运行的副本数量永远不会低于法定人数所需的数量。Web 前端可能希望确保提供负载的副本数量永远不会低于总数的特定百分比。	是，如果默认为两个以上的副本
服务账户（姓名）	将在其下运行的服务帐号 pod 的名称。	是
服务账户（注释）	应用于服务帐号的注释。通常用于服务账户的 IAM 角色功能	不是，IAM 服务账户角色 ARN 是在顶级亚马逊 EKS 附加组件 API 中设置的。此规则的一个例外情况是，如果您的插件有多个部署/控制器（例如 Flux），并且需要单独的 IRSA 角色 ARN。

参数	描述	应该有默认值吗？
优先级 ClassName	优先级表示一个 Pod 相对于其他 Pod 的重要性。如果无法调度 Pod，调度器会尝试抢占优先级（驱逐）优先级较低的 Pod，以便可以调度待处理的 Pod。	是。大多数插件对集群功能至关重要，默认情况下应设置优先级类别。
吊舱 SecurityContext	安全上下文定义 Pod 或容器的权限和访问控制设置。通常用于设置 fsGroup，这是 1.19 及更低版本集群中的 IRSA 所必需的。	不太可能，因为亚马逊 EKS 不再支持 Kubernetes v1.19
安全上下文	安全上下文定义 Pod 或容器的权限和访问控制设置。	是
更新策略	指定用于用新 Pod 替换旧 Pod 的策略。	是
名称覆盖	覆盖 pod 的名称。	否
吊舱 SecurityPolicy	对参数实施限制。	不是-PSP 已被弃用
额外 VolumeMounts /额外音量	用于非 Amazon EKS 集群中的 IRSA。	否

容器产品定价

本部分概述了容器产品的可用定价模式。您可以发布亚马逊弹性容器服务（亚马逊BYOL）、亚马逊 Elastic Kubernetes Service（亚马逊ECS）和的免费商品、自带许可模式 (EKS) 商品和付费商品。AWS Fargate您只能为每个产品设置一个价格。

> **Note**
>
> 您可以使用 AWS Marketplace Metering Service 为您的付费产品强制执行授权和计量用量。对于每个任务或每个容器组 (pod) 定价， AWS会自动计量用量。

您为集装箱产品设定的价格适用于所有商品 AWS 区域。每当您降低容器产品的价格时，将立即为您的买家实施新的价格。对于价格上涨，现有买家会在变更影响其计费前的 90 天收到有关变更的通知。新买家将按新金额计费。

> **Note**
>
> 对于新订阅用户，价格变动立即生效。对于现有订阅用户，价格变动在自发送价格变动通知之日起的 90 天期限之后的下一个月的第一天生效。例如，假设您在 3 月 16 日发送了价格变动通知。3 月 16 日之后的 90 天大约是 6 月 16 日。由于价格变动发生在 90 天期限之后的下一个月的第一天，因此变更的生效日期为 7 月 1 日。

容器定价模式

AWS Marketplace 集装箱产品有多种定价模式。

下表提供了有关基于容器的产品的定价模式的一般信息。

容器产品的定价模式

定价模式	描述
自带许可证 (BYOL)	BYOL AWS Marketplace 通过您与买家保持的外部账单关系进行管理。
每月	固定每月价格 固定每月价格，可为用户提供在下个月无限制使用产品的权限。 示例：您将产品价格设置为每月 99 USD。您的产品包括三个不同的容器镜像，这些镜像是使用 Amazon ECS 任务定义部署的。 在客户订阅您的产品后，他们立即支付 99 美元，每月重复一次，直到他们取消订阅。买家还可以无限制地使用该产品。买家还单独为任何运行任务的基础

定价模式	描述
	设施付费。订阅后，他们可以访问您的容器映像。他们可以在任何配置下从亚马逊ECS或亚马逊上的这些镜像EKS中启动和运行任意数量的容器。 如果买家在月中取消订阅，他们将无法访问存储容器镜像的 Amazon ECR AWS Marketplace 存储库。买家可能已经提取并存储了原始映像。但是，他们无法再提取您提供的新容器镜像版本 AWS Marketplace。买家将获得最后一个月未使用部分的退款。您的付款基于买家的使用量减去商定的费用。AWS Marketplace
自定义指标定价维度	根据您定义的维度（例如用户、节点、存储库或 GB）自定义计量价格，每个产品最多 24 个维度。 示例：您的产品按用户收费。您有管理员用户和普通用户，您可以将定价定义为管理员用户 2 美元，普通用户 1 美元。在上架产品时，您可以将它们设置为单独的维度。您按每天登录的用户收费，然后计量每天的用量。

定价模式	描述
每个任务或每个容器组 (Pod) 每小时价格	Amazon t ECS ask 或 Amazon EKS pod 每个 Amazon ECS 任务或每EKS个 Amazon Pod 的定价，我们用每小时设定的价格来衡量。 示例：您的产品包含三个不同的容器映像：控制器节点、Worker 节点和分析节点。由于没有控制器节点，您的产品就无法正常运行或无法使用，因此您可以决定为该映像收取使用费。您将价格设置为每小时 6 美元。 您可以修改控制器节点的容器镜像中的软件，使其与AWS Marketplace Metering ServiceRegisterUsage API操作集成。这可确保只有具有活动订阅的买家可以启动并运行该容器映像并根据其运行时间长度计量其用量。 对于每个运行的 Amazon EKS 控制器舱，每小时向买家收取 6 美元的费用。如果买家启动五个包含EKS控制器节点容器的 Amazon 控制器吊舱，则每小时需支付 30 美元（每个 Pod 6 美元）。买家还单独为容器组 (pod) 在其上运行的任何基础设施付费。 对于每小时定价，按秒计费，最低支付 1 分钟的费用。如果客户运行此控制器容器的时间长度为 20 分钟和 30 秒，他们将支付 20 x ($6/60) + 30 x ($6/60/60) = $2 + $0.05 = $2.05 的费用。您的付款基于买家的使用量减去商定的费用。 AWS Marketplace

定价模式	描述
按小时/用量付费与长期合同	以较低的价格签订的长期合同，可以预先支付或定期分期付款。可以将长期合同添加到具有自定义计费定价或每个任务和每个容器组 (pod) 定价的现有产品。当买家的消费量超过他们在长期合同中购买的数量时，买家将支付计量价格。 示例：对于计量定价模式，您可以为买家添加长期合同价格，以获得承诺预付的折扣。假设通常每消耗一部分单位收取 1 美元。每小时使用 1 个单位的买家每年将支付 8760 美元（`365 days x 24 hours x $1 per hour`）。您可以启用合同，允许买家在这 365 天内以该价格的一半（4380 美元）每小时使用 1 个单位。在这种情况下，买家承诺为一年期的合同预付款，价格从每单位 1 美元降至每单位 0.5 美元。您也可以让买家购买其中的多份合同。如果计量数量显示买家在一小时内消耗了 10 个单位，并且他们有两份合同，那么 2 份合同中将包括 2 个单位。另外 8 个单位将按每小时 1 美元的常规价格计费，该小时的总计为 8 美元。 例如，对于每任务或每容器组 (pod)，您可以为买家添加长期合同价格，以获得承诺预付的折扣。如果您通常为每个容器组 (pod) 收取 6 美元，则可以将长期合同期限设置为 365 天，价格为 13140 美元（`365 days x 24 hours x $3 per pod per hour`）。然后，一份合同将授权客户在这 365 天内每小时获得 1 个容器组 (pod)。客户可以选择购买多份合同。例如，客户可以购买两份合同，授权他们每小时获得 2 个容器组 (pod)。如果客户每小时运行的容器组 (pod) 个数多于授权的合同，则超出的容器组 (pod) 将按您的正常小时价格计费。 在这两种情况下，购买长期合同的买家都会预先被收取费用，无论是作为一次性付款还是定期安排的未来付款。买家也将按计量费率计算超出合同的任何额外使用费用。
容器合同定价	带有合同定价的集装箱 — 一种基于集装箱的产品，买方需要为此支付预付费用。

容器产品的合同定价

对于采用合同定价的基于集装箱的产品，根据您与客户之间的合同，提前或按照您定义的付款时间表向客户开具 AWS Marketplace 账单。之后，客户将有权使用那些资源。

要设置您的定价，请选择您向客户提供的一个或多个合同期限。您可以为每个合同期限输入不同的价格。您可以选择 1 个月、12 个月、24 个月和 36 个月的合同期限。对于专属优惠，您可以指定以月为单位的自定义期限（最多 60 个月）。

选择最能描述您的产品定价的类别。定价类别会显示在 AWS Marketplace 网站上。您可以从带宽（GB/s、MB/s）、数据（GB、MB、TB）、主机、请求、套餐或用户中进行选择。如果所有预定义的类别均无法满足您的需求，您可以选择更通用的单位类别。

该优惠允许向其中添加多达 24 个维度。每个维度都需要以下数据：

- 合同类别 – 如果产品在合同定价之外还支持基于消耗的计量，则合同类别用于衡量或计量您的产品。对于不基于消耗定价的合同产品，您可以选择与合同中的维度类别最相似的类别。如果没有值与合同中该维度的单位相似，请选择`Units`。
- 合同单位 – 如果产品支持基于消耗的计量，则合同单位与计量类别一起使用。根据所选类别，为与您的维度非常匹配的单位选择一个可用值。
- 合同维度允许多次购买 - 此字段用于指明优惠是分层定价优惠还是允许购买多个维度的非分层优惠。

  分层优惠 – 允许买家仅订阅优惠中的一个可用维度。分层优惠中的维度没有数量概念。签订具有特定维度的合同本质上表明买家已经选择了该维度所指示的特定特征。

  非分层优惠 – 允许客户在合同中购买多个维度，并允许他们购买每个此类维度的多个单位。

  将此字段的值设置为 true 表示该优惠是非分层优惠。将此字段的值设置为 false 表示该优惠是分层优惠。

使用产品装货表 (PLF) 为集装箱产品创建合同时，您必须为定价维度定义以下字段：

- 合同 DimensionX API 名称 — 应出现在买方账户中生成的许可证中的名称。 AWS License Manager 在`CheckoutlicenseAPI`调用`Name`中，此名称也用作 `Entitlement` in的值。
- 合同 DimensionX 显示名称 – 将显示在 AWS Marketplace 网站的产品详细信息和采购页面上的面向客户的维度名称。创建一个用户友好的名称。名称的最大长度为 24 个字符。在列表公开之后，将无法更改 `Name` 的值。
- 合同 DimensionX 描述 – 维度的面向客户的描述，提供有关产品的维度的其他信息，例如特定维度提供的功能。描述的最大长度为 70 个字符。
- 合同 DimensionX 数量 – 用于计算产品协议修改时的比例。对于所有合同优惠，此字段的值应设置为 1。不应对其进行编辑。
- 合同 DimensionX 1 个月费率 – 针对此维度的 1 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。

- 合同 DimensionX 12 个月费率 – 针对此维度的 12 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。
- 合同 DimensionX 24 个月费率 – 针对此维度的 24 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。
- 合同 DimensionX 36 个月费率 – 针对此维度的 36 个月权利收取的合同费率。对于非分层优惠，将按所购维度的每个单位收取此费率。该字段支持三个小数位数。

示例：数据存储应用程序

	1 个月价格	12 个月价格	24 个月价格	36 个月价格
未加密的数据 (GB)	1.50 美元/GB	16.00 美元/GB	30.00 美元/GB	60.00 美元/GB
加密的数据 (GB)	1.55 美元/GB	16.60 美元/GB	31.20 美元/GB	61.20 美元/GB

示例：日志监控产品

	1 个月价格	12 个月价格	24 个月价格	36 个月价格
基本（监控 10 台主机，监控 5 个容器）	100 USD	1000 美元	2000 美元	4000 美元
标准（监控 20 台主机，监控 10 个容器）	$200	2000 美元	4000 美元	8000 美元
专业（监控 40 台主机，监控 20 个容器）	400 美元	4000 美元	8000 美元	16,000 美元
额外主机的每小时监控成本	10 美元	100 USD	$200	400 美元
额外容器的每小时监控成本	10 美元	100 USD	$200	400 美元

> **Note**
>
> 价格可以采用以下期限：1 个月、12 个月、24 个月或 36 个月。可以选择为产品提供这些选项中的一个或多个选项。期限对于每个维度都必须相同。
>
> Example
>
> 例如，在您有`ReadOnlyUsers`和`AdminUsers`维度的情况下，如果您为其提供年度价格 ReadOnlyUsers，则还必须为`AdminUsers`其提供年度价格。

自动续订

当客户 AWS Marketplace 使用集装箱合同购买您的产品时，他们可以同意自动续订合同条款。客户继续按每月或按 1 年、2 年或 3 年期限支付权利费用。

客户可以随时修改续订设置。有关更多信息，请参阅《AWS Marketplace 买家指南》中的修改现有合同。

容器合同何时到期

容器合同产品具有合同到期日期。在合同终止时，会发生以下事件：

1. 您的集装箱产品会收到一条`entitlement-updated`通知，表明买家的权利已更改，并且 AWS Marketplace Entitlement Service 返回的回复为空白。
2. 您有一个小时来计量客户的任何剩余用量。之后，您不再能够发送此客户的计量记录。

容器产品计费、计量和许可集成

AWS Marketplace 与其他产品集成 AWS 服务 ，为您的集装箱产品提供计量和基于合同的定价。

采用 AWS Marketplace Metering Service按小时计量和自定义计量

要检查使用产品的权利以及对用量进行计量以进行计费，您可以使用 AWS Marketplace Metering Service。如果您想定义自己的定价单位并将使用量计给我们进行计费，请使用该MeterUsageAPI操作进行集成。如果您想根据使用的任务或容器数量对产品进行定价，并自动 AWS 计量该使用情况，请使用该RegisterUsageAPI操作进行集成。对于这两种类型的定价，您可以添加长期合同价格，而无需更改与 AWS Marketplace Metering Service集成的方式。

当您在中创建新的容器产品时 AWS Marketplace 管理门户，我们会提供一组产品标识符（产品代码和公钥），用于将您的产品与集成 AWS Marketplace Metering Service。

权利

通过与集成， AWS Marketplace Metering Service 您可以验证运行付费软件的客户是否订阅了您的产品 AWS Marketplace，从而保护您在容器启动时免受未经授权的使用。要验证权利，请使用MeterUsage或RegisterUsageAPI操作，具体取决于您的定价模式。对于每小时和每月固定定价模式，请使用`RegisterUsage`API操作。对于自定义计量定价模型，请使用`MeterUsage`API操作。

如果买家无权购买您的商品，则这些API操作会返回`CustomerNotEntitledException`异常。

> **Note**
>
> 如果买家在运行您的产品时取消订阅了产品，他们有权继续运行产品。但是，他们无法为您的产品启动其他容器。

集成指南

在创建和发布容器产品以及使用`MeterUsage`或`RegisterUsage`API操作进行授权和计量时，请记住以下准则：

- 请勿在软件或 Docker 容器镜像中配置 AWS 凭据。 AWS 当您的容器镜像在亚马逊ECS任务或亚马逊EKS容器中运行时，买家凭证将在运行时自动获取。
- 要从 Amazon 调用`MeterUsage`或`RegisterUsage`API操作EKS，必须使用支持的 AWS SDK。要测试`MeterUsage`或`RegisterUsage`集成亚马逊EKS，您必须运行运行 Kubernetes 1.13.x 或更高版本的亚马逊EKS集群。 AWS Identity and Access Management (IAM) 角色需要 Kubernetes 1.13 才能支持 Pod。IAM正在运行的 pod 需要角色才能获得在 Amazon 上调用这些操作所需的 AWS 证书EKS。
- 您可以进行本地开发，但您将获得 `PlatformNotSupportedException` 异常。当您在容器服务（亚马逊ECS、亚马逊和 Fargate）上启动 AWS 容器时EKS，不会发生此异常。

支持的 AWS 区域

有关所有 AWS Marketplace 支持的列表 AWS 区域，请参阅全球基础设施网站上的区域表。

获取 AWS 区域 用于计量的

将用于计量的容器与MeterUsage或RegisterUsageAPI操作集成时，请不要将配置 AWS SDK为使用特定的 AWS 区域。必须在运行时动态获取区域。

Example

例如，客户启动亚马逊ECS任务或 Amazon EKS Pod。在与启动 Amazon ECS 任务或 Amazon EKS Pod 的区域不同的区域中调用该RegisterUsageAPI操作。因此，该RegisterUsageAPI操作会引发InvalidRegionException错误。

AWS SDK语言无法以一致AWS_REGION的方式确定。如果您SDK没有自动获取AWS_REGION，则需要手动编写软件才能确定AWS_Region。例如，当环境变量或其他配置不存在时， AWS SDK for Java 会自动使用 Amazon EC2 实例元数据（具体而言ec2InstanceMetadata）来获取区域。在这种情况下，只有在 AWS_REGION 环境变量不存在时才调用 ec2InstanceMetadata。

有关如何在运行 AWS 区域 时动态获取的信息，请参阅适用于您的编程语言的《AWSSDK开发人员指南》。

防止计量修改

为买家引入修改或覆盖对 RegisterUsage 或 MeterUsage 的调用的方法可能导致不受欢迎的账单和付款问题。我们强烈建议您集成计量和授权逻辑。

在设计您的产品以防止计量修改时，请注意以下事项：

- 如果买家可以插入包含 CMD 或 ENTRYPOINT 指令的新映像层，请直接将 RegisterUsage 或 MeterUsage 集成到买家正在通过容器映像运行的软件中。否则，从基本映像中通过 CMD 或 ENTRYPOINT 执行的对 RegisterUsage 或 MeterUsage 的调用可能会被买家覆盖。
- 我们建议您管理您的软件用作输入 AWS Marketplace 的产品代码，RegisterUsage或者以买家无法修改MeterUsage的方式进行管理。但是，如果您的产品以客户可以覆盖的方式管理产品代码 AWS CloudFormation，例如 Helm chart 或 Kubernetes 清单，则必须维护可信 AWS Marketplace 产品代码列表。这样可确保您的软件作为 RegisterUsage 或 MeterUsage 的输入传递的产品代码有效。
- 如果任何受信任的产品代码适用于免费产品，请确保它们无法用于替代付费产品代码。

与的合同定价 AWS License Manager

对于采用合同定价的基于容器的产品，您可以使用 AWS License Manager 将许可证与您的产品关联起来。

AWS License Manager 是一种许可证管理工具，可让您的应用程序跟踪和更新客户购买的许可证（也称为授权）。本部分提供有关如何将您的产品与 AWS License Manager集成的信息。集成完成后，您可以在 AWS Marketplace上发布您的产品清单。

有关的更多信息 AWS License Manager，请参阅《AWS License Manager 用户指南》和《AWS CLI命令参考》一AWS License Manager节。

> **Note**
>
> - 合同到期后，客户无法启动新的容器实例。但是，在合同有效期内，他们可以启动任意数量的实例。这些许可证不绑定到特定的节点或实例。在任何节点上的任何容器上运行的任何软件都可以签出许可证，只要它具有分配的 AWS 凭证。
> - 专属优惠创建 – 卖家可以使用 AWS Marketplace 管理门户中的专属优惠创建工具为产品生成专属优惠。
> - 报告 – 您可以通过在 AWS Marketplace 管理门户中的报告部分设置 Amazon S3 存储桶来设置数据源。有关更多信息，请参阅 卖家报告、数据源和控制面板。

集成工作流

以下步骤显示了将容器产品与 AWS License Manager集成的工作流：

1. 卖家创建具有 AWS License Manager 集成功能的产品。
2. 卖家在上架商品 AWS Marketplace。
3. 买家在上面找到产品 AWS Marketplace 并购买。
4. 许可证通过买家的 AWS 账户发送给他们。
5. 买家通过启动亚马逊EC2实例、亚马逊ECS任务或亚马逊 EKS pod 软件来使用该软件。客户使用IAM角色进行部署。
6. 软件读取买方 AWS License Manager 账户中的许可证，发现购买的权利并相应地配置功能。

> **Note**
>
> License Manager 不进行任何跟踪或更新；这些是通过卖家的应用程序完成的。

使用 AWS Marketplace 计量服务按小时计

如果您的容器产品使用每小时每任务或每容器组 (pod) 定价，而不是自定义计量定价维度，则无需定义自定义计量维度。

`RegisterUsage`API操作计量每个亚马逊弹性容器服务 (AmazonECS) 任务或每个亚马逊 Elastic Kubernetes Service (EKSAmazon) Pod 每小时的软件使用量，使用量按比例分配给第二个。最低 1 分钟用量适用于有效期较短的任务或 pod。软件使用的连续计量由自动处理 AWS Marketplace Metering Control Plane。 除了调用`RegisterUsage`一次即可开始计量软件使用情况外，您的软件无需执行任何计量特定的操作。

`RegisterUsage` 必须在启动容器时立即调用。如果您在集装箱启动的前 6 个小时内没有注册集装箱，AWSMarketplace Metering Service 将不为前几个月提供任何计量保证。但是，计量将在当月继续进行，直到容器结束。

的 AWS Marketplace Metering Control Plane 继续向客户收取运行亚马逊ECS任务和 Amazon EKS Pod 的费用，无论客户的订阅状态如何。这样，您的软件就无需在任务或容器组 (pod) 首次成功启动后执行授权检查。

每小时计量先决条件

发布产品之前，您必须首先完成以下操作：

1. 在中创建新的容器产品 AWS Marketplace 管理门户，并记下其产品代码。

   有关更多信息，请参阅 创建容器产品。
2. 为运行应用程序的任务或 Pod 使用 AWS Identity and Access Management (IAM) 角色，并具有调用所需的IAM权限`RegisterUsage`。IAM托管策略`AWSMarketplaceMeteringRegisterUsage`具有这些权限。
3. （可选）如果您想查看日志记录，我们建议您在任务或 pod 定义中启用 AWS CloudTrail 日志记录。
4. 使用您定义的所有定价维度的记录对`RegisterUsage`API操作进行测试调用。

每小时计量的产品加载表单

在填写每小时计量的产品加载表单时，除了定义产品的其他必填字段和可选字段外，还要为您的产品填写以下字段：

- 每小时价格 - 产品每小时的价格。
- 维度长期费率 – 买家预先付款时，长期合同中的软件总价格。
- 长期期限（天） - 长期合同的期限（以天为单位）。

测试 `RegisterUsage` 的集成和预览模式

在将图片提交给发布之前，请使用该`RegisterUsage`API操作来 AWS Marketplace 测试您的集成。

预览模式的运行方式与生产模式完全相同，但预览模式不验证对您的产品的使用权。要`RegisterUsage`在预览模式下调用，请通过在 Amazon ECS 或 Amazon 上运行您的产品，`RegisterUsage`从容器图片中调用EKS。使用 AWS 账户 您正在使用的上架商品 AWS Marketplace。您的计量集成必须动态设置 AWS 区域，而不是对其进行硬编码。但是，在测试时，请至少在美国东部（弗吉尼亚北部）地区启动一个包含您的付费容器的 Amazon ECS 任务或包含您的付费容器的 Amazon EKS pod。这样， AWS Marketplace 运营团队就可以使用该区域的日志来验证您的工作。

> **Note**
>
> 如果您的产品同时支持亚马逊ECS和亚马逊EKS，您只需要在亚马逊EKS上线，我们就可以验证您的集成。

在使用所有必需的元数据和定价信息发布您的产品之前，您无法完全测试集成。如果需要， AWS Marketplace 目录运营团队可以在预览模式下验证您的计量记录是否收到。

`RegisterUsage` 中的错误处理

如果您的容器镜像与集成， AWS Marketplace Metering Service 并且`ThrottlingException`在容器启动之外收到异常，则应终止容器以防止未经授权的使用。

除此之外的`ThrottlingException`异常仅在初次调用`RegisterUsage`API操作时才会引发。`CustomerNotSubscribedException`即使客户在ECS任务或 EKS Pod 仍在运行时取消订阅，来自同一 Amazon 任务或 Amazon Pod 的后续调用也不会抛出。这些客户在取消订阅并跟踪其使用情况后仍需支付运行容器的费用。

下表描述了该RegisterUsageAPI操作可能引发的错误。每种 AWS SDK编程语言都有一套错误处理指南，您可以参考这些指南以获取更多信息。

错误	描述
InternalServiceErrorException	RegisterUsage 不可用。
CustomerNotEntitiledException	客户没有产品的有效订阅。
InvalidProductCodeException	作为请求的一部分传入的 ProductCode 值不存在。
InvalidPublicKeyException	作为请求的一部分传入的 PublicKeyVersion 值不存在。
PlatformNotSupportedException	AWS Marketplace 不支持从底层平台计量使用情况。仅支持 Amazon ECS EKS、Amazon 和 AWS Fargate 。
ThrottlingException	对 RegisterUsage 的调用受限。
InvalidRegionException	RegisterUsage 必须使用与启动 Amazon ECS 任务或 Amazon EKS pod 相同的 AWS 区域 方法进行调用。这可防止容器在调用 RegisterUsage 时选择区域（例如，withRegion(“us-east-1”) ）。

使用以下方法将您的容器产品与 AWS Marketplace 计量服务集成 AWS SDK for Java

以下步骤概述了使用与AWS Marketplace 计量服务RegisterUsage操作集成的实现示例。 AWS SDK for Java 有关完整源代码，请参阅RegisterUsage Java 示例。其中许多步骤适用于任何语言。

集成 AWS Marketplace 计量服务的示例步骤

1. 登录到 AWS Marketplace 管理门户。
2. 从资产中，选择容器以开始创建新容器产品。创建产品会生成产品的产品代码以与您的容器映像集成。有关设置IAM权限的信息，请参阅the section called “AWS Marketplace 计量和授权 API 权限”。

3. 下载公共 AWSJava 版本SDK。

> **Important**
>
> 要APIs从亚马逊调用计量EKS，您必须使用支持的， AWS SDK并在运行 Kubernetes 1.13 或更高版本的亚马逊EKS集群上运行。

4. （可选）如果您要与`RegisterUsage`操作集成，并且想要执行数字签名验证，则需要在应用程序类路径中配置BouncyCastle签名验证库。

   如果要使用 JSON Web Token (JWT)，则还必须在应用程序类路径中包含 JWTJava 库。使用JWT提供了一种更简单的签名验证方法，但不是必需的，您可以 BouncyCastle 改为使用独立版。无论您使用JWT还是 BouncyCastle，都需要使用诸如 Maven 之类的编译系统JWT在应用程序类路径中包含 BouncyCastle 或的传递依赖关系。

```
// 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. 从您的产品中的每个付费容器映像调用 `RegisterUsage`。`ProductCode` 和 `PublicKeyVersion` 是必需参数，所有其他输入都是可选的。以下是 `RegisterUsage` 的示例负载。

```
{
    "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**
>
> 在连接到 M AWS arketplace 计量服务时可能会出现暂时性问题。 AWS Marketplace 强烈建议实施最长 30 分钟的重试，并以指数级退缩，以避免短期中断或网络问题。

6. `RegisterUsage`使用 SHA -256 生成一个RSAPSS数字签名，您可以使用该签名来验证请求的真实性。该签名包括以下字段：`ProductCode`、`PublicKeyVersion` 和 `Nonce`。要验证数字签名，您必须保留请求中的这些字段。以下代码是 `RegisterUsage` 调用的示例响应。

```
{
"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. 重建包含`RegisterUsage`调用的容器镜像的新版本，标记容器，然后将其推送到任何与亚马逊ECS或亚马逊兼容的容器注册表EKS，例如亚马逊ECR或亚马逊ECR公众。如果您使用的是亚马逊ECR，请确保启动亚马逊ECS任务的账户或 Amazon EKS pod 具有访问亚马逊ECR存储库的权限。否则，启动失败。

8. 创建一个IAM角色来授予容器调用权限`RegisterUsage`，如以下代码所定义。您必须在亚马逊任务的任务IAM角色参数或 Amazon ECS EKS pod 定义中提供此角色。

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

9. 创建 Amazon ECS 任务或 Amazon EKS 容器定义，该定义引用已与您在步骤 7 中创建的IAM角色集成的容器 AWS Marketplace 并引用该角色。如果要查看 AWS CloudTrail 日志记录，则应在任务定义中启用日志记录。
10. 创建 Amazon ECS 或 Amazon EKS 集群来执行您的任务或容器。有关创建 Amazon ECS 集群的更多信息，请参阅《亚马逊弹性容器服务开发人员指南》中的创建集群。有关创建亚马逊EKS集群（使用 Kubernetes 版本 1.1.3.x 或更高版本）的更多信息，请参阅创建亚马逊集群。EKS
11. 在 us-east- AWS 区域 1 中配置亚马逊ECS或亚马逊EKS集群并启动您创建的亚马逊ECS任务定义或亚马逊EKS容器。只有在此测试过程中，在产品上线之前，您才必须使用此区域。
12. 当您从 `RegisterUsage` 获得有效的响应时，您可以开始创建您的容器产品。如有问题，请联系 AWS Marketplace 卖家运营团队。

RegisterUsage Java 示例

以下示例使用 AWS SDK for Java 和 AWS Marketplace 计量服务调用该`RegisterUsage`操作。签名验证是可选的，但如果您要执行签名验证，则必须包含所需的数字签名验证库。此示例仅用于说明。

```
import com.amazonaws.auth.PEM;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMetering;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMeteringClientBuilder;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageRequest;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageResult;
import com.amazonaws.util.json.Jackson;
import com.fasterxml.jackson.databind.JsonNode;
import com.nimbusds.jose.JWSObject;
import com.nimbusds.jose.JWSVerifier;
import com.nimbusds.jose.crypto.RSASSAVerifier;
```

```
import java.io.ByteArrayInputStream;
import java.nio.charset.StandardCharsets;
import java.security.PublicKey;
import java.security.Security;
import java.security.Signature;
import java.security.interfaces.RSAPublicKey;
import java.util.Base64;
import java.util.Optional;
import java.util.UUID;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

/**
 * Class for making calls out to AWS Marketplace Metering Service.
 */
class RegisterUsage {

    private static final String PRODUCT_CODE = ".......";

    private final AWSMarketplaceMetering registerUsageClient;
    private final SignatureVerifier signatureVerifier;
    private final int publicKeyVersion;

    public RegisterUsage(final SignatureVerifier signatureVerifier) {
        this.signatureVerifier = signatureVerifier;
        this.publicKeyVersion = PublicKeyProvider.PUBLIC_KEY_VERSION;
        this.registerUsageClient =
 AWSMarketplaceMeteringClientBuilder.standard().build();
    }

    /**
     * Shows how to call RegisterUsage client and verify digital signature.
     */
    public void callRegisterUsage() {
        RegisterUsageRequest request = new RegisterUsageRequest()
                .withProductCode(PRODUCT_CODE)
                .withPublicKeyVersion(publicKeyVersion)
                .withNonce(UUID.randomUUID().toString());

        // Execute call to RegisterUsage (only need to call once at container startup)
        RegisterUsageResult result = this.registerUsageClient.registerUsage(request);

        // Verify Digital Signature w/o JWT
        boolean isSignatureValid = this.signatureVerifier.verify(request, result);
        if (!isSignatureValid) {
```

```
            throw new RuntimeException("Revoke entitlement, digital signature
 invalid.");
        }
    }
}

/**
 * Signature verification class with both a JWT-library based verification
 * and a non-library based implementation.
 */
class SignatureVerifier {
    private static BouncyCastleProvider BC = new BouncyCastleProvider();

    private static final String SIGNATURE_ALGORITHM = "SHA256withRSA/PSS";

    private final PublicKey publicKey;

    public SignatureVerifier(PublicKeyProvider publicKeyProvider) {
        this.publicKey = publicKeyProvider.getPublicKey().orElse(null);
        Security.addProvider(BC);
    }

    /**
     * Example signature verification using the NimbusJOSEJWT library to verify the JWT
 Token.
     *
     * @param request RegisterUsage Request.
     * @param result  RegisterUsage Result.
     * @return true if the token matches.
     */
    public boolean verifyUsingNimbusJOSEJWT(final RegisterUsageRequest request, final
 RegisterUsageResult result) {
        if (!getPublicKey().isPresent()) {
            return false;
        }

        try {
            JWSVerifier verifier = new RSASSAVerifier((RSAPublicKey)
 getPublicKey().get());
            JWSObject jwsObject = JWSObject.parse(result.getSignature());
            return jwsObject.verify(verifier) &&
 validatePayload(jwsObject.getPayload().toString(), request, result);
        } catch (Exception e) {
            // log error
```

```
            return false;
        }
    }

    /**
     * Example signature verification without any JWT library support.
     *
     * @param request RegisterUsage Request.
     * @param result  RegisterUsage Result.
     * @return true if the token matches.
     */
    public boolean verify(final RegisterUsageRequest request, final RegisterUsageResult
result) {
        if (!getPublicKey().isPresent()) {
            return false;
        }
        try {
            String[] jwtParts = result.getSignature().split("\\.");
            String header = jwtParts[0];
            String payload = jwtParts[1];
            String payloadSignature = jwtParts[2];

            Signature signature = Signature.getInstance(SIGNATURE_ALGORITHM, BC);
            signature.initVerify(getPublicKey().get());
            signature.update(String.format("%s.%s", header,
payload).getBytes(StandardCharsets.UTF_8));
            boolean verified = signature.verify(Base64.getUrlDecoder()
                    .decode(payloadSignature.getBytes(StandardCharsets.UTF_8)));

            String decodedPayload = new String(Base64.getUrlDecoder().decode(payload));
            return verified && validatePayload(decodedPayload, request, result);
        } catch (Exception e) {
            // log error
            return false;
        }
    }

    /**
     * Validate each value in the returned payload matches values originally
     * supplied in the request to RegisterUsage. TimeToLiveInMillis and
     * PublicKeyExpirationTimestamp will have the values in the payload compared
     * to values in the signature
     */
```

```
    private boolean validatePayload(final String payload, final RegisterUsageRequest
 request,
                                    final RegisterUsageResult result) {
        try {
            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 计量服务为集装箱产品定制计量

AWS Marketplace 集装箱产品可以对每种产品多达 24 种不同的定价维度进行自定义计量。每个维度都可以具有与之相关的长期合同价格。要启用自定义计量，请将您的容器产品与 AWS Marketplace 计量服务集成。您可以为该使用量定义自己的定价单位和自定义计量，以便使用该`MeterUsage`API操作 AWS 进行计费。

价格维度在两个位置定义，一次是在产品加载表单中，一次是通过 `MeterUsage` API 操作。此双因素方法可确保后续的产品在向公众提供之前按预期工作。

要设置自定义计量，您需要选择使用类别、单位类型和定价维度：

- 用户类别 - 用户类别可帮助买家了解您的产品是什么以及如何使用它。
- 单位类型 - 单位类型定义计费的计量单位。例如，以GBps或为单位的带宽MBps、以主机数量为单位的带宽，或者以 MB、GB 或 TB 为单位的数据。
- 定价维度-定价维度表示您为其设置了单位价格的功能或服务（例如，用户vCPUs、扫描或部署的代理）。定价维度是公开的。但是，您仍然可以为公共产品定义私有和自带许可证 (BYOL) 优惠。请勿在计量记录中发送定价。您可以计量单位数量，我们将其与您在创建产品时定义的价格一起使用，以计算买家的账单。

  如果您的产品定价不符合任何预定义类别或单位类型，您可以选择通用单位类别。然后，使用维度描述来描述单位是什么。

或者，您可以按您跟踪的属性将用量分发到分配中。分配以标签形式呈现给买家。这些标签允许买家按标签值查看按用量划分的费用。例如，如果您按用户收费，并且用户具有“部门”属性，则可以使用键

为“部门”的标签创建使用分配，每个值一个分配。这不会更改您报告的价格、维度或总用量，但允许您的客户按与您的产品相应类别查看其成本。

我们建议您每小时发送一次计量记录。但是，您也可以汇总每日或每月的用量。如果您遇到中断，您可以聚合买家的软件用量，并在接下来的小时计量中发送。每小时不能发送多条记录。

> **Important**
>
> 免费试用和预付费权利按小时进行跟踪。因此，单独发送这些记录可能会导致买家被多收费用。

自定义计量先决条件

发布产品之前，您必须首先完成以下操作：

1. 在中创建新的容器产品 AWS Marketplace 管理门户，并记下其产品代码。
2. 填写产品加载表单以及必要的维度信息，并将其返回给我们进行处理。
3. 为运行应用程序的任务或 Pod 使用 AWS Identity and Access Management (IAM) 角色，并具有调用所需的IAM权限`MeterUsage`。IAM托管策略`AWSMarketplaceMeteringRegisterUsage`具有这些权限。
4. （可选）如果您想查看 AWS CloudTrail 日志记录，我们建议您在任务或 pod 定义中启用日志记录。
5. 使用您定义的所有定价维度的记录对`MeterUsage`API操作进行测试调用。

自定义计量的产品加载表单

填写产品加载表单进行自定义计量时，每种产品最多可以有 24 个维度。维度在以下字段中定义：

- 唯独名称 – 容器应用程序将计量记录发送到 AWS Marketplace Metering Service时使用的名称。此名称指示买家将使用的维度。此名称会显示在账单报告中。名称一旦设置便无法更改。
- 维度描述 – 维度的面向买家的描述。描述不能超过 70 个字符。产品向买家公开发布后，此字段无法更改。
- 维度费率 – 当买家按需支付费用时，此产品的单位软件价格。该字段支持三个小数位数。
- 维度长期费率 – 买家预先付款时，长期合同中的软件总价格。
- 长期期限（天）- 长期合同的期限（以天为单位）。

测试 MeterUsage 集成和预览模式

在将图片提交给发布之前，请使用该MeterUsage操作来 AWS Marketplace 测试您的集成。

预览模式的运行方式与生产模式完全相同，但预览模式不验证对您的产品的使用权。要以预览模式调MeterUsage用MeterUsage，请在亚马逊弹性容器服务 (亚马逊ECS) 或亚马逊 Elastic Kubernetes Service (Amazon) 上运行您的EKS商品，并使用您用于发布商品的容器镜像 AWS 账户 进行调用。AWS Marketplace您的计量集成必须动态设置 AWS 区域，而不是对其进行硬编码。但是，在测试时，请至少在美国东部（弗吉尼亚北部）地区启动一个包含您的付费容器的 Amazon ECS 任务或 Amazon EKS pod，以便 AWS Marketplace 运营团队可以使用该地区的日志验证您的工作。

> **Note**
>
> - 如果您的产品同时支持亚马逊ECS和亚马逊EKS，您只需要在亚马逊EKS上线，我们就可以验证您的集成。
> - 在向公众发布产品之前和添加新维度之后，对每个维度进行测试。如果您没有为与集装箱产品关联的每个维度发送计量记录，则会导致错误并导致请求失败。

在使用所有必需的元数据和定价信息发布您的产品之前，您无法完全测试集成。如果需要， AWS Marketplace 目录运营团队可以在预览模式下验证是否收到您的计量记录。

MeterUsage 中的错误处理

如果您的容器映像与 MeterUsage 操作集成并在容器启动时收到 ThrottlingException 之外的异常，您应终止容器以防止未经授权的使用。

仅在初始调用 MeterUsage 时引发 ThrottlingException 以外的异常。CustomerNotSubscribedException即使客户在ECS任务或 EKS Pod 仍在运行时取消订阅，来自同一 Amazon 任务或 Amazon Pod 的后续调用也不会抛出。这些客户在取消订阅并跟踪其使用情况后仍需支付运行容器的费用。

有关常见错误MeterUsage的详细说明，请参阅 “AWS Marketplace Metering Service API参考资料” MeterUsage。每种 AWS SDK编程语言都有一套错误处理指南，您可以参考这些指南以获取更多信息。

供应商计量标记（可选）

供应商计量标签可帮助独立软件供应商 (ISVs) 让购买者更精细地了解其软件使用情况，并可以帮助他们进行成本分配。

有许多方法可以标记买家的软件使用情况。一种方法是先询问买家他们希望在成本分配中看到什么。然后，您可以在为买家帐户跟踪的属性之间分配使用情况。属性的示例包括 AccountId、Business Unit、Cost Centers，以及您的产品的其他相关元数据。这些属性作为标签展示给买家。使用标签，买家可以在 AWS 账单控制台（https://console.aws.amazon.com/billing/）中查看按标签值划分为使用量的费用。供应商计量标记不会更改您报告的价格、维度或总用量。它允许您的客户按您产品的相应类别查看成本。

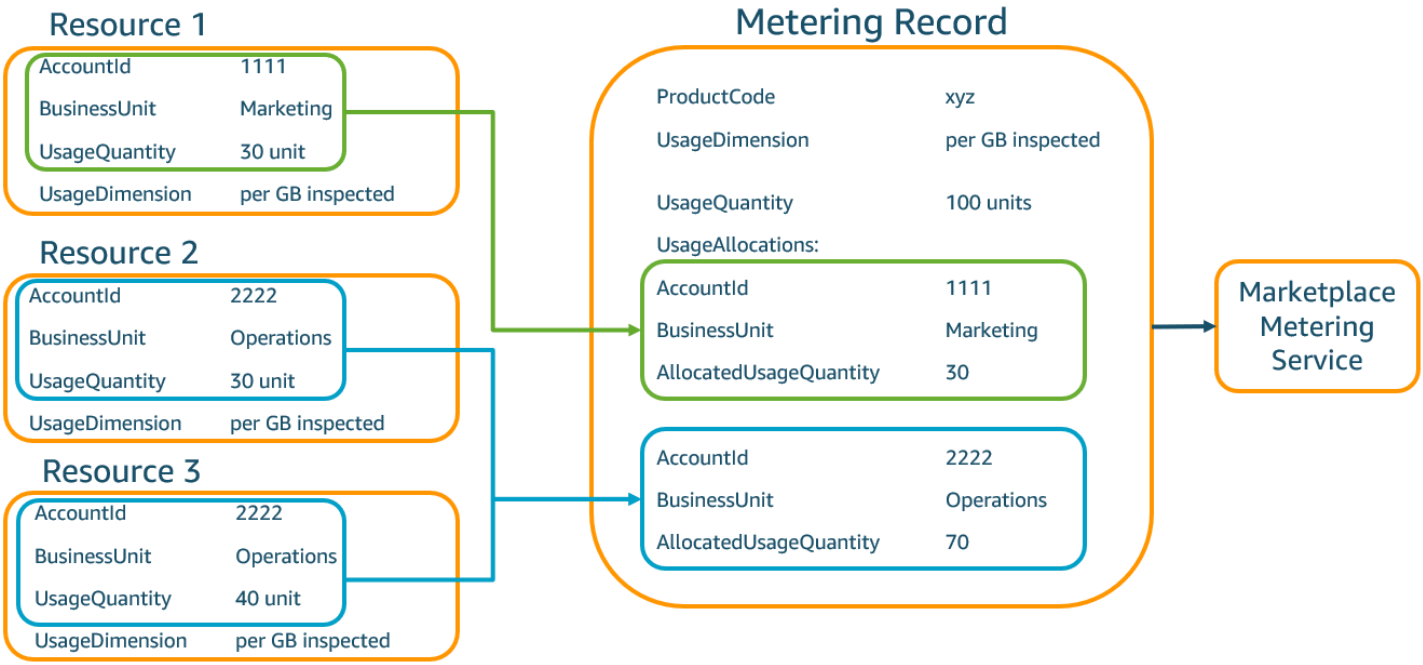
在常见情形中，买家会通过一个 AWS 账户订阅您的产品。买家还有许多与同一产品订阅相关的用户。您可以使用具有 AccountId 键的标签创建用量分配，然后将用量分配给每个用户。在这种情况下，买家可以在其账单与成本管理控制台中激活 AccountId 标签并分析个人用户的使用情况。

卖家体验

卖家可以汇总具有相同标签集的资源的计量记录，而不是汇总所有资源的使用情况。例如，卖家可以构造包含不同的 UsageAllocations 存储桶的计量记录。每个存储桶代表一组标签的 UsageQuantity，例如 AccountId 和 BusinessUnit。

在下图中，资源 1 具有一组唯一的 AccountId 和 BusinessUnit 标签，并作为单个条目出现在计量记录中。

资源 2 和资源 3 都具有相同的 AccountId 标签 2222 和相同的 BusinessUnit 标签 Operations。因此，它们在计量记录中合并为一个 UsageAllocations 条目。



卖家还可以将没有标签的资源组合成一个包含分配用量的 UsageAllocation，然后将其作为 UsageAllocations 中的一个条目发送。

限制包括：

- 标签数 – 5
- UsageAllocations（基数）的大小 – 2500

验证包括：

- 标签键和值允许的字符 – a-zA-Z0-9+ -=._:\/@
- UsageAllocation 列表中的最大标签数 – 5
- 两个 UsageAllocations 的标签不能相同（也就是说，标签键和值的组合相同）。如果是这样的话，他们必须使用相同的 UsageAllocation。
- UsageAllocation 的 AllocatedUsageQuantity 之和必须等于 UsageQuantity，即聚合用量。

买家体验

下表显示了买家激活 AccountId 和 BusinessUnit 供应商标签后的买家体验示例。

在此示例中，买家可以在其成本使用报告中看到分配的用量。供应商计量标签使用前缀 “aws:marketplace:isv”。在账单与成本管理中，买家可以再成本分配标签下的 AWS生成的成本分配标签中激活它们。

成本使用报告的第一行和最后一行与卖家向 Metering Service 发送的内容相关（如 卖家体验 示例所示）。

成本使用报告（简化）

ProductCode	买家	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	网络：每 (GB) 检查一次	70	2222	操作

ProductCode	买家	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	网络：每 (GB) 检查一次	30	3333	财务
xyz	111122223333	网络：每 (GB) 检查一次	20	4444	IT
xyz	111122223333	网络：每 (GB) 检查一次	20	5555	市场营销
xyz	111122223333	网络：每 (GB) 检查一次	30	1111	市场营销

有关代码示例，请参阅 带有用量分配标签的 MeterUsage 代码示例（可选）。

代码示例

以下代码示例旨在帮助您将容器产品与发布和维护产品 AWS Marketplace APIs所需的产品集成。

带有用量分配标签的 **MeterUsage** 代码示例（可选）

以下代码示例与具有消费定价模式的容器产品相关。Python 示例将带有相应使用量分配标签的计量记录发送给您的客户 AWS Marketplace ，以向您的客户收取 pay-as-you-go 费用。

```
# 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
)
```

有关的更多信息`MeterUsage`，请参阅 “AWS Marketplace Metering Service API参考” MeterUsage中的。

响应示例

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

AWS Marketplace Metering Service 使用 AWS SDK适用于 Java 的，将您的容器产品与集成

以下示例概述了使用与AWS Marketplace 计量服务`MeterUsage`操作集成的实现。 AWS SDK for Java 有关完整的详细信息，请参阅 `MeterUsage` Java 示例。以下步骤中的许多步骤适用于任何语言。

示例： AWS Marketplace 计量服务集成

1. 登录到 AWS Marketplace 管理门户。
2. 从资产中，选择容器以开始创建新容器产品。创建产品会生成产品的产品代码以与您的容器映像集成。有关设置 AWS Identity and Access Management (IAM) 权限的信息，请参阅the section called “AWS Marketplace 计量和授权 API 权限”。
3. 下载公共 AWSJava 版本SDK。

> **Important**
>
> 要从亚马逊 Elastic Kubernetes Service（EKS亚马逊）调用计量API操作，您必须使用 AWS SDK受支持的并在运行 Kubernetes 1.13 或更高版本的EKS亚马逊集群上运行。

4. 针对每个维度使用情况，每小时从任务或容器组 (pod) 调用一次 `MeterUsage` 操作。该API操作接受`Dimension`、`Resource`和的唯一组合的一条计量记录`Hour`。该资源可以是亚马逊弹性容器服务 (AmazonECS) 任务，也可以是亚马逊 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**
>
> 在连接时可能会出现暂时性问题 AWS Marketplace Metering Service。 AWS Marketplace 强烈建议实施最长 30 分钟的重试，并以指数级退缩，以避免短期中断或网络问题。

5. 重建包含`MeterUsage`调用的容器镜像的新版本，标记容器，然后将其推送到与亚马逊ECS或亚马逊兼容的任何 Docker 注册表EKS，例如亚马逊弹性容器注册表 (AmazonECR)。如果您使用的是亚马逊ECR，请确保启动亚马逊ECS任务的账户或 Amazon EKS pod 具有访问亚马逊ECR存储库的权限。否则，该操作将失败。
6. 创建一个IAM角色来授予容器调用权限`MeterUsage`，如以下代码示例所定义。您必须在亚马逊任务的任务角色参数或 Amazon ECS EKS pod 定义中提供此 AWS Identity and Access Management (IAM) 角色。

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

7. 创建 Amazon ECS 任务或 Amazon EKS 容器定义，引用已与您在步骤 6 中创建的IAM角色集成的容器 AWS Marketplace 并引用该角色。如果要查看日志记录，请在任务定义中启用 AWS CloudTrail 日志记录。
8. 创建 Amazon ECS 或 Amazon EKS 集群来运行您的任务或容器。有关创建 Amazon ECS 集群的更多信息，请参阅《亚马逊弹性容器服务开发人员指南》中的创建集群。有关创建亚马逊EKS集群（使用 Kubernetes 版本 1.1.3.x 或更高版本）的更多信息，请参阅创建亚马逊集群。EKS
9. 在 us-east-1 AWS 区域配置亚马逊或亚马逊EKS集群并启动您在步骤 8 中创建的亚马逊ECS任务定义或亚马逊EKS容器。ECS只有在此测试过程中，在产品上线之前，您才必须使用此区域。
10. 当您从正在为产品发布的每个维度的 `MeterUsage` 获得有效相应时，您可以开始创建容器产品。如有问题，请联系 AWS Marketplace 卖家运营团队。

`MeterUsage` Java 示例

以下代码示例使用 AWS SDK for Java 和 AWS Marketplace 计量服务来调用该`MeterUsage`操作。

使用以下示例代码调用 `MeterUsage` 操作，不设置 `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);
    }
}
```

使用以下示例代码调用 `MeterUsage` 操作，设置 `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();
    }
```

集装箱产品的合同定价 AWS License Manager

对于采用合同定价的基于容器的产品，请使用 AWS License Manager 将许可证与您的产品相关联。

AWS License Manager 是一种许可证管理工具，可让您的应用程序跟踪和更新客户购买的许可证（也称为授权）。本部分提供有关如何将您的产品与 AWS License Manager集成的信息。集成完成后，您可以在 AWS Marketplace上发布您的产品清单。

如果您要将 License Manager 与 AWS Marketplace 适用于 Amazon Anywhere、Amazon EKS Anywhere、亚马逊弹性计算云 (AmazonEC2) 或本地基础设施的 for Contain ECS ers Anywhere 产品集成，请按照中的说明进行操作将 “容器随处 AWS Marketplace 可用” 产品与 License Manager 集成。

有关的更多信息 AWS License Manager，请参阅《AWS License Manager 用户指南》和《AWS CLI命令参考》一AWS License Manager节。

许可模式

AWS Marketplace 与的集成 AWS License Manager 支持两种许可模式：

- 可配置许可模式
- 分层许可模式

可配置许可模式

可配置许可证模型（也称为可量化许可证模型）在买家获得许可证后授权买家获得特定数量的资源。

您可以设置定价维度和每单位价格。然后，买家可以选择他们想要购买的资源数量。

Example 定价维度和每单位价格

您可以设置定价维度（例如数据备份）和每单位价格（例如每单位 30 美元）

买家可以选择购买 5、10 或 20 个单位。

您的产品会跟踪和计量使用情况，以衡量消耗的资源数量。

在配置模式下，权利将通过以下两种方式之一进行计数：

- 消耗许可证
- 浮动许可证

消耗许可证

在使用时，许可证是从允许数量的许可证池中消耗。该权利已永久签出，无法返回到许可证池。

Example 处理有限数量的数据示例

用户有权处理 500 GB 的数据。当他们继续处理数据时，系统会从 500 GB 的池中消耗数量，直到所有 500 GB 的许可证都用完为止。

对于提款许可证，您可以使用`CheckoutLicense`API操作来查看已使用的许可证单位（授权）。

Example 每年备份到 S3 的多个单位示例

您拥有的存储产品允许将多达 1024 个单位的数据备份到 Amazon Simple Storage Service (Amazon S3) 中以存储一年的数据。您的应用程序可以使用多个 Amazon EC2 实例启动。您的应用程序具有跟踪和聚合数据的机制。每次备份时，您的软件都会使用产品 ID 调用该`CheckoutLicense`API操作以更新消耗的数量。

在此示例中，您的软件调用该`CheckoutLicense`API操作来检出 10 个单位的数据。当总容量达到客户购买的备份限制时，API呼叫就会失败。

请求

```
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=10, Unit=Count" \
--client-token "AKIAIOSFODNN7EXAMPLE"
```

响应

```
{"CheckoutType": "PERPETUAL",
"EntitlementsAllowed": [{
"Name": "IntermediateTier",
"Units": "None"
}],
"Expiration": "2021-04-22T19:02:36",
"IssuedAt": "2021-04-22T18:02:36",
"LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
"LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

浮动许可证

使用后，许可证将返回到允许数量的许可证池中。

对于浮动许可证，应用程序在使用资源时使用`CheckoutLicenseAPI`操作从授权池中签出授权。`CheckoutLicenseAPI`操作的响应包括许可证消耗令牌，该令牌是结账的唯一标识符。许可证消耗令牌可用于对已签出的权利执行其他操作，例如将其签回许可证池或延长签出时间。

要将授权签回池中，请在资源不再使用时使用该`CheckInLicenseAPI`操作。

```
aws license-manager check-in-license --license-consumption-token
 "f1603b3c1f574b7284db84..."
```

如果未能签入权利（比如应用程序崩溃），则权利将在 60 分钟后自动签回池中。如果资源的使用时间超过 60 分钟，则最佳做法是，只要资源正在使用中，就使用该`ExtendLicenseConsumptionAPI`操作将授权从池中签出。

```
aws license-manager extend-license-consumption --license-consumption-token
 "f1603b3c1f574b7284..."
```

Example 固定上限的用户数示例

某用户被授予应用程序 500 个并发用户的权利。当用户登录和注销时，用户会被消耗并返回到 500 个用户的池中。但是，应用程序无法从池中消耗超过 500 个用户，因为 500 个并发用户是固定上限。

对于浮动授权，您可以使用CheckInLicenseAPI操作将许可证单位返回到授权池。

Example 一年的并发用户数示例

您的产品根据并发用户数量定价。客户为 10 个用户购买了一份为期一年的许可证。客户通过提供 AWS Identity and Access Management (IAM) 权限来启动软件。当用户登录时，您的应用程序会调用该CheckoutLicenseAPI操作以将数量减少 1。当用户注销时，应用程序会通过调用CheckInLicenseAPI操作将该许可证返回到池中。如果您不调用 CheckInLicense，则许可证单位将在 1 小时后自动签入。

> **Note**
>
> 在以下请求中，key-fingerprint 不是占位符值，而是发布所有许可证时使用的指纹的实际值。

请求

```
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"
```

响应

```
{
  "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"
```

```
}
```

分层许可模式

分层许可模式在买家获得许可后，使买家有权使用特定级别或层的应用程序特征。

您可以为产品创建层，例如基本、中级和高级。然后，买家选择一个预定义的层。

应用程序无需跟踪或计量应用程序的使用情况。

在分层许可模式下，权利不计算在内，而是表示客户购买的服务层。

如果您想提供捆绑特征，则最好使用层。

Example 基本、中级和高级层

客户可以签署软件三个可能层之一的合同：基础、中级或高级。每个层都有自己的定价。您的软件可以通过调用`CheckoutLicense`API操作并在请求中指定所有可能的等级来识别客户已注册的等级。

请求响应包含与客户购买的层相对应的权利。基于这些信息，该软件可以预置适当的客户体验。

请求

```
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"
```

响应

```
{
  "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 集成先决条件

发布产品之前，您必须首先完成以下操作：

1. 在中创建新的容器产品 AWS Marketplace 管理门户，并记下其产品代码。

   有关更多信息，请参阅 创建容器产品。

2. 为运行应用程序的任务或容器使用具有调用`CheckoutLicenseExtendLicenseConsumption`、和`CheckInLicense`API操作所需的IAM权限的IAM角色。

   以下IAM策略详细说明了所需的IAM权限。

   ```
   {
      "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. 使用您定义的所有定价维度的记录对`RegisterUsage`API操作进行测试调用。

将容器产品与 License Manager 集成

要将基于容器的产品与 License Manager 集成，请执行以下操作：

1. 设置调用 License Manager 的IAM权限。有关更多信息，请参阅 AWS License Manager 集成先决条件。

2. 下载 AWS SDK.

> **Note**
>
> 不要在软件中配置 AWS 凭据。 AWS 当您的容器在亚马逊EC2实例、亚马逊ECS任务或亚马逊 EKS pod 中运行时，系统会在运行时自动获取买家证书。

3. 向您的产品添加许可证检查。

   无论在何处进行许可证检查，您的产品都可以在任何地方调用`CheckoutLicense`API操作人员。要查看许可证，您的产品必须了解：

   1. 许可证的可信颁发者 (AWS Marketplace)
   2. 应用程序的产品SKU（产品 ID）
   3. 查看此应用程序的权利

   根据您设置的定价许可证的类型，API电话会有所不同。

4. 在上发布您的产品清单 AWS Marketplace。

License Manager 的API操作

要管理存储在客户 License Manager 帐户中的许可证，您的软件可以使用以下API操作：

- `GetLicense`— 软件可以查询的。API它会检索已购买许可证的状态（即已过期或即将到期），并向客户发送状态通知。
- `CheckoutLicense` – 发现用户已购买的许可证。当用户消耗了一定数量的许可证时，您还可以使用该`CheckoutLicense`API操作来更新许可证数量。使用 `CheckoutLicense`，您可以继续签出客户使用的许可证数量。当客户耗尽所有许可证时，此调用将返回错误。有关建议运行 `CheckoutLicense` 的节奏的信息，请参阅the section called “许可证续订和升级”。
- `ExtendLicenseConsumption` – 如果是浮动维度，当软件签出许可证时，许可证会在 60 分钟后自动返回到池中。如果要延长许可证的签出时间，请使用该`ExtendLicenseConsumption`API操作将许可证再延长 60 分钟。
- `CheckInLicense`— 如果是浮动维度，当您要将许可证返回到授权池时，请使用`CheckInLicense`API操作。
- `ListReceivedLicenses`API— 列出买方购买的许可证。

许可证续订和升级

客户可以在 AWS Marketplace 管理门户上续订或升级许可证。在他们进行额外购买后， AWS Marketplace 会生成反映新权利的新版本的许可证。您的软件使用相同的API操作读取新的授权。您不必在 License Manager 集成方面执行任何不同的操作即可处理续订和升级。

由于许可证续订、升级、取消等，我们建议您的产品在使用产品时定期调用`CheckoutLicense`API操作。通过定期使用该`CheckoutLicense`API操作，产品可以检测权利的变化，例如升级和到期。

我们建议您每 15 分钟进行一次`CheckoutLicense`API呼叫。

将 “容器随处 AWS Marketplace 可用” 产品与 License Manager 集成

按照以下说明 AWS License Manager 与 AWS Marketplace 适用于 Amazon Anywhere、Amazon Anywhere、Amazon EKS 或本地基础设施的 for Contain EC2 er ECS Anywhere 产品集成。

有关 License Manager 与集成的一般信息 AWS Marketplace，包括可用的许可模式，请参阅集装箱产品的合同定价 AWS License Manager。有关 AWS License Manager的更多信息，请参阅《AWS License Manager 用户指南》和《AWS CLI 命令参考》中的 AWS License Manager 部分。

将 “容器随处 AWS Marketplace 可用” 产品与 License Manager 集成

按照以下说明将你的 for Container AWS Marketplace s Anywhere 产品与集成 AWS License Manager。

将 for Container AWS Marketplace s Anywhere 产品与 License M

1. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
2. 执行以下步骤，为您的容器产品创建产品 ID。您将在容器映像中使用此 ID，以便在后续步骤中进行许可证检查。

   a. 从菜单栏中，展开资产，然后选择容器。
   b. 为您的产品输入面向客户的名称，然后选择创建。您以后可以更改此名称。
   c. 记下产品 ID。之后在创建或更新产品定价详情时会使用它。

   > **Tip**
   > 如果您丢失了产品 ID，可以从 “资产” 菜单中选择 “容器”，在 “容器” 中找到它。 AWS Marketplace 管理门户 容器页面显示您的产品及其关联产品的列表IDs。

3. 下载最新的公共版本 AWS SDK，然后将其安装到您的容器应用程序中。你可以在 T ools to Build O AWS SDK n 中找到你喜欢的安装说明AWS。

> **Note**
>
> 要从 Amazon EKS Anywhere 或未提供的 Kubernetes 集群调用许可证管理器API操作 AWS，必须使用支持的。 AWS SDK要查看支持的列表 AWS SDKs，请参阅使用支持的 AWS SDK。

4. 使用自定义凭证提供程序创建 AWS License Manager 客户端，以便它可以为部署在 AWS 本地和本地的容器应用程序提供凭证。有关自定义凭证提供程序的完整源代码 `LicenseCredentialProvider`，请参阅以下各部分：

   - `LicenseManagerCredentialsProvider` - Java 实现。
   - `LicenseManagerCredentialsProvider` - Golang 实现。

   `LicenseCredentialsProvider`通过添加`LicenseManagerTokenCredentialsProvider`扩展 AWS SDK的默认凭证提供者链以供本地使用。这通过在本地环境中使用 License Manager OIDC 颁发的身份令牌来提供凭证。必须在应用程序类路径中包含 `LicenseCredentialsProvider` 的源代码。

   > **Note**
   >
   > 扩展`DefaultCredentialsProvider`允许同一个容器应用程序在本地环境中运行 AWS 和在本地环境中运行时获得证书。如果容器应用程序已经使用自定义凭证提供程序链而不是默认凭证提供程序链，则也可以通过将 `LicenseManagerTokenCredentialsProvider` 添加到自定义链中来进行扩展。

   以下代码段是使用 Java 创建 AWS License Manager 客户端的示例。

```
LicenseManagerClientBuilder clientBuilder =
 LicenseManagerClient.builder().credentialsProvider(LicenseCredentialsProvider.create());
```

5. 使用产品中每个付费容器镜像中的`aws license-manager checkout-license`命令调用该`CheckoutLicense`API操作。这将检查买家是否有权使用您的应用程序许可证。如果买家有权使用该应用程序，则 `CheckoutLicense` 成功并返回所请求的权利及其值。如果买家无权使用该应用程序，则 `CheckoutLicense` 会抛出异常。

调用该CheckoutLicenseAPI操作时需要以下参数：

- CheckoutType – 有效值为 PROVISIONAL 或 PERPETUAL。
  - PERPETUAL – 当已签出的权利池中的权利数量将用尽时使用。

    示例：买家有权处理 500 GB 的数据。当他们继续处理数据时，会消耗 500 GB 池的数量并将其耗尽。
  - PROVISIONAL 用于浮动许可证权利，其中权利从池中签出，并在使用后返回。

    示例：某用户被授予应用程序 500 个并发用户的权利。当用户登录或注销时，用户会被消耗或返回到 500 个用户的池中。要了解有关浮动许可证权利的更多信息，请参阅使用 License Manager 管理浮动许可证特权。
- ClientToken – 区分大小写的唯一标识符。我们建议UUID对每个唯一请求使用随机值。
- Entitlements – 待签出的权利列表。
  - 对于特征权利，请按以下方式提供 Name 和 Unit 属性。

```
{
  "Name": "<Entitlement_Name>",
  "Unit": "None"
}
```

  - 对于技术特权，请按以下方式提供 Name、Unit 和 Count 属性。

```
{
  "Name": "<Entitlement_Name>",
  "Unit": "<Entitlement_Unit>",
  "Value": <Desired_Count>
}
```

- KeyFingerprint – 由 AWS Marketplace 颁发的许可证的密钥指纹是 aws:294406891311:AWS/Marketplace:issuer-fingerprint。使用此密钥指纹可确保许可证由不可靠的实体颁发 AWS Marketplace ，而不是由不可靠的实体颁发。
- ProductSKU— 在之前的步骤 AWS Marketplace 管理门户 中生成的产品 ID。

以下代码段是使用CheckoutLicenseAPI操作进行调用的示例。 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**
>
> 要查看许可证，容器应用程序需要出站网络访问权限才能使用 License Manager。部署在本地的应用程序可能会遇到出站网络访问不可靠或缓慢的情况。这些应用程序在调用 License Manager 时应包括够的重试次数。有关更多信息，请参阅 针对本地部署集成 License Manager 的最佳实操。

6. 定期致电`CheckoutLicense`API运营部门，以确定由于续订、升级或取消而对客户的许可证进行的任何更改。 AWS Marketplace调用频率取决于应用程序。我们建议每天检查一次许可证，以便在没有任何买家干预的情况下自动获取更改。

   部署在本地的应用程序可能具有不可靠的出站网络访问权限，无法定期检查许可证。在这种情况下，应用程序应使用缓存许可证以获得足够的弹性。有关更多信息，请参阅 针对本地部署集成 License Manager 的最佳实操。

7. 将 `CheckoutLicense` 调用与容器应用程序集成后，基于更改生成 Docker 容器映像的新版本。

8. 更新应用程序的 Helm 图表以接受 Kubernetes 密钥作为可选输入，其中包含使用许可证管理器访问许可证的配置。APIs配置密钥将包含由 License Manager 颁发的身份令牌和一个 AWS Identity and Access Management 角色，前面描述的自定义凭据提供者将使用该角色来获取在本地部署容器应用程序APIs时调用 License Manager 的 AWS 凭证。另外，将 AWS 区域 作为输入添加，默认值为 `us-east-1`。

   在本地部署容器应用程序的买家可以通过容器产品的购买者体验创建 Kubernetes 秘密。 AWS Marketplace 提供 Kubernetes 密钥名称作为 `helm install` 命令的输入。配置密钥采用以下格式配置。

```
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. 更新集成的容器镜像的 Helm 图表中的应用程序部署模板， AWS License Manager 使其包含以下内容：

   - pod 的服务账户 — 在 Amazon 上部署 Helm 需要服务账户EKS。它用于通过在容器映像上为服务帐号设置IAM角色来获得调用 License Manager API 操作的权限。有关服务帐号IAM角色的更多信息，请参阅服务帐号的IAM角色。
   - 本地部署的许可证访问权限 — 需要许可证配置密钥才能提供凭据和相应权限，以便在本地环境中为 Helm 部署调用 License Manager API 操作。买家将根据 AWS Marketplace 买家体验生成许可证秘密并将其提供给 Helm。

   以下代码片段是示例部署规范，其中包含服务帐号、许可证配置和映像拉取密钥。

```
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**
>
> 许可证配置密钥是可选的。买家仅在本地部署时使用该值。对于 AWS 部署，部署规范必须包含 License Manager 集成映像的服务帐户。

10. EKS通过执行以下各节中的步骤，在本地和亚马逊上测试 License Manager 集成：
    a. 在本地测试 License Manager 集成
    b. 在亚马逊上测试 License Manager 集成 EKS
11. 成功验证本地 AWS 和本地的 License Manager 集成后，您可以按照中的步骤创建容器产品清单创建容器产品。

在本地测试 License Manager 集成

你可以使用 minikube 或任何其他设置来在本地测试任何 Kubernetes 集群上的 License Manager 集成。确保 Kubernetes 集群具有出站互联网访问权限以调用 License Manager 操作。API

要在本地测试 License Manager 集成，请执行以下操作：

1. 使用所需权利在测试卖家账户中创建测试许可证。要设置测试许可证，请参阅 “AWS License Manager API参考” CreateLicense中的。或者，使用以下脚本创建测试许可证，然后为测试买家账户创建许可授权，以使用该许可证。以下脚本使用测试卖家账户凭证。

```
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. 使用之前定义的密钥格式使用许可证令牌和IAM角色创建 Kubernetes 密钥。使用 License Manager `CreateToken` API 操作生成许可证令牌。然后，使用IAM`CreateRole`API操作创建具有权限和信任策略的IAM角色。请参阅以下示例脚本中的示例。以下脚本使用测试买家账户凭证。

```
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. 设置托管在外部的任何 Kubernetes 集群。 AWS使用它来测试容器应用程序是否可以 AWS License Manager API从其他环境连接到， AWS 以及自定义凭证提供程序是否已很好地集成到应用程序中。
4. 将之前生成的许可证令牌和IAM角色部署到本地 Kubernetes 集群中。

```
kubectl create secret generic "awsmp-license-access-config" \
--from-literal=license_token=${TOKEN} \
--from-literal=iam_role=${ROLE_ARN}
```

5. 使用密码名称作为输入通过 Helm 部署应用程序，并验证应用程序是否可以调用 License Manager API 操作来执行授权检查。有关 Helm 和部署规范的更改，请参阅将 “容器随处 AWS Marketplace 可用” 产品与 License Manager 集成中的步骤 9。

在亚马逊上测试 License Manager 集成 EKS

您也可以在亚马逊上测试 License Manager 的集成EKS。测试以确保应用程序可以在没有许可证配置密钥的情况下调用 License Manager API 操作。此外，请确保该服务帐号可用于为服务帐号设置IAM角色 (IRSA)，并为应用程序提供相关凭证。

在亚马逊上测试 License Manager 集成 EKS

1. 使用所需权利在测试卖家账户中创建测试许可证。请参阅CreateLicense API参考资料来设置您的测试许可证，或者使用以下脚本创建测试许可证，然后向测试买家账户创建使用许可证的许可授权。以下脚本使用测试卖家账户凭证。

```
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. 创建包含所需配置的测试 Amazon EKS 集群，或运行以下命令以使用默认配置。

```
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. 为现有集群创建服务账号并将其与IAM角色关联。以下命令使用创建IAM角色`AWSLicenseManagerConsumptionPolicy`。然后，该命令将其附加到应在其中部署 License Manager 集成映像的 Amazon EKS 集群的`test_sa`服务账户。因此，服务帐号可以获得相应的凭据来调用 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. 通过与前一个命令关联IAM角色的服务帐号中的 Helm 部署应用程序。验证应用程序是否可以调用 License Manager API 操作来执行授权检查。

使用 License Manager 管理浮动许可证特权

使用浮动许可证，当用户登录应用程序时，将从可用许可证池中消耗许可证。用户注销后，许可证将重新添加到可用许可证池中。

对于浮动许可证，应用程序使用该`CheckoutLicense`API操作在使用资源时从授权池中检出授权。`CheckoutLicense`API操作的响应包括许可证消耗令牌，该令牌是结账的唯一标识符。许可证消耗令牌可以对已签出的权利执行其他操作，例如将其签回许可证池或延长签出时间。

当资源不再使用时，应用程序会使用该`CheckInLicense`API操作将授权签回池中。

```
aws license-manager check-in-license \
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

如果许可证未能签回池中，例如，如果应用程序在操作过程中崩溃，则权利将在 60 分钟后自动签回池中。因此，如果资源的使用时间超过 60 分钟，则最佳实操是将权利保留在资源池之外。为此，只要资源正在使用，就使用该`ExtendLicenseConsumption`API操作。

```
aws license-manager extend-license-consumption \
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

针对本地部署集成 License Manager 的最佳实操

本地环境中的容器应用程序部署可能会遇到不可靠的出站网络访问。使用以下最佳实操来增加弹性，以避免因互联网连接不畅导致的潜在问题而导致买家的服务中断：

- 充分重试 — 临时网络问题可能会使您的应用程序无法连接到。 AWS License Manager实现最长 30 分钟的重试，并关闭指数级回退。这可以帮助避免短期中断或网络问题。
- 避免硬限制 – 部署在联网集群中的应用程序可以定期检查许可证，以确定由于升级或续订而导致的任何更改。由于出站访问不可靠，应用程序可能无法识别这些更改。应用程序应尽可能避免因无法通过 License Manager 检查许可证而中断向买家提供的服务。当许可证到期时，应用程序可能会依赖免费试用或开源体验，并且无法检查许可证是否有效。
- 通知客户 - 使用缓存许可证时，许可证的任何更改（包括续订或升级）都不会自动反映在正在运行的工作负载上。通知您的客户（他们必须暂时允许对应用程序进行出站访问），以便应用程序可以更新其缓存的许可证。例如，通过应用程序本身或通过其文档通知客户。同样，当回退到较低的功能集时，请通知客户其权利已用尽或许可证已过期。然后，他们可以选择升级或续订。

`LicenseManagerCredentialsProvider` - Java 实现。

`LicenseCredentialsProvider`通过添加`LicenseManagerTokenCredentialsProvider`扩展AWS SDK的默认凭证提供者链以供本地使用。

`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在本地环境中使用 License Manager OIDC 颁发的身份令牌提供凭证。必须在应用程序类路径中包含 LicenseCredentialsProvider 的源代码。

```
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 实现。

LicenseCredentialsProvider

LicenseCredentialsProvider通过添加LicenseManagerTokenCredentialsProvider扩展AWS SDK的默认凭证提供者链以供本地使用。

```
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`在本地环境中使用 License Manager OIDC 颁发的身份令牌提供凭证。必须在应用程序类路径中包含 `LicenseCredentialsProvider` 的源代码。

```
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
 }
}
```

亚马逊关于集装箱商品的SNS通知

要接收通知，您需要订阅商品创建期间提供给您的 AWS Marketplace 亚马逊简单通知服务 (AmazonSNS) 主题。这些主题提供有关客户产品订阅发生更改的通知。例如，您可以使用它们知道客户何时接受专属优惠。

> **Note**
>
> 在商品创建过程中，会为您的商品创建一个 Amazon SNS 主题。要订阅通知，您需要亚马逊SNS主题的亚马逊资源名称 (ARN)（例如`arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`）。服务器产品的卖家门户中不可用。ARN请联系 AWSMarketplace 运营团队申请ARN。

以下 Amazon SNS 主题适用于容器商品：

- Amazon SNS 主题：`aws-mp-subscription-notification` – 当买家订阅或取消订阅产品时，此主题会向您发送通知。此通知适用于按小时定价模式，包括长期按小时定价模式和按年计费模式。

Amazon SNS 主题：`aws-mp-subscription-notification`

`aws-mp-subscription-notification` 主题中的每条消息都采用以下格式。

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123",
    "isFreeTrialTermPresent":"true"
}
```

这些区域有：*<action-name>* 将因通知而异。可能的操作包括：

- `subscribe-success`
- `subscribe-fail`
- `unsubscribe-pending`
- `unsubscribe-success`

`offer-identifier` 仅当优惠为专属优惠 时，通知中才会显示。

为亚马逊SQS队列订阅亚马逊主题 SNS

我们建议通过 Amazon SQS 队列订阅所提供的SNS主题。有关创建SQS队列和为队列订阅主题的详细说明，请参阅《亚马逊简单通知服务开发者指南》中的 “为亚马逊SQS队列订阅亚马逊SNS主题”。

> **Note**
>
> 您只能订阅 AWS 账户 用于销售产品的用户 AWS Marketplace SNS的主题。但是，您可以将消息转发到其他帐户。有关更多信息，请参阅《亚马逊简单通知服务开发者指南》中的向其他账户中的亚马逊SQS队列发送亚马逊SNS消息。

轮询SQS队列以获取通知

在您为SQS队列订阅SNS主题后，消息将存储在中SQS。您需要定义一项服务，以不断轮询该队列，查找消息并相应地处理它们。

机器学习产品

作为卖家，您可以使用 AWS Marketplace 创建机器学习 (ML) 算法和模型，供买家部署 AWS。以下列出了两种类型的亚马逊 SageMaker 商品 AWS Marketplace：

模型包

一种预训练模型，用于进行预测，无需买家进一步培训。

算法

一种模型，要求买家在做出预测之前提供训练数据。包括训练算法。

买家可通过亚马逊 SageMaker 控制台或购买这些商品 AWS Marketplace。买家可以查看产品描述、文档、客户评价、定价及支持信息。当他们订阅模型包产品或算法产品时，它会添加到 SageMaker 控制台上的产品列表中。买家还可以使用 AWS SDKs、 AWS Command Line Interface (AWS CLI) 或 SageMaker 控制台来创建完全托管的REST推理端点或对批量数据进行推理。

要获得有关通过 Amazon 创建机器学习产品的支持 SageMaker，请联系AWS Marketplace 卖家运营团队。

机器学习产品入门

AWS Marketplace 使用 Amazon 支持两种机器学习产品类型 SageMaker。模型包产品和算法产品这两种类型都会生成用于预测的可部署推理模型。

SageMaker 模型包

A mazon SageMaker 模型包产品包含预训练模型。可以部署预训练的模型， SageMaker 以便实时或分批进行推断或预测。本产品包含经过训练的推理组件，其中包含模型构件（如有）。作为卖家，您可以使用自己的模型训练模型 SageMaker 或自带模型。

SageMaker 算法

买家可以使用SageMaker算法产品来执行完整的机器学习工作负载。一个算法产品有两个逻辑组件：训练和推理。在中 SageMaker，买家使用自己的数据集使用您的训练组件创建训练作业。训练组件中的算法完成后，它会生成机器学习模型的模型工件。 SageMaker 将模型工件保存在买家的亚马逊简单

存储服务 (Amazon S3) 存储桶中。 然后 SageMaker，购买者可以在中部署您的推理组件以及生成的模型工件，以实时或批量执行推理（或预测）。

部署推理模型

无论推理模型是根据模型包还是算法创建的，都有两种部署方法：

- 端点-此方法 SageMaker 用于部署模型和创建API端点。买家可以将此端点用作其后端服务的一部分，为其应用程序提供支持。当数据发送到端点时，将其 SageMaker 传递到模型容器并在API响应中返回结果。端点和容器会继续运行，直到被买家停止。

> **Note**
>
> 在中 AWS Marketplace，端点方法被称为实时推理，在 SageMaker 文档中，它被称为托管服务。有关更多信息，请参阅在 Amazon 中部署模型 SageMaker。

- 批量转换任务 – 在此方法中，买家将数据集存储在 Amazon S3 中进行推理。当批量转换任务启动时， SageMaker 部署模型，将数据从 S3 存储桶传递到模型的容器，然后将结果返回到 S3 存储桶。作业完成后， SageMaker 停止作业。有关更多信息，请参阅使用批量转换获取整个数据集的推理。

> **Note**
>
> 这两种方法对模型都是透明的，因为将数据 SageMaker传递给模型并将结果返回给买方。

安全性和知识产权

对于从中获得的模型和算法，亚马逊会 SageMaker 保护您的知识产权和买家数据 AWS Marketplace。

保护知识产权

创建产品时，代码将打包在 Docker 容器映像中。有关更多信息，请参阅本指南下文中的在里面准备好你的产品 SageMaker。上传容器映像时，映像和构件在传输过程中和静止状态都会被加密。在发布之前，还会对映像进行漏洞扫描。

为了帮助保护您的知识产权，仅 SageMaker 允许买家通过 AWS 服务 端点访问您的产品。买家无法直接访问或拉取容器镜像或模型构件，也无法访问底层基础设施。

无网络访问

与买家创建的 SageMaker 模型和算法不同，当买家发布您的产品时 AWS Marketplace，模型和算法是在没有网络访问权限的情况下部署的。 SageMaker在无法访问网络或 AWS 服务 端点的环境中部署映像。例如，容器映像无法对互联网、VPC 端点上的服务或任何其他 AWS 服务进行出站 API 调用。

客户数据安全

您的商品在买家的商品 SageMaker 中出售 AWS 账户。因此，当买家使用您的产品进行数据推理时，您作为卖家无法访问他们的数据。

对于算法产品，模型构件在每次训练作业结束后由您的训练映像输出。模型构件存储在买家的账户中。当买家使用您的推理映像部署模型时，会使用训练作业中的模型构件。要保护模型构件中可能包含的任何知识产权，请在输出之前对其进行加密。

> **Important**
>
> 此安全模型可防止您的代码在运行时访问互联网。因此，您的代码无法使用互联网上的资源或库，因此请将依赖项打包到 Docker 容器映像中。如果您选择对训练作业中输出的构件进行加密，这一点尤为重要。运行时无法通过互联网访问加密和解密构件的密钥。它们必须与您的映像打包在一起。

有关更多信息，请参阅 Amazon 安全 SageMaker。

机器学习产品

您可以从几种可用的 Amazon SageMaker 产品定价模式中进行选择。订阅您的产品的买家会自己 SageMaker 购买该产品 AWS 账户。买家的价格是在其 AWS 账户 中运行的资源的基础设施费用和您设定的产品定价的组合。

基础设施定价

买家需承担使用您的产品 SageMaker 时的所有基础设施成本。这些费用由亚马逊定价页面设定，AWS 并可在亚马逊定 SageMaker 价页面上找到。

软件定价

由您决定向买家 AWS Marketplace 收取使用您的产品的软件价格。在向中添加机器学习产品时，您可以设置定价和条款 AWS Marketplace。

每种实例类型的所有基础设施和软件价格均在买家订阅 AWS Marketplace 之前的产品列表页面上显示给买家。

免费定价

您可以选择免费提供您的产品。在这种情况下，买家只需支付基础设施成本。

每小时定价

您可以按每小时的价格向您的产品提供运行软件的每个实例的价格 SageMaker。您可以为运行软件的每种实例类型收取不同的小时价格。当买家运行您的软件时， AWS Marketplace 跟踪使用情况，然后相应地向买家开具账单。用量按比例计算到分钟。

对于模型包产品，买家可以通过两种不同的方式运行您的软件。他们可以持续托管端点以执行实时推理或对数据集运行批量转换作业。您可以为买家运行软件的两种方式设置不同的定价。

对于算法产品，除了确定执行推理的价格（如前所述）外，您还可以确定训练作业的每小时价格。您可以为训练图像支持的每种实例类型收取不同的小时价格。

按小时计费的年度合同

在合同选项中，您可以指定固定的预付费用，并在订阅时向客户开具全额合同的发票。在年度合同到期时，任何继续运行的实例都将按您设定的小时费率计费。

推理定价

当买家通过托管端点来运行您的软件以持续执行实时推理时，您可以选择设置每次推理的价格。

> **Note**
>
> 批量转换流程始终使用按小时定价。算法产品的训练作业也始终使用小时定价。您可以独立于推理定价来设置这些价格，也可以相互设置这些价格。

默认情况下，在推理定价中，每次调用您的终端节点都会向您的买家 AWS Marketplace 收费。但是，在某些情况下，您的软件会在一次调用（也称为迷你批处理）中处理一批推理。对于端点部署，您可以指定 AWS Marketplace 应向买家收取单次调用费用的自定义推理数量。为此，请在调用的HTTP响应标头中加入自定义计量标头，如以下示例所示。此示例显示了一个向买家收取三次推理费用的调用。

```
X-Amzn-Inference-Metering: {"Dimension": "inference.count", "ConsumedUnits": 3}
```

> **Note**
>
> 对于推理定价， AWS Marketplace 仅向买方收取HTTP响应代码所在2XX的请求。

免费试用

或者，您可以为产品创建免费试用并定义免费试用的天数。免费试用可能为 5-120 天。在免费试用期间，买家可以随意运行您的软件，无需为软件付费。在免费试用期间，买家仅需要支付基础设施费用。试用期结束后，他们将按正常的软件价格和基础设施成本收费。

> **Note**
>
> 您只能为按小时计费的优惠创建免费试用。您无法为具有推理定价的产品创建免费试用。

当买家订阅免费试用的产品时，他们会收到一封欢迎电子邮件。该消息包括免费试用期限、计算的到期日期以及取消订阅的详细信息。在到期日期的前 3 天，将发送提醒电子邮件。

如果您在中为产品提供免费试用 AWS Marketplace，则表示您同意免费试用的特定退款政策。

> **Note**
>
> 有关机器学习专属优惠的信息，请参阅专属优惠。

价格更改

作为卖家，您可以联系 AWSMarketplace 卖家运营团队来更改机器学习产品的定价。提供产品 ID 和新的定价详情。新价格在 90 天后生效。此外，您必须等待 90 天才能进行第二次价格变动。此限制也适用于向现有支持的实例列表中添加新的实例类型。例如，如果您在 2023 年 11 月 1 日提高了机器学习产品的价格，则可以在 2024 年 1 月 30 日之后添加新的实例类型或进行第二次价格更改。

在里面准备好你的产品 SageMaker

在发布商品之前 AWS Marketplace，您必须在 Amazon 上进行准备 SageMaker。准备产品需要执行三个步骤：

1. 将代码打包成映像 – 要准备模型包或算法产品，必须为产品创建 Docker 容器映像。
2. 上传映像 – 将您的代码打包到容器映像中并在本地对其进行测试后，上传映像并对其进行扫描以查找已知漏洞。先修复所有漏洞，然后再继续。
3. 创建您的亚马逊 SageMaker资源— 成功扫描图像后，它们可用于在中创建模型包或算法资源 SageMaker。

将代码打包成映像

机器学习产品 AWS Marketplace 使用 Amazon SageMaker 来创建和运行您为买家提供的机器学习逻辑。 SageMaker 运行包含您的逻辑的 Docker 容器镜像。 SageMaker在安全且可扩展的基础架构中运行这些容器。有关更多信息，请参阅 安全性和知识产权。

主题

- 我要创建哪种类型的容器映像？
- 模型包映像
- 算法映像

我要创建哪种类型的容器映像？

两种类型的容器映像，即推理映像和训练映像。

要创建模型包产品，您只需要一个推理映像。有关详细说明，请参阅 模型包映像。

要创建算法产品，您需要训练映像和推理映像。有关详细说明，请参阅 算法映像。

要将代码正确打包到容器镜像中，容器必须符合 SageMaker文件结构。容器必须公开正确的端点，以确保服务可以将数据传入和传出您的容器。以下各部分解释了此过程的细节。

> **Important**
>
> 出于安全目的，当买家订阅您的容器化产品时，Docker 容器在隔离（无互联网连接）的环境中运行。在您创建容器时，请勿通过互联网进行传出调用，否则将失败。对的调用 AWS 服务 也会失败。想要了解更多信息，请参阅 安全性和知识产权部分。

或者，在创建推理和训练映像时，可以使用可用的深度学习容器映像中的容器作为起点。这些映像已经用不同的机器学习框架正确打包。

模型包映像

Amazon SageMaker 模型包是一种预先训练的模型，可以进行预测，不需要买家接受任何进一步的培训。

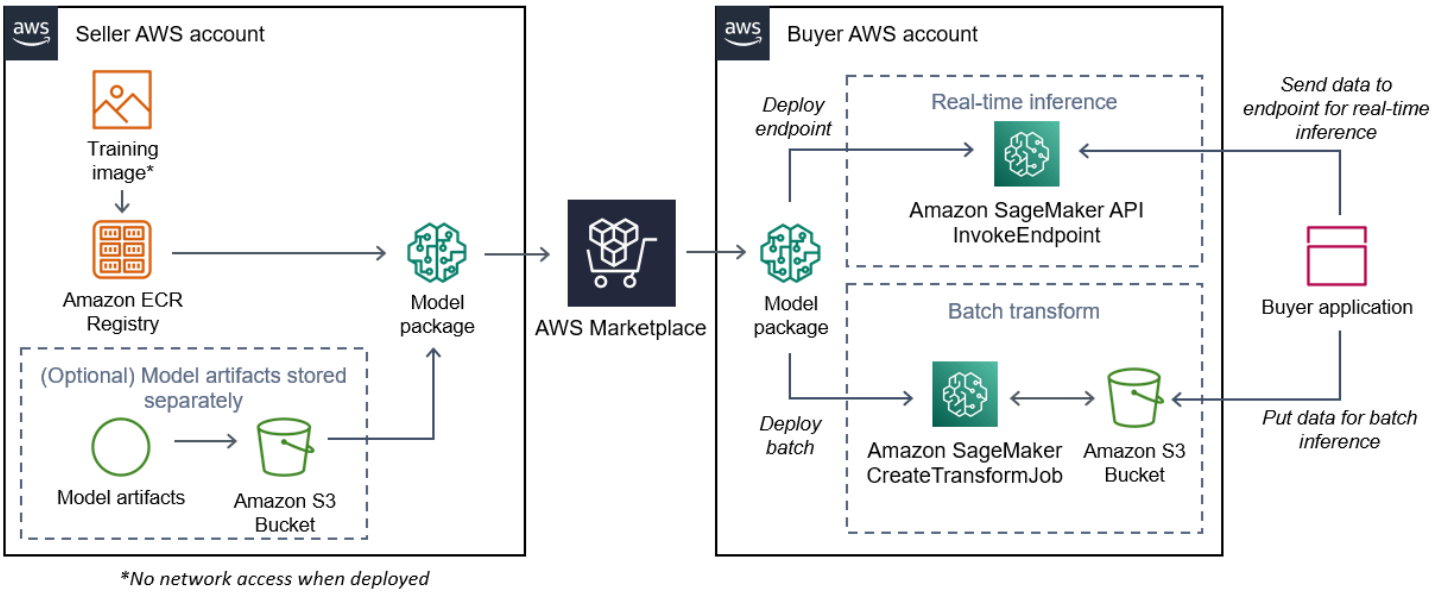
模型包包含以下组件：

- 存储在 Amazon Elastic Container Registry (Amazon ECR) 中的推理映像
- （可选）模型构件，单独存储在 Amazon S3 中

> **Note**
>
> 模型构件是模型用来进行预测的文件，通常是您自己的训练过程的结果。构件可以是模型所需的任何文件类型，但必须压缩为 .tar.gz。对于模型包，它们可以捆绑在您的推理图像中，也可以单独存储在 Amazon 中。 SageMaker存储在 Amazon S3 中的模型构件将在运行时加载到推理容器中。发布您的模型包时，这些项目会发布并存储在买家无法直接访问的 AWS Marketplace 自有的 Amazon S3 存储桶中。

> **Tip**
>
> 如果您的推理模型是使用 Gluon、Keras、mxNet、、、 TensorFlow-Lite 或 ONNX 等深度学习框架构建的 PyTorch TensorFlow，请考虑使用 Amazon Neo。 SageMaker Neo 可以自动优化部署到特定云实例类型（例如，`ml.c4` 和 `ml.p2` 等）的推理模型。有关更多信息，请参阅《Amazon SageMaker 开发者指南》中的使用 Neo 优化模型性能。

下图显示了发布和使用模型包产品的工作流程。



1. 卖家创建推理映像（部署后无法访问网络）并将其推送到 Amazon ECR 注册表。

   模型构件可以捆绑在推理映像中，也可以单独存储在 S3 中。
2. 然后，卖家在亚马逊创建模型包资源 SageMaker 并在上发布其机器学习产品 AWS Marketplace。
3. 买家订阅机器学习产品并部署模型。

> **Note**
>
> 该模型可以部署为实时推理端点，也可以部署为批处理作业，以一次性获取整个数据集的预测。有关更多信息，请参阅部署模型以进行推理。

4. SageMaker 运行推理图像。卖家提供的任何未捆绑在推理映像中的模型构件都将在运行时动态加载。
5. SageMaker 使用容器的 HTTP 端点将买方的推理数据传递到容器并返回预测结果。

为模型包创建推理映像

本部分提供了将推理代码打包成模型包产品的推理映像的演练。此过程包括以下步骤：

步骤

- 步骤 1：创建容器映像
- 第 2 步：在本地构建和测试映像

推理映像是包含您的推理逻辑的 Docker 映像。容器在运行时公开 HTTP 端点， SageMaker 以允许将数据传入和传出您的容器。

> **Note**
>
> 以下只是推理映像打包代码的一个示例。有关更多信息，请参阅将 Docker 容器与一起使用， SageMaker以及中的AWS Marketplace SageMaker 示例。 GitHub
> 为了简单起见，以下示例使用了 Web 服务 Flask，但不被视为生产就绪。

步骤 1：创建容器映像

要使推理映像与之兼容 SageMaker，Docker 镜像必须公开 HTTP 端点。当您的容器运行时，SageMaker 会将买方输入传递给容器的 HTTP 端点，以便进行推理。推理结果返回在 HTTP 响应的正文中。

以下演练在使用 Linux Ubuntu 发行版的开发环境中使用 Docker CLI。

- 创建 Web 服务器脚本
- 为容器运行创建脚本
- 创建 `Dockerfile`
- 打包或上传模型构件

创建 Web 服务器脚本

此示例使用名为 Flask 的 Python 服务器，但您可以使用任何适用于您的框架的 Web 服务器。

> **Note**
>
> 为了简单起见，这里使用 Flask。它不被视为生产就绪的 Web 服务器。

创建一个 Flask Web 服务器脚本，在使用的 TCP 端口 8080 上为两个 HTTP 端点提供服务。SageMaker 以下是两个预期的端点：

- `/ping`— SageMaker 向此端点发出 HTTP GET 请求，以检查您的容器是否准备就绪。当您的容器准备就绪时，它会使用 HTTP 200 响应代码来响应此端点的 HTTP GET 请求。

- `/invocations`— SageMaker 向此端点发出 HTTP POST 请求以进行推断。用于推理的输入数据在请求正文中发送。用户指定的内容类型在 HTTP 头中传递。响应的主体是推理输出。有关超时的详细信息，请参阅创建机器学习产品的要求和最佳实操。

./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)
```

在前面的示例中，没有实际的推理逻辑。对于实际的推理映像，请将推理逻辑添加到 Web 应用中，以便它处理输入并返回实际预测。

您的推理图像必须包含其所有必需的依赖项，因为它无法访问互联网，也无法调用任何 AWS 服务依赖项。

> **Note**
>
> 实时推断和批量推断都需要同样的代码

为容器运行创建脚本

创建一个名为的脚本`serve`，该脚本在 SageMaker 运行 Docker 容器镜像时运行。以下脚本启动 HTTP Web 服务器。

`./serve`

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker
flask run --host 0.0.0.0 --port 8080
```

创建 **`Dockerfile`**

在您的构建上下文中创建 `Dockerfile`。此示例使用 Ubuntu 18.04，但您可以从适用于您的框架的任何基础映像开始。

`./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 will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

`Dockerfile` 将之前创建的两个脚本添加到映像中。`serve` 脚本的目录已添加到 PATH 中，因此它可以在容器运行时运行。

打包或上传模型构件

提供从训练模型到推理映像的模型构件的两种方法如下：

- 与推理映像一起静态打包。
- 在运行时动态加载。由于它是动态加载的，因此您可以使用相同的映像来打包不同的机器学习模型。

如果要将模型构件与推理映像打包，请将构件包含在 `Dockerfile` 中。

如果您想动态加载模型构件，请将这些构件单独存储在 Amazon S3 的压缩文件 (.tar.gz) 中。创建模型包时，请指定压缩文件的位置，并在运行容器`/opt/ml/model/`时将内容 SageMaker 提取并复制到容器目录中。发布您的模型包时，这些构件会发布并存储在买家无法直接访问的 AWS Marketplace 所有 Amazon S3 存储桶中。

第 2 步：在本地构建和测试映像

在构建上下文中，现在存在以下文件：

- `./Dockerfile`
- `./web_app_serve.py`
- `./serve`
- 您的推理逻辑和（可选）依赖项

接下来，构建、运行和测试容器镜像。

构建映像：

在构建上下文中运行 Docker 命令来构建和标记映像。此示例使用标记 `my-inference-image`。

```
sudo docker build --tag my-inference-image ./
```

运行此 Docker 命令构建映像后，当 Docker 根据您的 `Dockerfile` 的每一行构建映像时，您该会看到输出。完成后，您应看到类似如下所示的内容：

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```

本地运行

构建完成后，您可以在本地测试映像。

```
sudo docker run \
  --rm \
  --publish 8080:8080/tcp \
  --detach \
  --name my-inference-container \
  my-inference-image \
  serve
```

下面是有关命令的详细信息：

- `--rm` – 容器停止后自动将其移除。
- `--publish 8080:8080/tcp`— 公开端口 8080 以模拟向其 SageMaker 发送 HTTP 请求的端口。
- `--detach` – 在后台运行容器。
- `--name my-inference-container` – 为正在运行的容器提供名称。
- `my-inference-image` – 运行构建的映像。
- `serve`— 运行运行容器时 SageMaker 运行的相同脚本。

运行此命令后，Docker 会根据您构建的推理映像创建一个容器并在后台运行它。容器运行 `serve` 脚本，该脚本会启动您的 Web 服务器以进行测试。

测试 ping HTTP 端点

SageMaker 运行您的容器时，它会定期对端点执行 ping 操作。 当端点返回状态码为 200 的 HTTP 响应时，它会发出信号 SageMaker ，表示容器已准备好进行推理。您可以通过运行以下命令进行测试，将测试端点并包含响应头。

```
curl --include http://127.0.0.1:8080/ping
```

示例输出如下：

```
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
```

测试推理 HTTP 端点

当容器通过向你的 ping 返回 200 状态码表示已准备就绪时，通过POST请求将推理数据 SageMaker 传递到 `/invocations` HTTP 终端节点。通过运行以下命令来测试推理点。

```
curl \
  --request POST \
  --data "hello world" \
  http://127.0.0.1:8080/invocations
```

示例输出如下：

```
{"prediction": "a", "text": "hello world"}
```

这两个 HTTP 端点正常工作后，推理图像现在与 SageMaker兼容。

> **Note**
>
> 您的模型包产品的模型可以通过两种方式部署：实时部署和批量部署。在这两个部署中，在运行 Docker 容器时 SageMaker 使用相同的 HTTP 端点。

要停止容器，请运行以下命令。

```
sudo docker container stop my-inference-container
```

当您的推理映像准备就绪并经过测试后，您可以继续上传映像。

算法映像

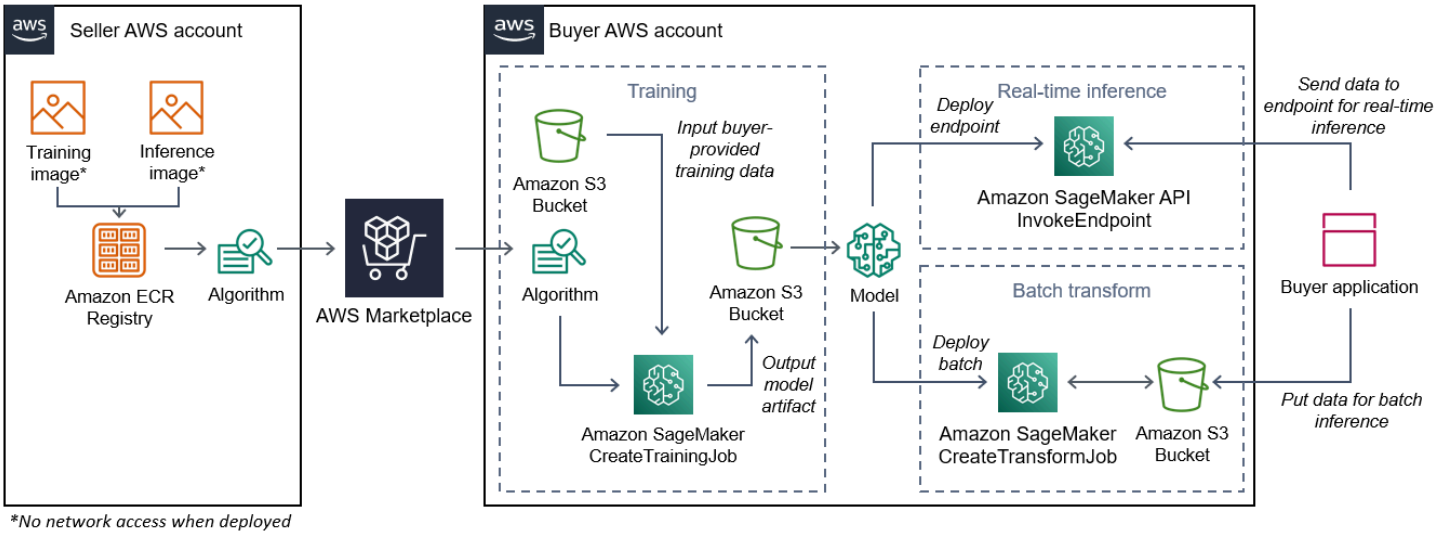
Amazon SageMaker 算法要求买家在做出预测之前带上自己的数据进行训练。

算法包括以下组件：

- 存储在 Amazon ECR 中的训练映像
- 存储在 Amazon Elastic Container Registry (Amazon ECR) 中的推理映像

> **Note**
>
> 对于算法产品，训练容器会生成模型构件，这些构件在模型部署时加载到推理容器中。

下图显示了发布和使用算法产品的工作流程。



1. 卖家创建训练映像和推理映像（部署后无法访问网络），然后将其上传到 Amazon ECR Registry。
2. 然后，卖家在 Amazon 中创建算法资源 SageMaker ，并在上发布其机器学习产品 AWS Marketplace。
3. 买家订阅机器学习产品。
4. 买方使用兼容的数据集和适当的超参数值创建训练作业。 SageMaker 运行训练图像并将训练数据和超参数加载到训练容器中。训练作业完成后，位于 `/opt/ml/model/` 中的模型构件会被压缩并复制到买家的 Amazon S3 存储桶中。
5. 买家使用存储在 Amazon S3 中的训练模型构件创建模型包，然后部署模型。
6. SageMaker 运行推理图像，提取压缩后的模型工件，并将文件加载到推理容器目录路径`/opt/ml/model/`中，用于推理的代码将在该路径中使用这些文件。
7. 无论模型部署为端点还是批量转换作业，都通过容器的 HTTP 端点代表买方将用于推理的数据 SageMaker 传递到容器并返回预测结果。

> **Note**
>
> 有关更多信息，请参阅训练模型。

为算法创建训练映像

本部分提供了将训练代码打包成训练映像的演练。创建算法产品需要训练映像。

训练映像 是包含您的训练算法的 Docker 映像。容器遵循特定的文件结构，允许将数据复制 SageMaker 到容器或从容器中复制数据。

发布算法产品时，需要同时提供训练映像和推理映像。创建训练映像后，必须创建推理映像。这两个映像可以合并为一个映像，也可以作为单独的映像保留。是合并映像还是将它们分开由您决定。通常，推理比训练简单，您可能需要单独的映像来提高推理性能。

> **Note**
>
> 以下只是训练映像打包代码的一个示例。有关更多信息，请参阅使用自己的算法和模型，AWS Marketplace并开启AWS Marketplace SageMaker 示例 GitHub。

步骤

- 步骤 1：创建容器映像
- 第 2 步：在本地构建和测试映像

步骤 1：创建容器映像

为了使训练图像与 Amazon 兼容 SageMaker，它必须符合特定的文件结构，以便 SageMaker 能够将训练数据和配置输入复制到容器中的特定路径。训练完成后，生成的模型工件将存储在从中 SageMaker 复制的容器中的特定目录路径中。

以下内容使用安装在 Linux 的 Ubuntu 发行版的开发环境中的 Docker CLI。

- 准备好程序以读取配置输入
- 准备好程序以读取数据输入
- 准备好程序以编写训练输出
- 为容器运行创建脚本

- 创建 Dockerfile

准备好程序以读取配置输入

如果您的训练程序需要买家提供的任何配置输入，则以下是运行时将这些内容复制到容器中的位置。如果需要，您的程序必须从这些特定的文件路径中读取。

- `/opt/ml/input/config` 是包含控制程序运行方式的信息的目录。
  - `hyperparameters.json` 是 JSON 格式的超参数名称和值字典。这些值是字符串，因此您可能需要对其进行转换。
  - `resourceConfig.json` 是一个 JSON 格式的文件，描述了用于分布式训练的网络布局。如果您的训练映像不支持分布式训练，则可以忽略此文件。

> **Note**
>
> 有关配置输入的更多信息，请参阅 Amazon 如何 SageMaker 提供培训信息。

准备好程序以读取数据输入

训练数据可以通过以下两种模式之一传递到容器。在容器中运行的训练程序会以这两种模式之一提取训练数据。

文件模式

- `/opt/ml/input/data/<channel_name>/` 包含该渠道的输入数据。渠道是根据对 `CreateTrainingJob` 操作的调用创建的，但通常重要的是渠道必须与算法的预期相匹配。每个渠道的文件都从 Amazon S3 复制到此目录，同时保留 Amazon S3 密钥结构所指示的树结构。

管道模式

- `/opt/ml/input/data/<channel_name>_<epoch_number>` 是给定纪元的管道。纪元从零开始，每次读取时都会增加一个。可以运行的纪元数没有限制，但是在读取下一个纪元之前，必须关闭每个管道。

准备好程序以编写训练输出

训练的输出将写入以下容器目录：

- `/opt/ml/model/` 是您编写模型或训练算法生成的模型构件的目录。您的模型可以采用任意格式。它可以是单个文件或整个目录树。 SageMaker 将此目录中的所有文件打包成压缩文件 (.tar.gz)。此文件位于 `DescribeTrainingJob` API 操作返回的 Amazon S3 位置。
- `/opt/ml/output/` 是一个目录，算法可以在其中写入描述任务失败原因的 `failure` 文件。此文件的内容将在 `DescribeTrainingJob` 结果的 `FailureReason` 字段中返回。对于成功的作业，无需编写此文件，将被忽略。

为容器运行创建脚本

创建一个在运行 Docker 容器镜像时 SageMaker 运行的 `train` shell 脚本。训练完成并将模型构件写入各自的目录后，退出脚本。

`./train`

```
#!/bin/bash

# Run your training program here
#
#
#
#
```

创建 **`Dockerfile`**

在您的构建上下文中创建 `Dockerfile`。此示例使用 Ubuntu 18.04 作为基础映像，但您可以从适用于您的框架的任何基础映像开始。

`./Dockerfile`

```
FROM ubuntu:18.04

# Add training dependencies and programs
#
#
#
#
#
# Add a script that SageMaker will run
# Set run permissions
# Prepend program directory to $PATH
COPY /train /opt/program/train
```

```
RUN chmod 755 /opt/program/train
ENV PATH=/opt/program:${PATH}
```

Dockerfile 将之前创建的 train 脚本添加到映像中。脚本的目录已添加到 PATH 中，因此它可以在容器运行时运行。

在前面的示例中，没有实际的训练逻辑。对于实际的训练映像，将训练依赖项添加到 Dockerfile 中，然后添加读取训练输入的逻辑，以训练和生成模型构件。

您的训练映像必须包含其所有必需的依赖项，因为它无法访问互联网。

有关更多信息，请参阅使用自己的算法和模型， AWS Marketplace并开启AWS Marketplace SageMaker 示例 GitHub。

第 2 步：在本地构建和测试映像

在构建上下文中，现在存在以下文件：

- ./Dockerfile
- ./train
- 您的训练依赖项和逻辑

接下来，您可以构建、运行和测试此容器映像。

构建映像：

在构建上下文中运行 Docker 命令来构建和标记映像。此示例使用标记 my-training-image。

```
sudo docker build --tag my-training-image ./
```

运行此 Docker 命令构建映像后，当 Docker 根据您的 Dockerfile 的每一行构建映像时，您该会看到输出。完成后，您应看到类似如下所示的内容：

```
Successfully built abcdef123456
Successfully tagged my-training-image:latest
```

本地运行

完成后，在本地测试映像，如以下示例中所示。

```
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
```

以下是命令的详细信息：

- `--rm` – 容器停止后自动将其移除。
- `--volume '<path_to_input>:/opt/ml/input:ro'` – 将测试输入目录设置为只读供容器使用。
- `--volume '<path_to_model>:/opt/ml/model'` – 训练测试完成后，绑定挂载模型构件在主机上的存储路径。
- `--volume '<path_to_output>:/opt/ml/output'` – 绑定挂载主机上 `failure` 文件中写入失败原因的路径。
- `--name my-training-container` – 为正在运行的容器提供名称。
- `my-training-image` – 运行构建的映像。
- `train`— 运行容器时运行的脚本与运行的脚本 SageMaker 相同。

运行此命令后，Docker 会根据您构建的训练映像创建一个容器并运行它。容器运行 `train` 脚本，从而启动您的训练程序。

训练程序完成且容器退出后，请检查输出模型构件是否正确。此外，请检查日志输出以确认它们没有生成您不想要的日志，同时确保提供了有关训练作业的足够信息。

这样就完成了算法产品的训练代码打包。由于算法产品还包括推理映像，因此请继续下一部分 为算法创建推理映像。

为算法创建推理映像

本部分提供了将推理代码打包成算法产品的推理映像的演练。

推理映像是包含您的推理逻辑的 Docker 映像。容器在运行时公开 HTTP 端点， SageMaker 以允许将数据传入和传出您的容器。

发布算法产品时，需要同时提供训练映像和推理映像。如果您尚未执行此操作，请参阅前面的部分为算法创建训练映像。这两个映像可以合并为一个映像，也可以作为单独的映像保留。是合并映像还是将它们分开由您决定。通常，推理比训练简单，您可能需要单独的映像来提高推理性能。

> **Note**
>
> 以下只是推理映像打包代码的一个示例。有关更多信息，请参阅使用自己的算法和模型，AWS Marketplace并开启AWS Marketplace SageMaker 示例 GitHub。
> 为了简单起见，以下示例使用了 Web 服务 Flask，但不被视为生产就绪。

步骤

- 步骤 1：创建推理映像
- 第 2 步：在本地构建和测试映像

步骤 1：创建推理映像

要使推理映像与之兼容 SageMaker，Docker 镜像必须公开 HTTP 端点。当您的容器运行时，SageMaker 会将买方提供的推理输入传递到容器的 HTTP 终端节点。推理结果返回在 HTTP 响应的正文中。

以下内容使用安装在 Linux 的 Ubuntu 发行版的开发环境中的 Docker CLI。

- 创建 Web 服务器脚本
- 为容器运行创建脚本
- 创建 `Dockerfile`
- 准备程序以动态加载模型构件

创建 Web 服务器脚本

此示例使用名为 Flask 的 Python 服务器，但您可以使用任何适用于您的框架的 Web 服务器。

> **Note**
>
> 为了简单起见，这里使用 Flask。它不被视为生产就绪的 Web 服务器。

创建 Flask Web 服务器脚本，该脚本在使用的 TCP 端口 8080 上为两个 HTTP 端点提供服务。SageMaker 以下是两个预期的端点：

- `/ping`— SageMaker 向此端点发出 HTTP GET 请求，以检查您的容器是否准备就绪。当您的容器准备就绪时，它会使用 HTTP 200 响应代码来响应此端点的 HTTP GET 请求。
- `/invocations`— SageMaker 向此端点发出 HTTP POST 请求以进行推断。用于推理的输入数据在请求正文中发送。用户指定的内容类型在 HTTP 头中传递。响应的主体是推理输出。

`./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)
```

在前面的示例中，没有实际的推理逻辑。对于实际的推理映像，请将推理逻辑添加到 Web 应用中，以便它处理输入并返回预测。

您的推理映像必须包含其所有必需的依赖项，因为它无法访问互联网。

为容器运行创建脚本

创建一个名为的脚本`serve`，该脚本在 SageMaker 运行 Docker 容器镜像时运行。在此脚本中，启动 HTTP Web 服务器。

`./serve`

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker
flask run --host 0.0.0.0 --port 8080
```

创建 **`Dockerfile`**

在您的构建上下文中创建 `Dockerfile`。此示例使用 Ubuntu 18.04，但您可以从适用于您的框架的任何基础映像开始。

`./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 will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

`Dockerfile` 将之前创建的两个脚本添加到映像中。`serve` 脚本的目录已添加到 PATH 中，因此它可以在容器运行时运行。

准备程序以动态加载模型构件

对于算法产品，买家使用自己的数据集和您的训练映像来生成独特的模型构件。训练过程完成后，您的训练容器会将模型工件输出到容器目录 `/opt/ml/model/`。 SageMaker 将该目录中的内容压缩成.tar.gz 文件并将其存储在买家的 Amazon S3 中。 AWS 账户

部署模型时， SageMaker 运行您的推理图像，从 Amazon S3 中买家账户中存储的.tar.gz 文件中提取模型工件，然后将其加载到目录中的推理容器中。`/opt/ml/model/`在运行时，您的推理容器代码使用模型数据。

> **Note**
>
> 要保护可能包含在模型构件文件中的知识产权，您可以选择在输出前对文件加密。有关更多信息，请参阅 安全性和知识产权。

第 2 步：在本地构建和测试映像

在构建上下文中，现在存在以下文件：

- `./Dockerfile`
- `./web_app_serve.py`
- `./serve`

接下来，您可以构建、运行和测试此容器映像。

构建映像：

运行 Docker 命令来构建和标记映像。此示例使用标签 `my-inference-image`。

```
sudo docker build --tag my-inference-image ./
```

运行此 Docker 命令构建映像后，当 Docker 根据您的 `Dockerfile` 的每一行构建映像时，您该会看到输出。完成后，您应看到类似如下所示的内容：

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```

本地运行

构建完成后，您可以在本地测试映像。

```
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
```

以下是命令的详细信息：

- `--rm` – 容器停止后自动将其移除。
- `--publish 8080:8080/tcp`— 公开端口 8080 以模拟向其 SageMaker 发送 HTTP 请求的端口。
- `--volume '<path_to_model>:/opt/ml/model:ro'` –将测试模型构件在主机上存储的路径绑定挂载为只读，以便容器中的推理代码可以使用它们。
- `--detach` – 在后台运行容器。
- `--name my-inference-container` – 为正在运行的容器提供名称。
- `my-inference-image` – 运行构建的映像。
- `serve`— 运行容器时运行的脚本与运行的脚本 SageMaker 相同。

运行此命令后，Docker 会根据推理映像创建一个容器并在后台运行它。容器运行 `serve` 脚本，该脚本会启动您的 Web 服务器以进行测试。

测试 ping HTTP 端点

SageMaker 运行您的容器时，它会定期对端点执行 ping 操作。 当端点返回状态码为 200 的 HTTP 响应时，它会发出信号 SageMaker ，表示容器已准备好进行推理。

运行以下命令以测试端点并包含响应头。

```
curl --include http://127.0.0.1:8080/ping
```

示例输出如以下示例中所示。

```
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
```

测试推理 HTTP 端点

当容器通过返回 200 状态码表示已准备就绪时，通过`POST`请求将推理数据 SageMaker 传递到 `/invocations` HTTP 端点。

运行以下命令以测试推理端点。

```
curl \
  --request POST \
  --data "hello world" \
  http://127.0.0.1:8080/invocations
```

示例输出如以下示例中所示：

```
{"prediction": "a", "text": "hello world"}
```

这两个 HTTP 端点正常工作后，推理图像现在与 SageMaker兼容。

> **Note**
>
> 您的算法产品的模型可以通过两种方式部署：实时部署和批量部署。对于这两个部署，在运行 Docker 容器时 SageMaker 使用相同的 HTTP 端点。

要停止容器，请运行以下命令。

```
sudo docker container stop my-inference-container
```

算法产品的训练和推理映像都准备好并经过测试后，请继续上传映像。

上传映像

本部分介绍如何将推理和训练映像上传到 Amazon Elastic Container Registry。Amazon ECR 是一项托管 Docker 注册表服务。 这是 Amazon 从中 SageMaker 提取图像以创建用于推理的模型包或训练作业的算法。这也是 AWS Marketplace 检索图像以发布模型包和算法产品的地方。

我必须上传哪些映像？

如果您要发布模型包，请仅上传推理映像。如果您要发布算法，请同时上传推理映像和训练映像。如果推理映像和训练映像合并在一起，则仅上传一次合并后的映像。

需要哪些 IAM 权限？

以下步骤假设本地计算机具有卖家中某个 AWS Identity and Access Management (IAM) 角色或用户的正确 AWS 证书 AWS 账户。角色或用户必须为两者 AWS Marketplace 和 Amazon ECR 制定正确的策略。例如，您可以使用以下 AWS 托管策略：

- AWSMarketplaceSellerProductsFullAccess— 用于访问 AWS Marketplace
- AmazonEC2ContainerRegistryFullAccess – 用于访问 Amazon ECR

将你的 Docker 客户端登录到 AWS

为要发布 AWS 区域 的设置变量（请参阅AWS 区域 支持发布）。对于本示例，请选择美国东部（俄亥俄州）区域。

```
region=us-east-2
```

运行以下命令以使用您的 AWS 账户 ID 设置变量。此示例假设当前 AWS Command Line Interface (AWS CLI) 证书属于卖家的 AWS 账户。

```
account=$(aws sts get-caller-identity --query Account --output text)
```

要使用您所在地区的 AWS 账户 Amazon ECR Docker 注册表对您的 Docker CLI 客户端进行身份验证，请运行以下命令。

```
aws ecr get-login-password \
--region ${region} \
| sudo docker login \
--username AWS \
--password-stdin \
${account}.dkr.ecr.${region}.amazonaws.com
```

创建存储库并上传映像

为上传映像的标签设置一个变量，为上传的映像存储库的名称设置另一个变量。

```
image=my-inference-image
repo=my-inference-image
```

> **Note**
>
> 在本指南的前几节中，推理和训练图像是生成的，它们分别被标记为my-inference-image和my-training-image。在本示例中，创建推理映像并将其上传到同名的存储库。

运行以下命令，在 Amazon ECR 中创建映像存储库。

```
aws ecr --region ${region} create-repository --repository-name "${repo}"
```

Amazon ECR 存储库位置的全名由以下部分组成： `<account-id>.dkr.ecr.<region>.amazonaws.com/<image-repository-name>`

要将映像推送到存储库，必须使用存储库位置的全名对其进行标记。

为映像存储库位置的全名和 `latest` 标签设置一个变量。

```
fullname="${account}.dkr.ecr.${region}.amazonaws.com/${repo}:latest"
```

用全名标记映像。

```
sudo docker tag ${image} ${fullname}
```

最后，将推理映像推送到 Amazon ECR 中的存储库。

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

上传完成后，该映像将出现在您要从中发布的区域的 Amazon ECR 控制台的存储库列表中。在前面的示例中，映像被推送到美国东部（俄亥俄州）区域的存储库中。

扫描上传的映像

在 Amazon ECR 控制台中 AWS 区域 ，选择您要发布的，然后打开上传图像的存储库。选择您上传的图像并开始扫描以检查是否存在已知漏洞。 AWS Marketplace 在发布之前，请检查您的亚马逊 SageMaker 资源中使用的容器映像的 Amazon ECR 扫描结果。在创建产品之前，必须修复存在严重程度为“严重”或“高”的漏洞的容器映像。

成功扫描映像后，它们可用于创建模型包或算法资源。

如果您认为自己的产品在扫描中存在误报错误，请联系 AWS Marketplace 卖家运营团队并提供有关错误的信息。

后续步骤

- 查看创建机器学习产品的要求和最佳实操中的文件大小限制
- 继续创建您的亚马逊 SageMaker资源

创建您的亚马逊 SageMaker资源

要发布模型包或算法产品，您必须在 Amazon 中创建相应的模型包资源或算法资源 SageMaker。

在为 AWS Marketplace 产品创建资源时，必须通过验证步骤对其进行认证。验证步骤要求您在发布模型包或算法资源之前提供数据对其进行测试。

> **Note**
>
> 如果您尚未为产品创建映像并将其上传到 Amazon Elastic Container Registry (Amazon ECR)，请参阅将代码打包成映像和上传映像，了解有关如何操作的信息。

创建模型包

以下是为 AWS Marketplace创建模型包的要求：

- 存储在 Amazon ECR 中的推理映像
- （可选）模型构件，单独存储在 Amazon S3 中
- 您用于推理的测试数据，存储在 Amazon Simple Storage Service (Amazon S3) 中

> **Note**
>
> 以下是关于创建模型包产品的信息。有关中模型包的更多信息 SageMaker，请参阅创建模型包资源。

创建模型包资源

以下过程将引导您完成模型包资源的创建。

第 1 步：创建模型包资源

1. 打开亚马逊 SageMaker 控制台。
2. 查看页面的右上角，确保您位于要发布的 AWS 区域。有关发布的信息，请参阅AWS 区域 支持发布部分。您在之前的步骤中上传到 Amazon ECR 的推理映像必须位于同一区域。
3. 在左侧导航菜单中，选择模型包。
4. 选择创建模型包。

创建包后，您需要设置推理包的规格。

步骤 2：设置推理规格

1. 为您的模型包提供一个名称（例如，*my-model-package*）。
2. 对于推理映像的位置，请输入上传到 Amazon ECR 的推理映像的 URI。您可以通过在 Amazon ECR 控制台中找到您的映像来检索 URI。
3. 如果训练模型构件与推理映像中的逻辑捆绑在一起，请将模型数据构件的位置留空。否则，请指定模型构件压缩文件 (.tar.gz) 的完整 Amazon S3 位置。
4. 使用下拉框为实时推理（也称为端点）和批量转换作业选择推理映像支持的实例类型。
5. 选择下一步。

在创建和发布模型包之前，必须进行验证以确保其按预期运行。这要求您使用您提供的用于推理的测试数据运行批处理转换作业。验证规范说明了 SageMaker 如何执行验证。

步骤 3：设置验证规格

1. 将发布此模型包设置为 AWS Marketplace是。如果您将其设置为否，则以后将无法发布此模型包。选择 “是” 将对您的模型包进行认证， AWS Marketplace 并且需要执行验证步骤。
2. 如果这是首次完成此过程，请选择为 IAM 角色创建新角色。Amazon 在部署您的模型包时 SageMaker 使用此角色。这包括一些操作，例如从 Amazon ECR 中拉取映像和从 Amazon S3 拉取构件。查看设置，然后选择创建角色。 在此处创建角色会 AmazonSageMakerFullAccess将 IAM 策略描述的权限授予您创建的角色。
3. 在验证配置文件中编辑 JSON。 有关允许值的详细信息，请参阅TransformJobDefinition。
   1. `TransformInput.DataSource.S3Uri`：设置为用于推理的测试数据的存储位置。
   2. `TransformInput.ContentType`：指定您的测试数据内容类型（例如`application/json`、`text/plain``image/png` 、或任何其他值）。 SageMaker 不验证实际的输入数据。此值将在 `Content-type` 头值中传递到您的容器 HTTP 端点。
   3. `TransformInput.CompressionType`：如果您在 Amazon S3 中用于推理的测试数据未压缩，则设置为 `None`。
   4. `TransformInput.SplitType`：设置为 `None` 以将 Amazon S3 中的每个对象作为一个整体传递以供推理。
   5. `TransformOutput.S3OutputPath`：设置为推理输出的存储位置。
   6. `TransformOutput.AssembleWith`：设置为 `None` 以将每个推理作为单独的对象输出到 Amazon S3 中。
4. 选择创建模型包。

SageMaker 从 Amazon ECR 中提取推理图像，将所有构件复制到推理容器，然后使用您的测试数据运行批量转换任务进行推理。验证成功后，状态将更改为已完成。

> **Note**
>
> 验证步骤不会使用您的测试数据来评估模型的准确性。验证步骤检查容器是否按预期运行和响应。

您已完成模型产品资源的创建。继续在 AWS Marketplace中发布产品。

创建算法

以下是在 AWS Marketplace中创建算法的要求：

- 存储在 Amazon ECR 中的推理映像
- 存储在 Amazon ECR 中的训练映像
- 用于训练的测试数据，存储在 Amazon S3 中
- 用于推理的测试数据，存储在 Amazon S3 中

> **Note**
>
> 以下演练创建了一个算法产品。有关更多信息，请参阅创建算法资源。

创建算法资源

以下过程将引导您完成在算法包中创建资源的过程。

步骤 1：创建算法资源

1. 打开亚马逊 SageMaker 控制台。
2. 查看页面右上角，确保您位于要发布的 AWS 区域（请参阅AWS 区域 支持发布）。您在之前的步骤中上传到 Amazon ECR 的训练和推理映像必须位于同一区域。
3. 在左导航窗格中，选择算法。
4. 选择创建算法。

创建算法包后，必须为模型的训练和调整设置规格。

第 2 步：设置训练和调整规格

1. 输入算法的名称（例如，`my-algorithm`）。
2. 对于训练映像，请粘贴上传到 Amazon ECR 的训练映像的完整 URI 位置。您可以通过在 Amazon ECR 控制台中找到您的映像来检索 URI。
3. 使用下拉框选择您的训练映像支持的训练实例类型。
4. 在渠道规格部分下，为算法支持的每个输入数据集添加一个渠道，最多可添加 20 个渠道的输入源。（有关更多信息，请参阅输入数据配置。）

5. 选择下一步。
6. 如果您的算法支持超参数和超参数调整，则必须指定调整参数。
7. 选择下一步。

> **Note**
> 我们强烈建议您的算法支持超参数调整并使适当的参数可调。这使数据科学家能够调整模型以获得最佳结果。

设置调整参数（如果有）后，必须为推理映像设置规格。

步骤 3：设置推理映像规格

1. 对于推理映像的位置，请粘贴上传到 Amazon ECR 的推理映像的 URI。您可以通过在 Amazon ECR 控制台中找到您的映像来检索 URI。
2. 使用下拉框为实时推理（也称为端点）和批量转换作业选择推理映像支持的实例类型。
3. 选择下一步。

在创建和发布算法之前，必须进行验证以确保算法按预期运行。这要求您同时运行包含用于训练的测试数据的训练作业，以及使用您提供的用于推理的测试数据的批量转换作业。验证规范说明了 SageMaker 如何执行验证。

步骤 4：设置验证规格

1. 将发布此算法设置 AWS Marketplace为是。如果将其设置为否，则以后将无法发布此算法。选择“是” 将验证您的算法， AWS Marketplace 并且需要验证规范。
2. 如果这是您第一次为其创建机器学习包 AWS Marketplace，请选择为 IAM 角色创建新角色。Amazon 在训练您的算法和部署后续模型包时 SageMaker 使用此角色。这包括从 Amazon ECR 拉取映像、在 Amazon S3 中存储构件以及从 Amazon S3 复制训练数据等操作。查看设置，然后选择创建角色。 在此处创建角色会 AmazonSageMakerFullAccess将 IAM 策略描述的权限授予您创建的角色。
3. 编辑训练作业定义的验证配置文件中的 JSON 文件。有关允许值的更多信息，请参阅 TrainingJobDefinition。

1. `InputDataConfig`：在此 JSON 数组中，为您在训练规格步骤中指定的每个渠道添加一个渠道对象。对于每个渠道，指定用于训练的测试数据的存储位置。
2. `OutputDataConfig`：训练完成后，训练容器目录路径 `/opt/ml/model/` 中的模型构件会被压缩并复制到 Amazon S3。指定存储压缩文件 (.tar.gz) 的 Amazon S3 位置。

4. 编辑转换作业定义的验证配置文件中的 JSON 文件。有关允许值的更多信息，请参阅 TransformJobDefinition。

   1. `TransformInput.DataSource.S3Uri`：设置为用于推理的测试数据的存储位置。
   2. `TransformInput.ContentType`：指定您的测试数据内容类型。例如，`application/json`、`text/plain`、`image/png` 或任何其他值。Amazon SageMaker 不验证实际输入数据。此值将在 `Content-type` 头值中传递到您的容器 HTTP 端点。
   3. `TransformInput.CompressionType`：如果您在 Amazon S3 中用于推理的测试数据未压缩，则设置为 `None`。
   4. `TransformInput.SplitType`：选择您希望在 S3 中拆分对象的方式。例如，`None` 将 Amazon S3 中的每个对象作为一个整体传递以进行推理。如需了解更多详情，请参阅 Amazon SageMaker API 参考 SplitType中的。
   5. `TransformOutput.S3OutputPath`：设置为推理输出的存储位置。
   6. `TransformOutput.AssembleWith`：设置为 `None` 以将每个推理作为单独的对象输出到 Amazon S3 中。

5. 选择创建算法包。

SageMaker 从 Amazon ECR 中提取训练图像，使用您的数据运行测试训练作业，并将模型工件存储在 Amazon S3 中。然后，它从 Amazon ECR 中拉取推理映像，将构件从 Amazon S3 复制到推理容器中，然后使用您的测试数据运行批量转换作业进行推理。验证成功后，状态将更改为已完成。

> **Note**
>
> 验证步骤不会使用您的测试数据来评估训练或模型的准确性。验证步骤检查容器是否按预期运行和响应。
>
> 验证步骤仅验证批处理。由您来验证实时处理是否适用于您的产品。

您已完成算法产品资源的创建。继续在 AWS Marketplace中发布产品。

在 AWS Marketplace中发布产品

在发布模型包或算法之前，需要满足以下条件：

- 已注册为 AWS Marketplace 卖家的。 AWS 账户 您可以在 AWS Marketplace 管理门户 中执行此操作。
- 在 AWS Marketplace 管理门户中的设置页面下填写完整的卖家资料。
- 要发布付费产品，您必须填写税务调查和银行表格。发布免费产品则无需填写。有关更多信息，请参阅卖家注册流程。
- 您必须拥有访问 AWS Marketplace 管理门户 和 Amazon 的权限 SageMaker。有关更多信息，请参阅 所需权限。

发布过程概述

发布过程分为四个步骤：

1. 提交产品 - 创建包含您的模型包或算法产品的描述、使用信息和其他详细信息的列表。在您提交产品进行发布后，大约需要一个小时才能更改状态，进入下一步。
2. 测试商品 — 使用 AWS 账户 您注册为 AWS Marketplace 卖家的商品预览中的商品 AWS Marketplace、订阅和测试商品。此外，其他允许的 AWS 账户 用户可以预览和测试该产品。如果需要进行任何更改，您可以返回并编辑列表详情。
3. 注销发布-当您的产品准备好上线时，请返回并选择 “注销并发布”。 AWS Marketplace 管理门户
4. 产品上线 – 您的产品现已在 AWS Marketplace上线。您可以通过发布包含更新或产品修复的新版本来维护您的产品。

所需权限

要发布 Amazon SageMaker 产品，您登录的 AWS Identity and Access Management 用户或角色需要以下一项或两项 IAM 操作：

- sagemaker: DescribeModelPackage — 用于列出模型包
- sagemaker: DescribeAlgorithm — 用于列出算法

要了解所需的 AWS Marketplace 权限或管理您的卖家账户，请参阅 AWS Marketplace 卖家政策和权限。

创建产品列表

以下是在模型包和算法产品中创建产品清单 AWS Marketplace 的演练。

> **Note**
>
> 在创建列表之前，请确保已在 创建机器学习产品的要求和最佳实操 中指定所需的资源。

此过程包含以下步骤：

步骤

- 步骤 1：创建新列表
- 第 2 步：提供一般产品信息
- 步骤 3：添加启动选项
- 步骤 4：设置定价和条款
- 第 5 步：提交产品以进行发布

步骤 1：创建新列表

要创建新的机器学习产品列表，请执行以下操作：

1. 登录您的卖家 AWS 账户 并导航至AWS Marketplace 管理门户。
2. 在顶部菜单中，导航到产品，然后导航到机器学习。
3. 选择创建新列表。

> **Note**
>
> 在新产品页面的产品摘要部分，您可以查看当前状态、隐私设置、产品类型、创建者和产品 ID。

第 2 步：提供一般产品信息

要提供一般产品信息，请执行以下操作：

1. 在一般产品信息部分的产品描述中，选择添加。

a. 对于产品可见性部分，选择下列选项之一：

- 公开 — 该产品最初将提供给有限的一组人 AWS 账户 进行测试。在您签署并发布产品后，该产品便可公开发现并可供所有客户订阅。
- 隐@@ 私 — 只有您指定的用户才能看到 AWS 账户 该产品。将来，您将无法公开此产品。

b. 输入产品标题、简短产品描述、产品概述、产品类别 1 和其他详细信息。您以后可以更改这些值。有关产品描述，请参阅创建机器学习产品的要求和最佳实操。

c. 完成后选择继续。

2. 对于促销资源，请提供产品徽标、搜索关键词和相关资源链接。您以后可以更改这些值。

- 完成后选择继续。

3. 对于支持信息，请选择是否为产品提供支持。

a. 如果您选择是，请提供支持和联系详细信息。您之后可以更改这些值。

b. 完成后选择继续。

4. 对于区域供货情况，请选择 AWS 区域 您要在其中列出商品的具体区域。

默认值为在所有目前和未来支持区域中提供。

- 完成后选择继续。

> **Note**
>
> 在提交草稿以供发布后，您无法更改此选择。

发布产品的下一步是提供启动选项，即您要销售的模型包或算法。

步骤 3：添加启动选项

要添加启动选项，请执行以下操作：

1. 在错误描述部分，对于输入 ARN，输入模型包或算法的 Amazon 资源名称 (ARN)。

您可以在 Amazon SageMaker 控制台模型包或算法页面中找到 ARN。

Example 模型包的 ARN

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

Example 算法的 ARN

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

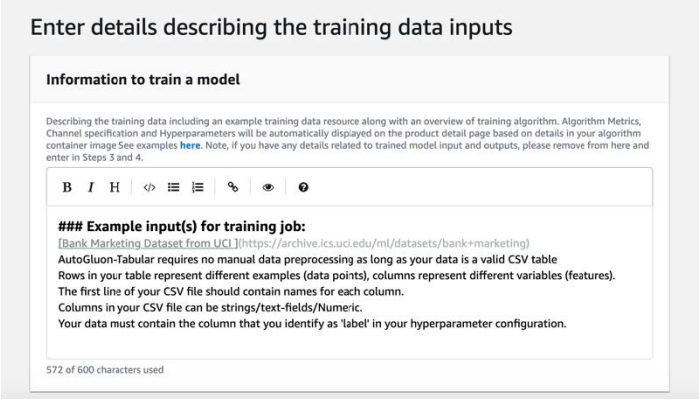
2. 选择添加。
3. 根据您发布的是模型包还是算法产品，以下步骤会有所不同。除了面向买家的版本号之外，您可以稍后更改版本详细信息。

   1. 对于步骤 1：输入版本详细信息和 Git 存储库链接，请提供版本号、发行说明以及示例 Jupyter 笔记本和 GitHub 存储库的 URL。
   2. 仅适用于算法产品，对于步骤 2：输入描述训练数据输入的详细信息，描述训练数据，并包括示例训练数据资源以及训练算法概述。

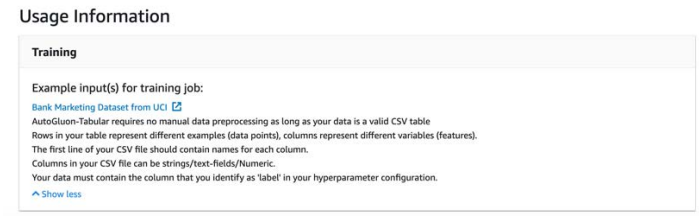
      算法指标、渠道规格和超参数将根据您在中创建算法资源时提供的值自动显示在 SageMaker产品详情页面上。

      以下示例展示了训练数据输入详细信息如何显示给卖家（您）和买家。

      Example 训练数据输入示例 – 卖家视图

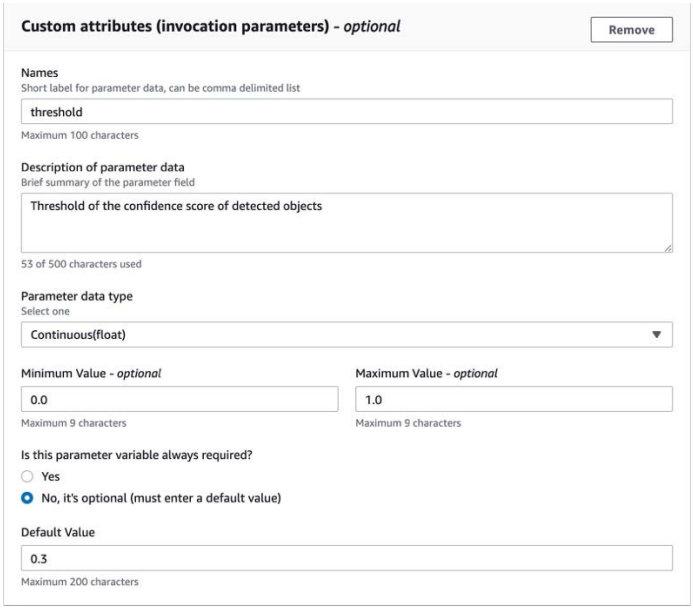


Example 训练数据输入示例 – 买家视图

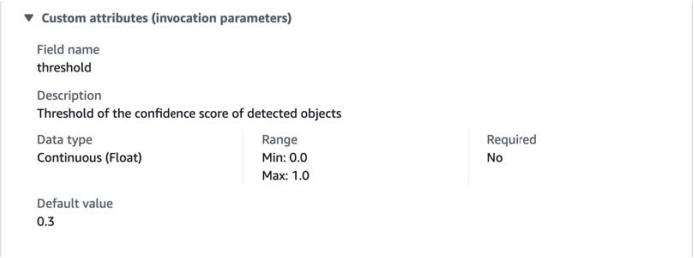


以下示例展示了自定义属性（调用参数）如何显示给卖家（您）和买家。

Example 自定义属性示例（调用参数）– 卖家视图



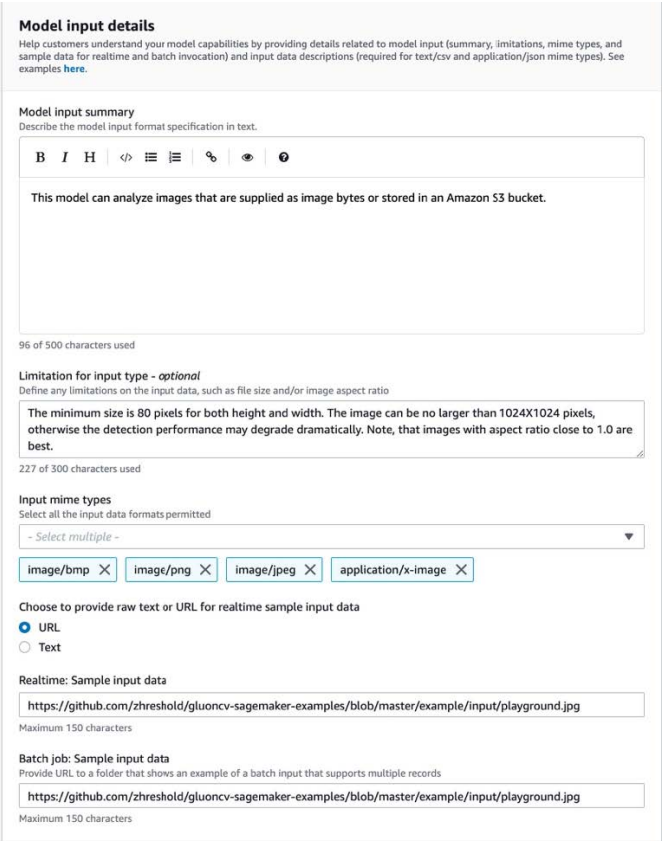
Example 自定义属性示例（调用参数）– 买家视图



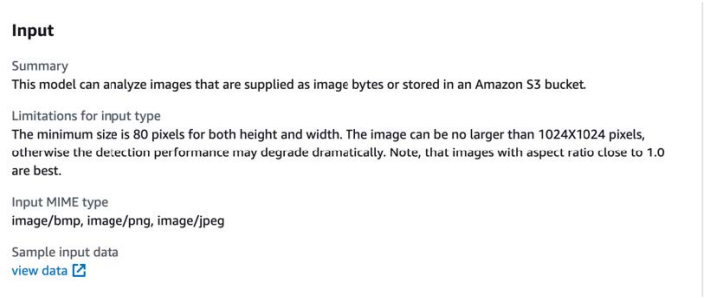
3. 对于步骤 3：输入输入详细信息，请提供模型或算法输入详细信息以及示例输入文件的 URL。

   以下示例展示了模型数据输入详细信息如何显示给卖家（您）和买家。

Example 模型数据输入示例 – 卖家视图



Example 模型数据输入示例 – 买家视图

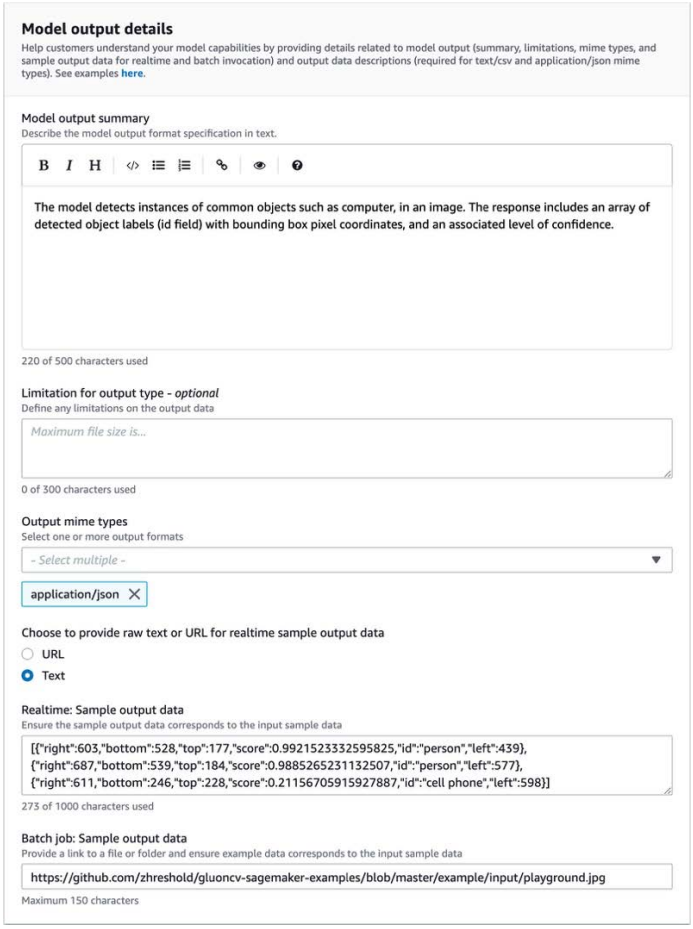


4. 对于步骤 4：输入输出详细信息，以文本或 URL 的形式提供模型或算法输出详细信息以及示例输出。

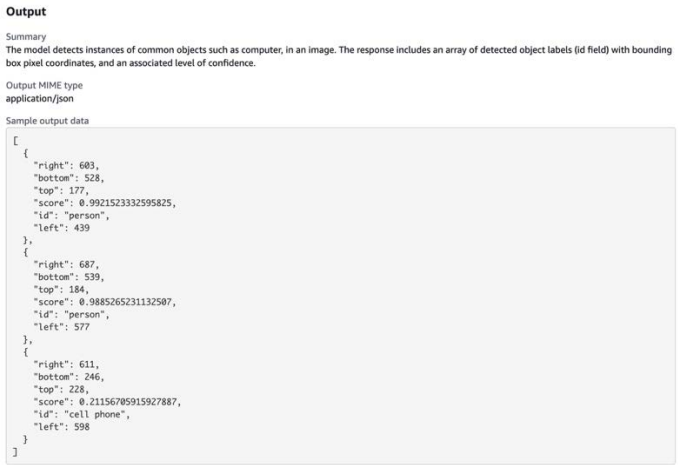
   有关使用信息，请参阅创建机器学习产品的要求和最佳实操。

   以下示例显示了模型数据输出详细信息如何显示给卖家（您）和买家。

Example 模型数据输出示例 – 卖家视图



Example 模型数据输出示例 – 买家视图



5. 对于步骤 5：查看支持的实例并创建，请设置推荐的实例。

- 如果这是模型包产品，请从支持的实例中选择推荐的实例类型，以用于批量转换和实时部署。
- 如果这是算法产品，还要选择推荐的实例类型以用于训练作业。

您不能选择您的模型包或算法资源不支持的实例类型。支持的实例类型是在您在 Amazon 中创建这些资源时选择的 SageMaker。

4. 完成后选择继续。

> **Note**
>
> 描述产品预期输入和输出的清晰使用信息（附示例）对于支持积极的买家体验至关重要。有关更多信息，请参阅 创建机器学习产品的要求和最佳实操。

发布产品的下一步是设置定价和条款。

步骤 4：设置定价和条款

要设置定价和条款，请执行以下操作：

1. 在定价和条款部分，选择添加优惠。
2. 设置您的定价。

   您可以免费提供软件、设置付费定价或启用免费试用期。有关更多信息，请参阅 机器学习产品。
3. 上传纯文本文件以用作最终用户许可协议 (EULA)。
4. 选择保存并关闭。

您已提供产品的所有信息。下一步是有限发布该产品，以便您可以进行测试。

第 5 步：提交产品以进行发布

要提交您的产品以进行发布，请执行以下操作：

1. 在 “新产品” 页面的 “提交以供发布” 部分的 “其他测试账户 — 可选” 下，为其他测试人员输入一个或多个 AWS 账户 ID。
2. 选择提交以进行发布。

这将通过创建预览列表来启动发布过程 AWS Marketplace ，您（和您的可选测试人员）可以在其中订阅并用于测试。

您现在已准备就绪，可测试产品。有关测试机器学习产品的更多信息，请参阅测试您的产品。

测试产品后，如果需要进行任何更改，则可以重复上述步骤。当您准备好向买家销售产品时，您可以签署发布。

测试您的产品

首次提交产品后，大约需要一个小时才能准备好您的预览列表。状态更改为 “测试商品” 后，您的卖家账户和其他允许上架的用户 AWS 账户 可以在中预览商品信息 AWS Marketplace、订阅商品并进行测试。

要查看列表的预览，请执行以下操作：

1. 在中 AWS Marketplace 管理门户，导航到产品概述页面。
2. 选择 转到暂存产品。
3. 如果您想进行更改，请选择编辑产品并按照与创建产品列表相同的步骤进行操作。
4. 当您准备好将产品公开发布以供所有买家查看时，请按照签署发布中的步骤操作。

要在发布前添加其他商品 AWS 账户 来测试您的商品，请联系AWS Marketplace 卖家运营团队并提供 AWS 账户 编码。允许列表中的账户在产品详细信息页面上的产品版本旁边会显示一个受限徽章。

签署发布

此步骤应在您写下描述、定价和使用信息，然后测试产品之后完成。

要签署发布，请执行以下操作

1. 登录您的卖家 AWS 账户 并导航至AWS Marketplace 管理门户。
2. 在顶部菜单中，导航到产品，然后导航到机器学习。
3. 导航到您的产品的产品概述。
4. 选择签署并发布。

更新您的产品

您可以使用中的 Mach ine Learnin AWS Marketplace 管理门户 g 列表页面通过以下方式更新您的模型包或算法产品：

- 添加新版本 - 您可以添加新的模型包或算法资源作为现有产品的新版本。
- 限制版本 - 您可以限制现有产品的先前版本。
- 删除产品 - 您可以删除整个产品。

添加新版本

要添加模型包或算法资源的新版本，请执行以下操作：

1. 导航到 AWS Marketplace 管理门户中的机器学习列表页面。
2. 导航到现有产品的产品概述。
3. 选择编辑产品。
4. 在启动选项下，选择编辑。
5. 要添加资源的 ARN，请导航到版本页面，然后选择添加新版本。

   有关添加启动选项的更多信息，请参阅创建产品列表。

> **Note**
>
> 使用信息因每个产品版本而异。向新版本添加使用信息时，请继续按照创建机器学习产品的要求和最佳实操中的说明进行。

当您的买家从产品 AWS Marketplace 列表中发布您的产品时，他们可以选择不同的版本。当您的买家从 Amazon SageMaker 控制台发布您的商品时，只能看到最新版本。

限制版本

要限制模型包或算法资源的新版本，请执行以下操作：

1. 导航到现有产品的产品概述。
2. 选择编辑产品。
3. 在启动选项下，选择编辑。

4. 在版本页面上，选择限制版本。
5. 返回产品概览，然后选择提交以进行发布。

> **Note**
> 已经订阅您产品的买家可以继续使用您的模型包或算法的受限版本。但是，新买家将无法将这些受限版本视为选项。

删除产品

要删除产品，请执行以下操作：

1. 在 AWS Marketplace 管理门户中的机器学习列表页面中导航到您的已发布产品列表。
2. 选择要删除的产品，然后在操作下拉列表中选择取消发布列表。
3. 请提供电子邮件地址和删除您的列表的理由，以便 AWS Marketplace 销售代表就您的请求与您联系。

> **Note**
> 当您从中移除产品时 AWS Marketplace，新买家将无法再订阅您的产品。但是，现有买家可继续使用您的产品，且您还必须至少提供 90 天的支持。如果您计划用其他产品替换未发布的产品，请在删除请求的详情中注明新产品。

创建机器学习产品的要求和最佳实操

务必使买家能够轻松测试您的模型包和算法产品。以下各部分描述了创建机器学习 (ML) 产品列表的要求以及机器学习产品的最佳实操。有关要求和建议的完整摘要，请参阅机器学习产品列表的要求和建议摘要。

> **Note**
> 如果您发布的商品不符合这些要求，销售 AWS Marketplace 代表可能会与您联系以帮助您满足这些要求。

主题

- 所需资产
- 机器学习产品的一般最佳实操
- 使用信息要求
- 输入和输出的要求
- Jupyter 笔记本的要求
- 机器学习产品列表的要求和建议摘要

所需资产

在创建机器学习产品列表之前，请确保您拥有以下必需的资产：

- Amazon 资源名称 (ARN) — 在中提供您要发布 AWS 区域 的模型包或算法资源（参见AWS 区域 支持发布）。ARN
  - a f ARN or a model 包的形式如下：`arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>`
  - 一个ARN算法的形式是这样的：`arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>`
- the section called “使用信息要求” – 提供有关输入、输出和代码示例的详细信息。
- the section called “输入和输出的要求” – 提供文件或文本。
- the section called “Jupyter 笔记本的要求” – 演示完整的产品使用情况。

机器学习产品的一般最佳实操

为您的机器学习产品提供以下信息：

- 对于产品描述，请包括以下内容：
  - 模型的作用
  - 目标客户
  - 最重要的用例
  - 模型训练方式及使用的数据量
  - 性能指标及使用的验证数据
  - 如果是医疗用途，则您的模型是否用于诊断

- 默认情况下，机器学习产品配置为公开可见。但是，您可以创建专属可见产品。有关更多信息，请参阅 创建产品列表。
- （可选）对于付费产品，提供 14-30 天的免费试用，让客户试用您的产品。有关更多信息，请参阅机器学习产品。
- （可选）对于模型包产品，如果您想在产品列表页面上启用实时产品演示，请联系 AWS Marketplace 卖家运营团队。产品演示允许潜在买家直接在列表页面上试用您的模型，而无需自己订阅或部署模型。

使用信息要求

描述产品预期输入和输出的清晰使用信息（附示例）对于提供积极的买家体验至关重要。

对于添加到产品列表中的每个新资源版本，都必须提供使用信息。

要为首次发布的新产品添加使用信息，请登录 AWS Marketplace 管理门户 控制台。从产品下拉列表中，选择机器学习。选择您的产品。在启动选项下的产品概述中，提供您的模型包或算法资源，然后选择添加。ARN

要编辑特定版本的现有使用信息，请选择启动选项下的编辑，然后选择编辑版本。

输入和输出的要求

清晰解释您的格式，并举例说明输入和输出，对于帮助您的买家了解和使用您的产品非常重要。这种理解可以帮助您的买家对输入数据进行任何必要的转换，以获得最佳的推理结果。

将您的 Amazon SageMaker 资源添加到您的产品清单时，系统会提示您输入以下信息。

推理输入和输出

对于推理输入，请提供实时端点和批量转换作业的输入格式。包括用于对数据进行任何必要预处理的代码片段。包括支持的MIME内容类型（例如 image /jpeg、image/png 、image/ bmp）、值描述（如果适用）和限制。包括托管在上的输入样本GitHub。

对于推理输出，请提供实时端点和批量转换作业的输出格式。如果适用，请包括输出MIME内容类型（例如 application/json、image/jpeg）和值描述。包括托管在上的输出示例GitHub。

对于示例，请提供适用于您的产品的输入文件。如果您的模型执行多分类器，请为每个类提供至少一个示例输入文件。

训练输入

在训练模型的信息部分，提供输入数据格式和代码片段，以便对数据进行任何必要的预处理。包括支持的MIME内容类型（例如，图像/jpeg、image/png 、image/ bmp）、值描述（如果适用）以及限制。确保包括托管在上的输入样本GitHub。

解释买家可以提供的可选和必备特征，并指定是否支持 PIPE 输入模式。如果支持分布式训练（使用大于 1 个CPU/GPU实例的训练），请指定此项。要进行调整，请列出推荐的超参数。

Jupyter 笔记本的要求

将您的 SageMaker 资源添加到产品列表时，请提供一个指向托管的 Jupyter 笔记本样本的链接 GitHub，该笔记本无需要求买家上传或查找任何数据即可演示完整的工作流程。

使用 AWS SDK for Python (Boto)。精心开发的示例笔记本可以让买家更轻松地尝试使用您的产品。

对于模型包产品，您的示例笔记本演示了输入数据的准备、实时推理端点的创建以及批量转换作业的性能。有关更多信息，请参阅上的 Model Package 列表和示例笔记本 GitHub。有关示例笔记本，请参阅 a uto_ insurance。这款笔记本可以正常工作 AWS 区域，无需输入任何参数，买家也无需查找样本数据。

> **Note**
>
> 开发不足的 Jupyter 笔记本示例无法显示多个可能的输入和数据预处理步骤，这可能会使买家难以完全了解您产品的价值主张。

对于算法产品，示例笔记本演示了完整的训练、调整、模型创建、实时推理端点的创建以及批量转换作业的性能。有关更多信息，请参阅算法列表和示例笔记本 GitHub。有关示例笔记本，请参阅 amazon_demo_pro duct 和 automl on。 GitHub这些示例笔记本可在所有区域使用，无需输入任何参数，也无需买家查找示例数据。

> **Note**
>
> 缺少示例训练数据可能会使您的买家无法成功运行 Jupyter 笔记本。开发不足的示例笔记本可能会使您的买家无法使用您的产品并阻碍其采用。

机器学习产品列表的要求和建议摘要

下表提供了机器学习产品列表页面要求和建议的摘要。

详细信息	对于模型包列表	对于算法列表
Product descriptions		
详细解释产品对支持的内容类型（例如，“检测图像中的 X”）的作用。	必需	必需
提供有关产品的有说服力和差异化的信息（避免使用诸如 “最佳” 或未经证实的说法之类的形容词）。	推荐	推荐
列出该产品最重要的用例。	必需	必需
描述其训练所依据的数据（来源和大小），并列出所有已知的限制。	必需	不适用
描述模型所依据的核心框架。	推荐	推荐
根据验证数据汇总模型性能指标（例如，“使用 Z 数据集进行基准测试的 XX.YY 精度百分比”）。	必需	不适用
根据建议的实例类型汇总模型延迟和/或吞吐量指标。	必需	不适用
描述算法类别。例如，“这种决策森林回归算法基于树结构分类器的集合，这些分类器是使用引导聚合的通用技术和随机选择的特征构建的。”	不适用	必需

详细信息	对于模型包列表	对于算法列表
Usage information		
为了进行推理，请提供实时端点和批量转换作业的输入格式。包括支持的MIME内容类型（例如，图像/jpeg、image/ png 、image/ bmp ）、值描述（如果适用）以及限制。请参阅 输入和输出的要求。	必需	必需
为了进行推理，请为实时端点和批量转换作业提供输入样本。样品必须托管在 GitHub。请参阅 输入和输出的要求。	必需	必需
为了进行推理，请提供实时端点和批量转换作业的输出格式。如果适用，请包括输出MIME内容类型（例如 application/json、image/jpeg ）和值描述。请参阅 输入和输出的要求。	必需	必需
为了进行推理，请为实时端点和批量转换作业提供输出样本。样品必须托管在 GitHub。请参阅 输入和输出的要求。	必需	必需
为了进行推理，请举一个使用端点或批处理转换作业的示例。使用 AWS Command Line Interface (AWS CLI) 命令或使用，包括一个代码示例 AWS SDK。	必需	必需

详细信息	对于模型包列表	对于算法列表
对于培训，请提供输入格式。包括支持的MIME内容类型（例如，image/jpeg、im age/pn g、image/bmp）、值描述（如果适用）和限制（例如，所需的最小数据行数）。请参阅 输入和输出的要求。	不适用	必需
要进行训练，请提供托管在上的输入样本 GitHub。请参阅 输入和输出的要求。	不适用	必需
要进行培训，请举一个执行训练作业的示例。描述支持的超参数、其范围及其总体影响。指定算法是否支持超参数调整、分布式训练或GPU实例。包括代码示例 AWS SDK，例如 AWS CLI 命令或使用。	不适用	必需
提供一本 Jupyter 笔记本，用于 GitHub 演示产品的完整使用情况。请参阅 Jupyter 笔记本的要求。	必需	必需
提供与产品使用相关的技术信息，包括用户手册和样本数据。	推荐	推荐

服务限制和限额

本部分介绍 AWS Marketplace中的机器学习 (ML) 产品的限制和限额。

网络隔离

出于安全目的，当买家订阅您的容器化产品时，Docker 容器在隔离（无网络访问）的环境中运行。在您创建容器时，请勿通过互联网进行传出调用，否则将失败。对的调用 AWS 服务 也会失败。

映像大小

您的 Docker 映像大小受 Amazon Elastic Container Registry (Amazon ECR) 服务限额的约束。Docker 映像大小会影响训练作业、批量转换作业和端点创建期间的启动时间。为了获得更好的性能，请保持最佳 Docker 映像大小。

存储大小

当您创建终端节点时，亚马逊会将亚马逊弹性区块存储 (Amazon EBS) Elastic Block Store 存储卷 SageMaker 附加到托管该终端节点的每个 ML 计算实例。（终端节点也称为实时推理或 Amazon SageMaker 托管服务。）存储卷的大小取决于实例类型。有关更多信息，请参阅 Amazon SageMaker 开发者指南中的主机实例存储卷。

有关批量转换，请参阅《Amazon SageMaker 开发者指南》中的 “批量转换中的存储”。

实例大小

SageMaker 提供了一系列针对不同机器学习用例进行了优化的实例类型。实例类型包括 CPU、GPU、内存和网络容量的不同组合。实例类型便于您灵活选择适合构建、训练和部署机器学习模型的资源组合。有关更多信息，请参阅 Amazon SageMaker ML 实例类型。

推理的有效负载大小

对于端点，将每次调用的输入数据的最大大小限制为 6 MB。此值无法调整。

对于批量转换，每次调用输入数据的最大大小为 100 MB。此值无法调整。

推理的处理时间

对于端点，每次调用的最大处理时间为 60 秒。此值无法调整。

对于批量转换，每次调用的最大处理时间为 60 分钟。此值无法调整。

服务限额

有关与训练和推理相关的配额的更多信息，请参阅 Amazon S SageMaker ervice Q uotas。

异步推理

中发布的模型包和算法 AWS Marketplace 无法部署到为 Amazon SageMaker 异步推理配置的终端节点。为异步推理配置的端点要求模型具有网络连接。所有 AWS Marketplace 型号均在网络隔离下运行。有关更多信息，请参阅无网络访问。

无服务器推理

中发布的模型包和算法 AWS Marketplace 无法部署到为 Amazon SageMaker Serverless Infer ence 配置的终端节点。为无服务器推理配置的端点要求模型具有网络连接。所有 AWS Marketplace 型号均在网络隔离下运行。有关更多信息，请参阅无网络访问。

托管的竞价型训练

对于来自的所有算法 AWS Marketplace，的值设置`MaxWaitTimeInSeconds`为 3,600 秒（60 分钟），即使实施了托管点训练的检查点。此值无法调整。

Docker 镜像和 AWS 账户

要发布，图片必须存储在卖家拥有的 Amazon ECR 存储库中。 AWS 账户 无法发布存储在他人拥有的存储库中的图像 AWS 账户。

根据内置算法发布模型包或 AWS Marketplace

无法发布使用 Amazon SageMaker 内置算法或订 AWS Marketplace 阅算法的训练作业创建的模型包。

您仍然可以使用训练作业中的模型构件，但是发布模型包需要您自己的推理映像。

AWS 区域 支持发布

AWS Marketplace 支持在符合以下条件 AWS 区域 的情况下发布模型包和算法资源：

- 亚马逊 SageMaker支持的地区
- 默认选择加入的可用区域（例如， describe-regions 返回 `"OptInStatus": "opt-in-not-required"`）

发布模型包或算法产品所需的所有资产都必须存储在您选择发布的同一区域。这包括以下这些：

- 在 Amazon 中创建的模型包和算法资源 SageMaker

- 上传到 Amazon ECR 存储库的推理和训练映像
- 存储在 Amazon Simple Storage Service (Amazon S3) 中并在模型包资源的模型部署期间动态加载的模型构件（如有）
- 存储在 Amazon S3 中的用于推理和训练验证的测试数据

您可以在支持的任何地区开发和培训您的产品 SageMaker。但是，在发布之前，您必须将所有资源复制到 AWS Marketplace 支持发布的区域，并在该区域中重新创建资源。

在发布过程中，无论您从哪个地区发布，您都可以选择要发布到的区域，并在其中发布您的产品。AWS 区域

故障排除

本部分为您在机器学习产品的发布过程中可能遇到的一些常见错误提供帮助。如果您的问题未列出，请联系 AWS Marketplace 卖家运营团队。

常规：当我在模型包或算法中添加我的模型包的 Amazon 资源名称 (ARN) 时，出现了 400 错误 AWS Marketplace 管理门户

如果您使用 Amazon SageMaker 控制台创建资源，则必须在 “发布此模型包” 流程的最后一页上选择 “是”， AWS Marketplace或者在 “发布此算法” 时选择 “是” AWS Marketplace。您不能选择否然后稍后再发布。选择是不会发布模型包或算法。但是，它会在创建模型包或算法资源时对其进行验证，这是在 AWS Marketplace中使用所必需的。

如果您使用 AWS 软件开发工具包创建模型包或创建算法，请确保将参数设置`CertifyForMarketplace`为`true`。

重新创建经过认证和验证的模型包或算法资源后，在 AWS Marketplace 管理门户中添加新的 ARN。

一般：当我在模型包或算法中添加模型包或算法的 ARN 时，出现了 404 错误 AWS Marketplace 管理门户

此错误可能是由于以下几个原因引起的：

- ARN 可能无效。确保您使用的是正确的 ARN。
  - 对于模型包，ARN 应类似于 `arn:aws:sagemaker:us-east-2:000123456789:model-package/my-model-package-name`。
  - 对于算法，ARN 应类似于 `arn:aws:sagemaker:us-east-2:000123456789:algorithm/my-algorithm`。

- 模型包或算法资源不是在与卖家账户 AWS 账户 相同的情况下创建的。确保所有用于发布的资源和资产都在您发布内容的卖家账户中。
- 您用于发布的用户或角色没有访问模型包或算法资源的正确 IAM 权限。确保您的用户或角色拥有以下权限：
  - 对于模型包，必须允许对模型包资源执行 `sagemaker:DescribeModelPackage` 操作。
  - 对于算法，必须允许对算法资源执行 `sagemaker:DescribeAlgorithm` 操作。

一般：我在中为我的算法产品指定定价时收到 500 错误 AWS Marketplace 管理门户

当您尝试发布仅包含训练映像而没有随附推理映像的算法资源时，可能会发生此错误。上发布的算法资源 AWS Marketplace 必须同时包含这两个组件。有关更多信息，请参阅 在里面准备好你的产品 SageMaker。

Amazon SageMaker：当我创建模型包或算法资源时，我收到 “客户端错误：注册表访问被拒绝” 失败消息

当用于创建模型包或算法的映像存储在属于另一个 AWS 账户的 Amazon ECR 存储库中时，可能会发生此错误。模型包或算法验证不支持跨账户映像。将该映像复制到您用来发布的 Amazon ECR 存储库中。 AWS 账户 然后，使用新的映像位置继续创建资源。

Amazon SageMaker：我收到 “未启动” 和 “客户端错误：未计划扫描...” 失败消息

当 SageMaker 无法启动对存储在 Amazon ECR 中的 Docker 容器映像的扫描失败时，就会发生此错误。如果发生这种情况，请打开 Amazon ECR 控制台，找到您的映像上传到的存储库，选择映像，然后选择扫描。

机器学习报告

AWS Marketplace 为您的 Amazon SageMaker 商品生成报告，其中包括有关买家、财务、使用情况和税费的数据。所有报告均可在 “报告” 页面 AWS Marketplace 管理门户 上找到。有关更多信息，请参阅卖家报告。

以下小节总结了如何报告您使用的 SageMaker 机器学习产品的财务状况。

每日业务报告

每日业务报告为每个买家和产品提供实例类型、使用时长、软件费用收入以及其他详细信息。买家通过唯一且匿名的客户参考 ID 进行识别。有关更多信息，请参阅每日业务报告。

月度收入报告

月度收入报告为您提供因使用您的软件而向买家收取的月度收入。有关更多信息，请参阅月度账单收入报告。

支付报告

月度支付报告提供了在软件费用结算周期内代表您收取的所有资金的明细。报告中反映的结算总金额应与存入您银行账户的金额一致。有关更多信息，请参阅支付报告。

其他报告和分析

有关其他可用报告，请参阅卖家报告。

您也可以使用 AWS Marketplace中的可用卖家传输数据源服务创建自定义报告。

基于 SaaS 的产品

使用软件即服务 (SaaS) 产品，您可以部署托管在 AWS 基础架构上的软件，并授予购买者在您的 AWS 环境中访问该软件的权限。您负责管理软件中的客户访问、账户创建、资源预置和账户管理。

有关将您的API网关APIs与 Amazon Gateway 集成的信息，请参阅《亚马逊API网关开发者指南》AWS Marketplace中的 “APIs通过销售您的API网关”。

如需有关 SaaS 产品的帮助，请联系 AWS Marketplace 卖家运营团队。

主题

- 开始使用 SaaS 产品
- 规划您的 SaaS 产品
- SaaS 产品准则
- SaaS 产品定价
- SaaS 免费试用
- SaaS 客户载入
- SaaS 产品的亚马逊SNS通知
- 访问 AWS Marketplace 计量和授权服务 APIs
- 报告
- SaaS 产品集成的代码示例
- AWS PrivateLink 与一起使用 AWS Marketplace

开始使用 SaaS 产品

本章概述了软件即服务 (SaaS) 产品如何为创建和维护它们的卖家工作。本部分介绍如何在 AWS Marketplace中使用 SaaS 产品。还介绍了如何根据SaaS产品的计费模式将您的SaaS产品与适当的 AWS Marketplace API操作集成。

先决条件

在开始之前，您必须满足以下先决条件：

1. 访问和使用 AWS Marketplace 管理门户。这是您用来注册为卖家并管理您销售的商品的工具 AWS Marketplace。有关更多信息，请参阅 AWS Marketplace 管理门户。

2. 注册为卖家并提交您的税务和银行信息。有关更多信息，请参阅 卖家注册过程。
3. 规划如何在 AWS Marketplace中创建和集成 SaaS 产品。有关更多信息，请参阅 规划您的 SaaS 产品。

主题

- SaaS 产品生命周期
- 创建 SaaS 产品
- 创建初始 SaaS 产品页面
- SaaS 产品设置
- 集成 SaaS 订阅产品
- 集成 SaaS 合同产品
- 将你的 SaaS 合同与 pay-as-you-go 产品集成
- 部署无服务器 SaaS 集成解决方案

SaaS 产品生命周期

当你在中创建 SaaS 产品时 AWS Marketplace，该产品最初发布时可见性有限，因此只有你的账户才能访问它。准备就绪后，您可以将其发布到 AWS Marketplace 目录中，以允许买家订阅和购买您的产品。

在 SaaS 产品页面上，您可以查看您的产品列表。根据其在产品生命周期中的阶段，产品将具有以下状态之一：

- 暂存 – 未完成的产品，您仍在为其添加信息。在第一次保存并退出自助服务体验时，成功的变更请求会使用您提交的完成步骤中的信息创建未发布的产品。在此状态下，您可以继续向产品添加信息或通过变更请求更改已提交的详细信息。
- 受限 – 产品在提交给系统并通过系统中的所有验证后即已完成。然后，产品将发布到受限状态。此时，该产品有一个详细信息页面，只有您的账户和您列入允许列表的人才能访问该页面。您可以通过详细信息页面测试您的产品。如需更多信息或帮助，请联系 AWS Marketplace 卖家运营团队。
- 公开 – 当您准备发布产品以便买家可以查看和订阅产品时，您可以使用更新可见性更改请求。此请求启动了 AWS Marketplace 卖家运营团队根据 AWS 政策审查和审核您的商品的工作流程。产品获得批准并处理变更请求后，产品将从受限状态更改为公开。有关 AWS 指南的信息，请参阅 SaaS 产品指南。

- 受限 – 果您想阻止新用户订阅您的产品，则可以使用更新可见性更改请求来限制该产品。受限状态表示现有用户可以继续使用该产品。但是，该产品将不再对公众可见，也不会再向新用户提供。

您可以在暂存、受限和公开状态下更新您的产品。有关更多信息，请参阅更新产品信息。

创建 SaaS 产品

要在中销售软件即服务 (SaaS) 产品 AWS Marketplace，您需要遵循以下主要步骤：

- 在中创建 SaaS 产品 AWS Marketplace。
- 将 SaaS 订阅、合同或合同与 pay-as-you-go产品集成 AWS Marketplace。
- 测试 pay-as-you-go产品与集成的订阅、合同或合同 AWS Marketplace。
- 提交您的产品以供发布。

下一节中的步骤说明了如何完成这些任务。

创建 SaaS 产品

创建 SaaS 产品

1. 决定上架一款 SaaS 产品

   有一款你想销售的 SaaS 产品 AWS Marketplace。查看并了解如何规划您的 SaaS 产品。

2. 确定定价和优惠类型

   SaaS 产品有三种优惠类型：订阅、合同和合同 pay-as-you-go。您选择的优惠类型会影响您的 SaaS 产品与集成的方式 AWS Marketplace。有关更多信息，请参阅 规划您的定价。

3. 收集资产

   收集提交产品所需的资产。资产包括：

   - 产品徽标 URL — 可公开访问的 Amazon S3URL，其中包含您所提供的产品徽标的清晰图像。
   - 最终用户许可协议 (EULA) URL-您的产品必须EULA具有可作为PDF文件提供的。您必须提供指向 Amazon S3 存储桶的链接，客户可以在其中查看您的产品 AWS Marketplace 页面EULA上的内容。
   - 产品注册 URL — 这是成功订阅您的产品后买家被重定向URL的地方。 AWS Marketplace
   - 有关产品的元数据 – 您在 AWS Marketplace 管理门户的产品创建向导中提供元数据。

- 产品的支持信息-此信息包括电子邮件地址和URLs产品支持渠道。

4. 提交您的产品进行集成

   使用您的卖家账户和 AWS Marketplace 管理门户创建初始 SaaS 产品页面。 AWS Marketplace 会将您的产品作为限量产品发布，这意味着它只能用于集成和测试。您的产品代码和 Amazon Simple Notification Service (SNS) 主题将在产品概述页面上提供。

   > **Note**
   >
   > 您的商品必须保持较低的价格，这样您和 AWS Marketplace 卖家运营团队就可以在不产生高额成本的情况下测试您的商品。当您请求公开产品时，我们会要求您提供产品的实际价格。

5. 与集成 AWS Marketplace

   您的产品必须支持客户载入和使用您的产品，包括在向他们提供访问权限之前验证他们的订阅，在某些情况下，还要计量他们的用量。您的集成方式 AWS Marketplace 取决于您为产品使用的优惠类型。有关集成的更多信息（基于优惠类型），请参阅以下主题：

   - 订阅集成
   - 合同集成
   - pay-as-you-go 集成合同

   将您的产品与集成的最后一步 AWS Marketplace 是对其进行测试，以确保集成工作正常。

6. 提交您的产品以供发布

   验证集成并准备好让产品上线后，选择更新可见性。AWSMarketplace 卖家运营团队将审核您的商品并更新价格，然后才能将可见性更新为公开。

   > **Note**
   >
   > AWS Marketplace 卖家运营部门使用手动流程来验证和更新 SaaS 产品。更新公众可见性的过程需要 7-10 个工作日，如果团队发现错误，则需要更长的时间。有关计时的更多信息，请参阅本指南中的时机和期望。

创建初始 SaaS 产品页面

使用您的软件即服务 (SaaS) 应用程序元数据，并使用在 AWS Marketplace 目录中创建初始 SaaS 产品。 AWS Marketplace 管理门户

要创建初始 SaaS 产品页面，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 选择创建 SaaS 产品，然后选择 SaaS 产品。
3. 生成 SaaS 产品 ID 和代码。您也可以添加可选标签以支持基于标签的授权。

> **Note**
>
> 有关基于标签的授权的信息，请参阅《AWS Identity and Access Management 用户指南》中的使用标签控制对 AWS 资源的访问权限。

4. 使用自助服务体验创建 AWS Marketplace 房源。添加产品信息、产品部署详细信息和公开优惠详情。或者，您可以将账户添加到允许列表中以测试产品。

> **Note**
>
> 如果您需要在完成步骤之前结束会话，请选择保存并退出选项，将您当前的选择保存到暂存区域。此选项会创建验证您提供的信息的请求。验证您的请求期间，您无法编辑产品。
> 如果您的请求成功，则可以通过选择恢复产品创建来继续创建产品。
> 如果您的请求失败，则是由于验证错误所致，该错误可在产品请求日志中看到。选择查看错误的请求，然后从操作下选择复制到新的来更正错误，然后重新提交请求。要更新之前的步骤，请打开产品详细信息页面并提交更改请求。

> **Note**
>
> 测试期间，您的价格将默认为每个维度 0.01 美元。这个价格允许您在受限状态下测试您的产品，而不会产生大笔账单。在公开产品时，您需要提供实际价格。

5. 选择提交。然后， AWS Marketplace 验证信息。如果验证成功，则以 “受限” 状态 AWS Marketplace 发布产品。验证成功后，您可以预览、集成和测试您的产品。

> **Note**
>
> 验证过程中，您无法编辑产品。当您的商品最初发布 AWS 账户 时，只有创建商品的用户和 AWS Marketplace 卖家运营团队的测试账户才能访问该商品。如果您从 SaaS 产品页面查看产品，则可以选择查看开启 AWS Marketplace以查看产品详情，因为它们将显示在 AWS Marketplace 买家面前。其他 AWS Marketplace 用户看不到此详细信息列表。

SaaS 产品设置

在中创建软件即服务 (SaaS) 产品后 AWS Marketplace，可以在必要时修改该产品的许多设置。有关更多信息，请参阅以下主题。

主题

- 管理更改请求
- 更新产品信息
- 更新允许名单 AWS 账户 IDs
- 更新产品可见性
- 更新定价条款
- 添加定价维度
- 更新定价维度
- 限制定价维度
- 确定买家将如何访问您的产品
- 按国家/地区更新可用性
- 更新产品的退款政策
- 更新最终用户许可协议 (EULA)

管理更改请求

在自助服务列表中，您可以使用更改请求对产品进行更改。您当前的请求可以在请求选项卡 AWS Marketplace 管理门户上找到。您可以通过导航栏下方的请求更改下拉列表提出新的请求。

要为 SaaS 产品创建更改请求，请执行以下操作：

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/，然后登录您的卖家账户。
2. 在产品选项卡中，从下拉列表中选择 SaaS。
3. 提交请求后，就会开始处理。变更请求将处于以下状态：审核中、正在准备更改和正在应用更改。
4. 请求的处理完成后，其状态将更改为以下值之一：
   - 成功 - 此状态表示您请求的更改已处理并且更改已反映在系统中。
   - 失败 - 此状态表示请求出现点问题，更改未得到处理。如果状态为失败，您可以选择请求以查找错误代码，这些错误代码提供了有关如何更正问题的建议。您可以对错误进行故障排除，并创建新的更改请求。为了加快处理速度，您可以使用复制到新请求功能来复制失败请求的详细信息。您可以进行必要的更改并重新提交请求。

以更新开头的更改请求将加载项目的当前详细信息。然后，您可以进行更新，从而覆盖现有的详细信息。添加和限制请求对专门用于在每次请求成功后预置的更新（在自助服务体验中选择保存并退出和提交操作之后）。也就是说，现有订阅用户可以继续使用该产品，直到他们的订阅或合同到期。但是，不能向处于受限状态的产品添加新的订阅用户。

更新产品信息

创建产品后，您可能需要在 AWS Marketplace中更改与其关联的信息。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/，然后登录您的卖家账户。
2. 在 SaaS 产品页面的 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新产品信息。
4. 更新以下任何您想要更改的字段：
   - 产品标题
   - SKU
   - 简短描述
   - 详细描述
   - 产品徽标 URL
   - 要点
   - 产品类别

- 关键词
- 产品视频 URL
- 资源
- 支持信息

> **Note**
>
> 有关徽标格式的详细信息，请参阅公司和产品徽标要求。

5. 要更新产品信息，请选择提交。
6. 验证请求是否显示在请求选项卡上，状态为正在审核。您可能需要刷新页面才能看到您的新请求。

更新允许名单 AWS 账户 IDs

您可以更改可以在 AWS 账户 IDs受限状态下查看您的产品的列表。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/，然后登录您的卖家账户。
2. 在 SaaS 产品页面的 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新允许列表。列表显示了当前 AWS 账户 IDs已列入许可名单的。
4. 在 “允许名单 AWS 账户” 字段中，输入 AWS 账户 IDs并使用逗号分隔它们。
5. 要更新允许名单 AWS 账户 IDs，请选择提交。

更新产品可见性

要更改哪些买家可以查看您的快速启动体验 AWS Marketplace，您可以使用 “更新” 可见性。

1. 打开 AWS Marketplace 管理门户，然后登录到卖家账户。
2. 在 SaaS 产品页面上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新可见性。

> **Note**
>
> 您可以使用此更改请求请求将产品从受限态更改为公开状态。但是，变更请求必须经过 AWS Marketplace 卖家运营团队的批准才能移至公开。

4. 当您向公众发布时，你需要提供产品的实际价格。此价格将在您的列表获得公开批准后应用。
5. 要提交请求以供审核，请选择提交。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

更新定价条款

如果您想更改 SaaS 产品的每个维度的定价，则可以使用更新定价条款。

> **Note**
>
> 任何维度的定价上调都将导致定价更新选项至少在接下来的 90 天内不可用。如果同时更新降价和上调，请先更新降价。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/，然后登录您的卖家账户。
2. 在 SaaS 产品页面的 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新公开优惠，然后选择更新定价条款。
4. 当前定价已在字段中预先填写。您可以删除当前价格，然后添加新价格。
5. 要提交请求以供审核，请选择提交。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

添加定价维度

您可以添加要用于向产品收费的维度。维度是买家在使用您的产品时需要支付的基本计量单位。

> **Note**
>
> 要更新现有定价维度的名称或描述，请参阅the section called “更新定价维度”。

1. 打开 AWS Marketplace 管理门户，然后登录到卖家账户。
2. 在 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新定价维度，然后选择添加定价维度。
4. 提供维度API标识符、显示名称和描述以向您的产品添加新维度，然后选择 “下一步”。

> **Note**
>
> 标API识符和名称在所有维度上都必须是唯一的。创建维度后，您无法更改API标识符和单位。

5. 为您添加的每个维度定义价格，然后选择下一步以查看您的更改。

> **Note**
>
> 您只能为为产品选择的定价模式添加维度（例如，合同、用量或含消费的合同）。对于受限产品，新增维度的价格设置为 0.01 美元。当产品准备好向公众提供时，您可以更新价格。

6. 选择提交，提交您的请求以供审核。
7. 在请求选项卡中，验证请求状态是否为正在审核。在请求完成后，状态将更改为成功。

更新定价维度

您可以更新要用于向产品收费的维度。维度是买家在使用您的产品时需要支付的基本计量单位。

1. 打开 AWS Marketplace 管理门户，然后登录到卖家账户。
2. 在 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新定价维度，然后选择更新维度信息。
4. 找到要更新的维度，然后选择名称或描述。
5. 提供新的名称或描述，然后选中复选标记以确认更新。

> **Note**
>
> 维度名称必须唯一。

6. 选择提交，提交您的请求以供审核。
7. 在请求选项卡中，验证请求状态是否为正在审核。在请求完成后，状态将更改为成功。

限制定价维度

您可以限制产品中当前列出的维度。此请求会将所选维度从产品中删除。

1. 打开 AWS Marketplace 管理门户，然后登录到卖家账户。
2. 在 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新定价维度，然后选择限制定价维度。
4. 对于受限及公开产品，系统会提示您使用联系我们按钮联系 AWS Marketplace 卖家运营团队。使用此表单提供要从产品列表中删除维度的详细信息。

确定买家将如何访问您的产品

您可以选择以下选项之一，作为买家访问您产品的方式：

- the section called “更新 SaaS URL 配送选项”— 客户使用来URL表示订阅您的产品后他们被重定向到的网站。 AWS Marketplace
- the section called “配置快速启动” – 客户使用简化的流程来配置和启动您的产品。您可以为具有“受限”或“公开”可见性的现有产品完成此配置。

更新 SaaS URL 配送选项

要更新用于配送您URL的 SaaS 产品的，请使用更新发货选项选项卡。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/，然后登录您的卖家账户。
2. 在 SaaS 产品页面的 SaaS 产品选项卡上，选择要修改的产品。
3. 从 “请求更改” 下拉列表中，选择 “更新公开报价”，然后选择 “编辑默认配送” URL。
4. 在 “发货 URL” 字段中，输入新URL的 SaaS 产品发货选项。
5. 要提交请求以供审核，请选择提交。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

配置快速启动

中列出的 SaaS 产品 AWS Marketplace 通常需要在订阅购买者的账户中部署 AWS 资源（例如IAM角色）。Quick Launch 允许您使用 AWS CloudFormation 模板为买家提供指导、 step-by-step 说明和资源部署。买家使用 CloudFormation 模板来配置和发布产品。

> **Tip**
> 要了解有关快速启动配置过程的更多信息，请参阅启用 SaaS 快速启动实验。

要配置客户可用于启动您的 SaaS 产品的快速启动体验，请使用交付选项选项卡。

1. 在 SaaS 产品页面的 SaaS 产品选项卡上，选择要修改的产品。

   > **Note**
   > 要配置快速启动体验，产品必须具有“受限”或“公开”可见性。

2. 在产品详细信息页面上，选择交付选项选项卡。
3. 对于快速启动，请选择激活并配置按钮。
4. 有关账户登录详情，请URL为您的网站提供一个，买家可以在其中登录或创建账户。这为买家体验URL打开了一个新的选项卡。然后，买家登录或创建一个账户，然后返回 AWS Marketplace 以启动模板。
5. 对于 AWS CloudFormation 模板，选择添加 AWS CloudFormation 模板按钮并提供以下信息：

   1. 标题-提供 CloudFormation 部署的名称。
   2. 描述 – 提供模板的描述。
   3. 堆栈名称 – 提供堆栈的名称。此名称是中买方的堆栈名称 CloudFormation。
   4. CloudFormation 模板 URL — 为模板提供亚马逊简单存储服务 (Amazon URL S3)。 AWS 将审阅此模板并 AWS 提供最终模板URL。

      > **Note**
      > 为了简化客户的启动过程，我们建议尽量减少与配置过程相关的模板数量。理想情况下，您需要一个模板来部署使用产品所需的资源。有关 CloudFormation 模板的问题，请联系您的 AWS Marketplace 业务发展合作伙伴或AWS Marketplace 卖家运营团队。

   5. 所需IAM权限-提供部署 CloudFormation 模板所需的权限。如果您想共享部署参数（这些参数作为密钥存储在 AWS Secrets Manager 中，以供买家使用），您的策略必须包含以下操作：

      - `secretsManager:ListSecrets`
      - `secretsManager:DescribeSecret`

- `secretsManager:ReplicateSecretToRegions`
- `secretsManager:GetSecretValue`

> **Note**
>
> 如果您的产品需要卖家提供的 CloudFormation 部署参数（例如API密钥和外部参数 IDs），请使用该`PutDeploymentParameter`操作与您的客户共享该参数。有关更多信息，请参阅《AWS Marketplace 部署服务API参考》PutDeploymentParameter中的。

6. （可选）有关手动配置说明，请为想要手动配置产品的买家提供说明。考虑包含产品入门指南和文档的链接。
7. 有关 Launch 的详细信息，请提供 CloudFormation 堆栈部署后买家将在URL何处访问产品。
8. （可选）对于快速启动的允许名单账户，请提供一个以逗号分隔的列表 AWS 账户 ，这些账户可以在有限的可见性下查看快速启动体验。
9. 选择提交按钮。快速启动体验的可见性状态将为“受限”，这意味着只有您的账户和列入许可名单的账户才能看见。由于可见性有限，您可以在订阅产品并选择 “设置您的帐户” 按钮后，使用配置和启动页面测试您的配置。
10. 准备就绪后，可以在 AWS Marketplace 目录中发布快速启动体验。在产品详细信息页面上，使用交付选项选项卡上的更新快速启动可见性按钮。

    当您将可见性更改为 “公开” 时， AWS Marketplace 卖家运营团队将审查配置、进行买家测试并发布体验。

    > **Note**
    >
    > 如果您在启用快速启动体验时需要支持，请联系 AWS Marketplace 卖家运营团队。

按国家/地区更新可用性

您可以定义提供产品的国家/地区。

1. 打开 AWS Marketplace 管理门户，然后登录到卖家账户。
2. 在 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新公开优惠，然后选择按国家/地区更新可用性。

4. 请选择以下选项之一：
   - 所有国家/地区 – 适用于所有受支持的国家/地区。
   - 所有国家/地区（部分除外） – 在所有受支持的国家/地区均可用，部分国家/地区除外。
   - 仅加入允许列表的国家/地区 – 提供产品的国家/地区的具体列表。
5. 选择提交，提交您的请求以供审核。
6. 在请求选项卡中，验证请求状态是否为正在审核。在请求完成后，状态将更改为成功。

更新产品的退款政策

您可以使用更新退款政策来更新产品的退款政策。

1. 打开 AWS Marketplace 管理门户 at https://aws.amazon.com/marketplace/管理/tour/，然后登录您的卖家账户。
2. 在 SaaS 产品页面的 SaaS 产品选项卡上，选择要修改的产品。
3. 从请求更改下拉列表中，选择更新公开优惠，然后选择更新退款政策。
4. 文本框中提供了当前退款政策的详细信息。根据需要查看和修改详细信息。提交请求会改写当前的退款政策。
5. 要提交请求以供审核，请选择提交。
6. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态将更新为成功或失败。

更新最终用户许可协议 (EULA)

您可以EULA为订阅您的产品的新用户更新您的信息。

1. 打开 AWS Marketplace 管理门户，然后登录到卖家账户。
2. 在 SaaS 产品选项卡上，选择要修改的产品。
3. 从 “请求更改” 下拉列表中，选择 “更新公开报价”，然后选择 “更新” EULA。
4. 您可以选择 AWSMarketplace 的标准合同 (SCMP) 或提交自定义合同EULA。对于定制EULA，您必须为合同提供 Amazon Simple Storage Service（Amazon S3）URL。

> **Note**
>
> 您的 Amazon S3 存储桶必须可公开访问。

5. 选择提交，提交您的请求以供审核。
6. 在请求选项卡中，验证请求状态是否为正在审核。在请求完成后，状态将更改为成功。

集成 SaaS 订阅产品

将您的产品与集成 AWS Marketplace 仅一步之遥创建 SaaS 产品。要将您的软件即服务 (SaaS) 订阅产品与集成 AWS Marketplace，您必须编写代码并证明它可以成功响应多种客户场景。以下各部分描述了这些场景以及如何应对这些场景，并概述了测试集成的情况。

> **Note**
>
> 在开始之前，请确保您已为自己的 software-as-a-service （SaaS）产品选择了正确的定价模式 AWS Marketplace。有关更多信息，请参阅 规划您的 SaaS 产品。

主题

- 场景：您的服务验证新客户
- 场景：计量用量
- 场景：监控用户订阅的更改
- 场景：验证客户订阅
- 测试您的 SaaS 订阅产品集成

场景：您的服务验证新客户

当客户订阅您的产品时，他们会被重定向到您的注册 URL，这是一个带有临时 `x-amzn-marketplace-token` 令牌的 HTTP POST 请求。通过下列方式来响应此请求：

1. 通过调用 AWS Marketplace Metering Service中的 `ResolveCustomer` API 操作将令牌交换为 `CustomerIdentifier`、`CustomerAWSAccountId` 和 `ProductCode`。
2. 在系统中保留 `CustomerIdentifier`、`CustomerAWSAccountID` 和 `ProductCode`，以备将来调用。您必须存储客户是否拥有有效的订阅，以及您需要的有关该客户的任何信息。
3. 作为对请求的响应，您必须展示用户的首次使用体验（如果适用于您的服务）。

场景：计量用量

当客户开始使用您的服务时，您必须每小时发送一次计量记录。有关如何计量的详细信息，请参阅用量的计量。

我们建议您使用 AWS CloudTrail 监控活动，以确保账单信息已发送至 AWS。在发送计量记录时，请记住以下几点：

- 将按小时去除重复的计量请求。
- 每小时发送的记录是累积的。
- 即使过去一小时内没有记录，我们也强烈建议实施最佳实操，即每小时发送一次计量记录（用量为 0）。

场景：监控用户订阅的更改

设置 Amazon Simple Queue Service (Amazon SQS) 队列，并订阅您的产品的 Amazon SNS 主题。您的 SNS 主题信息包含在您创建商品时从 AWS Marketplace 卖家运营团队收到的电子邮件中。有关更多信息，请参阅 创建 SaaS 产品。订阅 SNS 主题后，您将收到有关客户订阅变更的通知，包括为特定客户提供或撤消访问权限。

> **Note**
>
> Amazon SNS 主题 Amazon 资源名称 (ARN) 与 `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>` 类似。

您必须响应的通知有：

- `subscribe-success` – 客户已订阅，您可以成功地根据他们的客户 ID 进行计量。
- `unsubscribe-pending` – 客户正在取消订阅。您应该发送所有最后的计量记录。
- `unsubscribe-success` – 客户已取消订阅。将不再接受客户的计量记录。执行您关闭客户资源的实操，遵守您的保留策略。
- `subscribe-fail` – 客户订阅失败。您不应根据他们的客户 ID 进行计量或代表客户创建资源。

场景：验证客户订阅

在代表客户创建资源之前，请确认客户应该有权访问您的产品。存储您通过 Amazon SQS 收到的通知中客户的最新状态，以了解客户是否有权访问。

测试您的 SaaS 订阅产品集成

将 SaaS 订阅产品与集成后 AWS Marketplace，必须进行深入测试以确保集成成功。以下过程概述了验证产品集成的步骤。

> **Note**
>
> 使用您自己的账户订阅您的产品并测试集成是否成功。可以暂时降低价格，以便您能够测试购买流程，而不会在这些账户中产生较高的费用。有关暂时降低价格或允许其他测试账户访问您的产品的更多信息，请联系 AWS Marketplace 卖家运营团队。
> 产品推出后，该服务必须继续对新客户的这些情形做出响应。

1. 使用允许的账户通过订阅产品来测试客户体验。
2. 使用允许的账户进行订阅后，请确保该账户被重定向到注册 URL，并且重定向是一个包含临时令牌的 POST 请求。请确保您的应用程序保留客户 ID，以备将来调用。这测试了场景：您的服务验证新客户中的一部分。
3. 在上一步中验证测试账户后，将该账户注册到应用程序中。例如，您可以让测试客户填写表单来创建新的用户。或者，向这些客户提供其他后续步骤以获取对 SaaS 应用程序的访问权限。这测试了场景：您的服务验证新客户中的一部分。
4. 测试客户入职后，使用中的 `BatchMeterUsage` API 操作提出请求，将计量记录发送到以 AWS 用于计费。 AWS Marketplace Metering Service这测试了场景：计量用量。
5. 测试订阅变更。可能的情况包括取消订阅、成功订阅和订阅失败。这测试了场景：监控用户订阅的更改。
6. 确认订阅成功。在收到测试账户的 Amazon SNS 通知以及成功订阅消息后，可以开始计量。AWS Marketplace Metering Service 在您收到 Amazon SNS 通知之前发送到的记录不会被计量。这测试了场景：验证客户订阅。

   > **Note**
   >
   > 为了防止出现计费问题，我们强烈建议您以编程方式等待此通知，然后代表您的客户启动资源。

7. 完成所有集成要求并测试解决方案后，请通知 AWS Marketplace 卖家运营团队。他们将通过验证您是否已使用 `BatchMeterUsage` API 操作成功发送计量数据，来对解决方案运行一系列的最终测试。

集成和测试完成后，您可以进行最终审核并将您的产品公之于众 AWS Marketplace。有关更多信息，请参阅 创建 SaaS 产品。

集成 SaaS 合同产品

将您的产品与集成 AWS Marketplace 仅一步之遥创建 SaaS 产品。要将您的软件即服务 (SaaS) 合同产品与集成 AWS Marketplace，您必须编写代码并证明它可以成功响应多种客户场景。以下各部分描述了这些场景以及如何应对这些场景，并概述了测试集成的情况。

> **Note**
>
> 在开始之前，请确保您已为 AWS Marketplace中的软件即服务 (SaaS) 产品选择正确的定价模式。有关更多信息，请参阅 规划您的 SaaS 产品。

主题

- 场景：您的服务验证新客户
- 场景：您的服务处理客户请求
- 场景：监控用户订阅的更改
- 测试您的 SaaS 合同产品集成

场景：您的服务验证新客户

当客户订阅您的产品时，他们会被重定向到您的注册 URL，这是一个带有临时 `x-amzn-marketplace-token` 令牌的 HTTP POST 请求。通过下列方式来响应此请求：

1. 通过调用 AWS Marketplace Metering Service 中的 `ResolveCustomer` API 操作将令牌交换为 `CustomerIdentifier`、`CustomerAWSAccountId` 和 `ProductCode`。
2. 通过调用 AWS Marketplace Entitlement Service中的 `GetEntitlements` API 操作来验证客户可以访问的订阅和数量（如适用）。
3. 在系统中保留 `CustomerIdentifier`、`CustomerAWSAccountId` 和 `ProductCode`，以备将来调用。存储客户是否拥有有效的订阅，以及您需要的有关该客户的任何信息。

4. 作为对请求的响应，您必须展示用户的首次使用体验（如果适用于您的服务）。

场景：您的服务处理客户请求

当客户向您的服务提出请求时，您必须通过适当的操作或消息来响应以下情形：

- 他们在系统中没有客户 ID。这表明他们尚未订阅。您应该告知用户如何订阅。
- 他们有客户 ID，且 `GetEntitlements` API 操作返回了相应的权利。在这种情况下，您应该满足请求。
- 他们确实有客户 ID，但是 `GetEntitlements` API 操作未返回任何权利，或者返回的数量不足以满足请求。在这种情况下，您必须确定如何处理访问权限和管理他们的体验。

场景：监控用户订阅的更改

设置 Amazon Simple Queue Service (Amazon SQS) 队列，并订阅您的产品的 Amazon SNS 主题。您的 SNS 主题信息包含在您创建产品时从 AWS Marketplace 运营团队收到的电子邮件中。有关更多信息，请参阅 创建 SaaS 产品。订阅 SNS 主题后，您将收到有关客户权利变更的通知，包括为特定客户提供或撤消访问权限。

> **Note**
>
> SNS 主题 Amazon 资源名称 (ARN) 与 `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>` 类似。

您必须响应的唯一通知是：

- `entitlement-updated` – 客户权利已更改，您必须调用 `GetEntitlements` API 操作才能查看新状态。更新您的客户商店，如果适用（例如，客户的合同已失效），请执行关闭客户资源的实操，遵守您的保留策略。

> **Note**
>
> 有关更多信息，请参阅 检查权利。

测试您的 SaaS 合同产品集成

将 SaaS 合同产品与集成后 AWS Marketplace，必须进行深入测试以确保集成成功。以下过程概述了验证产品集成的步骤。

> **Note**
>
> 使用您自己的账户订阅您的产品并测试集成是否成功。可以暂时降低价格，以便您能够测试购买流程，而不会在这些账户中产生较高的费用。有关暂时降低价格或允许其他测试账户访问您的产品的更多信息，请联系我们。
>
> 产品推出后，该服务必须继续对新客户的这些情形做出响应。

1. 使用允许的账户通过获取产品合同来测试客户体验。
2. 在该账户获得合同后，请确保它被重定向到注册 URL，并且重定向是一个包含临时令牌的 POST 请求。确保您的应用程序保留客户 ID 以备将来调用之用，并正确处理客户拥有的权利。这测试了场景：您的服务验证新客户中的一部分。
3. 在上一步中验证测试账户后，将该账户注册到应用程序中。例如，您可以让测试客户填写表单来创建新的用户。或者，向这些客户提供其他后续步骤以获取对 SaaS 应用程序的访问权限。这测试了场景：您的服务验证新客户中的一部分。
4. 如果 `GetEntitlements` API 操作未返回任何权利，无论是在载入期间还是验证通过期间，您的应用程序都必须正确管理未授权用户的访问权限和体验。这测试了场景：您的服务处理客户请求。
5. 测试订阅变更。验证您的应用程序是否正确处理取消订阅、成功订阅和订阅失败的情况。这测试了场景：监控用户订阅的更改。
6. 完成所有集成要求并测试解决方案后，请通知 AWS Marketplace 运营团队。然后，他们将通过验证您是否已成功调用 `GetEntitlements` API 操作并充分载入新客户来测试解决方案。

集成和测试完成后，您可以进行最终审核并将您的产品公之于众 AWS Marketplace。有关更多信息，请参阅 创建 SaaS 产品。您也可以通过填写退款申请表来取消测试订阅。有关取消订阅的更多信息，请参阅the section called “AWS Marketplace 产品退款流程”。

将你的 SaaS 合同与 pay-as-you-go 产品集成

将您的产品与集成 AWS Marketplace 仅一步之遥创建 SaaS 产品。要将您的软件即服务 (SaaS) 合同产品与集成 AWS Marketplace，您必须编写代码并证明它可以成功响应多种客户场景。以下各部分描述了这些场景以及如何应对这些场景，并概述了测试集成的情况。

> **Note**
>
> 在开始之前，请确保您已为 AWS Marketplace中的软件即服务 (SaaS) 产品选择正确的定价模式。有关更多信息，请参阅 规划您的 SaaS 产品。

主题

- 场景：您的服务验证新客户
- 场景：您的服务处理客户请求
- 场景：计量用量
- 场景：监控用户权利的变化
- 测试您的 SaaS 合同产品集成

场景：您的服务验证新客户

当客户订阅您的产品时，他们会被重定向到您的注册 URL，这是一个带有临时 `x-amzn-marketplace-token` 令牌的 HTTP POST 请求。通过下列方式来响应此请求：

1. 通过调用 AWS Marketplace Metering Service中的 `ResolveCustomer` API 操作将令牌交换为 `CustomerIdentifier`、`CustomerAWSAccountId` 和 `ProductCode`。
2. 通过调用 AWS Marketplace Entitlement Service中的 `GetEntitlements` 操作来验证客户可以访问的订阅和数量（如适用）。
3. 在系统中保留 `CustomerIdentifier`、`CustomerAWSAccountId` 和 `ProductCode`，以备将来调用。存储客户是否拥有有效的订阅，以及您需要的有关该客户的任何信息。
4. 作为对请求的响应，您必须展示用户的首次使用体验（如果适用于您的服务）。

场景：您的服务处理客户请求

当客户向您的服务提出请求时，您必须通过适当的操作或消息来响应以下情形：

- 他们在系统中没有客户 ID。这表明他们尚未订阅。您应该给他们发送消息，描述如何订阅。
- 他们有客户 ID，且 `GetEntitlements` API 操作返回了相应的权利。在这种情况下，您应该满足请求。
- 他们确实有客户 ID，但是 `GetEntitlements` API 操作未返回任何权利，或者返回的数量不足以满足请求。在这种情况下，您必须确定如何处理访问权限和管理他们的体验。

场景：计量用量

当客户开始使用您的服务时，您必须每小时发送一次计量记录。有关如何计量的详细信息，请参阅用量的计量。

我们建议您使用 AWS CloudTrail 监控活动，以确保账单信息已发送至 AWS。在发送计量记录时，请记住以下几点：

- 将按小时去除重复的计量请求。
- 每小时发送的记录是累积的。
- 即使过去一小时内没有记录，我们也强烈建议实施最佳实操，即每小时发送一次计量记录（用量为 0）。

场景：监控用户权利的变化

设置 Amazon Simple Queue Service (Amazon SQS) 队列，并订阅您产品的 Amazon SNS 主题，有两个 SNS 主题，一个用于权利变更，另一个用于订阅变更。您的主题信息包含在您创建商品时从 AWS Marketplace 卖家运营团队收到的电子邮件中。有关更多信息，请参阅 创建 SaaS 产品。订阅 SNS 主题后，您将收到有关客户订阅变更的通知，包括为特定客户提供或撤消访问权限。

> **Note**
>
> 订阅变更的 SNS 主题 Amazon 资源名称 (ARN) 与 `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>` 类似。权利变更的 SNS 主题 ARN 与 `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>` 类似。

您必须响应的通知如下：

- `entitlement-updated`（在权利 SNS 主题中）– 客户权利已更改，您必须调用 `GetEntitlements` API 操作才能查看新状态。更新您的客户商店，如果适用（例如，客户的合同已失效），请执行关闭客户资源的实操，遵守您的保留策略。
- `subscribe-success`（在订阅 SNS 主题中）– 客户已订阅，您可以成功地根据他们的客户 ID 进行计量。
- `unsubscribe-pending`（在订阅 SNS 主题中）– 客户正在取消订阅。您应该发送所有最后的计量记录。

- `unsubscribe-success`（在订阅 SNS 主题中）– 客户已取消订阅。将不再接受客户的计量记录。执行您关闭客户资源的实操，遵守您的保留策略。
- `subscribe-fail`（在订阅 SNS 主题中）– 客户订阅失败。您不应根据他们的客户 ID 进行计量或代表客户启用资源。

> **Note**
>
> 有关更多信息，请参阅 检查权利。

测试您的 SaaS 合同产品集成

将合同与 pay-as-you-go 产品整合后 AWS Marketplace，必须进行深入测试以确保集成成功。以下过程概述了验证产品集成的步骤。

> **Note**
>
> 使用您自己的账户订阅您的产品并测试集成是否成功。可以暂时降低价格，以便您能够测试购买流程，而不会在这些账户中产生较高的费用。有关暂时降低价格或允许其他测试账户访问您的产品的更多信息，请联系 AWS Marketplace 卖家运营团队。
> 产品推出后，该服务必须继续对新客户的这些情形做出响应。

1. 使用允许的账户通过获取产品合同来测试客户体验。
2. 在该账户获得合同后，请确保它被重定向到注册 URL，并且重定向是一个包含临时令牌的 POST 请求。确保您的应用程序保留客户 ID 以备将来调用之用，并正确处理客户拥有的权利。这测试了场景：您的服务验证新客户中的一部分。
3. 在上一步中验证测试账户后，将该账户注册到应用程序中。例如，您可以让测试客户填写表单来创建新的用户。或者，向这些客户提供其他后续步骤以获取对 SaaS 应用程序的访问权限。这测试了场景：您的服务验证新客户中的一部分。
4. 如果 `GetEntitlements` API 操作未返回任何权利，无论是在载入期间还是验证通过期间，您的应用程序都必须正确管理未授权用户的访问权限和体验。这测试了场景：您的服务处理客户请求。
5. 测试客户入职后，使用中的 `BatchMeterUsage` API 操作提出请求，将计量记录发送到以 AWS 用于计费。 AWS Marketplace Metering Service这测试了场景：计量用量。
6. 测试订阅变更。验证您的应用程序是否正确处理取消订阅、成功订阅和订阅失败的情况。这测试了场景：监控用户权利的变化。

7. 完成所有集成要求并测试解决方案后，请通知 AWS Marketplace 卖家运营团队。然后，他们将通过验证您是否已成功调用 `GetEntitlements` API 操作并充分载入新客户来测试解决方案。他们还将验证您是否已通过 `BatchMeterUsage` API 操作成功发送计量记录。

集成和测试完成后，您可以进行最终审核并将您的产品公之于众 AWS Marketplace。有关更多信息，请参阅 创建 SaaS 产品。

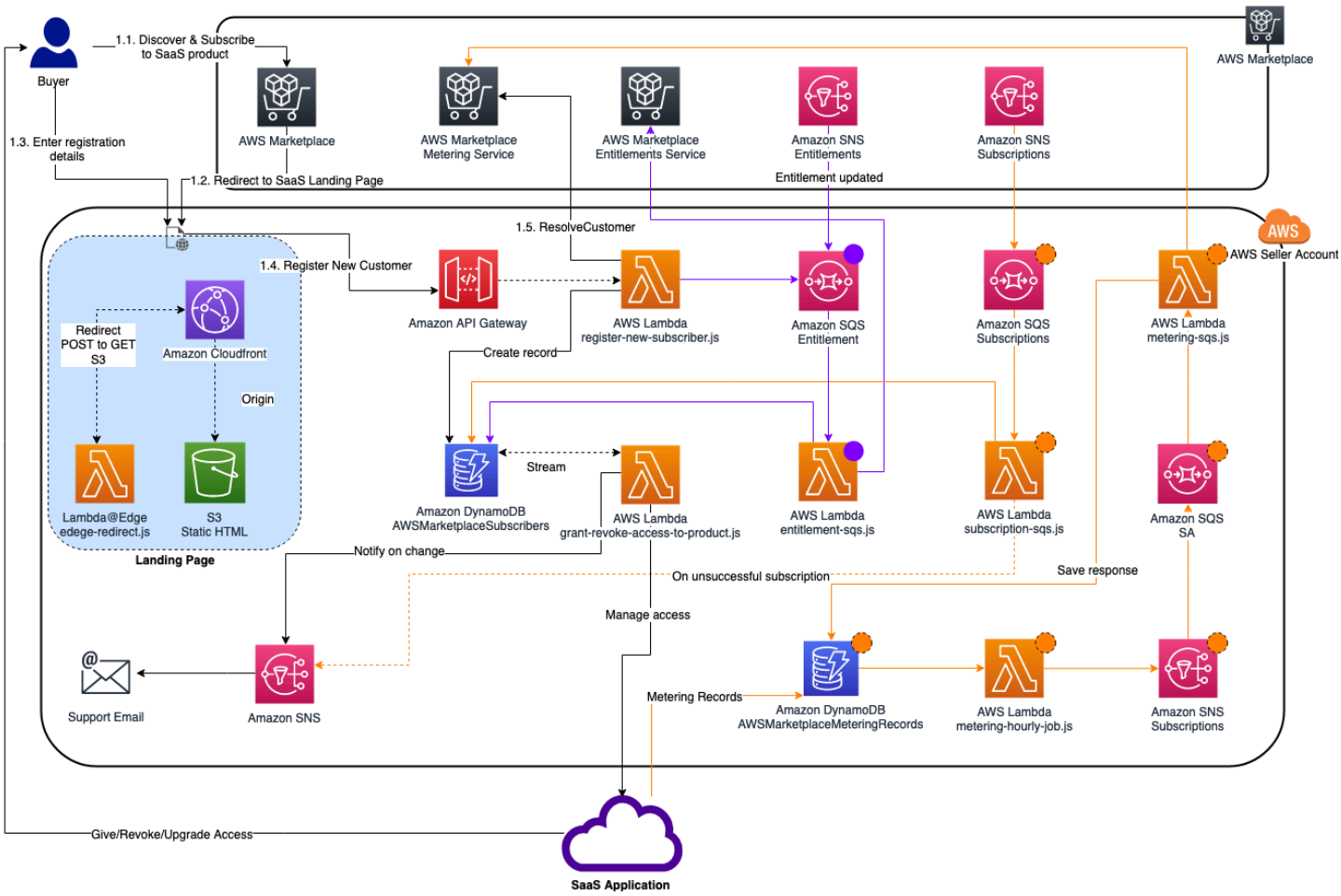
部署无服务器 SaaS 集成解决方案

AWS Marketplace 无服务器 SaaS 集成部署可满足成功集成供应商的 SaaS 解决方案及其相应列表所需的核心功能。 AWS Marketplace这些功能包括接受新客户注册、授予和撤销客户访问权限、更新客户权利以及报告计量用量。

此处精选的视频介绍了如何为 AWS Marketplace 无服务器 SaaS 集成部署 AWS 快速入门。通过这种自动参考部署，您可以轻松地将新的软件即服务 (SaaS) 解决方案集成到您的 AWS Marketplace 卖家账户，加快上市流程并显著缩短 go-to-market 时间。

AWS Marketplace 无服务器 SaaS 集成的 AWS 快速入门

下图显示了 AWS 环境中的 AWS Marketplace 无服务器 SaaS 集成如何设置以下事件工作流程。



有关如何在上部署 serverlass SaaS 集成的更多信息 AWS Cloud，请参阅 AWS Marketpl ace Serverless SaaS 集成快速入门参考部署指南。本快速入门参考指南 适用于想要轻量级无服务器解决方案来完成新 SaaS 列表所需集成的注册 AWS Marketplace 卖家。

规划您的 SaaS 产品

在向中添加软件即服务 (SaaS) 产品之前 AWS Marketplace，必须先进行一些规划。要让您的产品获得成功，这一步是至关重要的。缺少规划可能会导致计费问题，也可能导致您需要在 AWS Marketplace 中重新创建您的产品。

> **Important**
>
> 您产品的大多数设置一经配置便无法更改。如果您需要在中创建产品后对其进行更改 AWS Marketplace，则可能需要使用正确的设置创建新产品。

规划您的定价

上的 SaaS 产品有三种定价模式 AWS Marketplace。为您的产品选择正确的定价模式将是进行产品规划时要做出的最重要的决定。选择错误的定价模式可能会耽误数周时间。定价模式决定了您客户的支付选项以及您需要编写、测试和部署的账单集成代码。有关不同类型定价模式的信息，请参阅 SaaS 产品定价。

> **Note**
>
> 所有 SaaS 定价模式都支持免费试用。有关更多信息，请参阅 SaaS 免费试用。

规划您的账单集成

开启 SaaS 产品的好处之一 AWS Marketplace 是整合账单。要利用此好处，您必须与 AWS Marketplace Metering Service 或 AWS Marketplace Entitlement Service集成，具体取决于您选择的定价模式。这两项服务有助于确保您的账单和用量报告准确无误。

在规划集成后，您必须先测试与产品的集成，然后再推出产品。有关集成和测试的更多信息，请参阅访问 AWS Marketplace 计量和授权服务 APIs。

规划您的 Amazon SNS 集成

您可以为 SaaS 产品订阅两个 Amazon Simple Notification Service (Amazon SNS) 主题。有关更多信息，请参阅 SaaS 通知。这些消息可以帮助您以编程方式处理由客户发起 AWS 或由客户发起的订阅和合同的更改。Amazon SNS 通知可以是程序化触发器，使客户能够在您的产品注册网站上注册新账户。他们还可以拒绝订阅已过期的客户访问您的产品。您可以选择客户接收通知的方式，具体取决于您如何对这些通知的处理进行编程。

规划客户访问您的产品的方式

此部分介绍如何使您的产品可供买家访问。

规划 SaaS 产品注册网站

购买 SaaS 产品的客户需要具有对该产品的访问权限。您必须规划和实施希望客户访问产品的方式。SaaS 产品支持以下访问选项：

- 快速启动

- AWS PrivateLink
- 您自己的产品网站

要使用您的注册网站验证 AWS Marketplace 客户，请参阅 SaaS 客户载入。

使用“快速启动”以便客户访问您的产品

使用“快速启动”部署选项可以减少买家配置、部署和启动产品所需的时间和资源。“快速启动”可减少买家在此过程中必须访问的站点数量。有关更多信息，请参阅配置快速启动。

AWS PrivateLink 用于让客户访问您的 SaaS 产品

您可以使用 AWS PrivateLink 与一起使用 AWS Marketplace 将您的服务配置为 Amazon Virtual Private Cloud (Amazon VPC) 端点服务。您的客户可以创建一个 VPC 端点并跨 AWS Cloud 虚拟网络访问您的软件。或者，可以通过您拥有和维护的网站向在 Internet 上创建连接的客户提供对您的软件产品的访问权限。

使用您自己的网站

SaaS 产品托管于您的环境中，必须通过您管理和维护的公共端点（如网站）在 Internet 上访问它。通常，您有一个网站，客户可以使用该网站注册您的产品、登录以使用该产品以及访问对该产品的支持。

SaaS 产品准则

AWS Marketplace 维护所有 SaaS 产品和服务的这些指南， AWS Marketplace 以便为我们的客户推广一个安全、可靠和值得信赖的平台。

所有产品及其相关元数据在提交时都要经过审核，以确保它们符合或超过当前 AWS Marketplace 指南。将审核并调整这些准则，以满足我们不断变化的安全要求。此外， AWS Marketplace 持续审查商品，以验证它们是否符合这些指南的任何更改。如果产品不符合要求，我们可能会要求您更新您的产品，在某些情况下，您的产品可能暂时无法供新的订户进行订阅，直到问题得到解决。

产品设置准则

所有 SaaS 产品都必须遵循以下产品设置准则：

- 至少有一个定价维度的价格必须大于 0.00 美元。
- 所有定价维度都必须与实际软件相关，不得包括与该软件无关的任何其他产品或服务。
- 仅在这些 AWS GovCloud (US) 地区提供的 SaaS 产品必须在产品标题中包含`GovCloud`某处。

客户信息要求

所有 SaaS 产品必须遵循以下客户信息要求：

- SaaS 产品必须完全通过 AWS Marketplace上列出的维度计费。
- 您不得在任何时候收集您的 SaaS 产品的客户付款信息，包括信用卡和银行账户信息。

产品使用准则

所有 SaaS 产品都必须遵循以下产品使用准则：

- 在中订阅产品后 AWS Marketplace，客户应该能够在您的 SaaS 应用程序中创建帐户并获得对 Web 控制台的访问权限。如果客户无法立即获得应用程序的访问权限，您必须提供一条消息，其中包含有关客户何时获得访问权限的特定说明。创建账户后，必须向客户发送通知，确认其账户已创建，并提供明确的后续步骤。
- 如果客户已在 SaaS 应用程序中拥有账户，则他们必须能够从配送登录页面登录。
- 客户必须能够在 SaaS 应用程序中查看其订阅状态，包括任何相关合同或订阅使用信息。
- 客户必须能够轻松获得有关使用应用程序、故障排除和申请退款（如果适用）等问题的帮助。必须在交付登录页面上指定支持联系选项。
- 产品软件和元数据不得包含将用户重定向到 AWS Marketplace中未提供的其他云平台、其他产品、追加服务或免费试用优惠的语言。

  有关 SaaS 产品免费试用的信息，请参阅SaaS 免费试用。
- 如果您的产品是其他产品或其他ISV产品的附加产品，则您的产品描述必须表明它扩展了其他产品的功能，如果没有该产品，则您的产品的实用性非常有限。例如，本产品扩展了 <product name> 的功能，如果没有它，则该产品的应用将受到限制。请注意，<product name> 可能需要自己的许可才能使用此列表的全部功能。

架构准则

所有 SaaS 产品都必须遵循以下架构准则：

- 您的应用程序的一部分必须托管在您拥有的 AWS 账户 中。
- 所有应用程序组件都应托管在您管理的基础设施中。需要客户基础设施中额外资源的应用程序必须遵循以下准则：

- 以安全的方式配置资源，例如使用 AWS Security Token Service (AWS STS) 或 AWS Identity and Access Management (IAM)。
- 提供其他文档，包括所有配置的描述 AWS 服务、IAM政策声明，以及如何在客户账户中部署和使用IAM角色或用户。
- 在产品说明中加入一条通知，说明如果客户在 AWS Marketplace 交易之外产生了额外的 AWS 基础设施费用，则他们有责任支付额外的基础设施费用。
- 如果您的产品部署了代理，则必须向客户提供有关描述如何在其 AWS 账户中部署代理的说明。
- 需要在客户基础架构中运行资源的应用程序将接受额外的审查 AWS Marketplace，这可能需要 2-4 周的时间。

- 成功调用注册为提供商并提交 SaaS 发布请求的。 AWS Marketplace APIs AWS 账户 SaaS 定价模型决定了APIs应该调用哪个：
  - SaaS 合同 — GetEntitlements在 AWS Marketplace Entitlement Service.
  - SaaS 与消费签订合同 — GetEntitlements在 AWS Marketplace Entitlement Service 和BatchMeterUsage在 AWS Marketplace 计量服务中。
  - SaaS 订阅 — BatchMeterUsage在 AWS Marketplace 计量服务中。
- 仅在这些 AWS GovCloud (US) 地区提供的 SaaS 产品必须概述其他 AWS GovCloud (US) 地区 AWS 区域 和区域之间的架构边界、该产品的用例以及不建议用于该产品的工作负载。

SaaS 产品定价

买家购买您的软件即服务 (SaaS) 产品后 AWS Marketplace，会向您 AWS Marketplace 提供他们的账单标识符。您可以使用账单标识符来调用 AWS Marketplace Entitlement Service 和 AWS Marketplace Metering Service。然后，客户可以在您的 AWS 环境中或通过您创建的虚拟私有云 (VPC) 端点连接访问产品。

> **Note**
>
> 所有 SaaS 定价模式都支持免费试用。有关更多信息，请参阅 SaaS 免费试用。

SaaS 定价模式

定价模式	描述
SaaS 订阅	一种 pay-as-you-go 模式，我们向买家收取每小时使用您的 SaaS 产品的费用。有关更多信息，请参阅 SaaS 订阅定价。
SaaS 合同	您可以提前向买家收取软件使用费用，也可以为他们提供一个灵活的付款计划。客户还可以为其合同范围之外的额外使用付费。有关更多信息，请参阅 SaaS 合同定价。
与 SaaS 签订合同 pay-as-you-go	您可以提前向买家收取软件使用费用，也可以为他们提供一个灵活的付款计划。除合同价格外，还针对额外用量按计量费率向买家收取费用。有关更多信息，请参阅 SaaS 合同定价。

要在上线您的 SaaS 产品 AWS Marketplace，请决定是要提供 SaaS 订阅定价模式还是 SaaS 合同定价模式。

SaaS 订阅定价

对于软件即服务 (SaaS) 订阅，请根据您发送给我们的计量记录向您的客户开具 AWS Marketplace 账单。必须通过客户的账户中部署的软件每小时测量和报告一次所有费用。然后，所有使用量均按月计算，并使用与AMI基础 AWS Marketplace 产品相同的机制按月计费。我们对客户的产品用量收费的功能取决于是否从您收到计量记录。您有责任确保已成功发送并收到您的产品的计量记录。

必须先执行以下操作，然后才能发布采用订阅定价的 SaaS 产品：

1. 在中创建新的 SaaS 产品 AWS Marketplace 管理门户，选择 “新建 SaaS 订阅”。
2. 使用必要信息填写常规选项卡中的字段。记下产品代码。
3. 在定价选项卡的设置定价下，选择最准确描述产品定价的类别。定价类别会显示在 AWS Marketplace 网站上。您可以选择带宽 (GBps、MBps)、数据（GB、MB、TB）、主机（小时）、请求或用户（小时）。如果所有预定义的类别均无法满足您的需求，您可以选择更通用的单位类别。

接下来，定义您的定价维度。每个定价维度均表示一项特征或服务，可以为其设置每单位价格。维度的示例包括用户、扫描的主机和接收日志的 GB。您可以定义最多 24 个维度。对于您定义的每个维度，您必须添加以下信息：

- 维度API名称-向发送计量记录时使用的API名称AWS Marketplace Metering Service。此名称指示客户使用的维度。此名称会显示在账单报告中。由于您是唯一有权访问您的报告的人员，因此该名称无需是读者友好的名称。名称一旦设置便无法更改。
- 维度描述 - 面向客户的声明，用于描述产品的维度。描述不能超过 70 个字符，并且应便于用户理解。描述示例：管理员每小时和预置的每 Mbps 带宽。在发布产品后，您将无法更改该描述。
- D@@ imension Rat e — 本产品每FCP单位的软件费用，单位USD。该字段支持三个小数位数。

在 SaaS 订阅结束时

客户可以通过 AWS Management Console从 SaaS 订阅产品取消订阅。SaaS 订阅结束流程的要点包括以下几点：

1. 您的 SaaS 产品`unsubscribe-pending`将通过亚马逊SNS主题向该客户发送通知。
2. 您有一个小时来计量客户的任何剩余用量。
3. 之后，您将收到 `unsubscribe-success` 通知。此时，您不再能够发送此客户的计量记录。

您可以决定您希望如何在您的 SaaS 产品中为取消订阅的客户禁用功能。例如，您的产品可能要完成客户的现有工作，但要防止客户创建工作。您可能希望向客户显示一条消息，告知对方已禁止其使用。买家可以通过重新订阅您的产品。 AWS Marketplace

当 SaaS 订阅被取消时

SaaS 订阅取消流程的要点包括以下几点：

1. 客户可以在 AWS Marketplace 网站的 Your Marketplace 软件页面上取消对您的 SaaS 订阅产品的订阅。

   您的 SaaS 产品将通过亚马逊SNS主题向该客户发送通知。
2. 您有一个小时来计量客户的任何剩余用量。
3. 您从产品中通知客户正在取消。如果客户表示他们希望通过您的产品进行取消，请将客户定向到 AWS Marketplace。为了保证将来不会收取任何费用，客户应向确认取消。 AWS Marketplace

SaaS 合同定价

对于软件即服务 (SaaS) 合同，客户开始购买您的软件并与您达成协议。根据该协议，客户有权使用您的SaaS产品的指定数量。 AWS Marketplace 将这些权利传达给您的 SaaS 应用程序。此操作通过 AWS Marketplace Entitlement Service完成. 使用 SaaS 合同定价模式时，您的应用程序从不发送计量记录，相反，它通过调用. AWS Marketplace Entitlement Service来验证权利。您可以定义用量类别、维度和合同期限。

AWS Marketplace 根据您与客户之间的合同，提前或按照您定义的付款时间表向客户开具账单。之后，客户将有权使用那些资源。对于合同之外的其他用量，您的软件需要报告使用情况，并根据我们通过收到的计量记录向客户开具 AWS Marketplace 账单 AWS Marketplace Metering Service。

在发布采用合同定价的 SaaS 产品之前，您必须先执行以下操作：

1. 在中创建新的 SaaS 产品 AWS Marketplace 管理门户，然后选择新的 SaaS 合同。
2. 使用必要信息填写常规选项卡中的字段。记下产品代码。
3. 在定价选项卡上：
   a. 对于设置定价，选择您想要向客户提供的合同期限。您可以为每个合同期限输入不同的价格。您可以选择以下一个或多个选项：每月、1 年、2 年和 3 年。如果您要创建专属优惠，则可以选择以月为单位的自定义期限（最多 60 个月）。
   b. 对于选择想要提供的合同类型，从以下选项中选择您希望客户购买您的产品的方式：
      - 买家可以选择提供的一个或多个选项 – 客户可以为您提供的每个定价维度选择一个数量。
      - 买家只能从提供的多个套餐中选择一个套餐 – 客户可以从包含不同特征、服务和用量组合的选项中选择一个套餐。
   c. 选择最能准确描述您的产品定价的用量单位类别。定价类别会显示在 AWS Marketplace 网站上。您可以选择带宽 (GBps、MBps)、数据（GB、MB、TB）、主机（小时）、请求或用户（小时）。如果所有预定义的类别均无法满足您的需求，您可以选择更通用的单位类别。
4. 在选择类别后，定义您的定价维度。每个定价维度均表示一项特征或服务，可以为其设置每单位价格。维度的示例为用户、扫描的主机和接收日志的 GB。对于您定义的每个维度，您可以添加名称、描述、价格和API名称。将向客户显示名称、价格和描述。您可以使用该API名称进行跟踪和报告，AWS Marketplace 如下所示：
   - 在调用 AWS Marketplace Entitlement Service 检索客户已购买的维度时。
   - 在调用 AWS Marketplace Metering Service 指示客户使用的维度时。

   对于合同中的每个定价维度，可以选择让客户针对其合同范围之外的其他维度用量现用现付。您也可以在没有合同价格的情况下添加其他维度，而客户只需现用现付。

在使用该向导为 SaaS 产品创建合同时，您必须为定价维度定义以下字段：

- 维度API名称-调用权利时使用的名称API。此名称在账单报告中可见，并且报告不是面向外部的。API名称的最大长度为 15 个字符。名称一旦设置便无法更改。
- 维度显示名称 – 维度的面向客户的名称。此名称将帮助客户了解产品的维度。名称应是用户友好的，最大长度为 24 个字符。可以更改此值。
- 维度描述 – 维度的面向客户的描述，提供有关产品的维度的其他信息。描述的最大长度为 70 个字符。
- 维度 - 每月价格 – 针对此维度的 1 个月选项收取每单位的软件费用。该字段支持三个小数位数。
- 维度 - 1 年价格 – 针对此维度的 12 个月选项收取每单位的软件费用。该字段支持三个小数位数。这不是每月费用。此价格必须反映 12 个月的一次性收费价格。
- 维度 - 2 年价格 – 针对此维度的 24 个月选项收取每单位的软件费用。该字段支持三个小数位数。
- 维度 - 3 年价格 – 针对此维度的 36 个月选项收取每单位的软件费用。该字段支持三个小数位数。

示例：数据存储应用程序

	每月价格	12 个月价格	24 个月价格	额外使用的 P ay-as-you-go 价格
未加密的数据 (GB)	1.50 美元/GB	16.00 美元/GB	30.00 美元/GB	每小时 0.1 美元/GB
加密的数据 (GB)	1.55 美元/GB	16.60 美元/GB	31.20 美元/GB	每小时 0.11 美元/GB

示例：日志监控产品

	每月价格	12 个月价格	额外使用的 P ay-as-you-go 价格
基本（监控 10 台主机，监控 5 个容器）	100 USD	1000 美元	
标准（监控 20 台主机，监控 10 个容器）	$200	2000 美元	

	每月价格	12 个月价格	额外使用的 P ay-as-you-go 价格
专业（监控 40 台主机，监控 20 个容器）	400 美元	4000 美元	
额外主机的每小时监控成本			0.1 美元
额外容器的每小时监控成本			0.2 美元

> **Note**
>
> 价格可以采用以下期限：1 个月、12 个月、24 个月或 36 个月。可以选择为产品提供这些选项中的一个或多个选项。期限对于每个维度都必须相同。例如，假设您有 `ReadOnlyUsers` 和 `AdminUsers` 维度。如果您为 `ReadOnlyUsers` 提供按年定价，则还必须为 `AdminUsers` 提供年度价格。

SaaS 合同升级

客户可以将合同升级到更高值之一（更长期限除外）。例如，他们可以升级到更高数量或值更高的权限。将向客户提供现有合同的按比例分配的信用额。客户无法减小其现有合同的规模。他们只能在续订时减小规模或取消其续订。

权限由您的 SaaS 产品验证，该产品调用 AWS Marketplace Entitlement Service。

自动续订

当客户 AWS Marketplace 使用SaaS合同购买您的产品时，他们可以同意自动续订合同条款。客户继续按每月或按 1 年、2 年或 3 年期限支付权限费用。客户始终具有修改续订设置的选项。他们可以取消续订或按不同的数量和期限续订合同。

在 SaaS 合同终止时

SaaS 合同产品具有合同到期日期。在合同终止时，会发生以下事件：

1. 您的 SaaS 产品会收到一条 `entitlement-updated` 通知，表明买家的权利已更改。 AWS Marketplace Entitlement Service 返回空响应。
2. 您有 1 小时来计量客户的任何剩余用量。之后，您不再能够发送此客户的计量记录。

当 SaaS 合同被取消时

SaaS 合同取消流程的要点包括以下几点：

1. 客户可以通过 AWS Support请求取消和 SaaS 合同产品退款。

   客户必须在 48 小时内申请退款 AWS Support。

   全额退款或按比例退款通常在 3–5 个工作日内发放。
2. 您的 SaaS 产品将通过亚马逊SNS主题向该客户发送通知。
3. 对于任何额外的使用费，您有一小时的时间向客户发送最终计量记录。
4. 您从产品中通知客户正在取消。如果客户表示他们希望通过您的产品进行取消，请将客户定向到 AWS Marketplace。为了保证将来不会收取任何费用，客户应向确认取消。 AWS Marketplace

SaaS 免费试用

卖家可以在 (AMMP) 中创建软件即服务 (SaaS) 免费试用优惠。 AWS Marketplace 管理门户 客户可以使用 SaaS 免费试用选项在做出重大购买决策之前评估软件产品。客户订阅您的产品后，您的产品会像对付费客户一样执行权利检查。

每个 SaaS 产品 AWS 账户 只能免费试用一次。免费试用期间授予的免费使用金额不会在 AWS 组织中的关联账户之间共享。单个主付款人账户中的不同关联账户可以创建自己的个人免费试用。

> **Note**
>
> 如果您使用卖家数据传送服务 (SDDS)，您将在 Amazon Simple Storage Service (Amazon S3) 存储桶中收到协议详情试用报告。该报告包括协议详细信息，例如订阅用户名称和 ID、优惠 ID 以及协议开始和结束日期。作为卖家，您还会在创建新订阅时收到 Amazon Simple Notification Service (Amazon SNS) 通知。Amazon SNS 通知中包含一个用于标识免费试用协议的 `isFreeTrialTermPresent` 标志。

创建 SaaS 免费试用优惠

卖家可以在 AWS Marketplace 管理门户 (AMMP) 中创建 SaaS 免费试用优惠。

要创建 SaaS 免费试用优惠，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 在上 AWS Marketplace 管理门户，选择以下任一选项：
   - 创建或管理优惠
   - 优惠选项卡
3. 在优惠页面上，选择公开免费试用选项卡，查看所有 SaaS 免费试用。
4. 选择创建免费试用优惠。卖家可以为每个公开 SaaS 产品创建一个 SaaS 免费试用优惠。
5. 对于优惠基础知识，选择您的产品，然后选择下一步。
6. 在免费试用设置中：
   a. 输入免费试用期（天）。

      免费试用的期限为 7–90 天。

   b. 查看您现有公开优惠中的产品维度。

      您无法更改 SaaS 订阅免费试用的产品维度。

      您可以为 SaaS 合同免费试用更改每个维度的数量限制，并删除或添加维度。
7. 查看服务协议。

   对于 EULA 版本，您可以选择 AWS Marketplace标准合同或自定义 EULA，然后选择查看优惠。
8. 验证并查看优惠的所有信息，然后选择创建优惠。

取消 SaaS 免费试用优惠

卖家可以随时通过 AWS Marketplace 管理门户取消免费试用优惠。

要取消 SaaS 免费试用优惠，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 在上 AWS Marketplace 管理门户，选择以下任一选项：

- 创建或管理优惠
- 优惠选项卡

3. 在优惠页面上，选择优惠价。
4. 选择查看优惠。
5. 选择取消优惠。

取消优惠价后，此优惠的有效协议将一直有效，直至到期。无法为已取消的优惠价创建新协议。

SaaS 客户载入

通过软件即服务 (SaaS) 订阅和 SaaS 合同，您的客户可以通过您的环境订阅您的产品， AWS Marketplace 但在您的 AWS 环境中访问产品。订阅产品后，您的客户会被定向到您作为 SaaS 产品的一部分创建和管理的网站，以注册其账户和配置产品。

创建您的 SaaS 产品列表时，您需要提供指向注册登录页面 URL。在客户订阅后，我们使用该 URL 将客户重定向到您的注册登录页面。在软件的注册登录页面上，您可以收集为客户创建账户所需的任何信息。如果您打算通过电子邮件与客户联系以获得使用情况通知，我们建议您收集客户的电子邮件地址。

注册登录页面必须能够识别并接受表单数据中的`x-amzn-marketplace-token` AWS Marketplace 令牌以及客户的标识符进行计费。然后，它应将该代币值传递给 AWS Marketplace Metering Service ，以解析唯一的客户标识符、客户 AWS 账户 ID 和相应的产品代码。有关代码示例，请参阅 `ResolveCustomer` 代码示例。

> **Note**
>
> 注册令牌解析给特定的订阅客户，每个生成的令牌都在 4 小时后过期。只要调用者使用相同的令牌调用 API，它就会一直返回相同的响应值，直到令牌过期。

配置您的 SaaS 产品以接受新买家

您负责正确地配置 SaaS 软件以接受新客户并对他们进行适当的计量。以下过程概述了一种用于标识、实施和计量新客户对软件的访问的推荐方法：

1. 当客户在 AWS Marketplace 网站上访问您的产品页面时，他们会选择订阅您的产品。

2. 客户 AWS 账户 已订阅您的产品。这意味着从您的产品发送的订阅和计量记录将成为客户 AWS 账单的一部分。
3. 为客户生成一个注册令牌，其中包含其客户标识符和您的产品代码。
4. 客户将被重定向到软件的注册登录页面。本页面必须能够接受具有客户标识符的令牌。
5. 客户的浏览器向您的软件的注册登录页面 URL 发送 `POST` 请求。该请求包含一个 `POST` 参数 `x-amzn-marketplace-token`，其中包含客户的注册令牌。从注册网站的角度来看，客户已使用此参数提交表单。注册令牌是一个不透明的字符串。如果优惠类型为免费试用，则将在请求中添加第二个参数，即值为 `free-trial` 的 `x-amzn-marketplace-offer-type`。
6. 要使用此注册令牌兑换客户识别码、客户 AWS 账户 ID 和产品代码，您的网站必须致电 ResolveCustomerAWS Marketplace 计量服务。客户标识符不是客户 AWS 账户 ID，但它在产品之间是通用的，应作为客户记录的一部分保存到内部来源。产品代码是 AWS 提供给您的SaaS产品的唯一字符串。每种 AWS 产品都有一个唯一的产品代码，该代码是在注册时分配给您的。

   > **Note**
   >
   > 要查看 `ResolveCustomer` 调用示例，请参阅 `ResolveCustomer` 代码示例。

7. 指示客户在产品中创建账户或登录现有账户。

   > **Note**
   >
   > 如果在您的产品中设置或关联现有客户账户需要您的团队手动操作，则可以使用“联系我们”表单来收集客户的联系信息。收集他们的联系信息并解析其 AWS Marketplace 唯一客户标识符（如步骤 6 所获得的）后，向客户显示一条通知消息。在通知中，说明他们的账户正在设置中，并要求他们等待您与他们联系。向客户提供预期的周转时间和您的联系信息。还要向客户发送一封包含相同详细信息的电子邮件。

8. 客户现在已使用特定于该 SaaS 产品的凭证登录您的网站。在您的账户数据库中，对于每个客户可能都有一个条目。您的账户数据库中必须有一列 AWS 客户标识符，您可以使用在步骤 6 中获得的客户标识符填充该列。验证您的系统中没有任何其他账户共享此客户标识符。对于通过订阅多个产品的客户 AWS Marketplace，客户标识符将保持不变，每次订阅都有一个唯一的产品代码。
9. 在卖家注册过程中，您订阅 Amazon SNS 主题，用于在客户订阅或取消订阅您的产品时通知您。以下是 JSON 格式的 Amazon SNS 通知，用于告知您买家的操作：

   - 权利通知 - 对于定价模式中包含合同的产品，当买家创建新合同、升级合同、续订合同或合同到期时，您会收到通知。您的账户数据库必须有额外的一列来表示订阅状态。有关更多信息，请参阅 Amazon SNS 主题：`aws-mp-entitlement-notification`。

- 订阅通知 - 对于任何定价模式的产品，包括合同和订阅，当买家订阅或取消订阅产品时，您会收到通知。有关更多信息，请参阅 Amazon SNS 主题：`aws-mp-subscription-notification`。

建议您使用 Amazon Simple Queue Service (Amazon SQS) 来捕获这些消息。您在收到带有 `subscribe-success` 的订阅通知后，客户账户已准备就绪，可以计量了。系统不会计量您在此通知前发送的记录。有关如何执行此操作的信息，请参阅《Amazon Simple Notification Service 开发人员指南》中的步骤 2：授予 Amazon SNS 主题向 Amazon SQS 队列发送消息的权限。

> **Note**
>
> 除非您收到 `subscribe-success` 通知，否则请勿激活产品订阅。

10. 可以使用存储在数据库中的客户标识符通过 AWS Marketplace Metering Service 对用量计量或通过 AWS Marketplace Entitlement Service检查权利。

安全性和订购

作为卖家，您有责任仅信任立即返回的客户识别码 AWS 或您的系统已签名的客户识别码。我们建议您立即解析注册令牌，因为它可能会在约 1 个小时后过期。解析注册令牌后，将客户标识符作为签名的属性存储在客户的浏览器会话中，直到注册完成。

SaaS 产品的亚马逊SNS通知

要接收通知，您需要订阅商品创建期间提供给您的 AWS Marketplace 亚马逊简单通知服务 (AmazonSNS) 主题。这些主题提供有关客户产品订阅和合同权利发生更改的通知。这可以让您知道什么时候为特定客户提供和撤销访问权限。

> **Note**
>
> 在商品创建过程中，您将收到该SNS主题的实际亚马逊资源名称 (ARN)。例如：`arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`

以下亚马逊SNS主题适用于软件即服务 (SaaS) 产品：

- Amazon SNS 主题：`aws-mp-entitlement-notification` – 当买家创建新合同、升级、续订或合同到期时，本主题会通知您。这仅适用于定价模式中包含合同（也称为 SaaS 合同和带消费的 SaaS 合同（超额））的产品。
- Amazon SNS 主题：`aws-mp-subscription-notification` – 当买家订阅或取消订阅产品时，本主题会通知您，其中包括专属优惠的 `offer-identifier` 和 SaaS 免费试用的免费试用标志。这适用于所有定价模式，包括合同和订阅（也称为 SaaS 订阅、SaaS 合同和带消费的 SaaS 合同（超额））。

要详细了解您响应这些通知的场景，请参阅以下主题：

- 集成 SaaS 订阅产品
- 集成 SaaS 合同产品
- 将你的 SaaS 合同与 pay-as-you-go 产品集成

Amazon SNS 主题：**`aws-mp-entitlement-notification`**

`aws-mp-entitlement-notification` 主题中的每条消息都采用以下格式。

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
}
```

这些区域有：*`<action-name>`* 将永远如此 `entitlement-updated`。

> **Note**
>
> - 对于权利消息，无论采取何种操作（新建、升级、续订或已过期），消息都是一样的。需要随后调用 `GetEntitlement` 才能发现更新的内容。
> - 对于带消费的 SaaS 合同（超额），向卖家提供了`aws-mp-subscription-notificationSNS主题`。这是卖家在增加超额定价时会收到的额外通知。当卖家获得新客户时，卖家会收到一条订阅消息，表明这是新客户，而不仅仅是获得 `entitlement-updated`（这可能是指任何类型的操作）。

- 对于 future 日期的协议 (FDAs)，此主题在协议开始日期（而不是协议签署日期）启动。它也会在权利发生后续变更时启动，例如协议的取消、更换、续订或到期。

具有合同定价的产品（包括与之签订的合同 pay-as-you-go）必须回复这些消息。有关如何响应的更多信息，请参阅场景：监控用户订阅的更改。

Amazon SNS 主题：`aws-mp-subscription-notification`

`aws-mp-subscription-notification` 主题中的每条消息都采用以下格式。

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123",
    "isFreeTrialTermPresent":"true"
}
```

`offer-identifier` 仅当优惠为专属优惠 时，通知中才会显示。

`isFreeTrialTermPresent` 属性会指示买家的订阅是否为免费试用。此属性的JSON值不是布尔数据类型。而是将该值转换为字符串数据类型。有关更多信息，请参阅 SaaS 免费试用。

这些区域有：*`<action-name>`* 将因通知而异。可能的操作包括：

- `subscribe-success` – `subscribe-success` 消息表示卖家何时可以开始发送计量记录。如果买家接受了基于协议的报价，则此消息将与新报价一起再次发送。`offer-identifier`
- `subscribe-fail`— 如果生成`subscribe-fail`消息，即使买家已经从卖家的 SaaS 登录页面过渡到卖家的 SaaS 登录页面，付款也可能失败。 AWS Marketplace 卖家应等待 `subscribe-success` 消息后再允许消费产品。
- `unsubscribe-pending` – 当买家取消订阅时，会先发送一条 `unsubscribe-pending` 消息。这表明在买家完全取消之前，卖家只有有限的时间（大约一小时）来发送计量记录。
- `unsubscribe-success` – `unsubscribe-success` 消息表示取消已完成，之后将不再接受任何计量记录。

> **Note**
>
> - 如果买家取消订阅，然后在发送最终 `unsubscribe-success` 消息之前立即成功重新订阅，则不会发送最终 `unsubscribe-success` 消息，而是发送 `subscribe-success` 消息。
> - 对于 future 日期的协议 (FDAs)，`subscribe-success`操作将在协议开始日期（而不是协议签署日期）启动。

采用订阅定价（包括与之签订的合同 pay-as-you-go）的产品必须回复这些消息。有关如何响应的更多信息，请参阅以下主题。

- 集成 SaaS 订阅产品
- 将你的 SaaS 合同与 pay-as-you-go 产品集成

为SQS队列订阅主题 SNS

我们建议通过 Amazon SQS 队列订阅所提供的SNS主题。有关创建SQS队列和为队列订阅主题的详细说明，请参阅《亚马逊简单通知服务开发者指南》中的 “为亚马逊SQS队列订阅亚马逊SNS主题”。

> **Note**
>
> 您只能订阅 AWS 账户 用于销售产品的用户 AWS Marketplace SNS的主题。但是，您可以将消息转发到其他帐户。有关更多信息，请参阅《亚马逊简单通知服务开发者指南》中的向其他账户中的亚马逊SQS队列发送亚马逊SNS消息。

轮询SQS队列以获取通知

在您为SQS队列订阅SNS主题后，消息将存储在中SQS。您需要定义一项服务，以不断轮询该队列，查找消息并相应地处理它们。

访问 AWS Marketplace 计量和授权服务 APIs

本节概述了与 AWS Marketplace Metering 服务集成的过程 AWS Marketplace Entitlement Service，或者用于确保您的软件即服务 (SaaS) 产品的计费和客户使用情况报告的准确性。假定您提交了已发布为

受限状态的 SaaS 订阅产品或 SaaS 合同产品。在受限状态下，您可以使用测试账户来验证具有正确的配置和功能，但您的产品没有公开发布。

> **Note**
>
> 如果您的 SaaS 产品与其他以不同方式处理计量的 AWS 托管服务（例如 Amazon G SageMaker round Truth 或 AWS WAF）集成，则您无需与 AWS Marketplace 计量服务集成。您的产品只能在一个系统中进行计量，以避免向客户重复计费。

主题

- 用量的计量
- 检查权利
- SaaS 产品集成核对清单

有关设置和凭据的信息 AWS CLI，请参阅《AWS Command Line Interface 用户指南》 AWS CLI中的配置。如果你不熟悉 AWS PythonSDK，请参阅 Boto 3 快速入门。

用量的计量

对于软件即服务 (SaaS) 订阅，您可以计量所有使用情况，然后 AWS 根据您提供的计量记录向客户计费。对于 SaaS 合同，您只能计量客户合同权利中的用量。当您的应用程序计量客户的使用量时，您的应用程序会 AWS 提供累积的使用量。您的应用程序根据您在创建产品时定义的定价维度进行计量，例如传输的 GB 数据量或者给定小时内扫描的主机数。例如，如果您根据发送到应用程序的数据量收费，则可以测量数据量并每小时发送一次相应的计量记录。 AWS 使用计量数据以及您在创建产品时提供的价格来计算客户的账单。

> **Note**
>
> 或者，您可以将用量按您跟踪的属性分配。这些属性作为标签展示给买家。这些标签允许买家按标签值查看按用量划分的费用。例如，如果您按用户收费，并且用户具有 `Department` 属性，则可以使用键为 `Department` 的标签创建用量分配，每个值一个分配。这不会更改您报告的价格、维度或总用量，但允许您的客户按与您的产品相应类别查看其成本。有关更多信息，请参阅 供应商计量标记（可选）。

按小时计量

我们建议您以 AWS 每小时为单位向所有客户报告使用情况，每次最多 25 个。这使客户能够尽可能精细地了解其使用情况和成本。如果您在超过一小时的时间段（例如一天）内汇总使用量，请注意以下注意事项。

- AWS 只有在收到您的计量记录后，才能向客户开具产品使用费。您有责任确保已成功发送并收到您的产品的计量记录。您可以使用 AWS CloudTrail 来验证您发送的一条或多条记录是否准确。您还可以使用这些信息来执行一段时间的审计。有关更多信息，请参阅 使用记录 AWS Marketplace 计量 API呼叫 AWS CloudTrail。
- 如果这是定价模式为 “订阅”（不是定价模式 “合同” 或 “消费合同”）的SaaS，则买方可以随时取消订阅。当买家发起取消订阅操作时，卖家将收到`unsubscribe-pending`通知，并有 1 小时的时间将所有未报告的使用情况发送到最终通知之前。`unsubscribe-success`第二次通知之后的任何内容都将不予兑现。另外两种定价模式根据订阅时间设定了持续时间，买方无法在订阅期间取消订阅。他们只能关闭自动续订。如果不是自动续订，则在该期限结束时会发送相同的通知。
- 如果您不按小时发送计量记录，并且应用程序或网络出现故障，则您的记录将进一步落后。如果在订阅到期后恢复应用程序或网络中断，则可能会导致未报告的使用情况。
- 即使没有使用量可供报告，您也可以继续每小时发送一次计量记录，如果该小时没有使用量可供报告，则记录的数量为 0。请注意，在您报告买家使用维度（0 或以上）后，您无法修改记录。因此，最佳做法是报告前一小时的使用情况。在整个AWS账单周期中，报告的使用情况可在买家的 Billing and Billing and Cost Managem ent 控制台中查看。控制台在单独的行项目中显示产品的每个维度，包括该账单周期或迄今为止消耗的总单位数和该单项消费的总成本，即使维度价格为每单位0美元。如果启用，AWS成本和使用情况报告 (CUR) 将显示此详细信息，包括每单位费率。这些报告不是实时更新的。
- 在发布期间， AWS Marketplace 运营团队将在允许发布产品之前测试 SaaS 应用程序是否成功发送了计量记录。通常，该团队将对 SaaS 进行模拟注册，并确认已收到计量记录。

> **Note**
>
> 如果您的 SaaS 产品与其他以不同方式处理计量的 AWS 托管服务（例如 Amazon G SageMaker round Truth 或 AWS WAF）集成，则您无需与 AWS Marketplace 计量服务集成。您的产品只能在一个系统中进行计量，以避免向客户重复计费。请注意， AWS Marketplace 目前并未发布新 AWS WAF 产品。

将您的产品配置为计量用量

您可以使用 AWS Marketplace 计量服务中的`BatchMeterUsage`操作将计量记录传送到 AWS。记住以下内容：

- 我们要求卖家通过 `BatchMeterUsage` 操作来使用批处理。
- 我们按小时去除重复的计量请求。
  - 按照各产品/客户/小时/维度对请求去除重复项。
  - 您始终可以重试任何请求，但是，如果您计量了不同的数量，将按照原始数量计费。
  - 如果您针对同一个客户/维度/小时发送多个请求，则不会对记录进行汇总。
- 如果买家订阅了您的产品，则卖家可以发送包含最多 6 小时的时间戳的计量记录。如果客户取消订阅，卖家必须在客户取消订阅后的 1 小时内发送计量记录。
- `BatchMeterUsage` 有效负载不得超过 1MB。选择要在 `BatchMeterUsage` 请求中发送的用量记录数量，这样您就不会超过有效负载的大小。
- AWSMarketplace 计量服务可在AWS 一般参考中的AWS Marketplace 端点和配额中 AWS 区域 列出。在您请求产品时，默认为 SaaS 计量产品启用美国东部（弗吉尼亚州北部）区域。如果您打算使用其他区域，请联系 AWS Marketplace 卖家运营团队团队。有关更多信息，请参阅BatchMeterUsage。

有关代码示例，请参阅 SaaS 产品集成的代码示例。

示例：主机扫描

您的产品对计算硬件进行分析以查找已知的安全漏洞。客户手动启动或安排对其亚马逊弹性计算云 (AmazonEC2) 实例的扫描。当产品执行这些扫描时，它会记录每小时扫描的唯一主机数。在此示例中，您的产品使用主机类别。您可以为扫描的主机类型声明多个维度。例如，您可以针对小型、中型和大型主机按不同的价格收费。

示例：日志分析

您的 SaaS 产品会汇总由客户产品生成的日志、报告趋势和异常。当客户将日志上传到您的产品时，您可以测量收到的数据量（度量单位为兆字节、千兆字节或千吉字节）。在每个小时的第 10 分钟，cron 任务读取每位客户在上个小时的这一用量。任务生成批处理报告，并使用 `BatchMeterUsage` 操作将报告发送到 AWS。在此示例中，您的产品使用数据别。您的产品也可以按任何给定小时内存储的日志数据量进行计量。在这种情况下，您的产品可以按两个维度进行计量：此小时内收到的数据量和此小时内存储的总数据量。您可以继续计量存储的数据，直到客户删除此数据或数据到期为止。

供应商计量标记（可选）

供应商计量标签可帮助独立软件供应商 (ISVs) 让购买者更精细地了解其软件使用情况，并可以帮助他们进行成本分配。

有许多方法可以标记买家的软件使用情况。一种方法是先询问买家他们希望在成本分配中看到什么。然后，您可以在为买家帐户跟踪的属性之间分配使用情况。属性的示例包括 Account ID、Business Unit、Cost Centers，以及您的产品的其他相关元数据。这些属性作为标签展示给买家。使用标签，买家可以在 AWS 账单控制台（https://console.aws.amazon.com/billing/）中查看按标签值划分为使用量的费用。供应商计量标记不会更改您报告的价格、维度或总用量。它允许您的客户按您产品的相应类别查看成本。

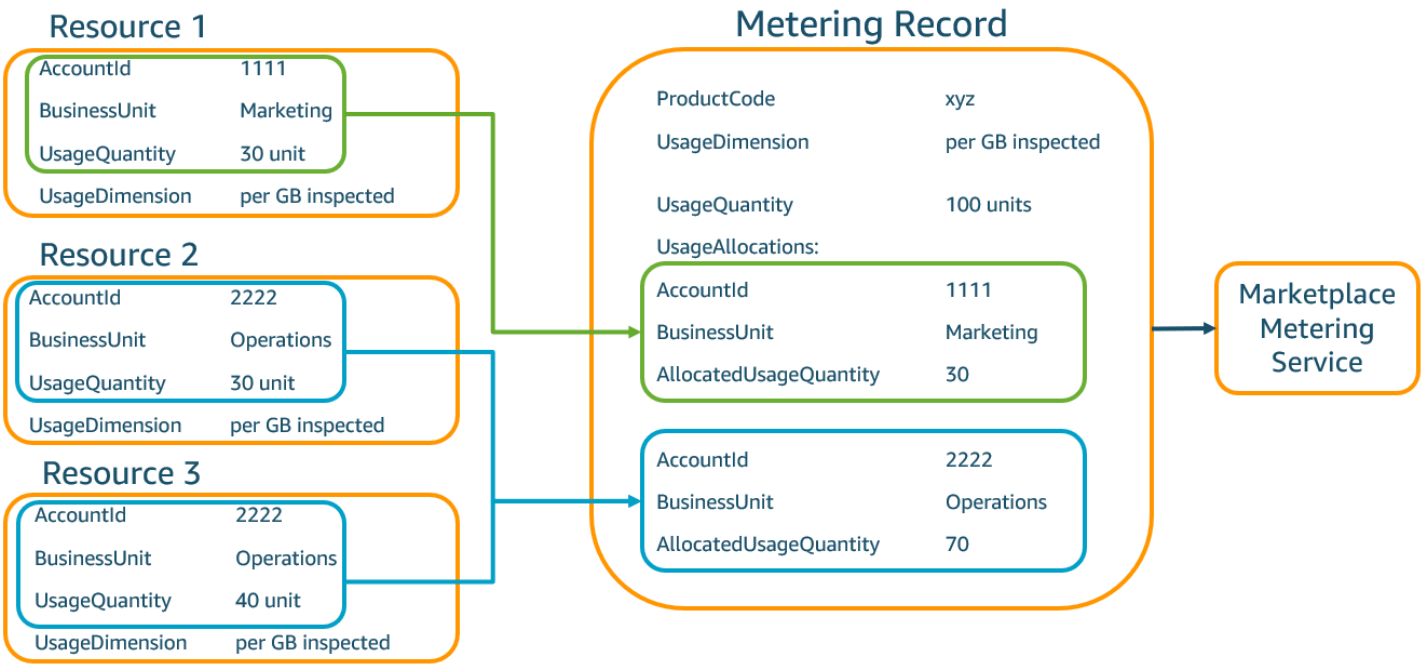
在常见情形中，买家会通过一个 AWS 账户订阅您的产品。买家还有许多与同一产品订阅相关的用户。您可以使用具有 Account ID 键的标签创建用量分配，然后将用量分配给每个用户。在这种情况下，买家可以在其账单与成本管理控制台中激活 Account ID 标签并分析个人用户的使用情况。

卖家体验

卖家可以汇总具有相同标签集的资源的计量记录，而不是汇总所有资源的使用情况。例如，卖家可以构造包含不同的 UsageAllocations 存储桶的计量记录。每个存储桶代表一组标签的 UsageQuantity，例如 AccountId 和 BusinessUnit。

在下图中，资源 1 具有一组唯一的 AccountId 和 BusinessUnit 标签，并作为单个条目出现在计量记录中。

资源 2 和资源 3 都具有相同的 AccountId 标签 2222 和相同的 BusinessUnit 标签 Operations。因此，它们在计量记录中合并为一个 UsageAllocations 条目。



卖家还可以将没有标签的资源组合成一个 `UsageAllocation`，然后将其作为 `UsageAllocations` 中的一个条目发送。

限制包括：

- 标签数 – 5
- `UsageAllocations`（基数）的大小 – 2500
- 最大请求大小 – 1 MB

验证包括：

- 标签键和值允许的字符 – a-zA-Z0-9+ -=._:\/@
- `UsageAllocation` 列表中的最大标签数 – 5
- 两个 `UsageAllocations` 的标签不能相同（也就是说，标签键和值的组合相同）。如果是这样的话，他们必须使用相同的 `UsageAllocation`。
- `UsageAllocation` 的 `AllocatedUsageQuantity` 之和必须等于 `UsageQuantity`，即聚合用量。
- 最大有效负载大小不能超过 1 MB。这包括输入属性键（例如，`UsageRecords`、`AllocatedUsageQuantity`、标签）。

> **Note**
>
> 为确保您没有违反有效负载限制，请根据业务要求创建具有最大大小的示例请求对象，将该对象转换为JSON字符串，然后获取以字节为单位的大小。确保单个API呼叫不会突破 1 MB 的限制。例如。如果一个带有 1 `UsageRecord` 的请求的最大大小为 200 KB，则作为请求的一部分发送的内容不要超过 5 `UsageRecords`（200KB * 5 = 1MB）。

买家体验

下表显示了买家激活 `AccountId` 和 `BusinessUnit` 供应商标签后的买家体验示例。

在此示例中，买家可以在其成本使用报告中看到分配的用量。供应商计量标签使用前缀 “`aws:marketplace:isv`”。在账单与成本管理中，买家可以再成本分配标签下的 AWS生成的成本分配标签中激活它们。

成本使用报告的第一行和最后一行与卖家向 Metering Service 发送的内容相关（如 卖家体验 示例所示）。

成本使用报告（简化）

ProductCode	买家	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	网络：每 (GB) 检查一次	70	2222	操作
xyz	111122223333	网络：每 (GB) 检查一次	30	3333	财务
xyz	111122223333	网络：每 (GB) 检查一次	20	4444	IT

ProductCode	买家	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	网络：每 (GB) 检查一次	20	5555	市场营销
xyz	111122223333	网络：每 (GB) 检查一次	30	1111	市场营销

有关代码示例，请参阅 带有用量分配标签的 `BatchMeterUsage` 代码示例（可选）。

检查权利

如果您的产品是 SaaS 合同产品，则您的产品会调用， AWS Marketplace Entitlement Service 以使用来检索客户的权利GetEntitlements。您的产品应根据 AWS Marketplace 授权服务验证该账户的后续使用情况。例如，如果客户在账户中预置 10 个用户，则您的产品应检查 AWS Marketplace 授权服务以获取该容量的使用权限。

要验证客户是否有权使用您的产品，请使用 AWS Marketplace 授权服务中的`GetEntitlements`操作。 AWS Marketplace 授权服务仅在美国东部（弗吉尼亚北部）地区提供，可通过以下方式访问`entitlement.marketplace.us-east-1.amazonaws.com`。

`GetEntitlements` 接受客户标识符和维度作为筛选器。`ProductCode` 是必填参数。该操作返回权限的分页列表。结果有一个 `ExpirationDate` 字段，显示该权利有效的最短时段。如果客户设置了自动续订，则 ExpirationDate 字段中的日期为续订日期。

有关代码示例，请参阅SaaS 产品集成的代码示例。

检索用户操作权利

以下示例可以帮助您更好地了解检索用户操作权利的流程。

示例：基于用户的产品

您提供的产品允许给定客户拥有一定数量的账户。客户可以访问控制面板来预置新用户（例如，分配凭证）。当客户配置新用户时，您的产品会调用 `GetEntitlements` 来验证容量是否存在。如果不是，您可以致电 AWS Marketplace 计量服务为其他用户开具账单。

示例：数据存储产品

您提供的产品让客户能够以加密或未加密格式存储一定量的数据。客户可以查看控制面板，该控制面板显示了您的产品中现有和分配的数据量。您的控制面板通过 `GetEntitlements` 检索分配数量。

SaaS 产品集成核对清单

在推出 SaaS 产品之前，请使用此核对表来验证您是否已完成所需的配置。

类别	要求
访问	提交了包含所需 AWS 账户的卖家注册表以供 AWS Marketplace 使用。
访问	完成了卖家注册，包括条款和条件、银行账户以及 W8 或 W9 税务表单。
访问	为注册账户配置跨 AWS Marketplace 账户角色。
产品	在 AWS Marketplace 管理门户中完成产品申请表。
产品	在 AMMP 中创建产品向导的 “备注” 选项卡中提供了用于测试的 AWS 账户 ID。
产品	在产品选项卡中以 .txt 格式提供 EULA 的 URL。
产品	已收到您的商品代码和 Amazon SNS 主题信息。 AWS Marketplace
产品	订阅 Amazon SNS 主题并创建了 Amazon SQS 队列以订阅 Amazon SNS 主题。

类别	要求
计费解决方案	对于 SaaS 订阅产品的每个客户，验证了您可以每小时发送计量记录到 `BatchMeterUsage` 操作。可以为 SaaS 合同产品，按照每位客户的额外用量发送计量记录。
计费解决方案	经过验证后，您可以验证 SaaS 合同产品的 AWS Marketplace 授权服务中的客户权利。
计费解决方案	验证了该成本按预期显示在为测试账户生成的账单上。
计费解决方案	测试了诸如无效客户 ID 和取消订阅等情况。
产品	已将产品请求提交回以 AWS Marketplace 供发布。
注册	实现了可接受 HTTP POST 请求的 HTTPS 注册页面。
注册	验证了您可接受新客户注册。
注册	验证了您未将注册令牌存储在 cookie 中。
注册	已验证您正在`ResolveCustomer` 使用`CustomerIdentifier` 从 AWS 令牌中获取`ProductCode` 和。
注册	经过验证后，您可以毫不延迟地解析收到 AWS 的注册令牌。
注册	测试了没有阻止如 Gmail 这样的电子邮件服务地址进行注册。
注册	测试了能接受不完整的注册和多次注册尝试。
订阅	测试了您可以处理 `unsubscribe-pending` 和 `unsubscribe-success` 消息。

类别	要求
订阅	验证了您发送在一小时内接收 `unsubscribe-pending` 消息的最终计量记录。
安全性	已验证 AWS 根账户没有 API 密钥、密码强且与硬件多因素身份验证 (MFA) 设备关联。所有管理访问权限均通过使用 AWS Identity and Access Management (IAM) 创建的身份进行。无共享账户。
安全性	验证了 IAM 角色用于所有编程式 Amazon Elastic Compute Cloud (Amazon EC2) 访问。凭证未硬编码到脚本、标头或源代码中。
安全性	验证了您维护着全面的日志记录和日志合并。
安全性	验证了您有明确定义的公有子网和私有子网边界，隔离应用程序服务，以及对数据库和文件系统的访问。区分敏感数据以及分隔公有和私有数据的不同数据类定义。
安全性	验证了您使用传输中和静态私有数据加密并按计划轮换密钥。
安全性	验证了您采用了安全事故工具和访问措施，以及定期安排的事件响应演习，以便于及时调查和恢复。
可靠性	验证了系统可适应不断变化的需求来向上扩展和向下扩展，并采用负载平衡，以确保高性能。系统还根据需要提供基于边缘的缓存。
可靠性	验证了已指定恢复时间和恢复点目标，并定期安排灾难恢复。组件故障可通过自动触发器和通知自我修复。

报告

AWS Marketplace 为您的 SaaS 产品生成报告，其中包括订阅者、财务、使用情况和税收方面的数据。有关更多信息，请参阅 the section called “卖家报告”。下表总结了报告 SaaS 产品的财务信息的方式。

报告	SaaS 内容
每日业务报告	预付合同费用显示在费用部分中。 计量使用费显示在用量部分。
月度收入报告	预付合同费用显示在年度订阅部分中。 计量使用费用显示在账单和收入数据部分中。
销售补偿报告	预付合同费用和每月额外使用费作为单独的行项目显示。
客户订阅用户报告	新的 SaaS 合同显示在 Annual subscriptions (年度订阅) 部分中。 新的 SaaS 订阅显示在 Hourly/monthly subscriptions (每小时/每月订阅) 部分中。

SaaS 产品集成的代码示例

以下代码示例可以帮助您将软件即服务 (SaaS) 产品与发布和维护产品所需的 AWS Marketplace API 集成。

主题

- ResolveCustomer 代码示例
- GetEntitlement 代码示例
- BatchMeterUsage 代码示例
- 带有用量分配标签的 BatchMeterUsage 代码示例（可选）

ResolveCustomer 代码示例

以下代码示例与所有定价模式相关。Python 示例将 x-amzn-marketplace-token 令牌交换为 CustomerIdentifier、ProductCode 和 CustomerAWSAccountId。CustomerAWSAccountId 是与订阅关联的 AWS 账户 ID。此代码在注册网站上的应用程序中运行（当您从 AWS Marketplace 管理门户重定向到该位置时）。重定向是一个包含令牌的 POST 请求。

有关更多信息ResolveCustomer，请参阅《AWS Marketplace 计量服务 API 参考》ResolveCustomer中的。

```
# 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
```

响应示例

```
{
    'CustomerIdentifier': 'string',
    'CustomerAWSAccountId':'string',
    'ProductCode': 'string'
}
```

GetEntitlement 代码示例

以下代码示例与具有合同的 SaaS 产品和具有消费定价模式的 SaaS 合同相关。Python 示例将验证客户是否具有有效权限。

有关更多信息`GetEntitlement`，请参阅《AWS Marketplace 授权服务 API 参考》GetEntitlement中的。

```
# Import AWS Python SDK
import boto3

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

# Filter entitlements for a specific customerID
#
# productCode is supplied after the AWS Marketplace Ops team has published
# the product to limited
#
# customerID is obtained from the ResolveCustomer response
entitlement = marketplaceClient.get_entitlements({
    'ProductCode': 'productCode',
    'Filter' : {
        'CUSTOMER_IDENTIFIER': [
            'customerID',
        ]
    },
    'NextToken' : 'string',
    'MaxResults': 123
})

# TODO: Verify the dimension a customer is subscribed to and the quantity,
# if applicable
```

响应示例

返回的值与您在 AWS Marketplace 管理门户中创建产品时所创建的维度对应。

```
{
   "Entitlements": [
      {
         "CustomerIdentifier": "string",
         "Dimension": "string",
         "ExpirationDate": number,
         "ProductCode": "string",
         "Value": {
            "BooleanValue": boolean,
            "DoubleValue": number,
            "IntegerValue": number,
```

```
            "StringValue": "string"
         }
      }
   ],
   "NextToken": "string"
}
```

BatchMeterUsage 代码示例

以下代码示例与具有消费定价模式的 SaaS 订阅和合同相关，但不适用于不带消费的 SaaS 合同产品。Python 示例向发送了一份计量记录 AWS Marketplace ，向您的客户收取 pay-as-you-go 费用。

```
# 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
#
# customerID is obtained from the ResolveCustomer response

# Import AWS Python SDK
import boto3

usageRecord = [
    {
        'Timestamp': datetime(2015, 1, 1),
        'CustomerIdentifier': 'customerID',
        'Dimension': 'string',
        'Quantity': 123
    }
]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(usageRecord, productCode)
```

有关更多信息`BatchMeterUsage`，请参阅《AWS Marketplace 计量服务 API 参考》BatchMeterUsage中的。

响应示例

```
{
    'Results': [
        {
            'UsageRecord': {
                'Timestamp': datetime(2015, 1, 1),
                'CustomerIdentifier': 'string',
                'Dimension': 'string',
                'Quantity': 123
            },
            'MeteringRecordId': 'string',
            'Status': 'Success' | 'CustomerNotSubscribed' | 'DuplicateRecord'
        },
    ],
    'UnprocessedRecords': [
        {
            'Timestamp': datetime(2015, 1, 1),
            'CustomerIdentifier': 'string',
            'Dimension': 'string',
            'Quantity': 123
        }
    ]
}
```

带有用量分配标签的 **BatchMeterUsage** 代码示例（可选）

以下代码示例与具有消费定价模式的 SaaS 订阅和合同相关，但不适用于不带消费的 SaaS 合同产品。Python 示例将带有相应使用量分配标签的计量记录发送给您的客户 AWS Marketplace ，以向您的客户收取 pay-as-you-go 费用。

```
# 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
#
# customerID is obtained from the ResolveCustomer response

# Import AWS Python SDK
import boto3
```

```
import time

usageRecords = [
    {
        "Timestamp": int(time.time()),
        "CustomerIdentifier": "customerID",
        "Dimension": "Dimension1",
        "Quantity":3,
        "UsageAllocations": [
            {
                "AllocatedUsageQuantity": 2,
                "Tags":
                    [
                        { "Key": "BusinessUnit", "Value": "IT" },
                        { "Key": "AccountId", "Value": "123456789" },
                    ]

            },
            {
                "AllocatedUsageQuantity": 1,
                "Tags":
                    [
                        { "Key": "BusinessUnit", "Value": "Finance" },
                        { "Key": "AccountId", "Value": "987654321" },
                    ]

            },
         ]
     }
]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(UsageRecords=usageRecords,
 ProductCode="testProduct")
```

有关的更多信息`BatchMeterUsage`，请参阅 AWS Marketplace Metering Service API 参考BatchMeterUsage中的。

响应示例

```
{
    "Results": [
```

```
        {
            "Timestamp": "1634691015",
            "CustomerIdentifier": "customerID",
            "Dimension": "Dimension1",
            "Quantity":3,
            "UsageAllocations": [
            {
                "AllocatedUsageQuantity": 2,
                "Tags":
                    [
                        { "Key": "BusinessUnit", "Value": "IT" },
                        { "Key": "AccountId", "Value": "123456789" },
                    ]

            },
            {
                "AllocatedUsageQuantity": 1,
                "Tags":
                    [
                        { "Key": "BusinessUnit", "Value": "Finance" },
                        { "Key": "AccountId", "Value": "987654321" },
                    ]

            },
         ]
            },
            "MeteringRecordId": "8fjef98ejf",
            "Status": "Success"
        },
    ],
    "UnprocessedRecords": [
        {
            "Timestamp": "1634691015",
            "CustomerIdentifier": "customerID",
            "Dimension": "Dimension1",
            "Quantity":3,
            "UsageAllocations": []
        }
    ]
}
```

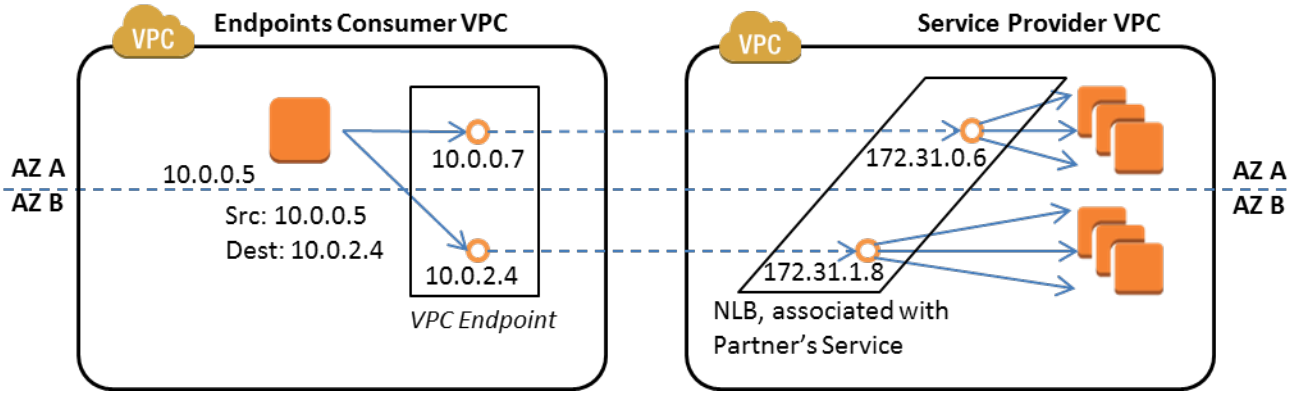
AWS PrivateLink 与一起使用 AWS Marketplace

AWS Marketplace 支持 AWS PrivateLink，这项技术允许您使用亚马逊网络为买家提供您销售的商品的访问权限 AWS Marketplace。本文档概述了使用 AWS PrivateLink 技术通过亚马逊 Virtual Private Cloud (VPC) 端点配置和交付产品的流程。

在本文档中，我们假设您具备多项 AWS 服务和 AWS Marketplace 环境的工作知识。

简介

作为 AWS Marketplace 卖家，您可以通过亚马逊VPC终端节点为买家提供访问您的服务的权限。此方法使用 AWS PrivateLink 技术，为买家提供在 Amazon 网络上访问您的服务的权限。如果您使用 AWS Marketplace 创建和交付此产品，则买家可以在中发现您的服务 AWS Marketplace。您的买家还可以在创建VPC终端节点的可用服务列表中找到您的产品。



VPC端点是一种虚拟设备，它使 AWS 客户能够在其VPC与其他 AWS 服务之间创建私有连接，而无需通过互联网、NAT设备、VPN连接或进行访问 AWS Direct Connect。您可以通过创建终端节点服务 AWS Marketplace ，让买家能够使用该技术连接到您的服务。这种连接方法对于买家来说更安全，因为买家是通过 Amazon 专用网络而不是 Inernet 访问您的服务。

对于您想要提供服务的每个区域，您可以创建或使用现有资源来配置、设置服务实例VPC、设置网络负载均衡器，并通过创建服务终端节点向网络负载均衡器注册服务。在完成这些步骤并测试产品之后，您可以将配置信息提供给 AWS Marketplace 卖家运营团队。

AWS 建议您提供一个私有DNS名称，供买家在创建VPC终端节点时使用。

当买家创建VPC终端节点时，他们可以选择启用私有DNS名称。通过选择此选项，购买者的VPC服务将配置私有托管区域。如果您提供私有DNS名称，则买家可以在配置VPC终端节点以连接到您的服务时使用该名称。在买家的私有托管区域中，私有DNS名称 (api.example.com) 将指向为你的终端节点服

务创建的随机生成的DNS名称（vpce-111111111111111-yyyyyyy.api.vpce.example.com）。买方的EC2实例在不同的名称上使用相同的统一DNS名称 (api.example.com)。VPCs此外，如果公用DNS名称和私有名称相同，则买方在从内部或外部访问您的服务时可以使用相同的公共名称VPC。

如需通过提供服务方面的帮助 AWS Marketplace，您可以联系AWS Marketplace 卖家运营团队。当 AWS Marketplace 买家订阅您的服务并创建VPC终端节点时，您的服务将显示在您的AWSMarketplace 服务下。 AWS Marketplace 卖家运营团队使用用户友好的DNS名称，以便在创建VPC端点时便于发现您的服务。

您的产品作为软件即服务 (SaaS) 产品创建。计量和计费与其他 AWS Marketplace SaaS产品相同。

配置产品

要将您的产品配置为可通过 Amazon VPC 终端节点上市，请执行以下操作：

1. 创建或使用现有的亚马逊VPC。
2. 为您的产品创建（或使用现有）Amazon EC2 实例。
3. 在您提供产品的每个地区创建网络负载均衡器。 AWS 建议您包括某个地区的所有可用区 (AZs)。
4. 使用 Amazon VPC 控制台CLI、或支持SDKs创建VPC终端节点服务。
5. 验证是否可以通过网络负载均衡器访问该服务。
6. 从 AWS Certificate Manager (ACM) 为您的用户友好型DNS名称申请证书。在ACM颁发证书之前，它会验证您是否拥有或控制证书请求中的域名。
7. 将您的用户友好DNS名称的子域名（例如 api.vpce.example.com）委托给卖家运营团队提供给您的域名服务器。 AWS Marketplace 在您的DNS系统中，您必须创建域名服务器 (NS) 资源记录，将此子域名指向 AWS Marketplace 卖家运营团队提供的亚马逊 Route 53 域名服务器，以便可以公开解析DNS名称（例如 vpce-0ac6c347a78c90f8.api.vpce.example.com）。
8. 允许访问您的买家 AWS 账户。

   注意：你可以使用支持的命令SDK或这个CLI命令来自动访问账户：aws vpcev2-permissions--service-id vpc modify-vpc-endpoint-service-svc-0123456789abcdef1 — arn: aws: iam:: 111111111111:root arn: aws: aws: aws: aws: aws: aws: aws: aws: aws: aws: aws: aws: aws: aws: aws: add-allowed-principals

将您的产品提交至 AWS Marketplace

在向发布服务的过程中 AWS Marketplace，您需要与 AWS Marketplace 卖家运营团队合作。要提交PrivateLink已启用的产品，请执行以下操作：

1. 通过电子邮件将以下信息发送给AWS Marketplace 卖家运营团队：
   a. 终端节点和用于创建终端节点的 AWS 账户。端点类似于：com.amazonaws.vpce.us-east-1.vpce-svc-0daa010345a21646
   b. 您的服务的用户友好DNS名称。这是 AWS Marketplace 买家用来访问您的商品的DNS名称。
   c. 您用于申请证书的 AWS 账户以及购买者用于访问VPC终端节点的私有DNS名称。

      AWS Marketplace 卖家运营团队会验证贵公司的身份以及用于您注册的服务（例如 api.vpce.example.com）的DNS名称。经过验证后，该DNS名称将覆盖默认的基本端点DNS名称。

买家访问VPC终端节点

AWS Marketplace 在以下情况下，正在创建VPC终端节点的买家可以发现您的服务：

- 您按照此页面前面描述的卖家流程创建了产品或使用现有产品。
- 买家订阅了您的服务。
- 您已将买家的 AWS 账户添加到允许的账户列表中。

当购买者创建VPC终端节点时，他们可以选择将私有托管区域与其关联起来VPC。托管区域包含服务的默认私有DNS名称记录集，该记录集解析为其VPC中的端点网络接口的私有 IP 地址。

任何买家托管的终端节点（包括 AWS Marketplace 服务）都可以为所有账户提供权限（“*” 权限）。但是，您在使用此方法时，除非按服务名称进行搜索，否则服务不会包含在 Describe 调用或控制台中。要在 Desc ribe 调用中显示服务，服务必须将买家的 AWS 账户明确添加到允许列表中。

要访问您的服务，买家需要执行以下操作：

1. 在上发现并订阅您的服务 AWS Marketplace.
2. 使用 AWS Command Line Interface (AWS CLI) API、或 Amazon VPC 控制台发现您的服务，然后在子网中建立一个VPC终端节点以连接到您的服务，然后AZs他们使用。端点在子网中显示为弹性网络接口。本地 IP 地址以及区域和区域DNS名称已分配给端点。

客户端名称 DNS	名称
区域性	Vpce<0dc9a211a78c90f8>.api.vpce.example.com

客户端名称 DNS	名称
IAD2(1a)	us-east-1a-Vpce<0dc9a211a78c90f8>.api.vpce.example.com
IAD2(1b)	us-east-1b-Vpce<0dc9a211a78c90f8>.api.vpce.example.com

如果您提供了默认私有DNS名称，并且买家在创建VPC终端节点时选择了启用私有DNS名称（与私有托管区域相关联），则买方会看到用于连接到您的服务的区域默认私有DNS名称。

名称	别名	托管区域别名 ID	（备注）
api.example.com	vpce<0dc9a211a78c90f8>. api.vpce.example.com	Z00 AABBCCDD	IAD1 IAD2

附录：核对清单

在将商品提交给 AWS Marketplace 卖家运营团队之前，请使用以下清单确保对商品进行配置和测试。

产品创建核对清单

- 创建（或使用现有的）VPC，然后对其进行配置。
- 在中创建和配置网络负载均衡器VPC。
- 通过创建VPC终端节点服务，向网络负载均衡器注册您的服务。
- 向 AWS Marketplace 卖家运营团队提供您用于配置VPC终端节点的 AWS 账户 ID。
- 向卖家运营团队提供默认终端节点服务名称（例如 com.amazonaws.vpce.us-east-1.vpce-svc-0bb070044a2164）。 AWS Marketplace
- 提供用户友好的服务DNS名称（必填）以覆盖随机生成的服务DNS名称。从中ACM申请用于用户友好型服务DNS名称的子域的SSL证书。向 AWS Marketplace 卖家运营团队提供这些证书和您申请证书时使用的 AWS 账户编号。
- 推荐：提供私人DNS名称。
- 创建一个流程，告知 AWS Marketplace 买家并允许他们选择使用 AWS PrivateLink 技术连接到您的服务。将买IDs家 AWS 账户添加到允许的账户列表中。

产品测试

- 验证您的服务是否已配置并且可以找到。
- 验证您的服务是否可以通过网络负载均衡器找到。
- 确认买家可以创建VPC终端节点并访问您的服务。使用您拥有的 AWS 帐户，该帐户不是您用于设置服务的帐户。

专业服务产品

作为卖家，您可以为 AWS Marketplace 买家提供专业服务。专业服务包括评估、迁移、支持、管理和培训他人如何使用 AWS Marketplace中的 AWS 服务和产品的服务。卖家创建描述他们所提供服务的产品报价，与买家协商以达成条款协议，然后通过 AWS Marketplace创建自定义服务优惠。

> **Note**
>
> 作为独立软件供应商 (ISV)、渠道合作伙伴或咨询合作伙伴，您也可以授权其他合作伙伴使用渠道合作伙伴私有优惠转售您的专业服务产品。有关更多信息，请参阅为渠道合作伙伴创造转售机会。ISV

买家可以在 AWS Marketplace 目录中找到专业服务产品，方法是在 “类别” 下选择 “专业服务”，在 “配送方式” 下选择 “专业服务”，然后按出版商、定价模式和定价单位细化搜索。他们需要为 AWS 账单中的服务付费。他们可以使用诸如 AWS Cost Explorer 集中付款和管理成本之类的工具。

有关专业服务产品的更多信息，请参阅：

- 专业服务产品入门
- 提供专业服务产品的详细信息
- 对专业服务产品的要求
- 为渠道合作伙伴创造转售机会 ISV

以下视频详细介绍了如何在 AWS Marketplace中管理专业服务产品。

在 AWS Marketplace 中管理专业服务产品。

获取帮助

如需专业服务产品方面的帮助，请联系您的业务发展合作伙伴 AWS Marketplace 或AWS Marketplace卖家运营团队。

开始使用专业服务产品

本主题介绍如何开始使用专业服务产品，并详述了创建您的第一个产品的步骤以及如何向客户提供该产品。您的产品定义告诉您的客户您提供的服务，以及他们为何应选择您的公司提供这些服务。 AWS

Marketplace 然后允许他们与您联系。您同意合同，然后创建专属优惠，允许他们以固定成本购买您的服务。

主题

- 先决条件
- 创建专业服务产品
- 创建专属优惠
- 编辑产品信息
- 编辑产品定价
- 编辑产品可见性
- 删除专业服务产品

以下视频详细介绍了如何在 AWS Marketplace中发布专业服务产品。

先决条件

要销售专业服务 AWS Marketplace，您必须满足以下先决条件：

- 拥有 AWS Marketplace 管理门户的访问权限。这是您用来注册为卖家并管理您销售的商品的工具 AWS Marketplace。要了解有关获取访问权限的更多信息 AWS Marketplace 管理门户，请参阅AWS Marketplace 卖家策略和权限。
- 注册为 AWS Marketplace 卖家并提交您的税务和银行信息。要了解有关成为卖家的更多信息，请参阅以卖方身份开始使用。
- 您必须提供与服务相关的专业 AWS 服务产品或至少一种公共产品 AWS Marketplace。您的产品必须直接支持这些产品，或者提供推动订阅这些产品的服务。

> **Note**
>
> 您的产品必须至少列在以下任一主要类别下：评估、实施、托管服务、高级支持或培训。
> 有关专业服务产品指南的更多信息，请参阅对专业服务产品的要求 。

创建专业服务产品

以下过程介绍了如何在 AWS Marketplace 管理门户中创建新的专业服务产品。

要创建专业服务产品，请执行以下操作：

1. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
2. 从产品菜单中，选择专业服务。此页面显示您已经创建的所有专业服务产品，以及您为创建或修改这些产品而提出的任何请求。
3. 在专业服务产品选项卡上，选择创建专业服务产品。
4. 在创建产品页面上，提供您的产品信息，然后选择提交。有关您必须提供的详细信息的更多信息，请参阅提供专业服务产品的详细信息。
5. （可选）从的 “产品” 菜单中 AWS Marketplace 管理门户，选择 “专业服务”，然后选择 “请求” 选项卡。确认您看到的产品请求包含正确的产品名称，且请求状态是否为正在审核。您的产品应在几分钟内以受限的预览模式创建。

> **Note**
>
> 您可以随时返回专业服务页面的请求选项卡，查看您的请求状态。创建过程中的任何错误都将出现在此处。您可以选择请求以查看请求详细信息或修复错误。

当您的产品最初创建时，只有您的 AWS 账户 （您用来创建产品的账户）才能访问该产品。如果您从 “专业服务” 页面查看商品，则可以选择 “查看开启”，查看 AWS Marketplace 面向买家的商品详情。AWS Marketplace除非您向其他 AWS Marketplace 用户提供私密优惠，否则此详情列表不向他们提供。

要了解如何公开该产品，请参阅编辑产品可见性。

创建专属优惠

当潜在买家查看您的产品时 AWS Marketplace，他们无法直接购买。当他们尝试订阅时，他们会被重定向到请求您的私人报价。 AWS Marketplace 向您的 AWS Marketplace 卖家账户根用户电子邮件地址发送一封电子邮件，告知您买家已申请私下报价。以下过程介绍如何响应此请求。

> **Note**
>
> 当您通过创建高于 250,000 美元的私募报价时 AWS Marketplace 管理门户，可能需要额外的批准。欲了解更多信息，请通过联系私募优惠成功团队 (POST)、供应商融资成功团队 (VFS) 或卖家报告团队AWS Marketplace 管理门户。

要为专业服务产品创建专属优惠，请执行以下操作：

1. 联系客户以解决您对请求的任何疑问。在 AWS Marketplace中创建专属优惠之前，请先就优惠条款达成一致。买家并非必须购买您的产品，因此需要在创建优惠之前达成同意。
2. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
3. 从菜单中选择优惠，然后选择创建专属优惠。
4. 在创建专属优惠页面上，选择要为其创建专属优惠的产品。仅包括当前可用的产品。
5. 输入您想要向其提供专属优惠的买家账户 ID。您可以在单个专属优惠中包含最多 25 个帐户。如果买家使用了请求优惠特征，则您收到的电子邮件中将包含请求账户的买家账户 ID。
6. 选择是否允许买家分期支付产品费用。通常，空头合约一次性支付。较长的合同可以选择分期付款，但这是您与买家达成的协议的一部分。选择下一步。
7. 填写优惠详情，包括以下信息：
   - 自定义优惠名称 - 提供足够的详细信息，以便您和买家都能认出该优惠。包括您的公司或产品名称以及产品描述。请勿包含任何个人身份信息，包括姓名、电话号码或地址。
   - 协议结束日期 - 协议服务终止的日期。例如，如果您提供的支持期为 1 年，请输入距离该服务推出日期还有 1 年的日期。
   - 产品维度 - 您提供的服务的价格和单位，如下所示：
     - 一次性付款优惠 - 您可以列出每个维度及其相关价格（例如，您可以提供名为白银、黄金和白金的维度）。买家可以根据自己的偏好进行选择和付款。
     - 包含付款计划的优惠 - 您必须选择一个维度，并提供包含每笔付款的金额和日期的付款计划。

     > **Note**
     > 如果您想创建零美元优惠，则必须选择我要启用零美元价格进行确认。此预防措施有助于防止您意外创建免费优惠。

   - 服务协议 - 定义您与客户签订的服务协议的文档。您上传的文档（文本或PDF格式）将一起附加到一个PDF文档中，因此请确保不需要文件名即可理解内容。
   - 优惠到期日期 - 优惠的到期日期。这决定了买家必须接受优惠的时间，与何时提供专业服务无关。您可以在优惠创建后延长优惠到期日期。
8. 编辑完选项后，选择下一步。
9. 在查看优惠页面上，确保优惠详情正确无误，然后选择创建优惠。

> **Note**
>
> 您的优惠可能需要一些时间才能发布。优惠发布后，您可以在管理优惠页面上查看优惠。如果您需要编辑优惠（尚未被接受），可以从该页面进行编辑。

10. 报价发布后，在 “管理私密报价” 页面上可用，从该报价的 “操作” 菜单中选择 “复制报价”URL，然后通过电子邮件将其发送给买家进行接受。

编辑产品信息

以下过程介绍如何编辑 AWS Marketplace 管理门户中现有专业服务产品的产品信息。

要编辑产品信息，请执行以下操作：

1. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
2. 从产品菜单中，选择专业服务。此页面显示您已经创建的所有专业服务产品，以及尚未解决的创建或修改这些产品的任何请求。
3. 请选择您想要编辑的现有产品。然后，从请求更改菜单中，选择更新产品信息。
4. 对详细信息进行更改。有关可编辑字段的更多信息，请参阅提供专业服务产品的详细信息。
5. 选择提交以创建请求。
6. （可选）如果您尚未进入专业服务页面的 “请求” 选项卡，请从 “产品” 菜单中选择 “专业服务”，然后选择 “请求” 选项卡。 AWS Marketplace 管理门户确认您看到的请求包含正确的产品名称，且请求状态是否为正在审核。您的产品将在几分钟内根据您要求的更改进行更新。如果出现错误，您可以在此处查看，并在修复错误后重新提交编辑内容。

编辑产品定价

以下过程介绍如何编辑 AWS Marketplace 管理门户中现有专业服务产品的定价信息。

要编辑产品定价，请执行以下操作：

1. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
2. 从产品菜单中，选择专业服务。此页面显示您已经创建的所有专业服务产品，以及您为创建或修改这些产品而提出的任何请求。
3. 选择您要编辑的现有产品，然后从请求更改菜单中选择更新定价维度。

> **Note**
>
> 您只能通过添加新的定价维度 AWS Marketplace 管理门户。要修改或删除之前创建的维度，请联系 AWS Marketplace 卖家运营团队提出您的请求。在您的请求中，请包括产品 ID 以及您想要更改或删除的维度的详细信息。

4. 根据需要添加任意新定价维度。有关定价字段的更多信息，请参阅提供专业服务产品的详细信息。
5. 选择提交以创建请求。
6. （可选）从的 “产品” 菜单中 AWS Marketplace 管理门户，选择 “专业服务”，然后选择 “请求” 选项卡。确认您看到的请求包含正确的产品名称，且请求状态是否为正在审核。您的产品将在几分钟内根据您要求的更改进行更新。如果出现错误，您可以在此处查看，并在修复错误后重新提交编辑内容。

编辑产品可见性

默认情况下，产品是在受限可见性下创建的，只有在您的账户中才能看到新产品。您可以添加其他测试账户，或者在 AWS Marketplace中公开显示该产品。以下过程介绍如何编辑 AWS Marketplace 管理门户中现有专业服务产品的可见性。

要编辑产品可见性，请执行以下操作：

1. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
2. 从产品菜单中，选择专业服务。此页面显示您已经创建的所有专业服务产品。
3. 选择您想要编辑或更新曝光度的现有产品。然后，从请求更改菜单中，选择更新产品可见性。
4. 选择一个选项来更新您的商品知名度，然后选择 “提交” 以提交您的审核请求。
5. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。如果状态为 “失败”，请选择请求名称以查看 “发现的问题”。

> **Note**
>
> 要使商品在公共 AWS Marketplace 目录中可见，需要 AWS Marketplace 卖家运营团队进行商品审查，以确保该商品符合商品指南（请参阅对专业服务产品的要求 ）。请求可能需要几天的时间才能完成。

删除专业服务产品

以下过程介绍了如何在 AWS Marketplace 管理门户中删除现有专业服务产品。

> **Note**
>
> 移除专业服务产品不会影响有效的私人优惠。

要删除产品，请执行以下操作：

1. 打开 Web 浏览器并登录 AWS Marketplace 管理门户。
2. 从产品菜单中，选择专业服务。此页面显示您已经创建的所有专业服务产品。
3. 选择您想要移除的现有商品。然后，从请求更改菜单中，选择更新产品可见性。
4. 选择 “受限” 作为可见性选项以更新您的商品曝光度，然后选择 “提交”。
5. 确认请求选项卡将请求状态显示为正在审核。请求完成后，状态变为成功。

> **Note**
>
> 请求可能需要几天的时间才能完成。具有有效报价的产品将移至受限状态，直到最后一个有效的订阅或合同完成并从中删除 AWS Marketplace。受限状态意味着现有用户可以继续使用该产品。但是，该产品将不再对公众可见，也不会再向新用户提供。

提供专业服务产品的详细信息

在上发布专业服务产品时 AWS Marketplace，必须提供产品元数据。本主题讨论了在您准备或编辑产品详情时有用的信息。

> **Note**
>
> 有关专业服务产品的准则和要求的信息，请参阅对专业服务产品的要求 。

主题

- 产品描述

- 其他 资源
- 支持信息
- 定价维度
- 产品可见性

产品描述

产品详细信息中的产品描述部分是您产品的核心。它向潜在买家描述您的产品，以便他们做出购买决定。产品详细信息的这一部分包括以下数据：

- 产品标题 - 您的产品的名称。它用于识别您的产品；它显示在产品页面和搜索结果中。为产品提供一个有意义的名称。它在内部必须是唯一的 AWS Marketplace。
- SKU —（可选）用于追踪您的商品 AWS Marketplace。此信息仅供您自己使用；买家看不到。
- 简@@ 短描述 — 对您的产品的简要描述，显示在产品目录的图块上和产品标题下方。 AWS Marketplace
- 详细描述 - 一种较长的格式化描述，用于向买家描述您的产品详情。列出产品特征、优点、用法以及与该产品相关的其他信息。使用可用的格式使信息更易于理解和扫描。
- 产品徽标 - 此字段是一个公开 S3 URL，指向代表您的产品的图片文件。文件必须为.png、.jpg 或.gif 格式，背景为透明或白色，小于 5MB，宽高在 110-10000 像素之间。徽标在产品提交期间上传并存储在 AWS Marketplace。修改网址的内容不会在提交 AWS Marketplace 后修改徽标。

> **Note**
>
> 您提供的 S3 URL 必须是公开的。这是文件所在的 S3 存储桶的属性。有关更多信息，请参阅《Amazon Simple Storage Service 控制台用户指南》中的如何编辑除 S3 存储桶的公开访问设置？。

- 要点 - 一组关于您的产品的一到三个简短要点，描述其主要特征或差异化优势。至少需要一个要点。
- 产品类别 - 您提供的服务类型。您必须至少选择一个，最多三个类别。有许多类别可供选择，但专业服务产品必须至少包括以下类别之一：

  评测

  评估客户当前的运营环境，为其组织找到合适的解决方案。

  实施

  帮助配置、设置和部署第三方软件。

高级支持

获得专为满足客户需求而设计的专家指导和帮助。

托管服务

代表客户进行nd-to-end 环境管理。

训练

专家提供的量身定制的研讨会、计划和教育工具，可帮助客户员工学习最佳实操。

- 搜索结果关键词 - 最多提供三个关键词，供买家搜索您的产品。您可以以逗号分隔的列表形式列出关键词，最多 250 个字符。
- 关联产品 — 可选 — 至少包括您的服务与之 AWS Marketplace 配合使用或支持的公共产品。 AWS Marketplace 在选择要在产品详情页面或这些商品的相关产品中显示的产品时，使用这些产品作为输入。

其他 资源

在产品详细信息的其他资源部分，您可以提供指向您为帮助客户而创建的资源的链接。这是一组可选的，由一到三个在线存储的可下载资源组成。资源示例包括产品信息表、白皮书或产品手册。为每个资源提供一个名称和一个 URL。

支持信息

本部分是一个格式化的文本字段，允许您描述您为服务提供的支持。

客户希望在使用服务、故障排除和申请退款（如适用）等问题上获得支持。支持描述应包含关于客户可以期待的支持级别的声明。考虑包括购买前问题和购买后问题的支持详情。

定价维度

专业服务的定价范围是您提供的套餐。例如，您可以提供白银、黄金和白金支持。或者您可以提供 10、20 或 50 小时的咨询服务。对于您要提供的每个维度（至少一个，最多 24 个），请指定名称和描述。当您通过直接与买家合作为产品创建专属优惠时，您可以为这些维度设定实际价格。

> **Note**
>
> 有关如何使用定价维度以及如何设置价格的信息，请参阅创建专属优惠。

产品可见性

已发布的产品只能显示在 AWS Marketplace 您自己的账户、一小部分测试账户或所有 AWS 账户中。默认情况下，该产品以专属形式发布。要更改产品的可见性，请参阅编辑产品可见性。

对专业服务产品的要求

AWS Marketplace 维持对所有产品和报价的要求 AWS Marketplace。这些要求有助于为我们的客户推广安全、可靠和值得信赖的精选数字目录。我们鼓励卖家审查其他控制措施和协议的实施情况，以满足其特定产品的需求。

AWS Marketplace 审核提交的所有产品及其相关元数据，以确保它们符合或超过当前 AWS Marketplace 要求。我们会审核并调整这些要求，以满足我们的安全要求。此外， AWS Marketplace 持续审查产品，以验证它们是否符合这些要求的任何更改。如果产品不合规， AWS Marketplace 会与您联系以更新您的产品。在某些情况下，您的产品可能暂时无法向新订阅用户提供，直到问题得到解决。

产品设置准则

所有专业服务产品必须符合以下产品设置准则：

- 所有定价维度都必须与实际提供的服务相关，并使用 AWS Marketplace计费。
- 您的产品必须列在这些类别之一下：评估、实施、托管服务、高级支持或培训。
- 除了所需的专业服务类别外，还应通过选择适合所提供服务的其他适当类别来正确分类您的产品。
- 您的产品徽标的设计不得与 AWS 徽标或来自无关第三方的任何徽标混淆。
- 您的产品详细信息不得包含令人反感或露骨的内容。他们必须遵守 AWS 可接受使用政策，网址为https://aws.amazon.com/aup/。
- 您的专业服务产品必须直接支持或提供与所列第三方软件产品相关的服务， AWS Marketplace 或者帮助客户实现与其采用或管理相关的具体成果 AWS Cloud。

客户信息要求

所有专业服务产品必须符合以下客户信息要求：

- 专业服务产品必须完全按照上 AWS Marketplace列出的维度计费。
- 您不得 AWS Marketplace 在任何时候为专业服务产品列表收集买家付款信息，包括信用卡和银行账户信息。

- 与您的专业服务产品一起提供 AWS 给您的任何订阅者或潜在订阅者信息，必须仅按照《 AWS Marketplace 卖家条款和条件》使用。

产品使用准则

所有专业服务产品都必须遵循以下产品使用准则：

- 客户通过专业服务列表与您联系后，您应在两个工作日内与他们联系。客户接受专属优惠后，除非专属优惠中另有说明，否则您必须在两天内与他们联系或向他们提供后续步骤。
- 客户必须能够轻松获得有关使用服务、故障排除和申请退款（如适用）等问题的帮助。必须在交付登录页面上指定支持联系选项。支持描述应包含关于客户可以期待的支持级别的声明。
- 您的产品概述应包括明确的价值主张、关键特征、详细文档的链接，以及对所提供服务的购买前和购买后支持的明确定义。
- 您的产品应有清晰明了的服务条款。

架构准则

所有专业服务产品都必须遵循以下架构准则：

- 专业服务产品必须与一项 AWS 服务或至少一种公共产品（最多四个）相关，在这些 AWS Marketplace 产品中，该产品直接为这些相关产品提供服务，或者提供的服务可以吸引更多订阅者使用这些相关产品。
- 如果所提供的服务需要客户基础设施中的额外资源，请遵循以下准则：
  - 以安全的方式配置资源，例如使用 AWS Security Token Service 或 AWS Identity and Access Management (IAM)。
  - 提供其他文档，包括对所有已配置 AWS 服务的描述、IAM政策声明以及如何在客户账户中部署和使用IAM角色或用户。
  - 在产品说明中加入一条通知，说明如果客户在 AWS Marketplace 交易之外产生了额外的 AWS 基础设施成本，则他们有责任支付额外的基础设施费用。
  - 如果您的产品部署了代理，请向客户提供有关描述如何在其 AWS 账户中部署代理的说明。

专业服务产品定价

您可以为专业服务产品设置以下产品定价模式：

- 专属优惠 - 对于卖家专属优惠，这些选项可用于多年和自定义期限合同。有关多年和自定义期限合同的更多信息，请参阅准备您的私人报价 和分期付款计划。

您只能为每个产品设置一个价格。

有关 AWS Marketplace 产品定价的更多信息，请参阅产品定价。

数据产品

AWS Data Exchange 是一项服务，可让 AWS 客户轻松安全地在中交换基于文件的数据集。 AWS Cloud作为提供商，AWS Data Exchange 无需构建和维护任何数据传输、授权或计费技术。AWS Data Exchange 中的提供商拥有安全、透明和可靠的渠道，可以更有效地接触 AWS 客户并向现有客户授予订阅权限。成为 AWS Data Exchange 提供商需要几个步骤来确定资格。

数据产品包含以下部分：

- 产品详细信息 – 此信息可帮助潜在订阅用户了解产品是什么。这包括名称、描述（简要和详细描述）、徽标图像和支持联系信息。产品详细信息由提供商填写。
- 产品优惠 – 为了使产品在 AWS Data Exchange 上可用，提供商必须定义公开优惠。这包括价格和持续时间、数据订阅协议、退款政策以及创建自定义优惠的选项。
- 数据集 – 一个产品可以包含一个或多个数据集。数据集是基于文件的数据内容的动态集。数据集是动态的，并通过使用修订版本进行版本化。每个修订版本可以包含多个资源。

有关更多信息，包括资格要求，请参阅《AWS Data Exchange 用户指南》中的 通过 AWS Data Exchange 提供数据产品。

提交要发布的产品

您可以使用产品提交流程在 AWS Marketplace上提供您的产品。产品可能非常简单，例如具有单一价格结构的 Amazon 机器映像 (AMI)、复杂、有多个AMIs AWS CloudFormation 模板，以及复杂的定价选项和付款时间表。您可以定义您的商品报价，并通过以下两种 AWS Marketplace 管理门户 方式之一提交报价：

- 使用产品选项卡 – 对于不那么复杂的产品，可以使用产品选项卡完整定义并提交请求。
- 使用 “资产” 选项卡 — 对于更复杂且需要更多定义的产品，您可以下载产品加载表 (PLF)，添加产品详细信息，然后使用文件上传选项上传填好的表单。

> **Note**
>
> 数据产品提供商必须使用 AWS Data Exchange 控制台发布产品。有关更多信息，请参阅《D AWSata Exchange 用户指南》中的发布新产品。

我们建议您首先使用产品选项卡来确定使用哪种方法。下表列出了可用于提交您的请求的配置和方法。第一列是产品的定价模型，其他三列描述了如何向客户部署产品。

定价模型	使用单节点推出的产品 AMI	推出产品时使用 AWS CloudFormation	作为软件即服务 (SaaS) 启动的产品
自带许可证 (BYOL)	产品选项卡	资产选项卡	
免费	产品选项卡	资产选项卡	
按小时付费	产品选项卡	资产选项卡	
按小时付费 – 年度	产品选项卡	资产选项卡	
按月付费	产品选项卡	资产选项卡	
按小时 – 月度	资产选项卡	资产选项卡	

定价模型	使用单节点推出的产品 AMI	推出产品时使用 AWS CloudFormation	作为软件即服务 (SaaS) 启动的产品
付费使用量（AWS Marketplace 计量服务）	产品选项卡	资产选项卡	
合同定价	产品选项卡		
SaaS 订阅			产品选项卡
SaaS 合同			产品选项卡
SaaS 传统			资产选项卡

您可以单独提交产品，或者，如果您使用产品加载表单，则可同时提交多个产品或产品更新。您不能使用 “产品” 选项卡提交多个商品。如果您不清楚要提交哪些产品以及如何提交，请先使用 “产品” 选项卡。如果您对提交有任何问题，请联系 AWS Marketplace 卖家运营团队。

使用“产品”选项卡

要访问产品选项卡，请登录 AWS Marketplace 管理门户。根据您管理的产品类型，从 “产品” 选项卡中选择 “服务器”、“SaaS” 或 “机器学习”。

将出现该产品类型的控制面板，并显示您当前的产品。如果您选择 “请求” 选项卡，控制面板将显示所有未处理的请求和已完成的请求历史记录。开始创建产品请求后，您可以保存正在进行的工作，如有必要，可以在多个不同的会话中创建请求。

当您提交产品请求时， AWS Marketplace 团队会对其进行审核。您可以在产品页面上监控您所请求的产品类型的请求状态。对于新商品，在您的发布申请获得批准后，您会收到一份限量版商品URL，用于预览和批准您提交的商品。您的产品在批准提交之前不会发布。

当您请求产品更新时，无需您审核和批准更改即可发布该更新。这包括添加或删除版本以及元数据更改。

您可在请求选项卡下跟踪您的请求状态。该选项卡显示以下内容之一：

- 草稿 - 您已启动请求流程，但尚未提交您的请求。
- 已提交 - 您已完成并提交您的请求，并且正在进行审核。

- 需要采取的行动- AWS Marketplace 团队已审核您的请求并需要更多信息。
- 需要批准 — AWS Marketplace 团队已URL为您的商品创建了限量商品信息。在发布URL之前 AWS Marketplace ，您必须审阅并批准或拒绝。如果批准，在发布网站时，状态将变为等待发布。如果拒绝，状态将恢复为草稿，以便您可以修改请求。
- 待发布 — 您已批准请求的模型并 AWS Marketplace 正在发布您的产品。
- 已过期 - 您已启动请求过程，但在 6 个月内未完成，因此，请求已过期。

如果某个条目的状态为已提交，您可以撤回提交。如果某个条目的状态为草稿，您可以删除请求。这样，您就可以重新开始。在删除草稿 条目时，该条目将移到请求历史记录选项卡中。

要在中添加您的商品 AWS GovCloud (US) AWS 区域，您必须拥有有效的 AWS GovCloud （美国）账户并遵守包括出口管制要求在内的 AWS GovCloud (US) 要求。

公司和产品徽标要求

您的公司徽标和产品徽标需要符合以下 AWS Marketplace准则，这样用户在浏览 AWS Marketplace 时可以获得一致的体验：

产品徽标规范 - 您的产品徽标图像应采用透明或白色背景，大小在 120 到 640 像素之间，长宽比为 1:1 或 2:1。

公司徽标规范 - 您的公司徽标图像应具有透明背景，大小为 220 x 220 像素，每一侧内沿允许 10 个像素的填充。

提交付费重新打包软件的要求

在提交重新打包软件的清单之前，您必须满足以下要求。在这种情况下，重新打包的软件包括开源软件 AMIs或由其他供应商创建的软件，例如AMI使用 Windows 的。

要求

- 产品标题必须注明重新包装所增加的价值。产品标题的示例包括：硬化、添加包装等。<Product><Product><Product1><Product2>
- 产品标题不得包含文档不支持的任何其他语言。例如，除非在您提供的产品详细信息中有所证实，否则产品标题不得使用已认证、原装或免费字样。
- 产品简短描述必须包括概述产品费用的明确声明。简短描述必须以此产品收取…相关费用开头。例如，如果某件产品包含卖家的支持费用，则产品描述应注明：此产品包含与之相关的卖家支持费用。

- 产品徽标必须与卖家注册过程中使用的公司徽标相同。只有当您使用官方软件徽标时，产品徽标才能与您的公司徽标不同，但您必须获得原始软件供应商的明确许可。如果获得明确许可，则必须在变更请求的备注部分中（或者使用产品加载表时，在文件上传页面的输入简要描述字段中）包含指向该文档的链接。
- 对于AMI商品，不得重复使用原始产品中的AMI名称。AMI名称必须以卖家名称开头并遵循以下格式：[卖家名称] [name-given-to-ami]。

  如果AMI名称不符合命名约定，则可以AMI从 AWS 控制台复制并重命名。有关更多信息，请参阅《亚马逊EC2AMIEC2用户指南》中的 “复制亚马逊”。

如果付费列表针对的不是贵公司创建的独立软件产品，并且该产品中没有添加知识产权（例如，捆绑其他软件库或添加特殊配置），则除了之前的要求外，还必须满足以下要求：

- 产品标题必须包含卖家名称（以及前面描述的增值）。卖家名称是卖家注册时使用的名称。例如，<seller> 为 <Product> 提供维护支持<。
- 产品详细描述的第一行必须以这是一款重新包装的软件产品，需要支付额外费用…开头（或者，如果它是开源的，则描述为这是一款重新打包的开源软件产品，需要支付额外费用…。然后，详细描述必须包括明确的陈述，概述您的收费情况，以及描述这些特征的其他详细信息。例如，对开源产品收取额外支持费用的SLA详细描述可能开头为：这是一款重新打包的开源软件产品，使用 {Details} 的支持需要支付额外费用。

对带有硬件组件的产品的要求

不允许销售硬件产品 AWS Marketplace。如果您提交的软件产品需要硬件组件（例如SIM卡、智能设备、物联网设备或传感器），则必须满足以下要求：

- 硬件组件不能出售 AWS Marketplace。
- 硬件组件的成本不能包含在产品的上市价格中。
- 清单的 “产品概述” 部分必须包含以下声明：必须单独购买此清单中可能需要的任何硬件。查看产品详情以了解更多信息。

AWS CloudFormation-已推出的产品（免费或付费）或基于使用量的付费产品 AMI

使用产品加载表 (PLF) 提交 AWS Marketplace 客户使用 AWS CloudFormation 模板发布的产品。PLF 可通过以下方式获得 AWS Marketplace 管理门户。

您可以按照以下主要步骤提交商品：

- 选择定价模式。
- 下载产品装货表 (PLF)，即微软 Excel 电子表格。
- 填写产品装货表。
- 按照表格中的说明提交您的商品。

有关完成每个步骤的更多信息，请按所列顺序展开各个部分。

选择定价模式

您必须为产品选择定价模式。您选择的型号控制您输入的定价信息PLF。有关支持的定价模式的列表，请参阅本指南AMI 产品定价中的。

正在下载新产品的 PLF

1. 启动 AWSMarketplace 管理门户。
2. 在 “资产” 选项卡的右侧窗格中，选择 “AMI与 CloudFormation 产品同一” 链接。

   表单出现在您的浏览器中。
3. 选择 “下载文件”，然后在 Excel 中编辑该文件。

   —或者—

   如果你有 Microsoft OneDrive，请选择 “编辑副本”。这样可以保存PLF到 OneDrive，你可以在那里对其进行编辑。

> **Note**
>
> 该电子表格包含多个示例产品。在提交表单之前，您必须将其删除。

PLF为现有产品下载

下载表格

- 启动 AWS Marketplace 控制面板，然后选择下载产品加载表。

该电子表格包含您的商品先前版本的数据。将该数据保留在原处，然后在下一个空白行中添加新产品。

填写 PLF

以下步骤说明了如何填写产品装货表 (PLF)。这些步骤适用于新产品和现有产品。

填写表格

1. 在 “退SKU款和取消政策” 列中，输入与您的商品相关的所有信息。

> **Note**
>
> 在 “产品访问说明” 列中，您必须提供详细、清晰的使用说明。请遵循本指南AMI和容器产品使用说明中列出的要求。

2. 在 “类型与端点的URL相关性” URL 列中，输入所需的信息AMI。

> **Important**
>
> 您必须AMI与 AWS Marketplace 共享您的信息。为此，请按照本指南正在提交AMIs给 AWS Marketplace中的步骤操作。

3. “最终用户许可协议 URL” 列提供了指向标准 AWS Marketplace 协议的链接。您可以接受该协议，也可以输入您喜欢使用的链接。EULA如果您提供链接，则该EULA链接必须允许客户从 Amazon S3 存储桶下载链接，例如链接。

   有关标准合约的更多信息，请参阅本指南中的标准化合同 AWS Marketplace中的。

4. 在 us-east-1 可供未来所有实例类型使用列中，在您打算使用的每个区域下**TRUE**输入**FALSE**或下方。 AWS

> **Note**
>
> GovCloud 各地区有额外要求。例如，您必须拥有一个 GovCloud 账户才能使用某个 GovCloud 区域。有关更多信息，请参阅《AWS GovCloud 用户指南》中的开始设置。

5. 在推荐的实例类型列中，接受推荐的实例类型或从列表中选择其他实例类型。确保实例类型在您要使用的区域中可用。

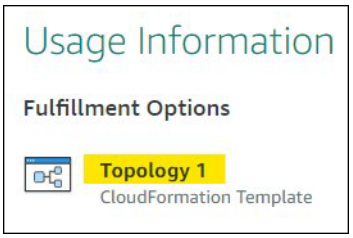
> **Note**
>
> - 大多数客户都接受推荐的实例类型。
> - 您必须确保该实例类型在与产品相同的地区可用。

6. 在推荐实例类型和要包括的国家 /地区之间的列中，**TRUE**在您的实例类型下方输入。这将激活实例类型。**FALSE**为其余类型输入。有关实例类型的更多信息，请参阅https://aws.amazon.com/ec2/实例类型/。
7. 在 “要包括的国家/地区” 和 “要排除的国家/地区” 列中，输入要包含或排除的任何国家/地区的双字母国家/地区代码，例如美国。
8. 在定价模式列中，输入产品的定价模式。

   以下列表描述了定价模型以及您必须填写的所有其他列。

   - BYOLURL— 输入许可证URL。无需输入定价信息。
   - 每小时-输入您设置为的任何实例类型的价格TRUE。将所有其他列留空。与此相关的列从 a1.medium 每小时价格到 z 1d.metal 每小时价格不等
   - 每小时年度定价 — 填写上一步中列出的列，再加上从 a1.medium 年价到 z1d .metal 年价开头的列。输入您设置为的任何实例类型的价格TRUE。您可以将所有其他列留空。
   - 使用情况-在 “FCP类别到维度 FCP 24 费率” 列中输入与使用维度相关的信息。
   - 合同-在 “合同类别到合同维度 24 36 个月费率” 列中，输入与合同维度相关的信息。
9. 在 “安全组规则 1 到安全组规则 12” 列中，输入有关您产品的安全组的信息。遵循`tcp,#,#,0.0.0.0/0`格式。例如，使用 f `tcp,22,22,0.0.0.0/0` or SSH 和 f `tcp,3389,3389,0.0.0.0/0` or RDP。
10. 在 “群集和AWS资源拓扑 1：群集标题和AWS资源拓扑 3：架构图” URL 列中，输入您的产品的CloudFormation数据。您必须输入以下数据：

- 拓扑标题-您的部署或配送选项的标题。标题显示在商品详情页面的 “配送选项” 部分。例如：



- 在 “估算价格” 列中，输入包含您的值的AWS计算器链接。
- 简短和详细描述-输入部署选项的描述。
- 模板 URL — 提供指向 Cloudformation 模板的可下载链接。
- 架构图 — 提供指向您的 CloudFormation拓扑架构图的可下载链接。 每个 Deployment 选项都必须有一个不同的图表，显示堆栈启动的内容。逻辑示意图必须符合中列出的要求架构示意图。

正在提交 PLF

以下步骤说明了如何提交已完成的PLF。

1. 登录到 AWS Marketplace 管理门户。
2. 在资产选项卡上，选择文件上传。
3. 在 “文件上传” 页面上，上传您的模板PLF和所有 AWS CloudFormation 模板。文件上传工具提供了一种安全传输机制和提交的文件历史记录。上传者会自动通知 AWS Marketplace，后者会审查您提交的内容是否符合政策、安全合规性、软件漏洞和产品可用性。如果团队对请求有任何疑问或问题，他们会向您发送一封电子邮件。

更新您的产品

对于使用产品加载表 (PLF) 创建的产品，您还可以使用PLF对这些产品进行更改。您可以对已完成的原始文件PLF进行更改，或者，如果它不可用，则可以从新的开始PLF。就像使用产品选项卡一样，您可以添加新版本，删除现有的版本以及更新定价、实例类型、区域可用性和元数据。要进行更新，请使用与新产品相同的方式准备任何更新的产品。准备好产品更新后，请按照以下步骤操作：

1. 使用现有的PLF，或启动 AWS Marketplace 管理门户，然后在 “资产” 选项卡上，选择 “文件上传”。在 “商品装货表和卖家指南” 下，您可以PLF为您的商品下载相应的。
2. 在中更新产品PLF。
3. 在 AWS Marketplace 管理门户 的资产选项卡上，选择文件上传。

4. 在 “文件上传” 页面上，上传您的更新模板PLF和所有 AWS CloudFormation 模板。文件上传工具提供了一种安全传输机制和提交的文件历史记录。上传者会自动通知 AWS Marketplace 团队开始处理您的请求。请包含提交的说明 (添加新版本，更改价格，更改元数据等)。

将审核您的产品提交的策略和安全合规性、软件漏洞和产品可用性。如果对请求有任何疑问或问题，AWS Marketplace 团队将通过电子邮件与您联系。对现有产品页面的更新将直接处理并发布，无需执行其他审核。

产品更改和更新

卖家可以随时提交对其产品的更改，将按照之前所述处理这些更改。不过，只能每隔 90 或 120 天进行某些更改，或者在实施待处理更改时进行更改。示例包括价格变动和 AWS 区域 /或实例类型变更。常见的更改包括：

- 新版本-软件的新版本以及补丁或更新的推出。根据您的要求，我们可以通知已订阅您 AWS Marketplace 内容的客户有关新版本的可用性或代表您发送升级说明。
- 元数据更改-产品信息（描述和使用说明）的更改。URLs
- 定价更改 - 对定价金额的更改。在完成请求后，将向当前客户发送通知。通知发送后，价格变动将在 90 天后的下一个月的第一天生效。例如，如果您在 3 月 16 日进行更改，那么 90 天后大约是 6 月 16 日，但价格变动发生在下个月的第一天。变更的实际日期为 7 月 1 日。
- 定价模式更改 - 对定价模式（例如按小时、免费、小时_年）的更改。并非支持所有定价模式更改，AWS Marketplace 团队必须审核并批准更改模型的所有请求。如果将免费模型更改为付费模型，将会对现有客户造成重大影响。一种替代方法是，提供具有额外特征的新产品并建议当前客户迁移。
- 区域或实例更改 - 添加或删除实例类型或区域。
- 产品下架-从中删除产品页面 AWS Marketplace ，以防止新客户订阅。在完成请求后，将向当前客户发送通知。

时限和预期

虽然我们努力尽快处理请求，但请求可能需要卖家和 AWS Marketplace 团队进行多次迭代和审核。使用以下内容作为完成该过程所需时间的指导：

- 总请求时间通常需要 2-4 个日历周。由于对产品元数据和软件进行了多次迭代和调整，更复杂的请求或产品可能需要更长的时间。

- 我们要求在任何计划中的活动或发布之前AMI至少提前 45 天完成产品申请，这样我们就可以相应地确定请求的优先顺序。

如果您对请求有任何问题，请联系 AWS Marketplace 卖家运营团队。

正在提交AMIs给 AWS Marketplace

所有AMIs构建和提交给的产品都 AWS Marketplace 必须遵守所有产品政策。我们建议您在提交AMI之前对您进行一些最后的检查：

- 从系统中删除所有用户凭证，例如，所有默认密码、授权的密钥、密钥对、安全密钥或其他凭证。
- 确保已禁用或锁定根登录。只允许使用 sudo 访问账户。
- 如果您要提交AMI要部署到该 AWS GovCloud (US) 地区，则需要拥有一个活跃的AWS GovCloud 账户并同意这些AWS GovCloud 要求，包括适用的出口管制要求。

AMI自助扫描

内提供自助AMI扫描功能 AWS Marketplace 管理门户。使用此功能，您可以快速启动扫描AMIs并接收扫描结果（通常在不到一个小时的时间内），并在一个位置获得清晰的反馈。

要开始AMI使用自助扫描进行共享和扫描

1. 导航到 “管理”/“ https://aws.amazon.com/marketplace/管理产品”/。
2. 选择AMI要共享的。
3. 查看您的扫描结果。

AMI成功扫描您的商品后，您可以按照当前流程上传商品加载表（PLF）将其提交给 AWS Marketplace 卖家运营团队。如果您有任何问题，请联系 AWS Marketplace 卖家运营团队。

要将您的包含AMI在自助扫描列表中，AMI必须位于`us-east-1`（弗吉尼亚北部）地区且归您的 AWS Marketplace 卖家账户所有。如果您需要向其他账户授予访问权限 AWS Marketplace 管理门户，则必须将这些账户注册为卖家。有关更多信息，请参阅 卖家注册过程。

AMI克隆和产品代码分配

提交后， AWS Marketplace 会AMIs为您指定应在其中提供软件的每个区域创建克隆。AMI在此克隆和发布过程中，将产品代码 AWS Marketplace 附加到克隆AMIs的中。该产品代码用于控制访问和计量用量。所有提交的内容都必须经过此AMI克隆过程。

最终核对清单

为了帮助避免产品发布延迟，请在提交产品请求之前使用此核对清单。

产品使用

- 生产就绪。
- 不按时间或其他限制来限制产品使用。
- 与一键履行体验兼容。
- 该软件包含使用该产品所需的一切，包括客户端应用程序。
- 默认用户利用随机化密码，并且/或者创建初始用户要求使用对实例唯一的值（如实例 ID）来验证买家已获授权可以使用此实例。

对于免费或付费产品：

- 使用产品不需要其他许可证。
- 付费重新打包的软件符合. AWS Marketplace 提交付费重新打包软件的要求
- 买家不必提供个人身份信息（如电子邮件地址）即可使用产品。

AMI准备

- 使用硬件虚拟机 (HVM) 虚拟化和 64 位架构。
- 不包含任何已知的漏洞、恶意软件或病毒。
- 买家拥有操作系统级别的管理权限。AMI
- 全面AMI进行AMI自助扫描。

对于 Windows AMIs

- 使用最新版本的`Ec2ConfigService`，如配置中所述 Windows 实例使用 EC2ConfigAmazon EC2 用户指南中的服务。

- `Ec2SetPasswordEc2WindowsActivate`、和`Ec2HandleUserData`插件已启用，如配置中所述 Windows 实例使用 EC2ConfigAmazon EC2 用户指南中的服务。
- 不存在来宾账户或远程桌面用户。

适用于Linux AMIs

- 根登录已锁定并禁用。
- 不包含已授权密钥、默认密码或其他凭证。
- 所有必填字段均已填写。
- 所有值都在指定字符限制之内。
- 所有URLs加载都没有错误。
- 产品图像至少为 110 像素宽，且宽高比介于 1:1 与 2:1 之间。
- 为所有已启用的实例类型指定定价（按小时、小时_月和小时_年度定价模式)。
- 指定月度定价（对于小时_月和月度定价模式）。

如果您对自动AMI构建有任何疑问或意见，请联系AWS Marketplace 卖家运营团队。

营销您的产品

您可以通过提高对产品页面的知名度 AWS Marketplace 以及将流量直接吸引到您的产品页面来为产品的成功做出贡献 AWS Marketplace。下面提供的信息和支持可帮助您推销在 AWS Marketplace中列出的一个或多个产品。

为期 180 天的学院 GTM

为期180天的GTM学院向所有 AWS Marketplace 卖家开放，它提供自助服务 go-to-market (GTM) 资源，以帮助您构建、激活和跟踪您的商品的需求挖掘活动。 AWS Marketplace您可以：

- 快速追踪对您的上架需求。
- 通过将 AWS Marketplace 价值主张整合到您的信息中，提高营销投资回报率 AWS 并增强客户信息。
- 您可以按照《AWS Marketplace 上市计划指南》内进行。

你可以从的 Marketpl ace 资源部分访问为期 180 天的GTM学院。AWS Marketplace 管理门户

宣传您的产品发布

我们鼓励您广泛宣布您的产品已在上市 AWS Marketplace。您可以通过新闻稿、推文、博客或所需的任何其他媒体渠道实现该目的。我们提供了您可能希望包含的示例文本，以及使用我们的商标和发布新闻稿的准则和说明。

在公开之前，我们将审核您的博客、推文和其他非新闻稿公告，以确保与 AWS 信息、品牌指南或声音保持一致。将您的审核请求提交给您的 AWS 客户经理。审核工作最多需要 10 个工作日才能完成。请在发布任何推文、博客和新闻稿时通知我们，我们将尽最大努力进行转发和推广。

AWS Marketplace 消息传递

在与客户沟通期间，您可能希望说明使用 AWS Marketplace购买您产品的用途、目标及优势。提到 AWS Marketplace时使用以下宣传用语。

什么是 AWS Marketplace？

AWS Marketplace 是一家在线商店，可让客户轻松查找、比较并立即开始使用上运行的软件和服务 AWS。访问者 AWS Marketplace 可以使用 1-Click 部署来快速启动预配置的软件，并且只需按小时或按月按使用量付费。 AWS 处理账单和付款，软件费用显示在客户的 AWS 账单上。

顾客为什么要去购物 AWS Marketplace？

寻找和部署软件可能具有挑战性。 AWS Marketplace 提供各种各样的商用和免费 IT 和商业软件，包括数据库和应用程序服务器等软件基础架构、物联网解决方案、开发人员工具和商业应用程序，均来自畅销产品。 AWS Marketplace 使客户能够比较选项、阅读评论并快速找到他们想要的软件。然后，他们可以使用一键式或 AWS Marketplace 管理门户将软件部署到自己的 Amazon Elastic Compute Cloud 实例。

在网站上明确标出了软件价格，客户可以使用已在 Amazon Web Services 中登记的付款方式立即购买大多数软件。软件费用与 AWS 基础设施费用显示在同一个月账单上。

为什么软件或 SaaS 卖家会继续销售 AWS Marketplace？

有了这些服务 AWS Marketplace，软件和软件即服务 (SaaS) 卖家 AWS 可以从提高客户认知度、简化部署和自动计费中受益。

AWS Marketplace 通过向成千上万的客户（从个人软件开发人员到大型企业）展示其产品，帮助软件和SaaS的软件和服务销售商 AWS 寻找新 AWS 客户。

Selin AWS Marketplace g on 使独立软件供应商 (ISVs) 能够为其软件增加按小时计费，而无需进行昂贵的代码更改。他们只需将 Amazon 系统映像 (AMI) 上传到 AWS 并提供每小时费用即可。账单由管理 AWS Marketplace，使卖家无需承担计量使用情况、管理客户账户和处理付款的责任，让软件开发人员有更多时间专注于开发出色的软件。

此外，客户还可以轻松部署预配置的软件映像，从而简化新客户的部署过程。

的评论 AWS Marketplace

AWS Marketplace 让买家能够提交对您的商品的评论。我们还为联合评论者（例如策划独立产品评论的 business-to-business 市场 G2）提供了整合其联合评论的能力。 AWS Marketplace

AWS Marketplace 买家评论必须符合买家用户指南中列出的评论指南。只有在对提交内容进行审核以确认其符合我们的审核标准之后， AWS Marketplace 才会发布评论提交。有关评论准则的更多信息，请参阅产品评论。联合审核组织使用自己独特的流程来验证其评论，并且不会在发布 AWS Marketplace 之前进行审核。如果您认为产品的联合评论不符合产品评论准则，或者产品评论包含令人反感的内容，请联系 AWS Marketplace 卖家运营团队。

审核者也可以根据五星评级系统为您的产品提供星级评分。系统计算星级评分的平均得分，从而提供您的产品的整体星级评分。联合评论也可以包括星级，但是联合评论中的星级评分与客户的星级评分不一致。 AWS

以下是有关产品评论特征的其他要点：

- 您不能从中删除商品评论 AWS Marketplace。不过，只要评论符合管理产品评论的审核标准，您就可以对任何评论发表评论。
- 如果您认为某条评论不符合评论准则或包含不当内容，可以联系 AWS Marketplace 卖家运营团队并说明您的问题。
- AWS 在中搜索商品的买家 AWS Marketplace 可以根据评分、经过验证的评论和外部来源的评论来搜索和筛选结果。 AWS 客户可以在搜索结果中看到来自外部的评分以及 AWS 客户评分。
- 您的商品的联合评论会自动添加到， AWS Marketplace 您无需支付任何费用。由于评论会自动添加，您不需要提交请求来添加联合评论。
- 如果您的产品没有任何联合评论，您可以联系联合审核者，并按照其流程审核您的产品。例如，对于 G2，您可以访问他们的网站并申请您的产品页面，以启动他们的审核流程。

链接到 AWS Marketplace

您的公司可能在网站上介绍和推广您的产品。我们鼓励您强调该产品可以继续运行 AWS 并且可以使用购买 AWS Marketplace。为了简化您的客户发现并部署您的软件的过程，我们提供了将您的客户链接到您的产品的说明。

使用 AWS Marketplace 品牌推广

> **Note**
>
> 您必须是 AWS 合作伙伴才能使用此处列出的指南和资产。有关成为 AWS 合作伙伴的信息，请参阅加入 AWS 合作伙伴网络。

AWS Marketplace 提供消息传递指导和品牌资产，以帮助您宣传您的软件、数据或服务。如果您想在上推广您的房源 AWS Marketplace，可以下载以下资源：

- AWS2023 年合作伙伴创意和信息传递指南
- 在 AWS Marketplace 品牌资产中提供

下载资源

1. 登录AWS 合作伙伴中心。
2. URL将以下内容粘贴到浏览器的地址栏中，然后打开页面：2023 年AWS合作伙伴创意和消息传递指南。
3. 选择下载。
4. 对在 AWS Marketplace 上架的品牌资产重复步骤 2 和步骤 3。

通过以下网址直接链接到您的产品 AWS Marketplace

添加指向您的网站或营销材料的深层链接，将客户引导至特定的产品页面 AWS Marketplace。使用以下链接格式：

```
https://aws.amazon.com/marketplace/pp/prodview-prodview ID
```

Replace（替换）*prodview ID* 使用您在 Marketplace 上的AWS商品的唯一字母数字代码。例如：

Example

```
https://aws.amazon.com/marketplace/pp/prodview-1234567890abcdef
```

URL当你在 a ws.amazon.com/marketplace 上搜索应用程序时，prodview ID 会显示在。如需有关查找您的 prodview ID 的帮助，请联系您的客户经理。

> **Note**
> 在使用链接之前对其进行测试，以确保它们将您的客户引导到正确的页面。

新闻稿

我们鼓励您 AWS Marketplace 通过自己喜欢的任何渠道宣布产品的上市情况。但是，在发布或发布任何公告之前，所有引用的新闻稿都 AWS Marketplace 必须经过亚马逊的审核和签署。虽然我们鼓励您发布公告，但我们不支持与 AWS Marketplace 卖家发布联合新闻稿。我们将在一定程度上 case-by-case 支持新闻稿，并引用以下内容 AWS。报价必须满足多个条件，包括但不限于：它宣布列出的新产品 AWS Marketplace 或服务，或者包括使用的客户推荐信 AWS Marketplace。

您必须起草所有新闻稿。我们建议使用以下标题：[插入产品名称] 现已上市 AWS Marketplace。请在此文档中使用该宣传信息以保持一致。

新闻稿应：

- 清楚准确地说明该公告与 Amazon.com 有何关联
- 明确你对客户的角色 AWS 以及与客户交往的角色
- 以客户为中心并强调客户利益

新闻稿不应：

- 使用合作伙伴、合作关系 或联盟 术语描述关系。我们更喜欢使用协议、合作 或关系。
- 引用 Amazon Web Services 高管的评价，除非之前已征得同意。
- 包含任何销售预测；除非在您的公司样板中引用该网站，否则，商户不应使用“.com”。
- 将您的组织称为 Amazon.com 的“关联方”，因为这可能会与 Amazon 合作者（我们的在线合作伙伴计划）发生混淆。
- 披露有关 Amazon.com 的专有信息或引用我们的股票代码。

请以文本格式将您的新闻稿提交给您的客户经理审核。此外，在使用任何商标之前，请查看亚马逊 Web Services AWS 商标指南。以下部分介绍了特定于 AWS Marketplace 商标的准则。

AWS Marketplace 商标使用指南

这些指南适用于您在事先获得亚马逊公司和/或其附属公司（“亚马逊”）批准的材料中使用 AWS Marketplace 徽标和商标（“商标”，统称为 “商标”）。必须始终严格遵守这些准则，如果使用商标的任何行为违反这些准则，则会自动终止与您使用商标相关的任何许可证。

1. 您只能将商标用于亚马逊明确授权的目的，并且您的使用必须：(i) 遵守与亚马逊签订的关于您使用任何商标的所有协议中的最新 up-to-date 版本（统称 “协议”）；(ii) 遵守这些指南的最新 up-to-date 版本；以及 (iii) 遵守亚马逊可能不时发布的适用于商标使用的任何其他条款、条件或政策。
2. 我们将提供一个批准的商标图像以供您使用。您不得以任何方式修改商标，包括但不限于：更改商标比例、颜色或字体，或者在商标中添加或删除任何元素。
3. 除了根据协议明确授权使用商标以外，您不得以暗示 Amazon 赞助或认可的任何方式使用商标。
4. 您不得使用商标贬低 Amazon、其产品或服务，或以某种方式使用商标而可能降低或以其他方式损害或玷污 Amazon 在商标中的声誉 (由 Amazon 自行决定)。

5. 商标必须单独出现，并在商标两侧与其他视觉、图形或文本元素之间留出一定的距离。在任何情况下，不得将商标放在任何影响商标可读性或显示效果的背景中。
6. 您必须在任何显示商标的材料中包含以下声明：“AWS Marketplace 并且该 AWS Marketplace 徽标是亚马逊公司或其附属公司的商标。
7. 您承认商标的所有权利是 Amazon 的独有财产，通过使用商标获得的所有声誉应使 Amazon 从中受益。您不得采取任何与 Amazon 的商标权利或所有权发生冲突的行为。

Amazon 保留单方面决定随时修改这些准则和/或批准的商标，并针对未经许可或未按照这些准则使用商标的情况采取相应措施的权利。如果您对这些准则有任何疑问，请与 trademarks@amazon.com 联系以获得帮助，或按下列地址写信给我们：

Amazon.com, Inc., Attention: Trademarks

PO Box 81226

Seattle, WA 98108-1226

卖家 AWS Marketplace 活动通知

AWS Marketplace 通过电子邮件、亚马逊 EventBridge 事件和亚马逊简单通知服务 (Amazon SNS) Semple Notification Service 主题提供及时的通知。

主题

- AWS Marketplace 事件的电子邮件通知
- 亚马逊 EventBridge 活动
- Amazon 简单通知服务 AWS Marketplace 事件通知

AWS Marketplace 事件的电子邮件通知

AWS Marketplace 使用您的 root 账户将自动电子邮件实时发送到您AWS账户的电子邮箱，为您提供有关 AWS Marketplace 优惠和协议的最新信息。

> **Note**
>
> 您可以为通知添加自定义电子邮件别名。有关更多信息，请参阅 the section called “添加或更新电子邮件地址”。

当买家接受报价时，卖家会收到电子邮件通知。购买者和独立软件供应商 (ISVs) 会收到公开订阅的通知。买家和渠道合作伙伴会收到有关渠道合作伙伴私人优惠的通知。ISVs

发送给渠道合作伙伴ISVs和渠道合作伙伴的通知包含以下详细信息：

- 购买日期、时间和时区
- 客户 AWS 账户 ID
- 产品名称
- 产品标识
- 优惠名称
- 优惠标识
- 协议标识
- 服务开始日期

- 服务结束日期
- 购买金额（针对合同和渠道合作伙伴）

> **Note**
>
> 如果您缺少 AWS Marketplace 电子邮件，请检查您的垃圾邮件文件夹或调整电子邮件设置。谷歌和雅虎等提供商可能会对其进行过滤。有关说明，请参阅防止有效电子邮件发送到垃圾邮件 (Google) 或在 Yahoo Mail 中屏蔽和解除封锁电子邮件地址。

事件类型

除机器学习产品外，所有产品和定价类型的电子邮件通知都支持以下事件类型：

- 买家请求专业服务产品。
- 重复扫描漏洞或定期扫描提醒。
- 经销商机会已创建、更新或已过期。
- 发布新的或更新的私人报价。
- 发布新的或更新的渠道合作伙伴私人报价。
- 通知买方和卖方接受报价。

管理通知

以下主题说明如何管理事件的电子邮件通知。

添加或更新电子邮件地址

您使用 AWS Marketplace 管理门户可添加最多 10 个电子邮件地址以接收自定义电子邮件通知。

要添加或更新电子邮件地址，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 从设置中选择通知选项卡。
3. 在自定义通知电子邮件下，选择添加电子邮件地址。
4. 对于收件人详细信息，请在电子邮件地址中输入自定义电子邮件地址。

5. （可选）选择添加新收件人以添加另一个电子邮件地址（总共最多 10 个）。
6. 选择提交。

取消收件人订阅通知

您可以删除电子邮件地址，这样收件人就可以取消订阅自定义电子邮件通知。

要取消收件人订阅通知，请执行以下操作：

1. 登录到 AWS Marketplace 管理门户。
2. 从设置中选择通知选项卡。
3. 在自定义通知电子邮件下，选择更新电子邮件地址。
4. 对于收件人的详细信息，请选择删除以删除电子邮件地址。
5. 选择提交。

   收件人将不再收到有关自定义事件的电子邮件通知。

> **Note**
>
> 您也可以使用电子邮件中的链接取消订阅。

亚马逊 EventBridge 活动

AWS Marketplace 已与亚马逊 EventBridge（以前称为亚马逊 CloudWatch 活动）集成。 EventBridge 是一项事件总线服务，可用于将应用程序与来自各种来源的数据连接起来。有关更多信息，请参阅 Amazon EventBridge 用户指南。

作为卖家，您会在创建报价 AWS Marketplace 时收到一个事件。该事件包含诸如 ID、到期日期和产品详情之类的详细信息。

主题

- AWS Marketplace 为API亚马逊 EventBridge 活动编目

AWS Marketplace 为API亚马逊 EventBridge 活动编目

本主题提供有关 EventBridge 控制台中 Marketplace Catalog 服务下的事件的详细信息。

卖家操作	收到的事件	相关主题
独立软件供应商 (ISV) 创建报价并使其可供购买	`Offer Released`	the section called “新优惠事件”
ISV渠道合作伙伴使用的产品来创建报价	`Offer Released`	the section called “新优惠事件”
渠道合作伙伴创建优惠	`Offer Released`	the section called “新优惠事件”
更改集成功	`Change Set Succeeded`	the section called “更改集事件”
更改集失败	`Change Set Failed`	the section called “更改集事件”
更改集取消	`Change Set Cancelled`	the section called “更改集事件”
在的产品上检测到安全漏洞ISV	`Products Security Report Created`	the section called “安全摘要报告的事件”

新优惠事件

当卖家创建优惠并使其可供购买时，他们会收到一个包含以下详细信息类型的事件：`Offer Released`。

> **Note**
>
> 有关创建 EventBridge 规则的信息，请参阅亚马逊 EventBridge 用户指南中的亚马逊 EventBridge 规则。

以下是由创建的新报价的事件正文示例ISV。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "Offer Released",
  "source": "aws.marketplacecatalog",
  "account": "123456789012",
  "time": "2023-08-26T00:00:00Z",
```

```
  "region": "us-east-1",
  "resources": [
    "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Offer/
offer-1234567890123"
  ],
  "detail": {
    "requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
    "catalog": "AWSMarketplace",
    "offer": {
      "id": "offer-1234567890123",
      "arn": "arn:aws:catalog:us-east-1:123456789012:Offer/offer-1234567890123",
      "name": "Offer Name",
      "expirationDate": "2025-08-26T00:00:00Z"
    },
    "product": {
      "id": "bbbbaaaa-abcd-1111-abcd-666666666666",
      "arn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
SaaSProduct/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"
    ]
  }
}
```

以下是渠道合作伙伴何时使用ISV某产品创建报价的事件正文示例。

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

以下是渠道合作伙伴创建优惠时的事件正文示例。

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

更改集事件

变更集完成后，卖家、渠道合作伙伴和 Private Marketplace 管理员会受到一个事件。当更改集完成且状态为 “成功”、“失败” 或 “已取消” 时， AWS Marketplace 目录会API发送一个事件。这些事件的来源是 `aws.marketplacecatalog`，可能的详细信息类型值包括 `Change Set Succeeded`、`Change Set Failed` 和 `Change Set Cancelled`。

> **Note**
> 有关变更集的信息，请参阅《AWS Marketplace 目录API参考》中的 “使用变更集”。

每个事件都包含更改请求的详细信息，例如更改集 ID、更改集名称、事件详细信息类型、失败代码（对于失败的请求）以及请求的开始和结束时间。这使您无需持续查询`DescribeChangeSet`操作或检查变更请求状态即可监控变更集。 AWS Marketplace 管理门户

> **Note**
> 有关创建 EventBridge 规则的信息，请参阅亚马逊 EventBridge 用户指南中的亚马逊 EventBridge 规则。

以下是 `Change Set Succeeded` 详细信息类型的事件主体示例。

```
{
  "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` 详细信息类型的事件主体示例。

```
{
  "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` 详细信息类型的事件主体示例。

```
{
  "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"
  }
}
```

安全摘要报告的事件

当在卖家的产品上检测到安全漏洞时，他们可以收到摘要报告事件，并定期提醒他们注意未解决的商品问题。这些事件的来源是`aws.marketplacecatalog`，细节类型是`Products Security Report Created`。

每个事件都包括检测到问题的产品和版本数量的摘要、受影响的最新版本数量以及为防止暂时限制这些产品或版本而需要采取解决方案的日期。

> **Note**
>
> 有关创建 EventBridge 规则的信息，请参阅亚马逊 EventBridge 用户指南中的亚马逊 EventBridge 规则。
> 有关管理安全事件的详细信息，请参阅博客上 AWS Marketplace博客文章中的 “如何提高产品目录的安全性”AWS 。

以下是 `Products Security Report Created` 详细信息类型的事件主体示例。

```
{
  "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"
   }
}
```

Amazon 简单通知服务 AWS Marketplace 事件通知

AWS Marketplace 还可以通过亚马逊简单通知服务 (Amazon SNS) Simple Notification Service 发送有关以下产品类型的买家订阅和合同权利变更的通知：

- 软件即服务 (SaaS) 产品
- 亚马逊机器映像 (AMI) 产品
- 容器产品

卖家报告、数据源和控制面板

AWS Marketplace 提供以下工具，用于收集和分析您的商品销售信息：

- 自动 AWS Marketplace 生成并提供给所有注册 AWS Marketplace 卖家@@ 的报告。
- 你可以用API它来访问这些报告的各个部分。
- 提供额外的客户信息的数据源，可用于找出在报告中列出的交易的客户信息。
- 由 Amazon QuickSight 提供支持的@@ 仪表板，其中包含图表、图表和见解，可帮助您访问和分析财务数据。

AWS Marketplace 在报告、数据源和仪表板中提供尽可能多的数据，同时遵守以下规定：

- 亚马逊用于保护客户数据的标准和原则。
- 买家在 AWS Marketplace上购买产品时接受的条款和条件。作为卖家，根据合同，您有义务安全管理买家数据并根据买家的要求删除数据。

> **Note**
>
> 要取消举报通知电子邮件，请联系AWS Marketplace 卖家运营团队。

卖家传输数据源服务

AWS Marketplace 提供数据源作为一种机制，用于将结构化、 up-to-date 产品和客户信息从 AWS Marketplace 系统发送到卖家 Amazon S3 存储桶，以便在卖家拥有的商业智能工具之间提取、转换和加载ETL（提取、转换和加载）。

交易数据以双时态结构传输和附加，因此卖家可以根据两个时间线存储和查询数据，且它们都具有时间戳

- 有效时间：某个事实实际发生的时间（“您所知道的”）
- 系统时间：将该事实记录到数据库的时间（“当您知道时”）。

在前一天的更新包含前一天的 24 小时数据UTC之后，每天午夜传送数据源。更新可以通过客户订阅、向客户开具发票或 AWS 付款来定义。

本部分概述了数据源，并说明了如何访问和使用它们。随后的部分将介绍每个数据源。

数据源的存储和结构

数据源收集以逗号分隔的值 (CSV) 文件并将其传送到您提供的加密 Amazon S3 存储桶。这些CSV文件具有以下特征：

- 它们遵循 4180 标准。
- 不带BOM字符编码为 UTF -8。
- 使用逗号作为值之间的分隔符。
- 字段由双引号进行转义。
- \n 是换行符。
- 日期按UTC时区报告，采用 ISO 8601 日期和时间格式，并在 1 秒内精确。
- 所有 `*_period_start_date` 和 `*_period_end_date` 值都包括在内，这意味着 `23:59:59` 是任何一天的最后一个可能的时间戳。
- 所有货币字段前面都有币种字段。
- 货币字段使用句点 (`.`) 字符作为小数分隔符，不使用逗号 (,) 作为千位分隔符。

数据源的生成和存储方式如下：

- 数据源在一天内生成，包含前一天 24 小时的数据。
- 在 Amazon S3 存储桶中，数据源使用以下格式按月进行组织：

  bucket-name/*data-feed-name_version*/year=*YYYY*/month=*MM*/data.csv
- 当生成每个每日数据源时，它会附加到该月的现有CSV文件中。当新的月份开始时，会为每个数据馈送生成一个新CSV文件。
- 数据馈送中的信息从 2010/01/01 到 2020/04/30（含）进行回填，并且可以在子文件夹的文件中找到。CSV `year=2010/month=01`

  您可能会注意到本月的给定数据源文件仅包含列标题，而不包含数据的情况。这意味着该月该源没有新条目。对于更新频率较低的数据源（如产品源），可能会发生这种情况。在这些情况下，在回填文件夹中提供数据。
- 在 Amazon S3 中，您可以创建 Amazon S3 生命周期策略来管理文件在存储桶中的保留时间。
- 您可以将 Amazon 配置SNS为在数据传输到加密的 S3 存储桶时通知您。有关如何配置通知的信息，请参阅《亚马逊简单通知服务开发者指南》SNS中的 “亚马逊入门”。

数据历史记录

每个数据源都包含记录数据历史记录的列。除了 `valid_to` 之外，这些列对于所有数据源都是通用的。它们作为通用历史记录架构包含在内，并且在查询数据时非常有用。

列名称	描述
valid_from	主键值相对于其他字段的值有效的第一个日期。
valid_to	此列仅显示在地址数据源中，且始终为空。
insert_date	将记录插入数据源中的日期。
update_date	上次更新记录的日期。
delete_date	该列始终为空。

下面显示了这些列的示例。

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` 字段共同构成了双时态数据模型。`valid_from` 字段的名字会表明该项目何时有效。如果项目已被编辑，源中可以有多条记录，每条记录都有不同 `update_date` 和相同的 `valid_from` 日期。例如，要查找某项的当前值，可以从具有最新 `valid_from` 日期的记录列表中查找具有最新 `update_date` 的记录。

在上面的示例中，该记录最初创建于 2018-12-12。然后在 2019-03-29 进行了更改（例如，记录中的地址发生了变更）。后来，在 2019-04-28，地址变更得到了更正（所以 `valid_from` 没有改变，但

update_date 变了）。更正地址（罕见事件）会追溯更改原始 valid_from 日期的记录，因此该字段不会更改。查找最新 valid_from 的查询将返回两条记录，具有最新 update_date 的记录会给出实际的当前记录。

访问数据源

要访问数据源，您需要将环境配置为接收加密的 Amazon S3 存储桶的数据源。您需要遵循多步流程来访问数据源，以下几组步骤说明了如何访问数据源。

1. 分配具有SQL和ETL（提取、转换、加载）经验的商业智能或数据工程师。此人还需要设置经验APIs。
2. 设置 Amazon 简单存储服务存储桶并订阅数据源。使用与您的 Marketplace 商品详情关联的AWS卖家账户编号。为此，您可以观看此 YouTube 视频或按照以下步骤操作。

   该视频和步骤说明了如何使用有助于简化配置的AWS CloudFormation 模板。

   a. 打开 Web 浏览器并登录 AWS Marketplace 管理门户，然后转到设置客户数据存储。
   b. 选择 “使用 AWS CloudFormation 模板创建资源”，在 AWS CloudFormation 控制台的另一个窗口中打开模板。
   c. 在模板中，指定以下内容，然后选择下一步：

      - 堆栈名称 – 您将创建的允许访问数据源的资源集合。
      - Amazon S3 存储桶名称 – 用于存储数据源的存储桶。
      - （可选）Amazon SNS 主题名称-用于在向 Amazon S3 存储桶 AWS 传输新数据时接收通知的主题。

   d. 在审核页面上，确认您的条目，然后选择创建堆栈。这将打开一个包含 CloudFormation 状态和详细信息的新页面。
   e. 在 “资源” 选项卡中，将 CloudFormation 页面中以下资源的亚马逊资源名称 (ARNs) 复制到 AWS Marketplace 设置客户数据存储页面上的字段中：

      - 用于存储数据源的 Amazon S3 存储桶
      - AWS KMS 用于加密 Amazon S3 存储桶的密钥
      - （可选）Amazon SNS 主题，用于在向 Amazon S3 存储桶 AWS 传送新数据时接收通知

   f. 在设置客户数据存储页面上，选择提交。
   g. （可选）编辑由 CloudFormation 模板创建的策略。有关更多信息，请参阅数据源策略。

   您现在已订阅数据源。下次生成数据源时，您可以访问数据。

3. 使用ETL（提取、转换、加载）操作将数据源连接到您的数据仓库或关系数据库。

> **Note**
>
> 数据工具具有不同的功能。您必须让商业智能工程师或数据工程师参与设置集成，使其与您的工具的功能相匹配。

4. 要运行或创建SQL查询，请将数据馈送配置为在数据工具中强制使用主键和外键。每个数据馈送代表一个唯一的表，您必须使用实体关系在数据架构中设置所有数据馈送。有关表和实体关系的更多信息，请参阅本指南数据源表概述中的。
5. 设置 Amazon 简单通知服务以自动刷新您的数据仓库或关系数据库。您可以将 Amazon SNS 通知配置为在来自每个唯一的 Feed 的数据传输到 S3 存储桶时发送提醒。如果卖家数据工具支持此功能，则当通过数据上传收到新数据时，可以利用这些通知自动刷新卖家数据仓库。请参阅《亚马逊简单通知服务开发者指南》SNS中的 “亚马逊入门”。

   示例通知：

```
{
        "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]"
        }
```

   通知可以有以下`executionStatus`状态：

   - `SKIPPED`— 卖家没有当天的新数据。
   - `COMPLETED`— 我们提供了包含新数据的 Feed。
   - `FAILED`— Feed 交付存在问题。
6. 通过运行SQL查询来验证设置。你可以在 /tree/main/ /queries 上使用本指南中的示例查询，也可以在 https://github.com/aws-samples/aws-marketplace-api-samples/tree/main/ /queries seller-data-feeds 上 GitHub使用查询。

> **Note**
>
> 本指南中的示例查询是为 AWS Athena 编写的。您可能需要修改查询才能与您的工具配合使用。

7. 确定业务用户想要在何处使用数据。例如，您可以：
   - 从您的数据仓库或SQL数据库中导出.csv 数据。
   - 将您的数据连接到 PowerBI 或 Tableau 等可视化工具。
   - 将数据映射到您的CRMERP、或财务工具，例如 Salesforce、Infor 或 Netsuite。

有关 AWS CloudFormation 模板的更多信息，请参阅《AWS CloudFormation 用户指南》中的使用 AWS CloudFormation 模板。

数据源策略

当 CloudFormation 模板创建您的 Amazon S3 存储桶时，它将为附加到该存储桶、 AWS KMS 密钥和亚马逊SNS主题的访问权限创建策略。这些策略允许 AWS Marketplace 报告服务使用数据源信息写入您的存储桶和SNS主题。每个策略都将包含如下所示的部分（此示例来自 Amazon S3 存储桶）。

```
        {
            "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:::datafeed-bucket",
                "arn:aws:s3:::datafeed-bucket/*"
            ]
        },
```

在本策略中，`reports.marketplace.amazonaws.com`是 AWS Marketplace 用于将数据推送到 Amazon S3 存储桶的服务主体。**`datafeed-bucket`**是您在 CloudFormation 模板中指定的存储桶。

当 AWS Marketplace 报告服务调用 Amazon S3 或 Amazon 时SNS，它将提供它打算写入存储桶的数据。 AWS KMS ARN为确保写入存储桶的唯一数据是以您的名义写入的数据，您可以在策略的条件中指定 `aws:SourceArn`。在以下示例中，必须替换 *`account-id`* 附上你的身份证 AWS 账户。

```
        {
           "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:::datafeed-amzn-s3-demo-bucket",
                "arn:aws:s3:::datafeed-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"]
                        }
                }
        },
```

取消订阅数据源

打开网络浏览器并登录 AWSMarketplace 管理门户。然后，前往联系我们页面，向 AWS Marketplace 卖家运营团队提交取消订阅请求。取消订阅请求可能需要 10 个工作日来处理。

使用数据源

当 Amazon S3 存储桶中存在数据时，您可以通过以下方式使用数据源：

- 下载. CSV来自您在中创建的 Amazon S3 存储桶中的文件，访问数据源以便您可以在电子表格中查看数据。
- 使用ETL（提取、转换和加载）、SQL查询、业务分析工具来收集和分析数据。

  您可以使用 AWS 服务来收集和分析数据，也可以使用任何可以进行分析的第三方工具。 CSV基于数据集。

示例：使用 AWS 服务收集和分析数据

以下过程假定您已配置环境以将数据源接收到 Amazon S3 存储桶，并且该存储桶包含数据源。

从数据源收集数据并进行分析

1. 在 AWS Glue 控制台 中，创建一个爬网程序以连接到存储数据源的 Amazon S3 存储桶、提取所需数据并在 AWS Glue Data Catalog中创建元数据表。

   有关的更多信息 AWS Glue，请参阅《AWS Glue 开发人员指南》。
2. 在 Athena 控制台中，对中的数据SQL运行查询。 AWS Glue Data Catalog

   有关 Athena 的更多信息，请参阅 《Amazon Athena 用户指南》。
3. 在 Amazon QuickSight 控制台中，创建分析，然后创建数据的可视化效果。

   有关亚马逊的更多信息 QuickSight，请参阅亚马逊 QuickSight 用户指南。

有关使用 AWS 服务收集和分析数据上传数据的一种方式的详细示例，请参阅博客上的使用卖家数据源配送服务、Amazon Athena 和 QuickSight 亚马逊创建卖家报告。 AWS Marketplace

数据源表概述

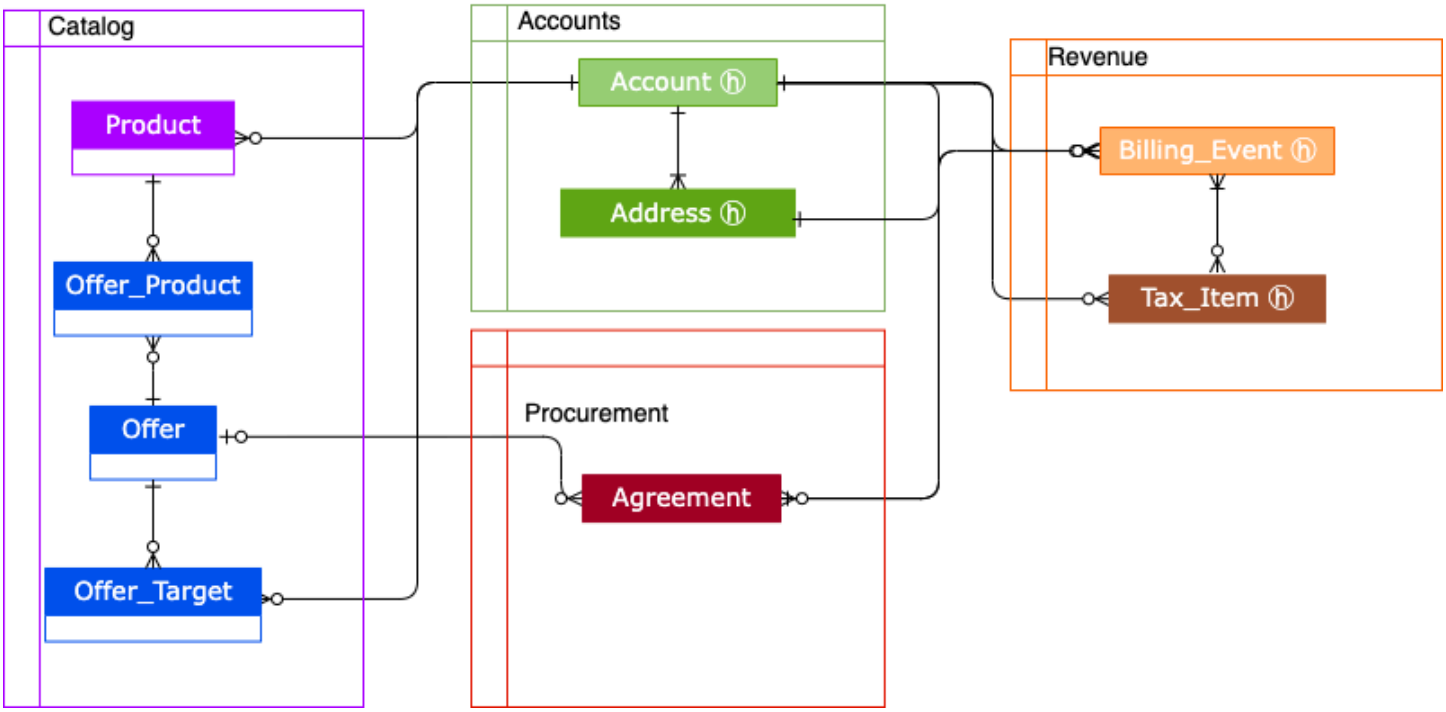
AWS Marketplace 提供的数据源是一组表格，您可以将这些表格连接在一起，为查询提供更多背景信息。

您的数据源中有三个一般域名或感兴趣的类别：

- 目录 - 包括有关您账户中产品和优惠的信息。

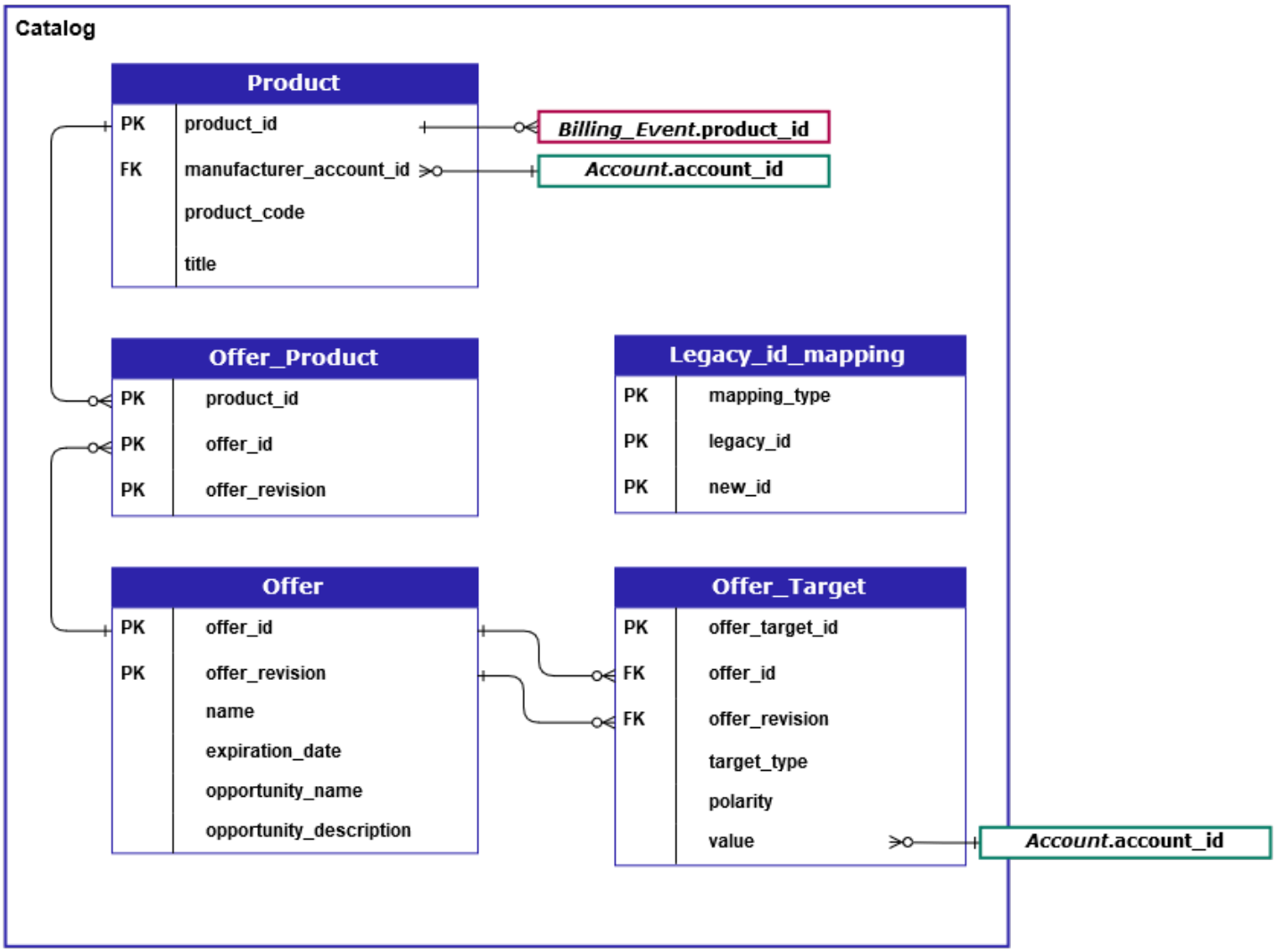
- 账户 — 包括有关提供或购买产品的账户 AWS Marketplace （您自己的账户或与您合作的各方账户，例如渠道合作伙伴或买家）的信息。
- 收入 - 包括有关账单、支付和税费的信息。
- 采购-包括有关您作为登记卖家创建的产品报价协议的信息。

下图显示了每个域中的表以及它们之间的关系。此图显示了目录、账户和收入域，包括其中的表格。



与目录相关的表

下图显示了目录域中各表之间的关系以及表中各字段的关系。



Product、Offer_Product、Offer、Offer_Target 和 Legacy_id_mapping 表位于目录域中。

Offer_Target 表包含目标的 account_id 的值字段，但仅当 target_type 的值为 account 时。

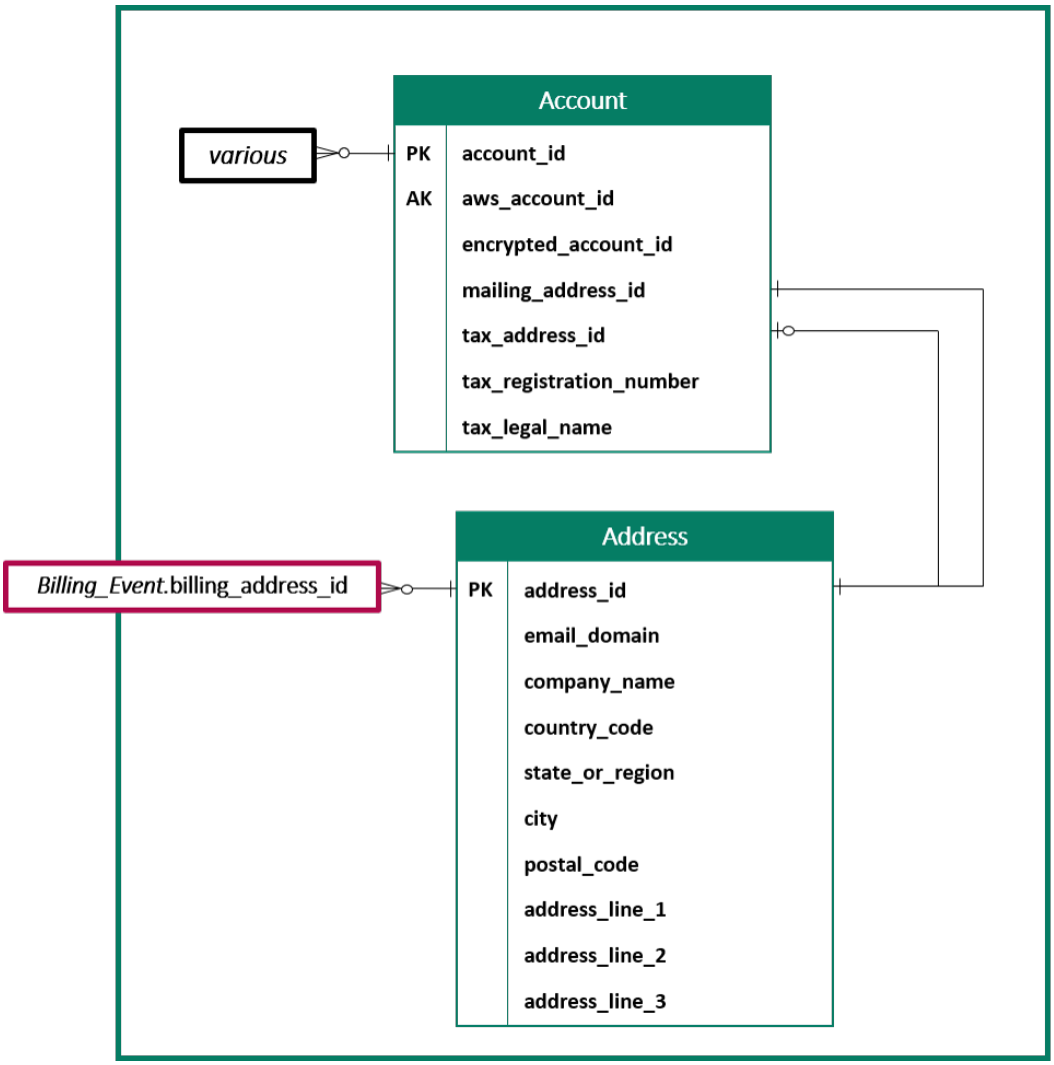
Legacy_id_mapping 表不用于当前数据。

> **Note**
>
> 有关这些表的更多信息，包括对表中每个字段的描述以及可以创建的连接，请参阅以下主题：
>
> - 产品数据源
> - 优惠产品数据源
> - 优惠数据源

- 优惠目标数据源
- 传统映射数据源

与账户相关的表

下图显示了账户域中 Account 和 Address 表之间的关系以及表中各字段的关系。

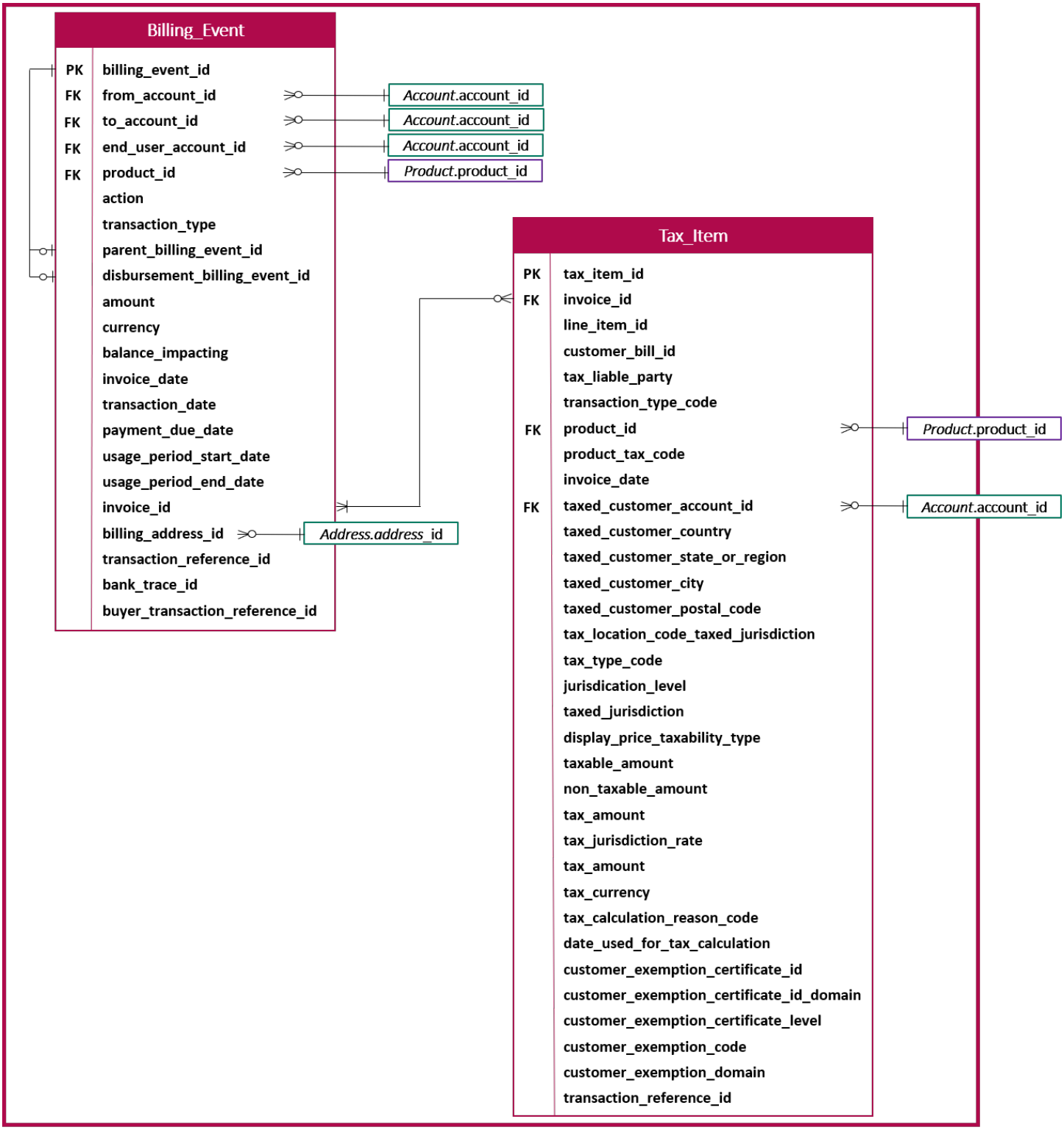


> **Note**
>
> 有关这些表的更多信息，包括对表中每个字段的描述以及您可以创建的联接，请参阅以下主题：
>
> - 账户数据源

- 地址数据源

与收入相关的表

下图显示了收入域中 Billing_Event 和 Tax_Item 表之间的关系以及表中各字段的关系。Billing_Event 表包括有关支付和账单事件的信息。


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**
>
> 有关这些表的更多信息，包括对表中每个字段的描述以及您可以创建的联接，请参阅以下主题：
>
> - 账单事件数据源
> - 税项数据源

与采购相关的表格

下图显示了 “采购” 域中 “协议” 表中的字段。

> **Note**
>
> 有关这些表的更多信息，包括对表中每个字段的描述以及可以创建的联接协议数据馈送，请参阅本指南中的。

以下各部分提供了每个域的实体关系 (ER) 图。每个 ER 图都显示表和每个表中的字段，以及可用于联接表的字段。

> **Note**
>
> 本部分中的 ER 图不包括所有数据源的公共字段。有关字段的更多信息，请参阅数据源的存储和结构。

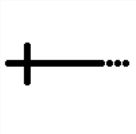
下表介绍了 ER 图中显示的符号。

符号	描述
PK	主键 - 表的主键。与 `valid_from` 和 `update_date` 字段一起使用时，它是唯一的。有关将这些字段一起使用的更多详细信息，请参阅数据历史记录。如果多个字段被标记为主键，则这些字段共同构成主键。

符号	描述
FK	外键 - 表示不同表中主键的字段。表中不一定是唯一的。 Note 在某些情况下，如果当前表中的记录在外表中没有相应的记录，则外键可能为空。
AK	备用密钥 - 可在表中用作密钥的密钥。遵循与主键相同的唯一性规则。
	连接器 - 字段之间的线代表一个连接，即两个可用于连接表的字段。线的末端代表连接的类型。此示例表示 one-to-many 连接。

连接器类型

下表显示每个连接器可以具有的端部类型。

连接器类型	描述
	一对 n – 具有此端的连接器表示在此侧只有一个值的连接。
	零或一到 n - 具有此端的连接器表示在此侧具有零或一个值的连接。

连接器类型	描述
	零或多对 n - 具有此端的连接器表示此侧具有零、一个或多个值的连接。
	一或多对 n - 具有此端的连接器表示在此侧具有一个或多个值的连接。

数据源查询示例

本节举例说明了使用提供的数据馈送进行复杂查询 AWS Marketplace。这些示例与您从 AWS Marketplace 管理门户中获得的卖家报告类似。您可以自定义这些查询以创建所需的其他报告。

示例查询

- 协议和续约
- 账单收入
- 未收或已支付的发票
- 已纳税发票
- 按产品划分的支出
- 销售补偿报告

协议和续约

要查找您的协议和续订数据，您可以运行一组查询，如下例所示。这些查询相互依赖，创建了 “协议和续订” 控制面板，即精细数据部分。您可以使用如图所示的示例，也可以根据您的数据和用例对其进行自定义。

查询中的注释解释了查询的作用以及如何对其进行修改。

```
Query currently under development.
```

账单收入

要查找您的发票数据，您可以运行一组查询，如下例所示。这些查询相互依赖，创建账单收入报告。您可以使用如图所示的示例，也可以根据您的数据和用例对其进行自定义。

查询中的注释解释了查询的作用以及如何对其进行修改。

```
-- 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)
```

未收或已支付的发票

要查找未收或已支付的发票，您可以运行一组查询，如下例所示。这些查询相互依赖，创建了收款和付款报告。您可以使用如图所示的示例，也可以根据您的数据和用例对其进行自定义。

查询中的注释解释了查询的作用以及如何对其进行修改。

```
-- 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)
```

已纳税发票

要查找您的纳税发票，您可以运行一组查询，如下例所示。这些查询相互依赖，创建税务报告。您可以使用如图所示的示例，也可以根据您的数据和用例对其进行自定义。

查询中的注释解释了查询的作用以及如何对其进行修改。

```
-- 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)
```

按产品划分的支出

要按产品查找支付的金额，您可以运行一组查询，如下所示。此示例与支付报告卖家报告相当。

这些示例查询相互依赖，创建了包含付款的最终产品详细信息列表。它还显示了如何在特定时间点获取产品信息。您可以如图所示使用此示例，也可以根据您的数据和用例对其进行自定义。

查询中的注释解释了查询的作用以及如何对其进行修改。

> **Note**
>
> 运行此查询时，我们假设数据是使用两个时间轴（`valid_from`和`update`列）提取的。有关轴的更多信息，请参见数据源的存储和结构。

```
-- 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;
```

销售补偿报告

要按客户查找账单收入，您可以运行一组查询，如下所示。此示例与销售补偿报告卖家报告相当。这些示例查询相互依赖，创建了客户详细信息的最终列表，其中包含向每位客户收取的软件使用费总额。您可以使用如图所示的查询，也可以根据您的数据和用例对其进行自定义。

查询中的注释解释了查询的作用以及如何对其进行修改。

> **Note**
>
> 运行此查询时，我们假设摄取的数据使用两个时间轴（`valid_from`和`update`列）。有关更多信息，请参阅 数据源的存储和结构。

```
    -- 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;
```

数据源

AWS Marketplace 提供大量数据馈送，以帮助卖家收集和分析有关您的商品销售的信息。所有注册 AWS Marketplace 卖家均可使用数据源。由于数据源是在一天内生成的，因此其中包含最新的可用数据。

以下各个部分介绍了每种数据源。

主题

- 账户数据源
- 地址数据源
- 协议数据馈送
- 账单事件数据源
- 传统映射数据源
- 优惠数据源
- 优惠产品数据源
- 优惠目标数据源
- 产品数据源
- 税项数据源

账户数据源

此数据源提供有关您与之交互的所有账户的信息：您自己的账户、与您合作的任何渠道合作伙伴、买家、付款人和所有纳税账户。

账户数据是不可变的，它不与版本号相关联。将附加对字段的更改，因此此数据源可能包含多个具有相同 `account_id` 和不同 `valid_from` 值的行。有关数据历史记录字段的信息，请参阅数据历史记录。

账户数据源每 24 小时刷新一次，因此每天提供的都是新数据。

下表说明了数据源列的名称和说明。

列名称	描述
account_id	账户的全球唯一标识符 (GUID)。 可用于连接 `Product`、`Offer_Target` 、`Billing_Event` 和 `Tax_Item` 数据源的字段。有关可用于连接的字段的信息，请参阅这些数据源。
aws_account_id	卖家 AWS 账户的账号，按 AWS 分区划分是唯一的。 AWS
encrypted_account_id	您的应用程序的个人买家的唯一已加密 ID。例如， AWS Marketplace 计量服务将的值用作`ResolveCustomer` 操作返回`CustomerIdentifier` 的值。`encrypted_account_id`
mailing_address_id	此账户的邮寄地址参考。
tax_address_id	此账户的纳税地址参考。
tax_registration_number	对于非美国账户，为此账户的税务登记号。
tax_legal_name	对于非美国账户，为法定公司名称。这是税务发票上使用的名称。

账户数据源示例

下面显示了账户数据源的示例。为了便于阅读，未显示数据历史记录列。有关数据历史记录字段的信息，请参阅数据历史记录。

account_id	aws_account_id	encrypted_account_id	mailing_address_id	tax_address_id	tax_registration_number	tax_legal_name
xk0CSmiAm6PQ	444456660000	Zf7 oMzheGWpH	25o3k46en6 fiiqtxwx8			

account_id	aws_account_id	encrypted_account_id	mailing_address_id	tax_address_id	tax_registration_number	tax_legal_name
4QqEog9iiaochIzuPlkMfba7a1oDlZ			e3 3 eViOfkaOiPalUiofjyFa			
7nyo5plyx81vx9ji04 OjwTRo1Ff8 88W8eEwTurbiQi	555567679999	373 q8vvuQUqm	5oj6mrrf2gvh2vj9hfqim800m vTjSzuLEHmyFY5Lr42s8	5oj6mrrf2gvh2vj9hfqim800m vTjSzuLEHmyFY5Lr42s8	SE823935083345	
VIeGa2t9j3MuxioH9wc8lsndXXCgGCGUreeXriocM5	73739998888	8 SPxAYmi8MwX	NLUc5UeiMlGFTrDWCoftDPhDUF1oaSd8xgl5QM8Db7	V5NYogwy0hBYBiWMhndGU4/fMggmuoTC2j7Pm8ZKKNNyT	DE469558025	

地址数据源

> **Important**
>
> AWS Marketplace 将于 2023 年 12 月停产使用卖家配送数据上传数据服务交付的 PIIFeed。使用 AddressFeed _V1 数据馈送来满足您的地址数据需求。如果您有任何疑问或需要支持，请联系 AWS Marketplace 卖家运营团队。

此数据源提供了您与之交互的所有账户的联系信息：您自己的账户、与您合作的任何渠道合作伙伴、买家、付款人和所有纳税账户。每次进行新交易时，都会扫描交易的客户地址；如果在数据源中不包含该地址，则会在数据源文件中添加新条目。

地址数据是不可变的。

地址数据源每 24 小时刷新一次，因此每天提供的都是新数据。

下表说明了数据源列的名称和说明。

列名称	描述
address_id	地址的唯一键。 可用于连接 `Billing_Event` 数据源的 `billing_address_id` 字段，或 `Account` 数据源的 `mailing_address_id` 或 `tax_address_id` 字段。
email_domain	为该账户登记的电子邮件地址域。
company_name	为该账户登记的公司名称。
country_code	为该地址登记的 ISO 3166 alpha-2 国家/地区代码。
state_or_region	为该地址登记的州/省或地区。
city	为该地址登记的城市。
postal_code	为该地址登记的邮政编码。
address_line_1	为该地址登记的地址第一行。
address_line_2	为该地址登记的地址第二行。
address_line_3	为该地址登记的地址第三行。

地址数据源示例

下面显示了地址数据源的示例。在数据源中，此信息以单个表格形式显示。为了便于阅读，数据显示在两个表格中，并且未显示数据历史记录列。有关数据历史记录字段的信息，请参阅数据历史记录。

address_id	email_domain	company_name	country_code	state_or_region	city	postal_code
v5nhby0wmhndgu4tc2j7pm8zknnnytBiYogwyAfMggmuo	a.com	Mateo Jackson's Company	DE		Hamburg	67568
g68xdbkzqdvvhzfbgw6yf5yos0a6nisvwhmh5 ViLjf	b.com	Mary Major's Company	美国	OH	Dayton	57684
nluc5gftrdwcoftdphduf1UeiMloasd8xgl5qm8db7	c.com	Our Seller	美国	NY	New York	89475

address_line_1	address_line_2	address_line_3
	19th Floor	

协议数据馈送

此数据源提供有关协议的信息，协议是提议者（记录在案的卖方）和接受者（AWS买方）之间为开始使用产品而签订的合同。此数据源提供有关您作为登记卖家创建的商品报价协议的信息。

协议数据源每 24 小时刷新一次。

下表列出并描述了数据馈送中的项目。

列	描述
valid_from	主键值相对于其他字段的值有效的第一个日期。
insert_date	将记录插入数据源中的日期。
update_date	上次更新记录的日期。
delete_date	该列始终为空。
Status	协议的当前状态。支持的状态包括： • 有效-协议的部分或全部条款已生效。 • 已终止 — 由于AWS启动的终止事件，协议在预先商定的结束日期之前结束。最常见的是付款失败。 • 已取消 — 接受者选择在协议结束日期之前终止协议。 • 已过期-协议在预先商定的结束日期结束。 • 已替换-使用替换报价替换协议。 • 续订-使用自动续订等功能，将协议续订为新协议。 • 已存档-协议已结束；未指定终止协议的确切原因，即 “已终止”、“已取消” 或 “已过期”。 • 已@@ 回退-由于出错，协议修订已被回退。较早的修订版现已启用。仅适用于无效的协议修订。 • 已@@ 取代-协议修订版不再处于活动状态，另一个协议修订版现已处于活动状态。仅适用于无效的协议修订。
估计的协议价值	客户在协议的整个生命周期内可能支付的已知总金额。

列	描述
货币代码	客户在协议的整个生命周期中可能支付的已知总金额的货币。
协议_id	协议的唯一标识符。
提案人_账户_id	提出此建议的卖家 PurchaseAgreement，由卖家账户的全球唯一标识符 (GUID) 表示。可用于加入账户数据源。
接受者_账户_id	接受这一点的买家 PurchaseAgreement，由买家账户的全球唯一标识符 (GUID) 表示。可用于加入账户数据源。
报价_修订_在接受时_修订	与本协议对应的报价的友好 ID。可用于加入报价和报价目标数据源。
start_time	协议开始的日期和时间。
end_time	协议结束的日期和时间。对于没有结束日期的 pay-as-you-go 协议，该字段为空。
验收时间	接受报价或创建协议的日期和时间。 • 可以追溯协议的 bring-your-own-license 日期 • 如果协议是使用 “远期协议” 功能创建的，则可能与起始日期不同
意图	上次修改协议时买方的意图。
之前_协议_id	先前协议的协议 ID。
状态_原因_代码	协议状态变更的原因。

协议数据馈送示例

valim	协议_id	提案人_账户_id	接受者_账户_id	offe	报价_修订_在接受时_修订	stare	end	验收时间	意图	之前_协议_id	stat	状态_原因_代码	估计的协议价值	货币代码
2021 20:0	agr 44c 78a 5t	88a xxxx xxxx x	88a xxxx xxxx x	offe krx xxx	1	2021 20:0	2021 20:0	2021 20:0	NE		AC		100	USD

账单事件数据源

此数据源提供有关账单事件的信息，包括开发票和付款。

例如，您可以使用此数据源来了解为买家开发票的时间和内容。您也可以使用示例SQL查询来分析来自此数据馈送的数据。

此数据源包含与您作为其记录卖家的账单事件相关的信息。对于通过渠道合作伙伴达成的协议，此数据源包含有关制造商和记录卖家之间的账单事件的信息。

账单事件数据源每 24 小时刷新一次，因此每天提供的都是新数据。

账单事件数据是不可变的。

下表说明了数据源列的名称和说明。

列名称	描述
billing_event_id	账单事件的标识符。此 ID 在卖家的环境中是唯一的。
from_account_id	启动账单事件的账户。如果 `transaction_type` 为 `SELLER_REV_SHARE`，则它是买家的付款人账户。这是账户数据源的外键。 可用于连接 `account_id` 数据源的 `Account` 字段。
to_account_id	接收产品的交易金额的账户。这是账户数据源的外键。 可用于连接 `account_id` 数据源的 `Account` 字段。
end_user_account_id	使用产品的账户。此账户可能与买家账户和付款人账户不同。 可用于连接 `account_id` 数据源的 `Account` 字段。
product_id	产品的标识符。这是产品数据源的外键。 可用于连接 `product_id` 数据源的 `Product` 字段。
action	此事件的操作类型。可能值如下所示： • `INVOICED` – 向买家开具了该金额的发票。 • `FORGIVEN` – 向买家开具了该金额的发票，并且 AWS 恢复了该费用。 • `DISBURSED` – 向卖家支付了该金额。这可以包括一个月的发票，也可以是按需付款。
transaction_type	交易的类型。有关示例，请参阅征税方案。可能值如下所示：

列名称	描述
	• SELLER_REV_SHARE – 一个正金额，这是卖家在与买家的协议中设置的价格。 • SELLER_TAX_SHARE – 一个正金额，这是添加到 SELLER_REV_SHARE 中以涵盖卖家所欠税款的金额。 • AWS_REV_SHARE – 一个负金额，这是列示费用。 • AWS_TAX_SHARE — 正金额；这是除此之外还征 AWS 收的税款SELLER_REV_SHARE 。此金额不影响卖家的余额。无需支付此金额，提供它是为了让卖家了解向买家开具发票并代表卖家汇款给主管部门的税款金额。 • *transaction_type* _REFUND – 买家请求的退款金额。 • *transaction_type* _CREDIT— 向买家 AWS 存入款项的金额。 • BALANCE_ADJUSTMENT — 为解决发票问题 AWS 而进行的调整。 • DISBURSEMENT – 如果 action 的值为 DISBURSED ，1 的值为 balance_impacting ，则这是支付给卖家的金额。如果 action 的值为 INVOICED，则此记录将全部或部分否定 parent_billing_event_id 记录。在这种情况下，将显示相关支付 disbursement_billing_event_id 且 balance_impacting 的值为 0。 • DISBURSEMENT_FAILURE – 否定交易。

列名称	描述
parent_billing_event_id	如果 broker_id 的值为 AWS_INC，action 的值为 DISBURSED 或 FORGIVEN，transaction_type 的值为 DISBURSEMENT，则 parent_billing_event_id 表示发起此账单事件的原始 billing_event_id 。如果 action 具有其他值，则此字段为空。 当 broker_id 的值为 AWS_EUROPE 时，parent_billing_event_id 表示在以下情况下启动此账单事件的原始 billing_event_id ： • action 的值为 FORGIVEN 或 INVOICED，且 transaction_type 的值为 AWS_REV_SHARE 、AWS_REV_SHARE_REFUND 或 SELLER_REV_SHARE_REFUND 。 • action 的值为 DISBURSED ，且 transaction_type 的值为 ANY（不包含 DISBURSEMENT_FAILURE ）。 • transaction_type 的值为 AWS_TAX_SHARE 、AWS_TAX_SHARE_REFUND 、SELLER_TAX_SHARE 或 SELLER_TAX_SHARE_REFUND 。 • action 的值为 DISBURSED ，且 transaction_type 的值为 DISBURSEMENT_FAILURE 。 当 broker_id 的值为 AWS_EUROPE ，则 parent_billing_event_id 表示在以下情景中上一次成功的支付账单事件的原始 billing_event_id ：

列名称	描述
	• action 的值为 DISBURSED ，且 transaction_type 的值为 DISBURSEMENT_FAILURE 。 当 broker_id 的值为 AWS_EUROPE 时，对于所有剩余情景，此字段均为空。
disbursement_billing_event_id	action 的值为 DISBURSED 且以下参数之一为 True 时的相关支付： • transaction_type like ('SELLER%') • transaction_type like ('AWS%') 其他所有情景下，此值为 Null。
amount	账单事件金额。
currency	ISO639 货币代码。
balance_impacting	在计算卖家付款时是否考虑该金额。值为 0 指示显示该金额以供参考，对余额没有影响。值为 1 指示在确定卖家付款时考虑该金额。
invoice_date	创建发票的日期。
payment_due_date	如果 action 的值为 INVOICED，则为发票的到期日期。
usage_period_start_date	记录中期间的开始日期。
usage_period_end_date	记录中期间的结束日期。
invoice_id	发 AWS 票编号。

列名称	描述
billing_address_id	地址数据源中的付款人账单地址参考。 可用于连接 `address_id` 数据源的 `Address` 字段。
transaction_reference_id	允许您交叉引用以下报告中的数据的标识符： • 支付报告 • 月度账单收入报告 • 销售补偿报告
bank_trace_id	对于支付交易（`transaction_type = DISBURSEMENT` 且操作为`DISBURSED`），是银行分配的追踪 ID。追踪 ID 可用于关联卖家银行提供的银行报告。
broker_id	为交易提供便利的商业实体的标识符。可能值如下所示： • `AWS_INC` — AWS, Inc.（总部设在美国）的标识符。 • `AWS_EUROPE` — 亚马逊 Web Services EMEASARL（总部设在卢森堡）的标识符。 • `NULL` – 前期没有明确 `broker_id` 的交易是由 `AWS_INC` 促成。
buyer_transaction_reference_id	一种标识符，它使用`GROUP BY`或中构造的窗口函数将账单馈送中的所有相关记录分组在一起 SQL。这些相关记录可以包含买家发票、卖家发票和上市费的增值税 (VATs)。

征税方案

买家和卖家所在国家/地区和州/省实施的征税模式决定了税款的收取和汇款方式。以下是可能的方案：

- 税款由收缴和汇出。 AWS在这些情况下，`transaction_type` 为 `AWS_TAX_SHARE`。
- 税款由卖家收取 AWS，支付给卖家，然后由卖家汇给税务机关。在这些情况下，`transaction_type` 为 `SELLER_TAX_SHARE`。
- 税款不是由收取的 AWS。卖家必须计算税款并将其汇款给税务机关。在这些情况下， AWS Marketplace 不进行税务计算或接收税务信息。卖家从收入份额中支付税款。

账单事件数据源示例

本部分显示开发票时和一个月之后的账单事件数据期间的示例。对于本部分中的所有表，请注意以下各项：

- 在数据源中，`billing_event_id` 值为 40 个字符的字母数字字符串。为了便于阅读，它们在这里显示为两个字符的字符串。
- 在数据源中，此信息以单个表格形式显示。为了便于阅读，数据显示在此处的多个表中，并且不显示所有列。

对于本部分中的示例，假定以下内容：

- Arnav 是买家。
  - 他的账户 ID 是 `737399998888`。
  - 他位于法国，需要遵守电商平台促进法的规定。有关更多信息，请参阅 Amazon Web Service 税务帮助。
  - 他购买了 `prod-o4grxfafcxxxx` 并因他每月使用该产品而获得了 120.60 美元的发票。
  - 他在当月内支付了发票。
- Jane 是制造商。
  - 她的账户 ID 是 `111122223333`。
- Paulo 是记录卖家。
  - 他的账户 ID 是 `777788889999`。
  - 他住在堪萨斯州，这里不受电商平台促进法约束。

记录卖家的账单事件数据源

作为记录卖家，Paulo 向买家 Arnav 开具发票。

下表显示了在 Paulo 向 Arnav 开发票时其数据源中的相关信息。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	action	transaction_type
I0	737399998888	777788889999	737399998888	prod-o4grxfafcxxxx	INVOICED	SELLER_REV_SHARE
I1	737399998888	AWS	737399998888	prod-o4grxfafcxxxx	INVOICED	AWS_TAX_SHARE
I2	777788889999	111122223333	737399998888	prod-o4grxfafcxxxx	INVOICED	SELLER_REV_SHARE
I3	777788889999	AWS	737399998888	prod-o4grxfafcxxxx	INVOICED	AWS_REV_SHARE

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		100	USD	2018-12-31T00:00:00Z	781216640
		20.6	USD	2018-12-31T00:00:00Z	781216640
		-80	USD	2018-12-31T00:04:07Z	788576665
		-0.2	USD	2018-12-31T00:04:07Z	788576665

下表显示了在 Arnav 支付发票后，月底时 Paulo 数据源中的相关信息。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	action	transaction_type
I10	737399998888	777788889999	737399998888		DISBURSED	SELLER_REV_SHARE
I12	777788889999	111122223333	737399998888		DISBURSED	SELLER_REV_SHARE
I13	777788889999	AWS	737399998888	prod-o4grxfafcxxxx	DISBURSED	AWS_REV_SHARE
I14	AWS	777788889999			DISBURSED	DISBURSEMENT

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
I0	I14	-100	USD	2018-12-31T00:00:00Z	781216640
I2	I14	80	USD	2018-12-31T00:04:07Z	788576665
I3	I14	0.2	USD	2018-12-31T00:04:07Z	788576665
		19.8	USD		

制造商的账单事件数据源

下表显示了在 Paulo 向 Arnav 开发票时 Jane 数据源中的相关信息。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	action	transaction_type
I5	777788889999	111122223333		prod-o4grxfafcxxxx	INVOICED	SELLER_REV_SHARE
I6	777788889999	111122223333		prod-o4grxfafcxxxx	INVOICED	SELLER_TAX_SHARE
I7	111122223333	AWS		prod-o4grxfafcxxxx	INVOICED	AWS_REV_SHARE

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
		73.5		2018-12-31T00:04:07Z	788576665
		6.5		2018-12-31T00:04:07Z	788576665
		-7.35		2018-12-31T00:04:07Z	788576665

下表显示了发票支付发票后，月底时 Jane 数据源中的相关信息。

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	action	transaction_type
I30	777788889999	111122223333		prod-o4grxfafcxxxx	DISBURSED	SELLER_REV_SHARE
I31	777788889999	111122223333		prod-o4grxfafcxxxx	DISBURSED	SELLER_TAX_SHARE

billing_event_id	from_account_id	to_account_id	end_user_account_id	product_id	action	transaction_type
I32	111122223333	AWS		prod-o4grxfafcxxxx	DISBURSED	AWS_REV_SHARE
I33	AWS	111122223333			DISBURSED	DISBURSEMENT

parent_billing_event_id	disbursement_billing_event_id	amount	currency	invoice_date	invoice_id
I5	I33	-73.5	USD		
I6	I33	-6.5	USD		
I7	I33	7.35	USD		
		72.65	USD		

示例查询

如使用数据源中所述，您可以使用 Athena 对收集并作为数据源存储在托管的 Amazon 存储桶中的数据运行查询。本部分提供了您执行此操作可能采取的常见方法的一些示例。所有示例都假定使用单一币种。

示例 1：开具发票的金额，含税

要了解向买家开具发票的金额（含税），您可以运行查询，如以下示例所示。

```
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'
  );
```

示例 2：代表卖家向买家开具发票的金额

要了解代表卖家向买家开具发票的金额，您可以运行查询，如以下示例所示。

```
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'
;
```

示例 3： AWS 可以代表卖家收取的金额

要了解 AWS 可以代表卖家收取的金额（减去任何退款、积分和免除的账户），您可以运行查询，如以下示例所示。

```
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')
;
```

示例 4：卖家可以收取的金额

要了解卖家可以收取的金额，您可以运行查询，如以下示例所示。此示例删除了房源费用和收取的 AWS 税费，并添加了任何特殊余额调整。

```
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')
;
```

您也可以使用以下查询来收集相同的信息，如以下示例所示。

```
SELECT sum(amount) FROM billing_event
WHERE
  balance_impacting = 1
  and action in ('INVOICED','FORGIVEN')
;
```

以下示例显示相同信息，但仅限于 2018 年交易，并假定所有买家都按照发票支付了款项。

```
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')
;
```

示例 5：支付金额

要了解已支付的金额，您可以运行查询，如以下示例所示。

```
select sum(amount) FROM billing_event
WHERE
  action ='DISBURSED'
  and transaction_type like 'DISBURSEMENT%'
;
```

示例 6：待支付金额

要了解待支付的金额，您可以运行查询，如以下示例所示。此查询删除了已付款的金额。

```
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
      )
    )
;
```

获取相同信息的另一种方法是运行查询来获取卖家余额，如以下示例所示。

```
SELECT sum(amount) FROM billing_event
WHERE
 balance_impacting = 1
;
```

以下查询扩展了我们的示例。它将结果限制为 2018 年交易，并返回有关交易的其他详细信息。

```
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
);
```

示例 7：一组发票的余额

要了解一组发票的总额，您可以运行查询，如以下示例所示。

```
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;
```

传统映射数据源

此数据源列出了产品 ID 和优惠 ID 如何映射到传统的全局唯一标识符 (GUID)。旧版 GUID 用于较早的报告中，新 ID 用于数据源和 AWS Marketplace API 中。

此数据源提供了有关您作为记录卖家创建的所有产品以及您有权转售的所有产品的信息。

传统映射数据源每 24 小时刷新一次，因此每天提供的都是新数据。

下表说明了数据源列的名称和说明。

列名称	描述
mapping_type	这是产品 ID 还是优惠 ID。
legacy_id	此产品或优惠的传统 ID。
new_id	此产品或优惠的友好 ID。此 ID 用作主键，并与所有当前 API 操作一起使用。

传统映射数据源示例

下面显示了传统映射数据源的示例。为了便于阅读，未显示数据历史记录列。有关数据历史记录字段的信息，请参阅数据历史记录。

mapping_type	legacy_id	new_id
优惠	8a806c74-dbd6-403e-9362-bb08f417ff37	offer-dacpxznflfwin
产品	1368541d-890b-4b6c-9bb9-4a55306ab642	prod-o4grxfafcxxxy
优惠	558d8382-6b3a-4c75-8345-a627b552f5f1	offer-gszhmle5npzip

优惠数据源

优惠数据源提供了有关您作为记录卖家创建的所有优惠的信息。如果单个优惠具有多个修订版，则所有修订版都包含在数据源中。

当您创建优惠修订版且公开字段中的数据发生更改时，会在数据源中为同一主键（`offer_id` 和 `offer_revision`）创建一条新记录。但 `valid_from` 字段具有不同的值。有关数据源历史记录列的更多信息，请参阅数据历史记录。

每 24 小时刷新一次优惠数据源，因此每天都提供新数据。

下表提供了数据源列的名称和说明。

列名称	描述
offer_id	优惠的友好标识符。 可用于连接 Offer_Product 数据源的 offer_id 字段。
offer_revision	优惠修订版。此字段与 offer_id 字段组合使用来构成主键。 offer_id 可用于连接 Target_Offer 数据源的 offer_id 和 offer_revision 字段。
name	卖家定义的优惠名称。
expiration_date	优惠到期的日期和时间。
opportunity_name	与此优惠相关的任何机会数据。如果优惠绑定到 AWS 机会，则会填充此字段。
opportunity_description	与此优惠相关的任何描述性信息。如果优惠绑定到 AWS 机会，则会填充此字段。
seller_account_id	卖家账户的全局唯一标识符 (GUID)。可用于连接账户数据源的 account_id 字段。
opportunity_id	只有当经销商销售您的产品时，才会填写机会的标识符。如果产品相同，则由不同渠道合作伙伴（或卖家）创建的所有优惠具有相同的 opportunity_id 。

优惠数据源示例

下面显示了优惠数据源的示例。为了便于阅读，未显示数据历史记录列。有关数据历史记录字段的信息，请参阅数据历史记录。

offer_id	offer_revision	name	expiration_date	opportunity_name	opportunity_description	seller_account_id	opportunity_id
offer-dacpxznflfwin	1	企业合同计划优惠	9999-01-01T00:00:00Z				
offer-gszhmle5npzip	1	卖家创建的专属优惠	2020-10-31T00:00:00Z				
offer-hmzhyle8nphlp	1	企业合同计划优惠	9999-01-01T00:00:00Z				

优惠产品数据源

一种优惠可以包含多种产品，一种产品可以包含在不同优惠中。此数据源列出了有关优惠与产品之间的关系的信息。

此数据源提供了有关您作为记录卖家创建的所有产品优惠的信息。

当您在优惠中添加或删除产品时，您将创建优惠修订版。

每 24 小时刷新一次优惠产品数据源，因此每天都提供新数据。

下表说明了数据源列的名称和说明。有关数据源历史记录列的信息，请参阅数据历史记录。

列名称	描述		
offer_id	此优惠的友好标识符。 可用于连接 `Offer` 数据源的 `offer_id` 字段。		

列名称	描述		
offer_revision	与 `offer_id` 字段组合使用来构成优惠修订版的外键。		
product_id	产品的友好标识符，这是此优惠公开的产品的外键。 可用于连接 `Product` 数据源的 `product_id` 字段。		

优惠产品数据源示例

下面显示了优惠产品数据源的示例。

offer_id	offer_revision	product_id
offer-dacpxznflfwin	10	prod-o4grxfafcxxxx
offer-gszhmle5npzip	24	prod-o4grxfafcxxxy

优惠目标数据源

此数据源为您作为记录卖家创建的所有优惠列出了优惠修订版的目标。如果单个优惠具有多个修订版，则所有修订版都包含在数据源中。

当您创建优惠修订版且公开字段中的数据发生更改时，会在数据源中为同一主键（`offer_id` 和 `offer_revision`）创建一条新记录，但 `valid_from` 字段的值不同。

优惠目标数据源每 24 小时刷新一次，因此每天提供的都是新数据。

下表说明了数据源列的名称和说明。

列名称	描述
offer_target_id	源的主键。
offer_id+offer_revision	优惠的标识符和修订版。这两列引用了与此目标相关的优惠。 可用于连接 `Target` 数据源的 `offer_id` 和 `offer_revision` 字段。
target_type	指示优惠接收者是 `BuyerAccounts` （这指示专属优惠）还是 `ParticipatingPrograms` 。
polarity	指示优惠是否打算面向 `target_type` 提供。可接受的值如下所示： • `PositiveTargeting` – 优惠适用于此 `target_type` 。 • `NegativeTargeting` – 优惠不适用于此 `target_type` 。
value	代表目标的字符串： AWS 账户 ID 或可与优惠一起使用的程序。例如，AWS Marketplace (SCMP) 的标准合同或AWS Marketplace 现场演示计划 (FDP)。

优惠目标数据源示例

下面显示了优惠目标数据源的示例。为了便于阅读，未显示数据历史记录列。有关数据历史记录字段的信息，请参阅数据历史记录。

offer_target_id	offer_id	offer_revision	target_type	polarity	value
925ddc73f6a373b7d5	offer-dacpxznflfwin	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract

offer_target_id	offer_id	offer_revision	target_type	polarity	value
544ea3210610803b600					
471ff22ae3165278f1fb960d3e14517bcd601	offer-gszhmle5npzip	1	ParticipatingPrograms	PositiveTargeting	FieldDemonstration
511ff22adfj65278f1fb960d3e14517bcd6e602	offer-gszhmle5npzip	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract

产品数据源

此数据源提供了有关您作为记录卖家创建的所有产品以及您有权转售的所有产品的信息。

产品数据是可变的。这意味着，当您更改以下某个字段的值时，将在数据源中创建一条新记录，其中包含不同的 `valid_from` 字段值。有关数据源历史记录列的更多信息，请参阅数据历史记录。

每 24 小时刷新一次产品数据源，因此每天都提供新数据。

下表说明了数据源列的名称和说明。

列名称	描述
product_id	产品的友好标识符。 可用于加入`AccountBilling_Event` 、和`Offer_Product` 数据馈送的`product_id` 字段。
manufacturer_account_id	产品拥有者的标识符。这是账户数据源的外键。

列名称	描述
	可用于连接 `Account` 数据源的 `account_id` 字段。
product_code	用于计量产品的现有授权产品代码。此值还用于将数据与报告合并在一起，或者用于引用 AWS Marketplace Metering Service 中提供的内容。
title	产品的标题。

产品数据源示例

下面显示了优惠目标数据源的示例。为了便于阅读，未显示数据历史记录列。有关数据历史记录字段的信息，请参阅数据历史记录。

product_id	manufacturer_account_id	product_code	title
prod-o4grxfafcxxxx	555568000000	product_code_1	Product1
prod-t3grxfafcxxxy	444457000000	product_code_2	Product2
prod-x8faxxfafcxxy	666678000000	product_code_3	Product3

税项数据源

此数据源提供有关客户发票的税务计算的信息。

对于给定客户发票 (`invoice_id`) 的给定产品 (`product_id`)，可能有多个行项目 (`line_item_id`)，每个税务管辖区一个或多个。例如，对于使用不同 AWS 实体（例如美国和爱尔兰）使用不同 AWS 区域规则的客户而言，会发生这种情况。要详细了解在哪里征 AWS 收销售税 VAT，或者GST对您的销售额征收销售税，并以, Inc. 的名义将此类税款汇给当地税务机关，请参阅亚马逊网络服务税务帮助。AWS

税项数据源每 24 小时刷新一次，因此每天提供的都是新数据。

税项数据是不可变的。

下表说明了数据源列的名称和说明。有关数据历史记录列的信息，请参阅数据历史记录。

列名称	描述
tax_item_id	税项记录的唯一标识符。
invoice_id	发 AWS 票编号。您可以将此值与 `product_id` 值结合使用，以查找相关的税务记账事件。
line_item_id	客户账单行项目的唯一标识符。退款交易具有与远期税务交易相同的行项目 ID。
customer_bill_id	客户账单的唯一标识符。买家可与卖家共享此标识符，以帮助识别和解决税务计算问题。
tax_liable_party	`AWS` 或 `Seller`。如果卖家是税务责任方，则会征收税款。如果 AWS 是纳税方，则由 AWS 其征收和汇出销售税。有关更多信息，请参阅 AWSMarketplace 卖家和税收征收。 如果未征收税款，则此处不显示值。卖家需要确定是否针对每张发票征收某些税务，因为卖家有征税责任。
transaction_type_code	交易的类型。可能值如下所示： • `AWS` – 远期税务交易 • `REFUND` – 全部或部分退款 • `TAXONLYREFUND` – 仅税费退款 退款交易具有与原始远期交易相同的行项目 ID。
product_id	产品的外键。 可用于连接 `product_id` 数据源的 `Product` 字段。

列名称	描述
product_tax_code	用于标识产品税费属性的标准代码。卖家在创建或修改产品时选择属性。
invoice_date	创建发票的日期。
taxed_customer_account_id	纳税的账户实体的外键。 可用于连接 `account_id` 数据源的 `Account` 字段。
taxed_customer_country	用于计税的地址的 ISO 3166 alpha 2 国家/地区代码。
taxed_customer_state_or_region	用于税务计算的州、地区或省。
taxed_customer_city	用于税务计算的城市。
taxed_customer_postal_code	用于税务计算的邮政编码。
tax_location_code_taxed_jurisdiction	与纳税位置关联的 vertex 地理编码。
tax_type_code	应用于交易的税费类型。可能的值为 `None`、`Sales` 和 `SellerUse` 。
jurisdiction_level	用于税务位置的地址的管辖区级别。可能的值为 `State`、`County`、`City` 和 `District`。
taxed_jurisdiction	税务管辖区的名称。
display_price_taxability_type	买家看到的价格是含税还是不含税。所有 AWS Marketplace 优惠均不含税。
taxable_amount	在此管辖区级别应纳税的交易金额。
nontaxable_amount	在此管辖区级别可免税的交易金额。
tax_jurisdiction_rate	在此管辖区级别应用的税率。
tax_amount	在此管辖区级别收取的税额。

列名称	描述
tax_currency	上述金额的 ISO 4217 alpha 3货币代码。
tax_calculation_reason_code	交易是应税、非应税、免税还是零税率，按管辖区级别划分。
date_used_for_tax_calculation	用于计算交易税费的日期。
customer_exemption_certificate_id	免税证明的证明 ID。
customer_exemption_certificate_id_domain	证书在 Amazon 系统中的存储位置。
customer_exemption_certificate_level	提供免税的管辖区级别。
customer_exemption_code	指定免税的代码，例如 RESALE。
customer_exemption_domain	用于捕获客户免税信息（如果有）的 Amazon 系统。
transaction_reference_id	允许您交叉引用以下报告中的数据的标识符： • 支付报告 • 月度账单收入报告 • 销售补偿报告

> **Note**
>
> 从 2021 年 8 月 5 日起，国际 Marketplace 服务商 AWS Marketplace 销售税将在纳税项目数据源中输入内容。这意味着，从 2021 年 8 月 5 日起，账单事件数据源中的每条 AWS TAX SELLER TAX _ SHARE SHARE 和 _ _ 记录都应在税项数据馈送中都有相应的记录。

税项数据源示例

下面显示了税项数据源的示例。在数据源中，此信息以单个表格形式显示。为了便于阅读，数据显示在此处的多个表中，并且不显示所有列。

tax_item_id	invoice_id	line_item_id	customer_bill_id
6p2ni6tu041xagvhbyanbgxl3xameha16txjoav_0001	781216640	71000000000000000000	22100000000000000000
6p2ni6tu041xagvhbyanbgxl3xameha16txjoav_0002	781216640	53000000000000000000	22100000000000000000
flr4jobxjzww8czdsrq4noue2uxd56j39wxw0k7_0001	250816266	76400000000000000000	57200000000000000000
gfkjjobxjzw56jgkrsrqgjtk52uxd56j39wgj567d_0002	280336288	76400000000000000000	57243900000000000000
wwk1qpvb8ran3geiw8e3mp6dgs2qj7wpkuwhgk1_0001	451431024	99300000000000000000	12300000000000000000
wwk1qpvb8ran3geiw8e3mp6dgs2qj7wpkuwhgk1_0002	451431024	99300000000000000000	31200000000000000000
fnohdid8kwgqq9lvii2k30spn3ftgwihbe8h75x_0001	229987654	92100000000000000000	63900000000000000000

tax_liable_party	transaction_type_code	product_id	product_tax_code	invoice_date
Seller	AWS	prod-o4grxfafcxxxx	AWSMP_SOFTWARE_RA	2018-12-31T00:00:00Z

tax_liable_party	transaction_type_code	product_id	product_tax_code	invoice_date
Seller	AWS	prod-o4grxfafcxxxx	AWSMP_SOFTWARE_RA	2018-12-31T00:00:00Z
Seller	AWS	prod-t3grxfafcxxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00:00:00Z
Seller	REFUND	prod-t3grxfafcxxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00:00:00Z
Seller	AWS	prod-x8faxxfafcxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00:00:00Z
Seller	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	美国	GA	MILTON	48573-4839

taxed_customer_account_id	taxed_customer_country	taxed_customer_state_or_region	taxed_customer_city	taxed_customer_postal_code
VIeGa2t9j3MuxioH9wc8lsndXXCgGCGUreeXriocM5	美国	GA	MILTON	48573-4839
7nyo5 plyx81vx9 ji04 O jwTRo 1Ff8 88W8 eEwTur biQi	美国	NC	DURHAM	27517-4834
7nyo5 plyx81vx9 ji04 O jwTRo 1Ff8 88W8 eEwTur biQi	美国	NC	DURHAM	27517-4834
7nyo5 plyx81vx9 ji04 O jwTRo 1Ff8 88W8 eEwTur biQi	美国	TX	NOT APPLICABLE	75844-1235
7nyo5 plyx81vx9 ji04 O jwTRo 1Ff8 88W8 eEwTur biQi	美国	TX	HOUSTON	75844-1235
192a0421313e41f069b52962ed7babf716291b688	美国	CT	NEW HAVEN	06002-2948

tax_location_code_taxed_jurisdiction	tax_type_code	jurisdiction_level	taxed_jurisdiction	display_price_taxability_type	taxable_amount	nontaxable_amount
460473664	Sales	State	GA	排除	100	0
66301164	Sales	County	FULTON	排除	0	100
692938178	SellerUse	状态	NC	排除	58.1	523.8
692938178	SellerUse	状态	NC	排除	-58.1	523.8
356794387	Sales	状态	TX	Exclusive	1105.14	0
528887443	Sales	城市	HOUSTON	排除	-36	0
171248162	Sales	状态	CT	排除	0	114.55

tax_jurisdication_rate	tax_amount	tax_currency	tax_calculation_reason_code	date_used_for_tax_calculation
0.206	20.6	USD	Taxable	2018-10-31T00:00:00Z
0	0	USD	NonTaxable	2018-10-31T00:00:00Z
0.1	5.8	USD	Taxable	2018-07-31T00:00:00Z
0.1	-5.8	USD	Taxable	2018-07-31T00:00:00Z
0.06	66.3	USD	Taxable	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	USD	NonTaxable	2018-07-31T00:00:00Z
0	0	USD	Exempt	2019-06-30T00:00:00Z

卖家报告

> **Important**
>
> 2024 年 8 月 30 日， AWS Marketplace 将停止使用几份报告和数据集：
>
> - Marketplace 旧版以逗号分隔 (csv) 卖家报告
> - 商务分析服务 (CAS) API
> - 关联的电子邮件通知
>
> 请查看报告页面是否有受影响的报告，或者如果您有任何疑问，请咨询在 2024 年 5 月 30 日收到通信的 AWS 管理员。

AWS Marketplace 提供包含产品使用情况、买家、账单和付款信息等信息的报告。所有注册 AWS Marketplace 卖家均可获得报告。

以下是有关报告生成的一些要点：

- 这些报告每日、每周或每月生成一次，具体取决于报告。
- 报告在 00:00 生成UTC，涵盖前一天的 24:00 UTC。
- 这些报告是作为 .csv 文件生成的。
- 您可以将 Amazon 配置SNS为在数据传输到加密的 S3 存储桶时通知您。配置通知后， AWS 将通知发送到与您在上注册的 AWS 帐户关联的电子邮件地址 AWS Marketplace。

有关如何配置通知的信息，请参阅《亚马逊简单通知服务开发者指南》SNS中的 “亚马逊入门”。

要取消接收通知电子邮件，请与 AWS Marketplace 卖家运营团队联系。

- 要了解每个报告，您可以下载示例报告。

访问报告

AWS Marketplace 提供了两种配置报告的方法：

- 使用接API口。AWS Marketplace 商务分析服务使您能够通过API界面自动访问报告中的数据。您可以自动提取信息并下载报告的一部分，而不是整个报告。该服务以异步方式将数据返回到亚马逊简单存储服务 (Amazon S3) 中的文件中，而不是像传统服务那样直接返回。API数据以机器可读格式传送，以便您可以将数据导入或加入到您的系统中。
- 使用 AWS Marketplace 管理门户 中的报告控制面板。此控制面板提供以前报告时段的报告。

您可以使用 AWS Identity and Access Management (IAM) 权限控制对报告的访问权限。

每日业务报告

每日业务报告提供以下方面的数据：，只有在有相关信息可用时，您才会收到此报告。

- AWS 买家每天如何使用您的商品
- 该使用产生的估计收入。

如果您没有收到此报告并认为应该收到此报告，请联系AWS Marketplace 卖家运营团队。如果您想取消报告的电子邮件通知，也可以联系该团队。

您可以通过 AWS Marketplace 管理门户 访问此报告。如果您已注册the section called “AWS Marketplace 商务分析服务”，则还可以使用访问您的报告 AWS SDK。

您可以对每个客户使用唯一标识符，以便随时间以及在不同的报告中标识客户。通过使用该标识符，您可以跟踪客户使用模式，以便估计客户支出以及深入了解免费试用和年度用量趋势。

发布计划

该报告于每日 00:00 UTC 发布，涵盖时间为前一天的 00:00 UTC 至 23:59 UTC。在每日业务报告部分的介绍中注明了计划的任何例外情况。

主题

- 第 1 部分：按实例类型列出的用量
- 第 2 部分：费用
- 第 3 部分：免费试用转换
- 第 4 部分：新实例
- 第 5 部分：新产品订阅用户
- 第 6 部分：取消的产品订阅用户

第 1 部分：按实例类型列出的用量

该部分为客户使用的每种实例类型列出一行数据。例如，如果客户在一种实例类型上使用一个产品，并在另一种实例类型上使用相同的产品，该报告为两种实例类型各包含一行。

列名称	描述
客户参考 ID	该唯一标识符并非账户 ID。它有助于跟踪客户的用量、收入和订阅。
用户状态	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。
用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
产品标题	产品的标题。
产品代码	产品的唯一标识符。
实例类型	与产品用量关联的实例类型：例如，t2.micro。
用量单位	客户在报告时段内使用的使用单位数。
用量单位类型	衡量客户用量的计量单位。例如，小时或天。
产品说明	产品的说明。例如，产品是按每小时的用量、免费试用或年度用量提供的。
估计收入	产品用量的估计收入。账单是在月末生成的。

列名称	描述
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
客户 AWS 账号	费用计入的账户的 ID。
客户国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

第 2 部分：费用

该部分包含与产品关联的基于费用的交易：例如，年度、月度、SaaS 合同产品费用和数据产品订阅费用。该部分中的数据涵盖生成报告之前 72 小时的 24 小时时段。例如，如果报告是在 5 月 24 日生成的，则数据涵盖 5 月 21 日的 24 小时时段。

列名称	描述
客户参考 ID	该唯一标识符并非账户 ID。它有助于跟踪客户的用量、收入和订阅。
用户状态	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。

列名称	描述
用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
产品标题	产品的标题。
产品代码	产品的唯一标识符。
使用额	使用费。如果具有退款，则该值为负数。如果此条目适用于 S AWS Marketplace aaS 合同，则该金额代表该维度的费用，而不是整个合同的费用。
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
费用描述	费用的原因：例如，月度费用、年度费用或退款。
客户 AWS 账号	费用计入的账户的 ID。
客户国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
客户省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
客户城市	与费用计入的账户关联的账单地址中的城市。
客户邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
客户电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 `liu-jie@example.com` ，则该条目为 `example.com` 。
开始日期	AWS Marketplace SaaS 合同或数据产品订阅的开始日期。

列名称	描述
结束日期	AWS Marketplace SaaS 合同或数据产品订阅的结束日期。
Quantity	合同指定的维度的单位数。
维度	合同指定的维度。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

第 3 部分：免费试用转换

该部分列出免费试用开始、转换和取消的数据，并涵盖之前的 24 小时时段。

列名称	描述
产品标题	产品的标题。
产品代码	表示产品的唯一标识符。
新免费试用	在报告时段内启动的新免费试用数。
当前免费试用总数	有效的免费试用订阅总数。
转换的免费试用	在报告时段内从免费试用转换为付费使用的订阅总数。
未转换的免费试用	结束免费试用但未转换为付费使用的订阅总数。
解决方案标题	解决方案的名称。

列名称	描述
解决方案 ID	解决方案的唯一标识符。

第 4 部分：新实例

本节列出了新EC2实例和实例类型的数据，并涵盖了之前的 24 小时时段。

列名称	描述
客户参考 ID	该唯一标识符并非账户 ID。它有助于跟踪客户的用量、收入和订阅。
用户状态	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。
用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
产品标题	产品的标题。
产品代码	产品的唯一标识符。
类型	Amazon EC2 实例类型。
计数	EC2实例的数量。
客户 AWS 账号	费用计入的账户的 ID。
客户国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
客户省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
客户城市	与费用计入的账户关联的账单地址中的城市。
客户邮政编码	与费用计入的账户关联的账单地址中的邮政编码。

列名称	描述
客户电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 `liu-jie@example.com` ，则该条目为 `example.com` 。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

第 5 部分：新产品订阅用户

该部分列出新买家的数据，并涵盖之前的 24 小时时段。

列名称	描述
客户参考 ID	该唯一标识符并非账户 ID。它有助于跟踪客户的用量、收入和订阅。
用户状态	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。
用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
产品标题	产品的标题。
产品代码	产品的唯一标识符。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。

列名称	描述
客户国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
客户省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
客户城市	与费用计入的账户关联的账单地址中的城市。
客户邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
客户电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 `liu-jie@example.com` ，则该条目为 `example.com` 。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

第 6 部分：取消的产品订阅用户

该部分列出买家取消的数据，并涵盖之前的 24 小时时段。

列名称	描述
客户参考 ID	该唯一标识符并非账户 ID。它有助于跟踪客户的用量、收入和订阅。
用户状态	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。

列名称	描述
用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
产品标题	产品的标题。
产品代码	产品的唯一标识符。
Subscribed Date	订阅的启动日期。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
客户 AWS 账号	费用计入的账户的 ID。
客户国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
客户省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
客户城市	与费用计入的账户关联的账单地址中的城市。
客户邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
客户电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 `liu-jie@example.com` ，则该条目为 `example.com` 。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

每日客户订阅用户报告

此报告列出了购买您产品的客户的数据。该报告不指定当前或过去的用量，仅指定客户订阅了您的产品。只有在具有相关信息时，您才会收到该报告。如果您没有收到此报告并认为应该收到此报告，请联系AWS Marketplace 卖家运营团队。您还可以联系 AWS Marketplace 卖家运营团队，取消此报告的电子邮件通知。

您可以通过 AWS Marketplace 管理门户 访问此报告。如果您已注册the section called “AWS Marketplace 商务分析服务”，则还可以使用访问您的报告 AWS SDK。

该报告包含两个部分：一个用于每小时和每月订阅，另一个用于年度订阅。该报告包括订阅您的商品的所有买家的 AWS 账户IDs列表。

发布计划

该报告于每日 00:00 UTC 发布，涵盖时间为前一天的 00:00 UTC 至 23:59 UTC。

主题

- 第 1 部分：每小时和每月订阅
- 第 2 部分：可变长度订阅

第 1 部分：每小时和每月订阅

本部分列出了截至前一天 23:59:59 的所有基于使用量的订阅的数据。UTC

列名称	描述
客户 AWS 账号	订阅产品的账户。
产品标题	产品的标题。
产品 ID	软件产品的唯一标识符。
产品代码	软件产品的唯一标识符。
订阅开始日期	订阅的开始日期，格式为 `YYYY-MM-DD` 。
优惠 ID	买家签订的优惠的标识符。

列名称	描述
优惠可见性	优惠是公开、专属还是企业合同优惠。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
经销商账户 ID	渠道合作伙伴经销商的唯一标识符。
经销商账户名称	渠道合作伙伴经销商的名称。

第 2 部分：可变长度订阅

本部分列出了截至前一天 23:59:5 UTC 9 的所有收费订阅的数据。

列名称	描述
客户 AWS 账号	订阅产品的账户的 ID。
产品标题	产品的标题。
产品 ID	软件产品的唯一标识符。
产品代码	软件产品的唯一标识符。此信息也作为 Amazon EC2 实例元数据的一部分提供。
订阅 ID	订阅的 ID。
订阅数量	客户购买的许可证总数。
订阅类型	订阅的类型。
订阅意图	此优惠是早期优惠的升级还是续订。
优惠 ID	买家签订的优惠的标识符。

列名称	描述
订阅开始日期	客户订阅产品的日期，格式为 YYYY-MM-DD 。
以前的优惠 ID	升级或续订优惠之前的优惠 ID（如果存在）。
优惠可见性	优惠是公开、专属还是企业合同优惠。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
经销商账户 ID	渠道合作伙伴经销商的唯一标识符。
经销商账户名称	渠道合作伙伴经销商的名称。

支付报告

> **Important**
>
> 该报告将于 2024 年 8 月 30 日发布。你可以使用以下替代方案：
>
> - 收款和付款控制面板
> - 收款_和_支出_报告查询。 SQL GitHub

支付报告提供有关我们在上次支付后收取并向您的银行账户支付的资金的信息。支付可能包括您的产品订阅的客户付款或退款，以及向客户收取或退还的一些税费。从客户收取资金后，您才会收到支付的资金。不同的客户有不同的付款条件 AWS，因此每个未收款年龄段的部分资金可能不是客户应付的。

退款显示为负数，因为在您批准退款后这笔款项退还给您的客户。

此报告可在 “报告” 选项卡 AWS Marketplace 管理门户 下找到。要创建与该报告类似的自己的报告 AWS Marketplace 卖家传输数据源服务，您可以使用（包括按产品划分的支出作为基础报告）进行自定义以满足您的需求。

> **Note**
>
> 要取消此报告的通知电子邮件，请联系AWS Marketplace 卖家运营团队。

发布计划

在开始将资金转入您的银行账户 3 到 5 天后，将发布此报告。通常，这是针对按月收到付款的卖家的报告。如果未启动支付，则不会生成任何支付报告。

主题

- 第 1 部分：按产品列出的支付金额
- 第 2 部分：按客户地理位置列出的支付金额
- 第 3 部分：按实例小时数列出的支付金额
- 第 4 部分：未收取的资金期限
- 第 5 部分：支付的资金期限
- 第 6 部分：逾期资金期限
- 第 7 部分：未收款项明细

第 1 部分：按产品列出的支付金额

该部分按产品列出支付的数据。

列名称	描述
产品	产品的标题。
产品代码	产品的唯一标识符。
SellerRev	向客户计费的产品用量或费用的金额。
AWSRefFee	AWS Marketplace 费用金额。
SellerRevRefund	如果在数据涵盖时段内处理了任何退款，则为向客户退还的订阅成本金额。

列名称	描述
AWSRefFeeRefund	如果在数据覆盖期内处理了任何退款，则退还的 AWS Marketplace 费用金额。
SellerRevCredit	AWS 存 AWS Marketplace 入客户账户的积分。
AWSRefFeeCredit	存 AWS Marketplace 入您账户的 AWS 积分。
Net Amount	我们向您支付的资金总额。此列等于该 SellerRev列减去该 AWSRefFee 列。向买家退款时，此列为负数，等于该 SellerRevRefund 列减去该 AWSRefFeeRefund列。
交易参考 ID	交易的唯一标识符，可帮助您关联 AWS Marketplace 报表中的交易。
S ellerUSSales 税	为该交易计费的美国销售和使用税总额。
S ellerUSSales TaxRefund	如果处理了退款，则为该交易退还的美国销售和使用税总额。
客户 AWS 账号	费用计入的账户的 ID。
客户国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
客户省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
客户城市	与费用计入的账户关联的账单地址中的城市。
客户邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
客户电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 `liu-jie@example.com` ，则该条目为 `example.com` 。
解决方案标题	解决方案的名称。

列名称	描述
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

第 2 部分：按客户地理位置列出的支付金额

该部分按客户地理位置列出支付的数据。

列名称	描述
结算 ID	支付的唯一标识符。
结算期间开始日期	支付周期的起始日期和时间。
结算期间结束日期	支付周期的结束日期和时间。
存入日期	进行支付的日期和时间。
已支付金额	支付总额。
国家/地区代码	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
省/自治区/直辖市或区域	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
城市	与费用计入的账户关联的账单地址中的城市。
邮政编码	与软件费用计入的账户关联的账单地址中的邮政编码。

列名称	描述
按纳税地点的净额	按税务位置向卖家支付的资金总额减去 AWS Marketplace 费用、退款以及美国销售和使用税。
按纳税地点的总金额	按税务位置向卖家支付的资金总额。
卖家美国销售税	代表卖家为该交易计费的美国销售和使用税总额。（也就是说，美国销售和税务报告中的相关记录显示 “纳税方” == “SELLER”。）
卖家美国销售税退税	当代表卖家收取了此类税款时，如果已处理退款，则为该交易退还的美国销售和使用税总额。（也就是说，美国销售和税务报告中的相关记录显示 “纳税方” == “SELLER”。）

第 3 部分：按实例小时数列出的支付金额

本部分按亚马逊EC2实例工时列出了支付数据。

列名称	描述
产品	产品的标题。
产品代码	产品的唯一标识符。
用量类型描述	用量说明，包括产品类型、区域和实例类型。
费率	产品类型、区域和实例类型的每小时费率。
用户数	使用产品类型、区域和实例类型的唯一客户数。
实例小时数	实例针对产品类型、区域和实例类型使用的小时数。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。

第 4 部分：未收取的资金期限

该部分列出未收取的资金的数据，按期限划分。未收取的资金可能包括尚未到期的金额。

列名称	描述
未收取（< 31 天待处理）	已计费但未收取的时间少于 31 天的资金总额。
未收取（31-60 天待处理）	已计费但未收取的时间为 31–60 天的资金总额。
未收取（61-90 天待处理）	已计费但未收取的时间为 61-90 天的资金总额。
未收取（91-120 天待处理）	已计费但未收取的时间为 91-120 天的资金总额。
未收取（> 120 天待处理）	已计费但未收取的时间超过 120 天的资金总额。
未收取（总计）	所有已计费但未收取的资金总额。

第 5 部分：支付的资金期限

该部分列出在上次支付后收取的资金的数据。

列名称	描述
已收取（< 31 天待处理）	在 0–31 天范围内计费并收取的资金总额。
已收取（31-60 天待处理）	在 31-60 天范围内计费并收取的资金总额。
已收取（61-90 天待处理）	在 61-90 天范围内计费并收取的资金总额。
已收取（91-120 天待处理）	在 91-120 天范围内计费并收取的资金总额。
已收取（> 120 天待处理）	在超过 120 天的时间内计费并收取的资金总额。
已收取（总计）	所有收取的资金总额。

第 6 部分：逾期资金期限

该部分列出已产生并且客户可支付但未按照客户与 AWS签订的协议进行支付的资金的数据。

列名称	描述
逾期（< 31 天）	过去 0–31 天产生并到期但客户未支付的资金总额。
逾期（31-60 天）	过去 31-60 天产生并到期但客户未支付的资金总额。
逾期（61-90 天）	过去 61-90 天产生并到期但客户未支付的资金总额。
逾期（91-120 天）	过去 91-120 天产生并到期但客户未支付的资金总额。
逾期（> 120 天）	过去 121 天或更长时间产生并到期但客户未支付的资金总额。
逾期（总计）	已产生并到期但客户未支付的资金总额。

第 7 部分：未收款项明细

此部分列出了所有未收款项（按付款到期日排序）。

列名称	描述
付款人AWS账号	软件费用计入的账户。
产品代码	产品的唯一标识符。
总收入	使用产品的计费金额，或使用产品的费用。
AWS 收入份额	结算时AWS从账单金额中扣除的费用金额。
总退款	交易的任何退款总额。

列名称	描述
AWS 退款份额	为交易退还的AWS费用部分。
净收入	此交易的账单净额减去AWS费用、退款以及美国销售和使用税。
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
AR 期间	交易的月份和年份，格式为 YYYY-MM。
交易参考 ID	代表交易的唯一标识符，可用于关联 AWS Marketplace 报表中的交易。
机会名称	已注册的机会的唯一标识符。
机会描述	注册的机会中的任何元数据。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。
付款截止日期	格式为 YYYY-MM-DD 的付款截止日期。

月度账单收入报告

> **Important**
>
> 本报告的三个部分将于 2024 年 8 月 30 日发布。您可以在第 1-3 节中使用以下替代方案：
>
> - 账单收入控制面板
> - billed_revenue_report 查询SQL开启。 GitHub

第 4 节：具有灵活付款时间表的合同将在 2024 年 8 月 30 日之后仍然有效。

每月账单收入报告为您提供有关每月账单收入的权威信息，以用于会计和其他财务报告用途。该报告显示了向客户收取产品的每小时、每年或每月使用 AWS 费账单的总金额。该报告包含 4 个部分：小时用量和月度费用、可变长度订阅、现场演示用量和灵活付款的账单金额。

> **Important**
>
> 该报告中的金额仅反映向客户计费的收入，而不反映我们收取的金额。

此报告可在 “报告” 选项卡 AWS Marketplace 管理门户 下找到。如果您注册了 AWS Marketplace 商务分析服务，则可以使用API调用来下拉此报告的各个部分。有关更多信息，请参阅 the section called “AWS Marketplace 商务分析服务”。

> **Note**
>
> 要取消此报告的通知电子邮件，请联系AWS Marketplace 卖家运营团队。

发布计划

该报告每月在每月的第十五天 00:00 UTC 发布。

账单和收入数据涵盖当月 15 日至下月 14日。

长期订阅涵盖该月的第一天 00:00 UTC 至该月的最后一天，即上个月的 23:59 UTC。

例如，5月15日发布的报告涵盖了4月1日 00:00 UTC 至4月30日 23 UTC :59。

主题

- 第 1 部分：账单和收入数据
- 第 2 部分：可变长度订阅
- 第 3 部分： AWS 现场演示用法
- 第 4 部分：具有灵活付款计划的合同

第 1 部分：账单和收入数据

该部分列出用量账单、退款、费用以及收取的美国销售和使用税的数据。

列名称	描述
客户参考 ID	该唯一标识符并非账户 ID。它有助于跟踪客户的用量、收入和订阅。
国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
城市	与费用计入的账户关联的账单地址中的城市。
邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
产品标题	产品的标题。
产品代码	产品的唯一标识符。
客户账单金额	向客户计费的产品用量或月度费用的金额。
AWS 上市费	要从账单金额中扣除的 AWS Marketplace 费用金额。
退款金额	如果在数据涵盖时段内处理了任何退款，则为向客户退还的订阅成本总额。
AWS 费用退款	如果在数据覆盖期内处理了任何退款，则退还的部分 AWS Marketplace 费用。
费用	经销商的产品成本：例如，经销商在销售您的产品时向您支付的费用。
合作伙伴收入金额	扣除 AWS Marketplace 费用、退款以及美国销售和使用税后的总账单金额。

列名称	描述
货币	交易的货币。例如，如果交易是以美元达成的，则该条目为 USD。
交易参考 ID	交易的唯一标识符，可帮助您关联 AWS Marketplace 报表中的交易。
美国销售税客户账单金额	代表卖家为该交易计费的美国销售和使用税总额。（也就是说，美国销售和税务报告中的相关记录显示 “纳税方” == “SELLER”。）
美国销售税退税金额	当代表卖家收取了此类税款时，如果已处理退款，则为该交易退还的美国销售和使用税总额。（也就是说，美国销售和税务报告中的相关记录显示 “纳税方” == “SELLER”。）
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
客户 AWS 账号	费用计入的账户的 ID。
客户电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 *liu-jie@example.com* ，条目是`example.com` 。
机会名称	已注册的机会的唯一标识符。
机会描述	注册的机会的元数据。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

第 2 部分：可变长度订阅

该部分列出基于费用的收费数据。

列名称	描述
客户参考 ID	该唯一标识符并非账户 ID。它有助于跟踪客户的用量、收入和订阅。
国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
城市	与费用计入的账户关联的账单地址中的城市。
邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
产品标题	产品的标题。
产品代码	产品的唯一标识符。
订阅数量	作为可变长度订阅购买一部分指定的总许可证数。
订阅开始日期	可变长度订阅购买的开始日期。
订阅结束日期	可变长度订阅购买的结束日期。
订阅实例类型	与可变长度订阅购买关联的实例类型。
客户账单金额	为用量和/或月度费用计费的金额。
AWS 上市费	从账单金额中扣除的 AWS Marketplace 费用金额。
退款金额	如果在数据涵盖时段内处理了任何退款，则为向客户退还的总金额。

列名称	描述
AWS 费用退款	如果在数据覆盖期内处理了任何退款，则退还的部分 AWS Marketplace 费用。
费用	经销商的产品成本：例如，经销商在销售您的产品时向您支付的费用。
合作伙伴收入金额	此笔交易的账单总额，扣除 AWS Marketplace 费用、退款以及美国销售和使用税。
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
交易参考 ID	交易的唯一标识符，可帮助您关联 AWS Marketplace 报表中的交易。
美国销售税客户账单金额	代表卖家为该交易计费的美国销售和使用税总额。（也就是说，美国销售和税务报告中的相关记录显示 “纳税方” == “SELLER”。）
美国销售税退税金额	当代表卖家收取了此类税款时，如果已处理退款，则为该交易退还的美国销售和使用税总额。（也就是说，美国销售和税务报告中的相关记录显示 “纳税方” == “SELLER”。）
客户 AWS 账号	费用计入的账户的 ID。
客户电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 `liu-jie@example.com` ，则该条目为 `example.com` 。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
合同开始日期	AWS Marketplace SaaS 合同的开始日期。
合同结束日期	AWS Marketplace SaaS 合同的结束日期。

列名称	描述
机会名称	已注册的机会的唯一标识符。
机会描述	注册的机会的元数据。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

第 3 部分： AWS 现场演示用法

该部分列出了产品 AWS 现场演示使用情况的数据。您可以配置您的产品，以允许我们向潜在客户演示您的产品。此处列出了演示中的任何用量。

列名称	描述
产品标题	产品的标题。
产品代码	产品的唯一标识符。
实例类型	与现场演示关联的 Amazon EC2 实例类型。
用量单位	与产品关联的使用单位数。
用量单位类型	与使用单位数关联的使用单位：例如，小时。

第 4 部分：具有灵活付款计划的合同

该部分列出您在上一个报告时段使用灵活付款计划创建的所有合同的数据。

列名称	描述
客户 AWS 账号	费用计入的付款人账户的 ID。
客户国家/地区	与费用计入的付款人账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
客户省/自治区/直辖市	与费用计入的付款人账户关联的账单地址中的省/自治区/直辖市。
客户城市	与费用计入的付款人账户关联的账单地址中的城市。
客户ZIP代码	与费用计入的付款人账户关联的账单地址中的邮政编码。
客户电子邮件域	与费用计入的付款人账户关联的电子邮件域。例如，如果电子邮件地址是 *liu-jie@example.com* ，条目是example.com 。
用户参考 ID	费用计入的付款人账户。
用户 AWS 账号	订阅产品的账户的 ID。
产品 ID	产品的唯一标识符。
产品标题	产品的标题。
产品类型	产品的类型。
AWS Marketplace 优惠编号	买家签订的优惠的标识符。
合同创建日期	合同创建日期，这是账户订阅产品的日期。
合同到期日期	合同的到期日期。
合约总价值 (USD)	合约的总价值USD。
付款次数 #	为合同计划的付款次数。

列名称	描述
发票日期	创建发票的日期。
发票金额 (USD)	发票上的账单金额。USD
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。

销售补偿报告

> **Important**
>
> 该报告将于 2024 年 8 月 30 日发布。你可以使用以下替代方案：
>
> - 账单收入控制面板
> - billed_revenue_report 查询SQL开启。 GitHub

此报告列出每月账单收入以及不在标准 the section called “月度账单收入报告” 中的其他客户信息。该报告列出了向客户收取产品的每小时、每年或每月使用 AWS 费账单的总金额。

> **Important**
>
> 此报告中的金额仅反映向客户计费的收入，而不反映从客户收取的金额。

根据我们与您签订的保密协议，或者如果不存在此类协议，则根据 AWS Marketplace 卖家的条款和条件，销售报酬报告和作为本计划的一部分与您共享的信息构成亚马逊的机密信息。您只能将此信息用于通过按公司名称、地理位置和 AWS 账户 ID 将 AWS Marketplace 收入映射到销售代表来补偿销售代表。您可以与需要了解此信息的员工分享此信息，以了解应付给他们的佣金的来源。您对此类信息的使用和共享必须遵守我们与您签订的保密协议中的义务以及针对 AWS Marketplace 卖家的条款和条件，包括但不限于 AWS Marketplace 卖家条款和条件的第 3.8 节。

要创建与该报告类似的自己的报告 AWS Marketplace 卖家传输数据源服务，您可以使用（包括销售补偿报告作为基础报告）进行自定义以满足您的需求。

发布计划

该报告每月发布一次，在每月的第十五天 00:00 UTC。该报告涵盖上一个日历月，从该月最后一个日UTC历日的 00:00 到 23:59 UTC。例如，5月15日发布的报告涵盖了4月1日 00:00 UTC 至4月30日 23:5 UTC 9。

账单收入

此报告的账单收入部分包括上个日历月的基于用量和基于费用的费用。以下是列名称和说明。

> **Note**
>
> 在此报告中，列示费用是根据分层列示费用确定的交易收益（授权转售商转售的授权转售产品的交易收益除外）的百分比。

列名称	描述
客户 AWS 账号	费用计入的账户。
国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。此报告使用 ISO 3166-1 alpha-2 标准。
省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
城市	与费用计入的账户关联的账单地址中的城市。
邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 `liu-jie@example.com` ，则该条目为 `example.com` 。
产品代码	产品的唯一标识符。
产品标题	产品的标题。
总收入	使用产品的计费金额，或使用产品的月度费用。

列名称	描述
AWS 收入分成	结算时 AWS 从账单金额中扣除的费用金额。它显示在 the section called “支付报告” 中。
总退款	交易的任何退款总额。
AWS 退款分享	为交易退还的 AWS 费用部分。
净收入	此次交易的账单净额减去 AWS 费用、退款以及美国销售和使用税。
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
AR 期间	交易的月份和年份，格式为 YYYY-MM。
交易参考 ID	代表交易的唯一标识符，可用于关联 AWS Marketplace 报表间的交易。
机会名称	已注册的机会的唯一标识符。
机会描述	注册的机会中的任何元数据。
解决方案标题	解决方案的名称。
解决方案 ID	解决方案的唯一标识符。
付款人参考 ID	该唯一标识符并非账户 ID。它与费用计入的账户关联。它有助于跟踪所有 AWS Marketplace 财务报告中客户的使用情况、收入和订阅情况。
付款人地址 ID	表示客户地址的唯一标识符。

补充报告

AWS Marketplace 通过卖家拥有的 卖家传输数据源服务 Amazon S3 账户提供补充报告，这些账户关联到与 AWS Marketplace 卖家 AWS Marketplace 商品相关的卖家账户编号。有关更多信息，请参阅创建目标 Amazon S3 存储桶。

如果前一天有新订阅用户，则补充报告将在每天 16:00 UTC 发布。这些报告涵盖了前一天 13:59 UTC 至次日 UTC 16:01。

协议详细信息报告

协议详细信息报告可帮助您向使用软件即服务 (SaaS) 免费试用的客户提供支持。该报告包括协议详细信息，例如订阅用户名称、订阅用户 ID、优惠 ID、协议开始和协议结束日期。

只有在具有相关信息时，您才会收到该报告。如果您认为自己应该收到该报告但却没有收到，请与 AWS Marketplace 卖家运营团队联系。

您可以通过与 AWS Marketplace 卖家账户编号关联的 Amazon S3 存储桶访问此报告。

下表列出了协议详细信息报告的列名和描述。

SaaS 合同免费试用报告数据

名称	描述
vendor_display_name	销售产品的供应商名称。
vendor_aws_account_id	与销售产品的供应商关联的标识。
subscriber_aws_account_id	与订阅产品的人相关的标识。 AWS 账户
customer_id	软件产品的唯一标识符。
product_title	产品的标题。
offer_id	买家签订的优惠的标识符。
offer_visibility	优惠是公开、专属还是企业合同优惠。
reseller_name	渠道合作伙伴经销商的名称。
reseller_aws_account_id	渠道合作伙伴经销商的唯一标识符。
agreement_id	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
agreement_acceptance_date	协议被接受的日期。
agreement_start_date	协议的开始日期。

名称	描述
agreement_end_date	协议的结束日期。对于计量/即用即付/订阅，该值设置为 1-JAN-9999。
is_free_trial_offer	表示优惠或协议是否为免费试用优惠的标志。
is_upgraded_after_free_trial	表示协议是否已升级为付费合同的标志。
total_contract_value	合同的总价值。

卖家控制面板

AWS Marketplace 提供由 Amazon 支持的控制面板， QuickSight其中包含图表、图表和见解，可帮助您访问和分析财务和销售数据。卖家控制面板包括：

the section called “财务运营控制面板”

- the section called “账单收入控制面板” – 提供有关账单收入的信息，用于会计和其他财务报告目的。
- the section called “收取和支付控制面板”— 提供有关自上次付款以来 AWS 收款并支付到您的银行账户的资金的信息。
- the section called “税务控制面板”— 提供有关卖家交易税费的信息。

the section called “销售运营控制面板”

- the section called “协议和续订控制面板” – 在 AWS Marketplace中签署协议后的 24 小时内提供有关协议和续订的信息。
- the section called “使用情况控制面板” – 为使用 SaaS 和基于服务器使用情况的产品的客户提供可视化和精细数据。

控制面板可供拥有相应权限的 AWS Marketplace 卖家使用。

访问控制面板

默认情况下，卖家账户的 AWS Marketplace 系统管理员可以访问中 “Insights” 选项卡上的所有控制面板 AWS Marketplace 管理门户。系统管理员可以创建 AWS Identity and Access Management (IAM) 策略，为卖方公司的其他用户提供对特定仪表板的访问权限。

> **Note**
>
> 2023 年 9 月，我们将不再支持通过传统IAM权限访问卖家控制面板。使用以下代码示例中的新 Amazon 资源名称 (ARN) 格式更新您的IAM权限。

有关创建策略的信息，请参阅创建IAM策略。

控制面板策略

使用以下策略之一提供对账单收入控制面板以及收取和支付控制面板的访问权限。

您可以使用以下代码示例，根据当前和未来的数据 Feed 提供对当前和未来 AWS Marketplace 资源（包括仪表板和报告）的访问权限：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/*",
    ]
  }]
}
```

或者，您可以通过添加特定ARN部分来提供对一个或多个仪表板的访问权限，如以下代码示例所示。例如，要仅提供对账单收入控制面板、协议和续订控制面板以及使用情况控制面板的访问权限，请从以下代码示例中删除此行：`arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/BillingEvent_V1/Dashboard/CollectionsAndDisbursements_V1`

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

```
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/
BillingEvent_V1/Dashboard/BilledRevenue_V1",
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/
BillingEvent_V1/Dashboard/CollectionsAndDisbursements_V1",
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/
Agreement_V1/Dashboard/AgreementsAndRenewals_V1"
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/Usage_V1/
Dashboard/Usage_V1"
      "arn:aws:aws-marketplace::<awsAccountID>:AWSMarketplace/ReportingData/TaxItem_V1/
Dashboard/Tax_V1"
    ]
  }]
}
```

> **Note**
>
> 有关创建 AWS Identity and Access Management (IAM) 策略的信息，请参阅《AWS Identity and Access Management 用户指南》中的创建IAM策略。

财务运营控制面板

AWS Marketplace 提供多个仪表板来帮助您跟踪财务数据。

主题

- 账单收入控制面板
- 收取和支付控制面板
- 税务控制面板

账单收入控制面板

账单收入控制面板提供有关 AWS Marketplace中所有账单销售额的信息。该报告预计将节省时间，并且比传统的月度账单收入报告大约早 45 天发布。旧版报告在每个月的第 15 天发布，这会延迟对前一个月账单的可见性。

您可以导出和下载数据（以.csv 格式或 Microsoft Excel 文件）来自 Amazon QuickSight 控制面板上的任何图表、图表和表格。有关更多信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

有关运营和财务流程，请查看以下主题。

主题

- 账单收入控制面板的刷新频率
- 第 1 部分：控件
- 第 2 部分：发票日期范围
- 第 3 部分：指标
- 第 4 部分：趋势
- 第 5 部分：细分
- 第 6 部分：精细数据

账单收入控制面板的刷新频率

仪表板每天午夜更新UTC。如果发票是在5月10日午夜之前创建的UTC，则5月11日的更新将显示该发票。如果发票是在5月10日午夜之后创建的UTC，则5月12日的更新将显示该发票。如果从上游系统收到的最新发票或付款数据延迟，则最新数据可能会延迟 1-2 天才能反映在控制面板上。

如果您没有看到预期的客户发票，请在联系您的 AWS Marketplace 业务发展联系人之前按照以下步骤操作。

解决账单收入控制面板缺失的客户发票问题

1. 使用以下一种或多种工具确认买家已接受优惠：
   - 每日客户订阅用户报告
   - AWS Marketplace 事件通知
   - AWS Marketplace 管理门户 “协议” 选项卡
2. （对于专属优惠）在您确认客户接受优惠后，请查看优惠发票清单和金额：
   - 如需了解私人优惠，请查看 AWS Marketplace 管理门户 优惠选项卡。
   - 对于渠道合作伙伴的私有优惠，请查看 AWS Marketplace 管理门户 合作伙伴选项卡，查看合作伙伴机会中的自定义付款时间表。
3. 考虑一下交易是否采用用量定价模式，即在每个月的第二天和第三天向客户收取上个月的使用费用。

第 1 部分：控件

控制面板的这一部分提供了筛选条件，用于调整您的账单收入控制面板数据。例如，您可以从AWS Marketplace 事件通知的字段上选择筛选条件，以确认特定客户账户 ID、订阅者公司名称或优惠 ID 的账单。您也可以向分析添加筛选条件，例如您想要包含在视觉对象中的日期范围。在控件中选择的筛选条件会更新指标、趋势、细分和精细数据部分中显示的数据。

有关筛选的更多信息，请参阅《亚马逊 QuickSight 用户指南》 QuickSight中的在亚马逊上筛选数据。

控件描述

控件名称	描述
订阅者 AWS 账户 ID	订阅产品的账户的 ID。
订阅用户公司名称	订阅产品的账户的名称。
产品标题	产品的标题。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
AWS记录在案的卖家	为交易提供便利的商业实体的标识符。可能的值包括： • AWS_INC： AWS, Inc.（总部位于美国）的标识符。 • AWS_EUROPE：亚马逊 Web Services EMEASARL（总部设在卢森堡）的标识符。 • AWS_AUSTRALIA：Amazon Web Services Australia Pty Ltd 的标识符 • AWS_JAPAN：日本 Amazon Web Services G.K. 的标识符
付款人身份证 AWS 账户	费用计入的账户的 ID。

控件名称	描述
付款人公司名称	费用计入的企业名称。
经销公司	授权销售软件制造商产品的经销商账户的公司名称。
经销商 ID AWS 账户	从中批发购买产品或服务以转售ISV给客户的账户的 ID。
转售授权 ID	已注册的机会的唯一标识符。
转售授权名称	已注册的机会的唯一名称。
订阅用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。
订阅用户省/自治区/直辖市或区域	与订阅产品的账户关联的账单地址中的省/自治区/直辖市或区域。
交易参考 ID	交易的唯一标识符，可帮助您关联 AWS Marketplace 旧版报告中的交易。

第 2 部分：发票日期范围

控制面板的此部分提供了筛选条件，可根据账单日期是在指定日期之前还是之后还是在某个日期范围内来细化记录。在发票日期范围内选择的筛选条件会更新指标、趋势、细分和精细数据部分中显示的数据。默认发票日期范围设置为拉取最近六个月的数据。

第 3 部分：指标

仪表板的此部分显示关键绩效指标 (KPI)，用于可视化关键账单收入数字的比较。系统会显示指定发票日期范围内的总收入、退款总额、上架费、批发成本（如果适用）和卖家净收入。KPI您可以通过更新选“发票日期范围”字段中的日期标准来更新日期范围。

第 4 部分：趋势

控制面板的此部分提供指定日期范围内的账单收入趋势。您可以按指定日期聚合（例如日、、或 year-over-year）查看趋势 month-over-month quarter-over-quarter，以深入了解账单收入。计费收入趋势信息按汇总层面或按优惠可见性类型提供：

- 账单趋势 – 提供日期聚合筛选条件中选定日期范围内的总收入、卖家净收入、批发成本（如适用）和退款的快照。
- 按产品/服务可见性划分的计费趋势 – 按优惠可见性类型提供一段时间内专属优惠、公开优惠（或自助服务）和企业计划的优惠数量和总收入的快照。

第 5 部分：细分

控制面板的此部分为您提供有关您的业务的关键指标，包括订阅用户、记录卖家、订阅用户地理位置和产品标题。您可以按总收入、付款人数、订阅用户数量、退款总额、上架费、卖家净收入和批发成本进行筛选。

第 6 部分：精细数据

控制面板的此部分显示所有已计费的销售额，包括向客户收取产品的每小时、每年或每月使用 AWS 费账单的总金额。 AWS 使用以下三个频率向客户开具账单：

- 接受订阅日期（预付账单）
- 自定义付款计划（由灵活付款计划程序创建的专属优惠）
- 在当月第二天和第三天对上月用量计量

> **Note**
>
> 2021 年 4 月 1 日之前创建的发票可能没有关联的协议编号、优惠编号、订阅者 AWS 账户 ID 或订阅者公司名称。

在向客户开具账单后 24 小时，客户费用将显示在此精细数据部分中。例如，如果客户在 11 月 3 日被收费，那么控制面板将在 11 月 4 日报告这张发票。有关如何从 QuickSight 表中导出和下载数据的更多信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

精细数据描述

列	描述
发票日期	向客户收取产品订阅费用的日期。
付款截止日期	付款到期日采用 YYYY-MM-DD 格式。

列	描述
付款条款	客户的发 AWS票付款条款。
发票 ID	为向客户收取的费用分配的 AWS ID。
上架费发票 ID	当通过AWSEMEASARL日本或澳大利亚的法人实体（登记在案的卖家）进行 AWS Marketplace 订阅时，销售运营商必须向卖家收取卖家上市费的增值税 (VAT)，并提供符合税收要求的发票。对于适用的交易，对上市费进行VAT评估的发票编号与软件或产品订阅发票编号不同。来自AWS, Inc. 的交易的值为 “missing_listing_fee_invoice_ID”，因为上市费发票编号不适用。
订阅用户公司名称	订阅产品的账户的名称。
订阅者 AWS 账户 ID	订阅产品的账户的 ID。
订阅用户电子邮件域	与订阅产品的账户关联的电子邮件域。例如，如果电子邮件地址是 liu-jie@example.com，则该条目为 example.com。
订阅用户城市	与订阅产品的账户关联的账单地址中的城市。
订阅用户省/自治区/直辖市或区域	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。
订阅用户国家/地区	与订阅产品的账户关联的账单地址中的国家/地区。
订阅用户邮政编码	与订阅产品的账户关联的账单地址中的邮政编码。
产品标题	产品的标题。
优惠名称	卖家定义的优惠名称。
优惠 ID	买家签订的优惠的标识符。

列	描述
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
协议开始日期	客户开始订阅产品的日期，格式为 MM-DD-。YYYY如果这是未来日期的协议，则该日期可能与接受日期不同。
协议接受日期	客户订阅产品的日期，格式为 MM-DD-。YYYY
协议结束日期	合同到期的日期，格式为 MM-DD-。YYYY对于计量/订pay-as-you-go 阅，此日期设置为 -1-9999。JAN
使用期结束日期	产品使用期的结束日期。
使用期开始日期	产品使用期的开始日期。
支付状态	与发票关联的状态，用于确认自上次付款以来AWS已向您的银行账户收款并支付了款项。相关发票的已支付资金已收取并支付。相关发票的未支付资金未收取也未支付。
支付日期	AWS 开始向卖家银行付款的日期。
支付银行跟踪 ID	对于支付，跟踪 ID 由银行分配。付款银行追踪编号可用于将卖家银行提供的存款通知和报告与报告中的发票关联起来。 AWS Marketplace
总收入	向客户计费的产品用量或月度费用的金额。
总退款	如果在数据涵盖时段内处理了任何退款，则为向客户退还的订阅成本总额。
上架费用	要从账单金额中扣除的 AWS Marketplace 费用金额。

列	描述
上架费用退款	如果在数据覆盖期内处理了任何退款，则退还的部分 AWS Marketplace 费用。
上架费用百分比	要从账单金额中扣除的 AWS Marketplace 费用百分比。
卖家纳税份额	该交易收取的美国销售和使用税总额。
卖家纳税份额退款	如果处理了退款，则为该交易退还的美国销售和使用税总额。
AWS税收份额	代表卖家为该交易计费的美国销售和使用税总额。
AWS税收份额退款	当代表卖家收取了此类税款时，如果已处理退款，则为该交易退还的美国销售和使用税总额。
批发成本	仅适用于渠道合作伙伴专属优惠。经销商的产品成本。例如，经销商在销售制造商产品时向制造商支付的费用。批发成本等于标价乘以折扣百分比。
批发成本退款	仅适用于渠道合作伙伴专属优惠。经销商退还的产品成本。
卖家净收入	扣除 AWS Marketplace 费用、退款以及美国销售和使用税后的总账单金额。
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
交易参考 ID	代表交易的唯一标识符，可用于关联 AWS Marketplace 报表间的交易。

列	描述
AWS记录在案的卖家	为交易提供便利的商业实体的标识符。可能值如下所示： • AWS_INC:AWS, Inc.（总部设在美国）的标识符 • AWS_EUROPE: 亚马逊 Web Services EMEASARL（总部设在卢森堡）的标识符 • AWSAUSTRALIA: Amazon Web Services Australia Pty Ltd 的标识符 • AWS_JAPAN: 日本 Amazon Web Services G.K. 的标识符
转售授权 ID	已注册的机会的唯一标识符。
转售授权名称	已注册的机会的唯一名称。
转售授权说明	已注册机会的ISV定义描述。
转售公司名称	以批发成本购买产品或服务以转售ISV给客户的账户名称。
付款人身份证 AWS 账户	以批发价购买产品或服务以转售ISV给客户的账户的 ID。
付款人电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 liu-jie@example.com，则该条目为 example.com。
付款人城市	与费用计入的账户关联的账单地址中的城市。
付款人省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
付款人国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。

列	描述
付款人邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
ISV 账户 ID	产品或服务所有者的标识符。
ISV公司名	产品或服务所有者的企业名称。
产品 ID	软件产品的友好型唯一标识符。

收取和支付控制面板

收款和付款控制面板提供自上次付款以来 AWS 收款和支付到您的银行账户的资金的数据。它还提供所有未结和未付发票列表。

支付包括您的产品订阅的客户付款或退款，以及向客户收取或退还的一些税费。在向客户收取发票上的全部款项之前，您不会收到支付资金。控制面板上的退款显示为负数，因为在您批准退款后这笔款项退还给您的客户。

通过使用收取和支付控制面板，您可以更及时地收到客户支付报告。与旧版支付报告相比，预计可节省大约四天，旧版是在支付发送五天后创建的。

> **Note**
>
> 买家的付款条件不同 AWS，因此某些未收款年龄段的资金可能不会从客户那里支付。

收取和支付控制面板为运营和财务流程提供信息。控制面板每天刷新。有关更多信息，请参阅以下主题。

主题

- 收取和支付控制面板的刷新频率
- 第 1 部分：控件
- 第 2 部分：选择日期类别
- 第 3 部分：指标
- 第 4 部分：趋势
- 第 5 部分：细分

- 第 6 部分：精细数据

收取和支付控制面板的刷新频率

收取和支付控制面板仅在北美工作日更新。您预计会在收到银行存款后的 1 天内看到已支付的发票。

第 1 部分：控件

控制面板的这一部分提供了筛选条件，用于调整您的控制面板数据。例如，您可以在AWS Marketplace事件通知的字段上选择筛选条件，以确认特定客户账户 ID、订阅者公司名称或优惠 ID 的付款。您还可以按支付状态进行筛选，以了解所有已支付给您的发票或未结和未付的发票。您可以向分析添加筛选条件，例如您想要包含在视觉对象中的日期范围。在控件中选择的筛选条件会更新指标、趋势、细分和精细数据部分中显示的数据。

有关筛选的更多信息，请参阅《亚马逊 QuickSight 用户指南》 QuickSight中的在亚马逊上筛选数据。

控件描述

控件名称	描述
订阅者 AWS 账户 ID	订阅产品的账户的 ID。
订阅用户公司名称	订阅产品的账户的名称。
产品标题	产品的标题。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
AWS记录在案的卖家	为交易提供便利的商业实体的标识符。可能的值包括： • AWS_INC： AWS, Inc.（总部位于美国）的标识符。 • AWS_EUROPE：亚马逊 Web Services EMEASARL（总部设在卢森堡）的标识符。

控件名称	描述
	• AWS_AUSTRALIA：Amazon Web Services Australia Pty Ltd（AWS 澳大利亚）的标识符 • AWS_JAPAN：日本 Amazon Web Services G.K. 的标识符
支付状态	与发票关联的状态，用于确认自上次付款以来 AWS 已向您的银行账户收款并支付了款项。相关发票的已支付资金已收取并支付。相关发票的未支付资金未收取也未支付。
付款人身份证 AWS 账户	费用计入的账户的 ID。
付款人公司名称	费用计入的企业名称。
经销公司	授权销售软件制造商产品的经销商账户的公司名称。
经销商 ID AWS 账户	从中批发购买产品或服务以转售ISV给客户的账户的 ID。
转售授权 ID	已注册的机会的唯一标识符。
转售授权名称	已注册的机会的唯一名称。
订阅用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。
订阅用户省/自治区/直辖市或区域	与订阅产品的账户关联的账单地址中的省/自治区/直辖市或区域。
交易参考 ID	交易的唯一标识符，可帮助您关联 AWS Marketplace 旧版报告中的交易。
支付银行跟踪 ID	对于支付，跟踪 ID 由银行分配。银行追踪 ID 可用于将卖家银行提供的存款通知和报告与 AWS Marketplace 报告中的发票关联起来。

第 2 部分：选择日期类别

控制面板的此部分提供了筛选条件，可根据两个不同的日期维度以及日期字段值是在指定日期之前还是之后还是在某个日期范围内来细化记录。日期维度是付款到期日或最后一次支付日期。日期类别筛选条件会更新指标、趋势、细分和精细数据部分中显示的数据。默认日期类别是最后一次支付日期，它提取最近六个月的数据。

第 3 部分：指标

控制面板的此部分显示一个关键绩效指标 (KPI)，用于可视化已支付和未支付收入数字之间的比较。系统会显示指定日期类别和日期范围内的总收入、净收入、批发成本（如果适用）、支付金额和未支付金额。KPI

第 4 部分：趋势

控制面板的此部分提供了指定日期范围内的支付和逾期趋势视图。您可以按指定日期聚合（例如日、月、季度或年）查看趋势，以深入了解您的 AWS Marketplace 馆藏运行状况。趋势视图包括：

- 支出趋势 - 提供平均支付天数和相关净收入的快照。该趋势衡量的是从发票日期到支付日期之间的天数，以报告收取效率。您可以从日期聚合筛选器中选择日期范围。
- 已支付付款期限 - 提供净收入的快照和按标准账龄应收账款时段（例如未到期、1 到 30 天和 31 到 60 天）分类的已支付发票数量。该趋势衡量从付款到期日到支付日期之间的天数，以报告支付是否在客户的付款期限内。
- 未支付付款的期限 - 提供净收入以及未结和未付发票数量的快照，按逾期时段（例如未到期、1 到 30 天和 31 到 60 天）整理。未支付的资金可能包括尚未到期的金额。该趋势衡量从当天的日期到付款到期日期之间的天数，以显示收到的应收账款。

第 5 部分：细分

控制面板的此部分按优惠 ID、产品标题、付款人公司名称、订阅用户公司名称、经销商名称（如果他们参与渠道合作伙伴的专属优惠）、付款人地理位置和订阅用户地理位置为您提供应收账款视图。使用细分来衡量每个类别的已支付应收账款与未支付的应收账款。

第 6 部分：精细数据

控制面板的此部分按产品、客户和优惠详情显示所有支付和未收取款项。

> **Note**
>
> 2021 年 4 月 1 日之前创建的发票可能没有关联的协议编号、优惠编号、订阅者 AWS 账户 ID 或订阅者公司名称。

有关如何从 QuickSight 表中导出和下载数据的信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

精细数据描述

列	描述
发票日期	向客户收取产品订阅费用的日期。
付款截止日期	付款到期日采用 YYYY-MM-DD 格式。
付款条款	客户的发 AWS 票付款条款。
发票 ID	为向客户收取的费用分配的 AWS ID。
上架费发票 ID	当通过 AWS EMEASARL日本或澳大利亚的法人实体（记录在案的卖家）进行 AWS Marketplace 订阅交易时，销售的商城运营商（例如 AWS EMEASARL）必须向卖家收取卖家VAT上架费。对于适用的交易，对上市费进行VAT评估的发票编号与软件或产品订阅发票编号不同。
订阅用户公司名称	订阅产品的账户的名称。
订阅者 AWS 账户 ID	订阅产品的账户的 ID。
订阅用户电子邮件域	与订阅产品的账户关联的电子邮件域。例如，如果电子邮件地址是 liu-jie@example.com，则该条目为 example.com。
订阅用户城市	与订阅产品的账户关联的账单地址中的城市。

列	描述
订阅用户省/自治区/直辖市或区域	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。
订阅用户国家/地区	与订阅产品的账户关联的账单地址中的国家/地区。
订阅用户邮政编码	与订阅产品的账户关联的账单地址中的邮政编码。
产品标题	产品的标题。
优惠名称	卖家定义的优惠名称。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
协议开始日期	客户开始订阅产品的日期，格式为 MM-DD-。YYYY如果这是未来日期的协议，则该日期可能与接受日期不同。
协议结束日期	合同到期的日期，格式为 MM-DD-。YYYY对于计量/订pay-as-you-go 阅，此日期设置为 -1-9999。JAN
协议接受日期	客户订阅产品的日期，格式为 MM-DD-。YYYY
使用期结束日期	产品使用期的结束日期。
使用期开始日期	产品使用期的开始日期。

列	描述
支付状态	与发票关联的状态，用于确认自上次付款以来 AWS 已向您的银行账户收款并支付了款项。相关发票的已支付资金已收取并支付。相关发票的未支付资金未收取也未支付。
支付日期	AWS 开始向卖家银行付款的日期。
支付银行跟踪 ID	对于支付，跟踪 ID 由银行分配。银行追踪编号可用于将卖家银行提供的存款通知和报告与报告中的发票关联起来。 AWS Marketplace
总收入	向客户计费的产品用量或月度费用的金额。
总退款	如果在数据涵盖时段内处理了任何退款，则为向客户退还的订阅成本总额。
上架费用	要从账单金额中扣除的 AWS Marketplace 费用金额。
上架费用退款	如果在数据覆盖期内处理了任何退款，则退还的部分 AWS Marketplace 费用。
上架费用百分比	要从账单金额中扣除的 AWS Marketplace 费用百分比。
卖家纳税份额	该交易收取的美国销售和使用税总额。
卖家纳税份额退款	如果处理了退款，则为该交易退还的美国销售和使用税总额。
批发成本	仅适用于渠道合作伙伴专属优惠。经销商的产品成本。例如，经销商在销售制造商产品时向制造商支付的费用。批发成本等于标价乘以折扣百分比。
批发成本退款	仅适用于渠道合作伙伴专属优惠。经销商退还的产品成本。

列	描述
卖家净收入	扣除 AWS Marketplace 费用、退款以及美国销售和使用税后的总账单金额。
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
交易参考 ID	代表交易的唯一标识符，可用于关联 AWS Marketplace 报表间的交易。
AWS记录在案的卖家	为交易提供便利的商业实体的标识符。可能值如下所示： • AWS_INC： AWS, Inc.（总部设在美国）的标识符 • AWS_EUROPE：亚马逊 Web Services 的标识符 EMEASARL（总部设在卢森堡） • Amazon Web Services Australia Pty Ltd（AWS 澳大利亚） • Amazon Web Services Japan G.K.
转售授权 ID	已注册的机会的唯一标识符。
转售授权名称	已注册的机会的唯一名称。
转售授权说明	已注册机会的ISV定义描述。
经销商公司名称	以批发成本购买产品或服务以转售ISV给客户的账户名称。
经销商 ID AWS 账户	以批发价购买产品或服务以转售ISV给客户的账户的 ID。
付款人公司名称	费用计入的账户的 ID。
付款人身份证 AWS 账户	费用计入的账户的 ID。

列	描述
付款人电子邮件域	与费用计入的账户关联的电子邮件域。例如，如果电子邮件地址是 liu-jie@example.com，则该条目为 example.com。
付款人城市	与费用计入的账户关联的账单地址中的城市。
付款人省/自治区/直辖市	与费用计入的账户关联的账单地址中的省/自治区/直辖市。
付款人国家/地区	与费用计入的账户关联的 2 字符国家/地区代码。
付款人邮政编码	与费用计入的账户关联的账单地址中的邮政编码。
ISV 账户 ID	产品或服务所有者的标识符。
ISV公司名	产品或服务所有者的企业名称。
产品 ID	软件产品的友好型唯一标识符。
已支付净收入	向卖家支付的交易总额。
未支付净收入	未向卖家支付的交易总额
支付期	描述应收账款收款的类别范围（例如，未到期，1 到 30 天和 31 到 60 天）。

税务控制面板

税务仪表板为以下交易提供有关美国和国际税收的可视化和细粒度数据。 AWS Marketplace可以从 “财务运营” 下的 “ AWS Marketplace 管理门户 见解” 选项卡访问此控制面板。

您可以从 Amazon QuickSight 控制面板上的任何图表、图表和表格中导出和下载数据（.csv 或 MicrosoftExcel文件）。有关更多信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

有关运营和财务流程，请查看以下主题。

主题

- 税务仪表板的刷新频率
- 第 1 部分：控件
- 第 2 部分：筛选日期
- 第 3 部分：指标
- 第 4 节：税收趋势
- 第 5 部分：细分
- 第 6 部分：精细数据
- 美国豁免代码
- 欧盟豁免代码

税务仪表板的刷新频率

仪表板每天更新。如果从上游系统收到的最新交易数据延迟，则最新数据可能会延迟 1-2 天才能反映在仪表板上。

第 1 部分：控件

控制面板的这一部分提供了筛选条件来完善您的税务数据。您也可以向分析添加筛选条件，例如您想要包含在视觉对象中的日期范围。在控件中选择的筛选条件会更新指标、趋势、细分和精细数据部分中显示的数据。

有关筛选的更多信息，请参阅《亚马逊 QuickSight 用户指南》 QuickSight中的在亚马逊上筛选数据。

控件描述

控件名称	描述
发票 ID	为向客户收取的费用分配的 AWS ID。
已纳税的客户 ID AWS 账户	为订阅产品而征税的账户的 AWS ID。
AWS 记录在案的卖家	为交易提供便利的商业实体的标识符。
已纳税的客户所在国家	与纳税客户关联的双字符国家/地区代码
对客户所在州或地区征税	与纳税客户关联的账单地址、州或地区。

控件名称	描述
已纳税的客户城市	与纳税客户关联的账单地址城市
Taxable	来源应纳税、非应税和退税交易。
纳税方	AWS 或 Seller。如果卖家是税务责任方，则会征收税款。如果 AWS 是纳税方，则由 AWS其征收和汇出销售税。有关更多信息，请参阅卖家 AWS Marketplace 税务帮助。 如果未征税，则不显示任何价值。卖方必须确定是否为每张发票征收了一些税款，因为卖方有责任征税。
税收类型	应用于交易的税费类型。可能的值为None、Sales、和SellerUse 、Tax registration type 、VAT（增值税）、CNPJ、IGST、CT。

第 2 部分：筛选日期

控制面板的这一部分提供了筛选条件，用于根据指定日期范围内的发票日期来优化记录。在发票日期内选择的筛选条件会更新指标、趋势、细分和精细数据部分中显示的数据。例如，如果您选择日期筛选条件Past 30 days，则所有部分都将反映与过去 30 天内纳税的发票相关的数据。

第 3 部分：指标

控制面板的此部分显示关键绩效指标 (KPI)，用于可视化与税收相关的指标，包括应纳税金额、非应纳税金额、税额、已纳税发票数量和交易总数。您可以通过更新 “筛选器” 部分中的日期筛选器来更新日期范围。

第 4 节：税收趋势

控制面板的此部分提供指定日期范围内的税收趋势。您可以按指定日期汇总（例如每日、、或 year-over-year）查看趋势 month-over-month quarter-over-quarter，以深入了解税收情况。以下税收趋势信息可用：

- 税收趋势-金额- 按发票日期提供选定时间段的应纳税额和税额的快照。

- 税收趋势-发票数量- 按发票日期提供选定时间段内发票数量的快照。

第 5 部分：细分

控制面板的这一部分按发票编号、税种类型、产品标题和客户地理位置为您提供企业的税务指标。

第 6 部分：精细数据

控制面板的这一部分显示了贵组织中 AWS Marketplace交易的详细税务数据。

精细数据描述

列	描述
发票 ID	为向客户收取的费用分配的 AWS ID。
订单项目 ID	行项目的唯一标识符。退款交易具有与远期税务交易相同的行项目 ID。
客户账单编号	客户账单编号
纳税方	AWS 或 Seller。如果卖家是税务责任方，则会征收税款。如果 AWS 是纳税方，则由 AWS其征收和汇出销售税。有关更多信息，请参阅卖家 AWS Marketplace 税务帮助。 如果未征税，则不显示任何价值。卖方必须确定是否为每张发票征收了一些税款，因为卖方有责任征税。
交易类型代码	交易的类型代码。值包括： • AWS— 远期税收交易。 • REFUND— 全额或部分退款。 • TAXONLYREFUND — 仅限税款的退款。 退款交易具有与原始远期交易相同的行项目 ID。

列	描述
产品 ID	软件产品的友好型唯一标识符。
产品标题	购买的产品的名称。
商品税务代码	用于标识产品税费属性的标准代码。您可以在创建或修改产品时选择这些属性。
发票日期	向客户收取产品订阅账单和纳税的日期。
已纳税的客户 ID AWS 账户	纳税客户的 AWS 账户 ID。
已纳税的客户所在国家	与纳税客户关联的双字符国家/地区代码。
对客户所在州或地区征税	与纳税客户关联的账单地址、州或地区。
已纳税的客户城市	与纳税客户关联的账单地址城市。
已纳税的客户邮政编码	用于税务计算的邮政编码。
税收类型	应用于交易的税费类型。可能的值为`NoneSales`、和`SellerUse`、`Tax registration type`、`VAT`、`CNPJ`、`IGST`、`CT`。
管辖级别	用于税务位置的地址的管辖区级别。可能的值为`State`、`County`、`City` 和 `District`。
税收管辖权	纳税管辖区的名称。
显示价格应纳税类型	向客户显示的价格的纳税类型。所有 AWS Marketplace 产品均为独家优惠。
税收管辖税率	在管辖区级别应用的税率。
税额	在管辖区级别收取的税费。
税收货币	所收税款的货币。例如，如果交易以美元为单位，则条目为USD。

列	描述
计税原因代码	交易是应税、非应税、免税还是零税率，按管辖区级别划分。
用于计税的日期	用于计算交易税费的日期。
客户豁免证书 ID	免税证明的证明 ID。
客户豁免证书 ID 域	在 Amazon 系统中存储证明的位置。
客户豁免证书级别	提供免税的管辖区级别。
买家豁免码	指定豁免的代码。例如，RESALE。
交易参考 ID	交易的唯一标识符，可帮助您关联 AWS Marketplace 报表中的交易。
AWS 记录在案的卖家	为交易提供便利的商业实体的标识符。

> **Note**
>
> 总税额和买家费用总额已移至账单收入控制面板中的AWS 税收份额和总收入字段。

美国豁免代码

豁免码	描述
501C	501C 组织
AP	农业生产
CO	慈善组织
DIRECT	直接付款许可证
DISTSBT	Distributor

豁免码	描述
DP	联合国/外交官
DV	残疾退伍军人
EDI	教育机构
FG	联邦政府
HCP	医疗保健提供者
HO	医院（非营利组织或州立医院）
INSUR	保险
IPM	IPM
LB	Library
MPU	多重使用点豁免；数码产品/软件豁免
NA	美洲原住民
NAI	美洲原住民个人
NP	非营利组织
NPR	非营利性宗教组织
OT	其他
RESALE	经销商
SBE	小型企业豁免
SLG	州/地方政府

欧盟豁免代码

豁免码	描述
SpecialOrg	免收费用的 Organization VAT。
SpecialZone	免税区域。
SplitPayment	买方直接向税务机关支付发票上VAT注明的任何款项。

销售运营控制面板

AWS Marketplace 提供多个仪表板来帮助您跟踪销售数据。

主题

- 协议和续订控制面板
- 使用情况控制面板

协议和续订控制面板

在 AWS Marketplace中签署系以后，协议和续订控制面板可在 24 小时内提供有关协议和续订的信息。仪表板有助于跟踪即将到期的订阅以启用续订。

要打开控制面板，请启动 AWS Marketplace 管理门户，选择 “销售运营”，然后选择 “见解” 选项卡。

主题

- 第 1 部分：控件
- 第 2 部分：选择日期类别
- 第 3 部分：指标
- 第 4 部分：订阅趋势
- 第 5 部分：细分
- 第 6 部分：精细数据

第 1 部分：控件

控制面板的这一部分提供了筛选条件，用于调整您的协议和续订控制面板数据。您可以选择以下筛选条件：

> **Note**
>
> 有关筛选的更多信息，请参阅《亚马逊 QuickSight 用户指南》 QuickSight中的在亚马逊上筛选数据。

控件描述

控件名称	描述
订阅者 AWS 账户 ID	订阅产品的账户的 ID。
订阅用户公司名称	订阅产品的账户的名称。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
CPPO 标志	是/否标志，表示协议是否使用渠道合作伙伴的专属优惠达成。如果是，则记录卖家是渠道合作伙伴。如果否，则记录卖家是软件制造商（独立软件供应商）。
ISVAWS账号	产品或服务所有者的账户。
ISV公司名	产品或服务所有者的企业名称。
旧版产品 ID	软件产品的旧版唯一标识符。
产品标题	产品的标题。
转售授权 ID	向渠道合作伙伴提供的转售授权ISV的 ID。

控件名称	描述
转售授权名称	向渠道合作伙伴提供的转售授权ISV的名称。
经销商 ID AWS 账户	从中批发购买产品或服务以转售ISV给客户的账户的 ID。
经销商公司名称	授权销售软件制造商产品的经销商账户的公司名称。
订阅用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。
订阅用户省/自治区/直辖市或区域	与订阅产品的账户关联的账单地址中的省/自治区/直辖市或区域。
订阅用户城市	与订阅产品的账户关联的账单地址中的城市。

第 2 部分：选择日期类别

控制面板的这一部分提供了筛选条件，用于根据指定日期范围内的协议结束日期、协议结束期或协议开始日期来调整记录。在选择日期类别中选择的筛选条件会更新指标、趋势、细分和精细数据部分中显示的数据。

有关最近签署的协议的信息，您可以按协议开始日期进行筛选。如果您有兴趣了解协议续订，则可以使用协议结束日期进行筛选。

第 3 部分：指标

仪表板的此部分显示关键绩效指标 (KPI)，用于可视化关键协议数据的比较。显示有效协议的数量和已终止协议的数量。KPI您可以通过更新选“择日期类别”字段中的日期标准来更新日期范围。

第 4 部分：订阅趋势

控制面板的此部分提供指定日期范围内的订阅趋势。您可以按指定日期聚合（例如每日、、或 year-over-year）查看趋势 month-over-month quarter-over-quarter，以深入了解协议。将整体显示协议和续订趋势信息。

- 订阅趋势 – 根据所选日期聚合筛选条件和所选日期类别筛选条件，提供有效协议数量和已终止协议数量的快照。要根据协议开始日期查看订阅趋势，可以在选择日期类别筛选条件中选择协议开始日期。要根据协议开始结束查看订阅趋势，可以在选择日期类别筛选条件中选择协议结束日期。
- 续订明细 – 提供在特定天数内已结束或即将结束的续订的快照。您可以查看多个时间段的续订明细，例如在 1 到 30 天前、31 到 60 天前或超过 60 天前结束的协议。您还可以筛选即将在 0 到 30 天、31 到 60 天后结束的协议，依此类推。

第 5 部分：细分

控制面板的这一部分为您提供跨公司名称ISVs、活跃订阅者和经销商的业务指标。您可以查看协议数量、已结束的协议IDs、报价IDsIDs、续订的协议IDs和订阅者。

第 6 部分：精细数据

仪表板的这一部分显示了协议、报价、产品、协议提议者、订阅者、转售授权、经销商和的详细数据。ISVs

> **Note**
>
> 有关如何从 QuickSight 表中导出和下载数据的更多信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

精细数据描述

列	描述
订阅用户公司名称	订阅用户公司的名称。
订阅者AWS账户 ID	订阅产品的账户的 ID。
订阅用户电子邮件域	与订阅产品的账户关联的电子邮件域。例如，如果电子邮件地址是 liu-jie@example.com，则该条目为 example.com。
订阅用户国家/地区	与订阅产品的账户关联的账单地址中的国家/地区。

列	描述
订阅用户省/自治区/直辖市或区域	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。
订阅用户城市	与订阅产品的账户关联的账单地址中的城市。
订阅用户邮政编码	与订阅产品的账户关联的账单地址中的邮政编码。
产品标题	产品的标题。
产品 ID	软件产品的友好型唯一标识符。
旧版产品 ID	软件产品的旧版唯一标识符。
优惠名称	卖家定义的优惠名称。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
协议开始日期	客户开始订阅产品的日期，格式为 MM-DD-。YYYY如果这是未来日期的协议，则该日期可能与接受日期不同。
协议接受日期	客户订阅产品的日期，格式为 MM-DD-。YYYY
协议结束日期	合同到期的日期，格式为 MM-DD-。YYYY对于计量/订pay-as-you-go 阅，此日期设置为 -1-9999。JAN

列	描述
Status	协议的当前状态。支持的状态包括： • 有效-协议的部分或全部条款已生效。 • 已终止 — 由于AWS启动的终止事件，协议在预先商定的结束日期之前结束。最常见的是付款失败。 • 已取消 — 接受者选择在协议结束日期之前终止协议。 • 已过期-协议在预先商定的结束日期结束。 • 已替换-使用替换报价替换协议。 • 续订-使用自动续订等功能，将协议续订为新协议。 • 已存档-协议已结束；未指定终止协议的确切原因，即 “已终止”、“已取消” 或 “已过期”。 • 已@@ 回退-由于出错，协议修订已被回退。较早的修订版现已启用。仅适用于无效的协议修订。 • 已@@ 取代-协议修订版不再处于活动状态，另一个协议修订版现已处于活动状态。仅适用于无效的协议修订。 • 已@@ 回退-由于出错，协议修订已被回退。较早的修订版现已启用。仅适用于无效的协议修订。
预计协议价值	协议的估计成本，也称为 “合同总价值” 或 “购买金额”。这适用于以下产品类型：SaaS、专业服务和服务器，以及合同或年度定价类型。
货币代码	与协议的预估成本相关的报价货币。
经销商公司名称	以批发成本购买产品或服务以转售ISV给客户的账户名称。

列	描述
经销商AWS账户 ID	以批发价购买产品或服务以转售ISV给客户的账户的 ID。
转售授权 ID	已注册的机会的唯一标识符。
转售授权名称	已注册的机会的唯一名称。
转售授权说明	已注册机会的ISV定义描述。
CPPO 标志	Amazon 中的计算字段 QuickSight。
协议意图	如果当前协议是先前协议的续订或升级，则填充之前的 agrement_id 列中提供的协议。可能的值包括： • 升级：对于专属优惠，升级是指任何修改（升级、降级、修改、扩展、续订）的广义术语。对于公开优惠，升级是指更改条款（使用不同的优惠）的具体协议，从而产生新协议，并且特定协议已存档。 • 续订：仅适用于公开优惠。当续订协议的开始日期为特定协议的结束日期，并且优惠 ID 未更改但协议 ID 已更改时。例如，协议 A 已变成协议 B。 • 自动续订：仅适用于公开优惠。当当前协议结束，创建新协议时。买家已开启自动续订。 • 新：买家已接受条款（使用优惠）并订阅了新协议。
之前的协议 ID	如果本协议已续订或升级，则为协议参考。
下一个协议 ID	如果当前协议已续订或升级，则为下一个协议的协议 ID。
以前的优惠 ID	如果当前协议已续订或升级，则与之前的协议关联的优惠 ID。

列	描述
下一个优惠 ID	如果当前协议已续订或升级，则与下一份协议关联的优惠 ID。
下一份协议 ID 接受日期	买家接受下一份协议的日期。
投标人AWS账户 ID	产品或服务所有者的标识符。
提议者公司名称	产品或服务所有者的企业名称。
ISVAWS账号	卖家的 AWS 标识符。
ISV公司名	卖家的公司名称。
协议到期（以天为单位）	QuickSight 中的计算字段。从今天起协议到期的剩余天数。
协议终止期间	中的计算字段 QuickSight。协议终止的时间段。

使用情况控制面板

使用情况控制面板为使用 SaaS 和基于服务器使用情况的产品的客户提供可视化和细粒度数据。 AWS Marketplace 卖家可以使用此控制面板跟踪客户对基于使用情况的产品的消费，从而就产品支持、定价、从公开报价转换为私有报价以及产品停产做出决定。仪表板提供过去 6 个月的数据，这是一个滚动窗口。

要打开控制面板，请启动 AWS Marketplace 管理门户，选择 “销售运营”，然后选择 “见解” 选项卡。

您可以将数据导出和下载为.csv 或 Microsoft Excel 来自 Amazon QuickSight 控制面板上任何图表、图表和表格的文件。有关更多信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

> **Note**
>
> 此控制面板显示在创建优惠时指定的所有维度键的使用情况。例如，对于 Amazon Machine Image (AMI) 产品，维度键是实例类型，报价中指定的所有实例类型都将显示在此控制面板中，即使它们的定价为 0 美元。要查看在创建优惠时未指定的维度的产品使用情况，请考虑重新发布该产品以包含您需要的维度。

主题

- 使用情况控制面板的刷新频率
- 第 1 部分：控件
- 第 2 部分：筛选条件
- 第 3 部分：指标
- 第 4 部分：趋势
- 第 5 部分：细分
- 第 6 部分：精细数据

使用情况控制面板的刷新频率

仪表板每天下午 4 点PST（午夜UTC）更新。请注意，使用情况数据接收自上游数据来源，可能会遇到延迟，为清楚起见，您可以查看具体的使用日期和控制面板上的使用情况报告日期，将两者进行比较。

有关运营和财务流程，请查看以下主题。

第 1 部分：控件

控制面板的这个部分提供筛选条件，用于筛选您的使用数据。例如，您可以选择以下筛选条件：

> **Note**
>
> 有关筛选的更多信息，请参阅亚马逊 QuickSight 用户指南中的 QuickSight在亚马逊中筛选数据。

控件描述

控件名称	描述
最终用户公司名称	使用产品的账户的名称。
最终用户 AWS 账户 ID	使用产品的账户的 ID。
最终用户所在国家/地区	与使用产品的账户关联的 2 字符国家/地区代码。

控件名称	描述
产品标题	产品的标题。
产品代码	用于计量产品的现有授权产品代码。此值还用于将数据与报告合并，或者用于引用 Marketplace Metering AWS Metering 服务中提供的内容。
优惠 ID	买家签订的优惠的标识符。
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
维度键	与产品使用情况相关的资源类型。维度键适用于 SaaS 和基于服务器使用情况的产品。
订阅用户公司名称	订阅产品的账户的名称。
订阅者 AWS 账户 ID	订阅产品的账户的 ID。
订阅用户国家/地区	与订阅产品的账户关联的 2 字符国家/地区代码。
经销商公司名称	授权销售产品制造商产品的经销商账户的名称。
经销商 ID AWS 账户	从中批发购买产品或服务以转售ISV给客户的账户的 ID。
转售授权 ID	从中批发购买产品或服务以转售ISV给客户的账户的 ID。
CPPO 标志	是/否标志，表示协议是否使用渠道合作伙伴的专属优惠达成。如果是，则记录卖家是渠道合作伙伴。如果否，则记录卖家是产品制造商（独立软件供应商）。

第 2 部分：筛选条件

控制面板的这个部分提供筛选条件，用于根据使用日期筛选记录。在这些筛选条件中选择的值会更新指标、趋势、细分和精细数据部分中显示的数据。默认选择是提取最近 6 个月的使用情况数据。

第 3 部分：指标

仪表板的这一部分显示了一个关键绩效指标 (KPI)，用于可视化与消耗相关的指标：估计的使用量单位、使用量的客户和含使用情况的产品。您可以通过更新筛选条件部分中的使用日期标准来更新日期范围。请注意，关键指标将显示所有单位类型的数据。

第 4 部分：趋势

控制面板的这个部分提供指定日期范围内的使用趋势。您可以按指定日期聚合（例如每日、、或）查看趋势 quarter-over-quarter， year-over-year 以深入了解使用情况。 month-over-month也可以选择使用单位类型，以图形方式查看其使用趋势。

第 5 部分：细分

控制面板的这一部分为您提供企业的估计使用量指标，包括公司名称、产品标题、维度密钥和所选单位类型的报价IDs。您也可以选择要查看的条目数。

第 6 部分：精细数据

控制面板的这一部分显示了使用情况、优惠、产品、订阅者、付款人、最终用户、转售授权、经销商和独立软件供应商（ ）的精细数据。ISVs请注意，精细数据表显示所有单位类型的数据。

在月底账单最终确定之前，应将收入视为估计值。用于结算上个月使用量的发票将在下个月的第二天或第三天提交给买家（例如，在 11 月 1 日至 11 月 30 日之间计量使用量的客户将在 12 月 2 日或 12 月 3 日收到使用量发票）。计量的使用量可能会在实际使用日期的几天后到达此控制面板，因此使用日期和使用情况报告日期可能会有所不同。这意味着您可能需要在整个月中每天访问控制面板以进行当月的 up-to-date 跟踪。有关权威的客户账单信息，请参阅财务运营选项卡中的账单收入控制面板。

> **Note**
>
> 有关筛选的更多信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

精细数据描述

列	描述
使用日期	客户使用产品的日期。
使用情况报告日期	客户对产品的使用显示在见解控制面板中的日期。
最终用户公司名称	使用产品的账户的名称。
最终用户AWS账户 ID	使用产品的账户的 ID。
最终用户电子邮件域	与使用产品的账户关联的电子邮件域。例如，如果电子邮件地址为 abc@example.com，则该条目为 example.com。
最终用户城市	与使用产品的账户关联的城市。
最终用户省/自治区/直辖市	与使用产品的账户关联的省/自治区/直辖市。
最终用户所在国家/地区	与使用产品的账户关联的 2 字符国家/地区代码。
最终用户邮政编码	与使用产品的账户关联的账单地址中的邮政编码。
产品标题	产品的标题。
旧版产品 ID	产品的旧版唯一标识符。
产品 ID	产品的友好型唯一标识符。
产品代码	用于计量产品的现有授权产品代码。此值还用于将数据与报告联接，或引用 AWS Marketplace Metering Service中提供的内容。
优惠 ID	买家签订的优惠的标识符。
优惠名称	卖家定义的优惠名称。

列	描述
优惠可见性	优惠是公开、专属还是企业合同优惠。
协议 ID	提议者和接受者之间签署的开始使用产品的协议的唯一协议数据源参考。
协议接受日期	客户订阅产品的日期时间戳。UTC
协议开始日期	客户产品订阅开始UTC时的日期时间戳。如果这是未来日期的协议，则该日期可能与接受日期不同。
协议结束日期	合同到期的日期。UTC对于计量/订pay-as-you-go阅，此日期设置为 9999 年 1 月 1 日上午 12:00。
维度键	与产品使用情况相关的资源类型。维度键适用于 SaaS 和基于服务器使用情况的产品。
区域	买家部署 Amazon EC2 实例的区域。
估计使用量	为产品记录的使用量。
使用量单位类型	记录使用量的单位类型。
估计收入	产品使用量带来的估计收入。在月底账单最终确定之前，应将收入视为估计值。用于结算上个月使用量的发票将在下个月的第二天或第三天提交给买家。
货币	交易的货币。例如，如果交易以美元为单位，则条目为USD。
订阅用户公司名称	订阅产品的账户的名称。
订阅者 AWS 账户 ID	订阅产品的账户的 ID。

列	描述
订阅用户电子邮件域	与订阅产品的账户关联的电子邮件域。例如，如果电子邮件地址为 abc@example.com，则该条目为 example.com。
订阅用户城市	与订阅产品的账户关联的账单地址中的城市。
订阅用户省/自治区/直辖市或区域	与订阅产品的账户关联的账单地址中的省/自治区/直辖市。
订阅用户国家/地区	与订阅产品的账户关联的账单地址中的国家/地区。
订阅用户邮政编码	与订阅产品的账户关联的账单地址中的邮政编码。
付款人公司名称	为产品付款的账户名称。
付款人身份证 AWS 账户	为产品付款的账户 ID。
付款人电子邮件域	与为产品付款的账户关联的电子邮件域。例如，如果电子邮件地址为 abc@example.com，则该条目为 example.com。
付款人城市	与为产品付款的账户关联的账单地址中的城市。
付款人省/自治区/直辖市	与为产品付款的账户关联的账单地址中的省/自治区/直辖市。
付款人国家/地区	与为产品付款的账户关联的账单地址中的国家/地区。
付款人邮政编码	与为产品付款的账户关联的账单地址中的邮政编码。
经销商公司名称	以批发成本购买产品或服务以转售ISV给客户的账户名称。

列	描述
经销商 ID AWS 账户	以批发价购买产品或服务以转售ISV给客户的账户的 ID。
转售授权 ID	已注册的转售机会的唯一标识符。
转售授权名称	已注册的转售机会的唯一名称。
转售授权说明	对已注册的转售机会的说明。
CPPO 标志	是/否标志，表示协议是否使用渠道合作伙伴的专属优惠达成。如果是，则记录卖家是渠道合作伙伴。如果不是，则记录在案的卖家是商品制造商 (ISV)。
ISV公司名	产品或服务所有者的名称。
ISV AWS 账户 身份证	产品或服务所有者的标识符。

用于营销的仪表板

AWS Marketplace 提供多个仪表板来帮助您跟踪营销数据。

主题

- 客户协议控制面板
- 房源绩效控制面板

客户协议控制面板

客户协议控制面板概述了在中订阅您的产品的协议和客户 AWS Marketplace。 控制面板提供有关您的新协议和有效协议的数据，以及客户趋势和概况。

> **Note**
>
> - 要解锁此控制面板，您必须注册AWS Marketplace 卖家 Prime 计划。

- 要打开控制面板，请启动 AWS Marketplace 管理门户，选择 “见解” 选项卡，然后选择 “营销”。

有关使用 AWS Marketplace 仪表板的更多信息卖家控制面板，请参阅本节前面的。

主题

- 第 1 部分：更新IAM权限
- 第 2 部分：筛选条件
- 第 3 部分：日期过滤器深入探索
- 第 4 节：公开发行和私募协议
- 第 5 部分：卖家指标

第 1 部分：更新IAM权限

要访问买AWS Marketplace 家协议控制面板，您必须注册 Seller Prime。

注册后，您必须更新现有IAM政策。从 2024 年 9 月起，我们将不再支持使用旧IAM权限访问卖家控制面板。要更新您的IAM权限，请使用以下示例中显示的 Amazon 资源名称 (ARN) 格式。

此示例允许访问当前和将来的 AWS Marketplace 资源，包括仪表板和报告。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

此示例通过包含单个仪表板来授予对单个仪表板的访问权限ARN。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::123456789012:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/CustomerAgreements_V1"
  }]
}
```

> **Note**
>
> 有关旧IAM权限的更多信息，请参阅《IAM用户指南》中的创建IAM策略。

第 2 部分：筛选条件

您可以使用以下筛选器来完善您的协议数据。

控件名称	描述
产品标题	产品的标题。
日期过滤器	包括过去 30、60 和 90 天、过去 12 个月 (TTM) 和年初至今 (YTD) 的数据。选择 “自定义” 以定义特定的开始和结束日期。
客户行业	为订阅卖家产品的客户 AWS 定义的行业。
客户群体	为订阅卖家商品的买家 AWS 定义的细分。

有关筛选的更多信息，请参阅《亚马逊 QuickSight 用户指南》 QuickSight中的在亚马逊上筛选数据。

第 3 部分：日期过滤器深入探索

此日期筛选器适用于客户协议仪表板上的所有指标。

例如，当您使用默认 year-to-date 筛选值时，会显示以下指标：

指标	描述
公开发行协议数量	将公开发售作为报价知名度、且至少有 1 天有效期的协议数量YTD。
私募报价协议计数	以私募报价作为报价曝光度且至少有 1 天有效期的协议数量YTD。
新的公开发行协议计数	以公开报价作为报价知名度的协议数量，其接受日期在内YTD。
新的私募要约协议计数	以私募报价作为报价可见度且接受日期在内的协议数量YTD。
活跃客户	内含至少 1 个有效协议的唯一客户（标识为订阅者AWS账户 ID）的数量YTD。有效协议定义为在该时间段内至少有 1 个有效日的协议。
新的付费客户	以订阅者AWS账户 ID 为首个账单月的唯一客户数量YTD。

第 4 节：公开发行和私募协议

控制面板的此部分显示您的协议概述。关键绩效指标 (KPIs) 包括公开发行协议的数量、私募协议的数量、新的公开发行协议的数量以及新的私募协议的数量。您可以看到音量和百分比中的 year-over-year 或 period-over-period 变化。您可以通过更新筛选器部分中的日期筛选器来更新日期范围。

协议是提议者（产品或服务所有者）和接受者（客户）之间签订的开始使用产品的合同。

> **Note**
>
> 新的协议指标包括有效、已过期、已取消和已终止的协议。

下表列出并描述了协议状态。

Status	描述
处于活动状态	协议的部分或全部条款已生效。
已过期	该协议在预先商定的结束日期结束。
已取消	接受者选择在协议结束日期之前终止协议。
已终止	由于AWS启动的终止事件（例如付款失败），协议在预先商定的结束日期之前结束。
已续订	使用自动续订等功能，将该协议续订为新协议。
已替换	该协议已被替换报价所取代。

第 5 部分：卖家指标

控制面板的此部分提供指定日期范围内的客户趋势。KPIs包括活跃客户数量和新付费客户数量。 月度趋势 — 根据图表中的筛选条件提供活跃客户数量和新付费客户数量的月度视图。 客户指标 — 根据所选指标筛选条件，提供有关活跃客户数量和新付费客户数量的数据。您可以选择客户群或行业，以了解每个客户群或客户行业对总客户指标的贡献。

> **Note**
>
> 协议具有给定状态。有关状态的信息，请参阅上一节中的表格。

下表列出并描述了这些指标。

指标	描述
公开发售协议	标识为IDs具有公开报价可见性的协议的数量；协议期限与选定的日期范围至少有一天重叠；长度在协议开始日期和协议结束日期之间确定。
私人报价协议	标识为IDs具有私募报价可见性的协议的数量；协议期限与选定的日期范围至少有一天重叠；期限在协议开始日期和协议结束日期之间确定。
新的公开募股协议	标识为具有公开报价可见性的协议 ID 的协议数量；协议接受日期在选定的日期范围内。
新的私募协议	标识为IDs具有私募报价可见性的协议的数量；协议接受日期在选定的日期范围内。
活跃客户	标识为订阅者AWS账户IDs且至少有一份有效协议的客户数量；协议期限至少有一天与所选日期范围重叠
新的付费客户	被标识为订阅者AWS账户IDs且其第一个账单月份在选定日期范围内的客户数量。

有关协议和收入的更多信息，请参阅本节中的协议和续订控制面板和账单收入控制面板。

房源绩效控制面板

商品信息业绩控制面板提供您的 M AWS arketplace 上架商品的概述和详细数据。控制面板提供有关流量以及您的客户订阅您的产品所采取的步骤的数据。控制面板还提供了按营销渠道划分的详细流量。

> **Note**
>
> - 要解锁此控制面板，您必须注册AWS Marketplace 卖家 Prime 计划。
> - 要打开控制面板，请启动 AWS Marketplace 管理门户，选择 “见解” 选项卡，然后选择 “营销”。

有关使用 AWS Marketplace 仪表板的更多信息卖家控制面板，请参阅本节前面的。

主题

- 第 1 部分：更新IAM权限
- 第 2 部分：过滤器
- 第 3 部分：日期过滤器深入探索
- 第 4 部分：摘要指标
- 第 5 部分：每月趋势和渠道转化率
- 第 6 部分：按渠道划分的流量趋势
- 第 7 部分：每月流量和协议指标
- 第 8 部分：独立访问者的网络流量来源
- 第 9 部分：网络流量来源明细

第 1 部分：更新IAM权限

注册后，您必须更新您的IAM权限。自 2024 年 9 月起，我们将不再支持通过传统IAM权限访问卖家控制面板。要更新您的权限，请使用以下示例中显示的 Amazon 资源名称 (ARN) 格式。

此示例允许访问当前和将来的 AWS Marketplace 资源，包括仪表板和报告。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

此示例通过包含单个仪表板来授予对单个仪表板的访问权限ARN。

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

```
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::123456789012:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/ListingPerformance_V1"
  }]
}
```

有关创建 Ident AWS ity and Access Management (IAM) IAM策略的信息，请参阅IAM用户指南中的创建策略。

第 2 部分：过滤器

您可以使用以下筛选器来优化数据。

控件名称	描述
产品标题	产品的标题。
日期	包括过去 30、60 和 90 天、过去 12 个月 (TTM) 和年初至今 (YTD)。您可以选择 “自定义” 来定义特定的开始和结束日期。

有关筛选的更多信息，请参阅《亚马逊 QuickSight 用户指南》 QuickSight中的在亚马逊上筛选数据。

第 3 部分：日期过滤器深入探索

此日期筛选器适用于房源绩效仪表板上的所有指标。

例如，当您使用默认 year-to-date 筛选值时，会显示以下指标：

指标	描述
独一无二的访客	访问过 AWS Marketplace 列表页面的独立个人数量YTD。
新的公开募股协议	将公开发售作为报价知名度、且在有效期内至少有 1 天的协议数量YTD。
新的独立访客	首次访问 AWS Marketplace 上架页面的独立个人数量YTD。

指标	描述
返回独立访客	访问过 AWS Marketplace 列表页面、之前在选定年份开始之前访问过并在中再次访问过的个人数量YTD。

第 4 部分：摘要指标

控制面板的此部分显示了您的流量和公开报价协议的摘要可视化。关键绩效指标 (KPIs) 包括独立访客数量、新的公开发售协议数量、新的独立访客数量以及回访的独立访客数量。您可以看到音量和百分比中的 year-over-year 或 period-over-period 变化。您可以通过更新筛选器部分日期字段中的日期条件来更新日期范围。

指标	描述
独立访客	访问过 AWS Marketplace 商品页面的独立个人数量。
协议	提议者（产品或服务所有者）和接受者（客户）之间签订的开始使用产品的合同。
新的公开募股协议	接受日期在选定日期范围内的以公开报价为报价可见性的协议数量。包括处于激活状态、已过期、已取消和已终止状态。
状态	有效-协议的部分或全部条款已生效。已过期-协议在预先商定的结束日期结束。已取消-接受者选择在协议结束日期之前终止协议。已终止-由于AWS启动的终止事件（例如付款失败），协议在预先商定的结束日期之前结束。续订-使用自动续订等功能，将协议续订为新协议。已替换-协议已被替换报价所取代。

第 5 部分：每月趋势和渠道转化率

控制面板的此部分提供指定日期范围内的流量和协议趋势。关键绩效指标包括独立访客数量和新协议的数量，指的是新的公开募股协议。 按月统计的唯一访客- 提供独立访客数量的月度浏览量。每月的独立访客总数可能超过摘要指标部分中的独立访客总数，因为一个独立访客可以访问多个月。 按月划分的新协议- 提供新公开发行协议数量的月度视图。 漏斗转换- 提供 step-by-step 转化视图。它包括列表页面、采购页面、配置页面和配送页面。 您可以选择独立访客、页面浏览量和唯一帐户作为条形图显示

的样子。当您将鼠标悬停在每个条形上时，您可以看到转换百分比定义为当前条形的值，即前一柱的百分比。

指标	描述
独立访客	访问过 AWS Marketplace 商品页面的独立个人数量。
页面浏览量	AWS Marketplace 列表页面的访问次数。
独特的账户	与访客关联的唯一 AWS 账户 ID 的数量。此指标仅在访客进行身份验证后才适用。

第 6 部分：按渠道划分的流量趋势

仪表板的这一部分深入探讨了每月的流量趋势。您可以选择独立访客和页面浏览量作为条形图显示的样子。您可以在频道筛选器中选择一个或多个频道，以查看所选频道的月度趋势。

第 7 部分：每月流量和协议指标

仪表板的此部分提供独立访客、新协议、转化率和协议总数的月度视图。转化率定义为新协议除以每个月的独立访客。

即使您为部分月份选择日期范围，表中的数据也代表整个月。例如，如果您筛选数据以显示从 2024 年 1 月 15 日到 2024 年 3 月 15 日这段时间，则该表仍将包含 2024 年 1 月 1 日至 2024 年 3 月 31 日的完整月度数据。

表中的数据按月排序，最近一个月显示在最前面。您可以选择按任何可用列对表格进行排序，方法是单击相应的列标题，并可以选择按升序或降序对数据进行排序。

第 8 部分：独立访问者的网络流量来源

仪表板的这一部分按营销来源深入探讨了独特访客。这些类别包括卖家自然发现、卖家主导的营销活动、 AWS 自然发现和AWS以卖家为主导的活动。这些类别由与访客或访问相关的跟踪代码标识。

流量来源	描述
卖家自然发现	通过您网站上的号召性用语、发现点或应用内通知到达您房源的独特访客。

流量来源	描述
卖家主导的营销活动	通过您的付费广告或促销渠道进入您房源的独特访客。
AWS 有机发现	通过自然 AWS 网站流量、AWS主机搜索和 M AWS arketplace 搜索等AWS渠道访问您房源的独特访客。
AWS 营销活动	通过AWS主导的广告系列和付费促销活动进入您房源的独特访客。

第 9 部分：网络流量来源明细

网络流量来源细分表提供了推动流量的营销活动的精细明细分。

- 流量来源-包括 AWS由主导和卖家主导的两个主要类别。
- 流量类型-包括卖家自然发现、卖家主导的营销活动、 AWS 自然发现和主 AWS导的营销活动。
- 渠道-包括各种营销策略，包括付费展示、付费搜索、付费社交媒体、电子邮件、自然广告、内部网站流量、外部链接等。
- 发布商-指发布广告活动的实体，包括 Facebook、Google、 LinkedIn、 AWS、卖家主导等。
- 促销-指已为其生成专用跟踪代码的特定活动。一个例子可以是`psm_linked_post_free-trial-global-ver-a`。

指标	描述
归因协议	每次促销活动的访问量所贡献的协议总数。
转换率	每项促销活动的归因协议除以唯一访客。

您可以将数据以.csv 或 Microsoft Excel 文件的形式从亚马逊 QuickSight 控制面板上的任何图表、图表和表格中导出和下载。有关更多信息，请参阅 Amazon QuickSight 用户指南中的从视觉对象导出数据。

AWS Marketplace 供应商见解

AWS Marketplace Vendor Insights是一项功能，可简化组织执行的软件风险评估，以保护他们信任并符合其标准的采购软件。借助 AWS Marketplace Vendor Insights，买家可以通过单个控制台近乎实时地监控产品的安全状况。 AWS Marketplace 供应商见解可以简化买方的采购流程，并有可能增加卖方的销售额。它通过提供软件产品安全和合规信息的控制面板来减轻买家的评估工作量。

Vendo AWS Marketplace r Insights 控制面板中的所有安全和合规信息均基于从以下来源收集的证据：

- 卖方的自我证明，包括 AWS Marketplace 供应商洞察安全自我评估和共识评估计划问卷 () CAIQ
- 行业标准审计报告（例如，国际标准化组织 ISO 27001）
- AWS Audit Manager，它可以自动从卖家的生产环境中收集证据

AWS Marketplace Vendor Insights 收集有关产品的合规性工件和安全控制信息，并将其显示在仪表板中。控制面板从卖家的自我评估中获取数据，从审计报告中获取证据，并从 AWS 账户中获取实时证据。这些数据流入安全控制措施，然后传输到控制面板供买家查看。实时证据持续更新来自多个来源的数据以呈现最新信息。 AWS Config 已在卖家环境中启用。有关配置、已启用的备份和其他信息的数据会自动更新。例如，假设产品的访问控制合规，并且 Amazon Simple Storage Service (Amazon S3) 存储桶已公开。控制面板将显示控件的状态从合规更改为尚不确定。

在使用 V AWS Marketplace endor Insights AWS 账户 之前，您必须在中设置基准资源和基础架构。设置完成后，Vendo AWS Marketplace r Insights 可以在中为您的软件即服务 (SaaS) 产品收集信息并生成安全配置文件 AWS Marketplace。

内容

- 了解 AWS Marketplace 供应商见解
- 设置 AWS Marketplace 供应商见解
- 查看您的 AWS Marketplace 供应商见解档案
- 在 “ AWS Marketplace 供应商见解” 中管理快照
- 通过 AWS Marketplace Vendor Insights 控制访问权限

了解 AWS Marketplace 供应商见解

AWS Marketplace Vendor Insights 会收集产品的合规性工件和安全控制信息，并将其显示在仪表板中。控制面板从产品所有者的自我评估中获取数据，从审计报告中获取证据，并从 AWS 账户中获取实时证据。这些数据流入安全控制措施，然后传输到控制面板供买家查看。

仪表板显示 AWS Marketplace 供应商洞察从多个安全控制类别收集的循证信息。这提供了对安全配置文件的近乎实时的见解，并减少了买家和卖家之间的讨论。买家可以在几个小时内验证卖家的信息，完成评估。 AWS Marketplace Vendor Insights 为卖家提供了一种 up-to-date 自动保存安全和合规态势信息的机制。他们可以按需与买家共享，无需随机回复问卷。

AWS Marketplace 供应商见解从三个来源收集循证信息：

- 您的供应商自我评估 — 支持的自我评估包括 AWS Marketplace 供应商洞察安全自我评估和共识评估计划问卷 ()。CAIQ
- 您的生产账户 – 在多个控制措施中，有 25 个控制措施支持从您的生产账户收集实时证据。通过使用一个或多个 AWS Config 规则评估 AWS 资源的配置设置，可以生成每个控件的实时证据。 AWS Audit Manager 收集证据并准备好供 AWS Marketplace 供应商见解使用。入职 AWS CloudFormation 模板可自动执行启用现场证据收集所需的先决条件。 AWS Config 已在卖家环境中启用。有关配置、已启用的备份和其他信息的数据会自动更新。例如，假设产品的访问控制合规，并且 Amazon S3 存储桶已公开。控制面板将显示控件的状态从合规更改为尚不确定。
  - 开机 AWS Config 并开始 AWS Audit Manager 服务。
  - 创建 AWS Config 规则和 AWS Audit Manager 自动评估。
  - 配置 AWS Identity and Access Management (IAM) 角色，以便 AWS Marketplace 供应商见解可以提取评估结果。
- 您的 ISO 27001 和 II SOC2 类报告-控制类别映射到国际标准化组织 (ISO) 和系统和组织控制 (SOC2) 报告中的控制。当您与 AWS Marketplace Vendor Insights 共享这些报告时，它可以从这些报告中提取相关证据，并将其显示在控制面板上。

设置 AWS Marketplace 供应商见解

以下过程描述了在 AWS Marketplace 软件即服务 (SaaS) 列表上设置 “ AWS Marketplace 供应商见解” 的高级步骤。

在 SaaS 列表上设置 “ AWS Marketplace 供应商见解”

1. the section called “创建安全配置文件”.

2. （可选）the section called “上传认证文件”。
3. the section called “上传自我评估”.
4. （可选）the section called “启用 AWS Audit Manager 自动评估”。

创建安全配置文件

安全配置文件可让您的买家详细了解您的软件产品的安全状况。安全配置文件使用相关的数据源，包括自我评估、认证和 AWS Audit Manager 自动评估。

> **Note**
>
> 您可以创建有限数量的安全配置文件。要创建更多安全配置文件，可以请求增加限额。有关更多信息，请参阅 AWS 一般参考 中的 AWS 服务限额。

要创建安全配置文件，请执行以下操作：

1. 使用有权访问 AWS Marketplace 卖家账户的IAM用户或角色登录。
2. 选择产品，然后选择 SaaS 以导航到 SaaS 产品页面。
3. 选择产品。
4. 选择 Vendor Insights 选项卡，然后选择联系支持团队以添加安全配置文件。
5. 填写表单，然后选择提交。

   AWS Marketplace 卖家运营团队将创建安全配置文件。安全配置文件准备就绪后，他们将向表单上标识的收件人发送电子邮件通知。

上传认证文件

认证是一种数据源，可在多个维度上提供产品安全状况的证据。 AWS Marketplace 供应商见解支持以下认证：

- 美联储RAMP认证 — 验证是否符合美国政府的云安全标准
- GDPR合规报告 — 证明遵守了《通用数据保护条例》(GDPR) 的要求，保护了个人数据和个人的隐私权
- HIPAA合规报告 — 证明遵守了《健康保险流通与责任法案》(HIPAA) 法规，保护受保护的健康信息

- ISO/IEC27001 审计报告 — 确认符合国际标准化组织 (ISO) /国际电工委员会 (IEC) 27001，强调信息安全标准
- PCIDSS审计报告 — 证明符合安全标准委员会制定的PCI安全标准
- SOC2 第 2 类审计报告 — 确认遵守服务组织控制 (SOC) 数据隐私和安全控制

要上传认证文件，请执行以下操作：

1. 在 Vendor Insights 选项卡上，导航至数据来源部分。
2. 在认证下，选择上传认证文件。
3. 在认证详情下，提供所需的信息并上传认证文件。
4. （可选）在标签下，添加新标签。

> **Note**
>
> 有关标签的信息，请参阅《标记AWS资源用户指南》中的为AWS资源添加标签。

5. 选择上传认证文件。

> **Note**
>
> 该认证会自动与当前的安全配置文件关联。您还可以关联已经上传的认证文件。在产品详细信息页面上，在认证下选择关联认证，从列表中选择认证文件，然后选择关联认证。上传认证文件后，可以使用产品详细信息页面上的下载认证文件按钮进行下载。也可以使用更新认证文件按钮更新认证文件详细信息。

认证状态将更改为，ValidationPending直到认证详细信息得到验证。在处理数据源期间和之后，会出现另一种状态：

- 可用-数据源已上传，系统验证成功完成。
- AccessDenied— AWS Marketplace 供应商见解无法再访问数据源的外部来源参考以供读取。
- ResourceNotFound— 数据源的外部源引用不再可供读 VendorInsights 取。
- ResourceNotSupported— 数据源已上传，但提供的数据源尚不支持。有关验证错误的详细信息，请参阅状态消息。
- ValidationPending— 数据源已上传，但系统验证仍在运行。现阶段没有适合您的操作项目。状态已更新为 “可用” ResourceNotSupported、或 ValidationFailed。

- ValidationFailed— 数据源已上传，但由于一个或多个原因，系统验证失败。有关验证错误的详细信息，请参阅状态消息。

上传自我评估

自我评估是一种数据源，可提供产品安全状况的证据。 AWS Marketplace 供应商见解支持以下自我评估：

- AWS Marketplace 供应商洞察自我评估
- 共识评估倡议问卷 (CAIQ). 有关更多信息，请参阅云安全联盟网站上的 “什么是 CAIQ”。

要上传自我评估，请执行以下操作：

1. 在 https://console.aws.amazon.com/marketplac AWS Marketplace e 上打开控制台。
2. 在 Vendor Insights 选项卡上，导航至数据来源部分。
3. 在自我评估下，选择上传自我评估。
4. 在自我评估详情下，填写以下信息：
   a. 名称 – 输入自我评估的名称。
   b. 类型-从列表中选择评估类型。

   > **Note**
   > 如果您选择了 Vendor Insights 安全自我评估，请选择下载模板以下载自我评估。为电子表格中的每个答案选择是、否或不适用。

5. 要上传已完成的评估，请选择上传自我评估。
6. （可选）在标签下，添加新标签。

   > **Note**
   > 有关标签的信息，请参阅《标记 AWS 资源用户指南》中的为 AWS 资源添加标签。

7. 选择上传自我评估。

> **Note**
>
> 自我评估会自动与当前的安全配置文件相关联。您还可以关联已经上传的自我评估。在产品详细信息页面上，在自我评估下选择关联自我评估，从列表中选择自我评估，然后选择关联自我评估。
>
> 上传自我评估后，可以使用产品详细信息页面上的下载自我评估按钮进行下载。也可以使用更新自我评估按钮更新自我评估详细信息。

状态更新为以下项之一：

- 可用-数据源已上传，系统验证成功完成。
- AccessDenied— 数据源的外部源引用不再可供读 VendorInsights 取。
- ResourceNotFound— 数据源的外部源引用不再可供读 VendorInsights 取。
- ResourceNotSupported— 数据源已上传，但提供的数据源尚不支持。有关验证错误的详细信息，请参阅状态消息。
- ValidationPending— 数据源已上传，但系统验证仍在运行。现阶段没有适合您的操作项目。状态已更新为 “可用” ResourceNotSupported、或 ValidationFailed。
- ValidationFailed— 数据源已上传，但由于一个或多个原因，系统验证失败。有关验证错误的详细信息，请参阅状态消息。

启用 AWS Audit Manager 自动评估

AWS Marketplace Vendor Insights 使用多个 AWS 服务 来自动为您的安全配置文件收集证据。

您需要以下资源 AWS 服务 和资源才能进行自动评估：

- AWS Audit Manager— 为了简化 V AWS Marketplace endor Insights 的设置 StackSets，我们使用了 AWS CloudFormation Stacks 和，它们负责配置和配置必要的资源。堆栈集可创建包含由 AWS Config自动填充控制措施的自动评估。

  有关的更多信息 AWS Audit Manager，请参阅《AWS Audit Manager 用户指南》。
- AWS Config— 堆栈集部署 AWS Config 一致性包以设置必要的 AWS Config 规则。这些规则允许 Audit Manager 自动评估为其中 AWS 服务 部署的其他人收集实时证据 AWS 账户。有关 AWS Config 功能的更多信息，请参阅AWS Config 开发人员指南。

> **Note**
>
> 您可能会注意到，与随后的几个月 AWS Config 相比，在记录的最初一个月中，您的账户活动有所增加。在最初的引导过程中， AWS Config 请查看您账户中您选择 AWS Config 要记录的所有资源。
>
> 如果您运行临时工作负载，则可能会看到活动增加， AWS Config 因为它记录了与创建和删除这些临时资源相关的配置更改。临时工作负载 是临时使用在需要时加载和运行的计算资源。
>
> 临时工作负载的示例包括亚马逊弹性计算云 (AmazonEC2) 竞价型实例、亚马逊EMR作业和。 AWS Auto Scaling AWS Lambda为了避免因运行临时工作负载而增加活动，您可以在关闭的情况下在单独的帐户中运行这些类型的工作负载。 AWS Config 这种方法可以避免增加配置记录和规则评估。

- Amazon S3 – 堆栈集创建以下两种 Amazon Simple Storage Service (Amazon S3) 存储桶：
  - vendor-insights-stack-set-output-bucket-{账号} — 此存储桶包含堆栈集运行的输出。 AWS Marketplace 卖家运营团队使用输出来完成您的自动数据源创建流程。
  - vendor-insights-assessment-reports-bucket-{账号} — 向此 Amazon S3 存储桶 AWS Audit Manager 发布评估报告。有关发布评估报告的更多信息，请参阅《AWS Audit Manager 用户指南》中的评估报告。

  有关 Amazon S3 特征的更多信息，请参阅 Amazon S3 用户指南。
- IAM— 入职堆栈集在您的账户中配置以下 AWS Identity and Access Management (IAM) 角色：
  - 部署 `VendorInsightsPrerequisiteCFT.yml` 模板后，它会创建管理员角色 `AWSVendorInsightsOnboardingStackSetsAdmin` 和运行角色 `AWSVendorInsightsOnboardingStackSetsExecution`。堆栈集使用管理员角色将所需的堆栈同时部署到多个 AWS 区域 中。作为 Vendo AWS Marketplace r Insights 设置过程的一部分，管理员角色扮演执行角色，部署必要的父堆栈和嵌套堆栈。有关自行管理权限的更多信息，请参阅《AWS CloudFormation 用户指南》中的授予自行管理权限。
  - 该`AWSVendorInsightsRole`角色为 V AWS Marketplace endor Insights 提供了阅读 AWS Audit Manager 资源中评估的权限。 AWS Marketplace 供应商见解会在您的 AWS Marketplace 供应商见解资料中显示评估中发现的证据。
  - 为 `AWSVendorInsightsOnboardingDelegationRole` V AWS Marketplace endor Insights 提供了列出和读取`vendor-insights-stack-set-output-bucket`存储桶中对象的权限。此功能允许 AWS Marketplace 目录运营团队协助您设置 “ AWS Marketplace 供应商见解” 档案。

- 该`AWSAuditManagerAdministratorAccess`角色提供管理权限，用于启用或禁用 AWS Audit Manager、更新设置以及管理评估、控制和框架。您或您的团队可以代入此角色，在 AWS Audit Manager中采取措施进行自动评估。

要启用 AWS Audit Manager 自动评估，您必须部署入职堆栈。

部署载入堆栈

为了简化 V AWS Marketplace endor Insights 的设置 StackSets，我们使用了 AWS CloudFormation Stacks 和，它们负责配置和配置必要的资源。如果您有多个账户或多个 AWS 区域 SaaS 解决方案，StackSets 请允许您从中央管理账户部署入职堆栈。

有关的更多信息 CloudFormation StackSets，请参阅《AWS CloudFormation 用户指南》 AWS CloudFormation StackSets中的 “使用”。

AWS Marketplace 供应商见解设置要求您使用以下 CloudFormation 模板：

- `VendorInsightsPrerequisiteCFT`— 设置在您的账户 CloudFormation StackSets 中运行所需的管理员角色和权限。在您的卖家账户中创建此堆栈。
- `VendorInsightsOnboardingCFT`— 设置所需的权限 AWS 服务 并配置相应的IAM权限。这些权限允许 V AWS Marketplace endor Insights 收集在您中运行的 SaaS 产品的数据， AWS 账户 并在您的 AWS Marketplace 供应商见解资料中显示这些数据。在您的卖家账户和通过托管 SaaS 解决方案的生产账户中创建此堆栈 StackSets。

创建 VendorInsightsPrerequisiteCFT 堆栈

通过运行`VendorInsightsPrerequisiteCFT` CloudFormation 堆栈，您可以设置IAM权限以开始加载堆栈集。

要创建 VendorInsightsPrerequisiteCFT 堆栈，请执行以下操作：

1. 查看并从 GitHub网站上的 “供应商见解模板AWS 样本存储库” 文件夹中下载最新`VendorInsightsPrerequisiteCFT.yml`文件。
2. AWS Management Console 使用您的 AWS Marketplace 卖家账户登录，然后在 https://console.aws.amazon.com/cloudformat ion 上打开 AWS CloudFormation 控制台。
3. 在 CloudFormation 控制台导航窗格中，选择堆栈，然后从下拉列表中选择创建堆栈和使用新资源（标准）。（如果导航窗格不可见，请在左上角选择并展开导航窗格。）

4. 在指定模板下，选择上传模板文件。要上传您下载的 `VendorInsightsPrerequisiteCFT.yml` 文件，请使用选择文件。然后选择下一步。
5. 输入堆栈的名称，然后选择下一步。
6. （可选）根据需要配置堆栈选项。

   选择下一步。
7. 在审核页面上，审核您的选择。要进行更改，请在要更改的区域中选择编辑。在创建堆栈之前，必须选中功能区域中的确认复选框。

   选择提交。
8. 创建堆栈后，选择资源选项卡并记下已创建的以下角色：

   - `AWSVendorInsightsOnboardingStackSetsAdmin`
   - `AWSVendorInsightsOnboardingStackSetsExecution`

创建 VendorInsightsOnboardingCFT 堆栈集

通过运行`VendorInsightsOnboardingCFT` CloudFormation 堆栈集，您可以设置所需的权限 AWS 服务 并配置相应的IAM权限。这允许 AWS Marketplace 供应商洞察收集在您中运行的 SaaS 产品的数据， AWS 账户 并将其显示在您的 AWS Marketplace 供应商洞察资料中。

如果您有多账户解决方案，或者您有单独的卖家账户和生产账户，则必须将此堆栈部署到多个账户。StackSets 允许您使用创建先决条件堆栈的管理帐户执行此操作。

堆栈集要使用自行管理的权限部署。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的创建具有自行管理权限的堆栈集。

要创建 VendorInsightsOnboardingCFT 堆栈集，请执行以下操作：

1. 查看并从 GitHub网站上的 “供应商见解模板AWS 样本存储库” 文件夹中下载最新`VendorInsightsOnboardingCFT.yml`文件。
2. AWS Management Console 使用您的 AWS Marketplace 卖家账户登录，然后在 https://console.aws.amazon.com/cloudformat ion 上打开 AWS CloudFormation 控制台...
3. 在 CloudFormation 控制台导航窗格中，选择创建 StackSet。（如果导航窗格不可见，请在左上角选择并展开导航窗格。）
4. 在 “权限” 下，为管理员IAM角色选择角色名称，然后从下拉列表中选择角色名称。AWSVendorInsightsOnboardingStackSetsAdmin

5. 输入**`AWSVendorInsightsOnboardingStackSetsExecution`**作为IAM执行角色名称。
6. 在指定模板下，选择上传模板文件。要上传您下载的 `VendorInsightsOnboardingCFT.yml` 文件，请使用选择文件，然后选择下一步。
7. 提供以下 StackSet 参数，然后选择 “下一步”。
   - `CreateVendorInsightsAutomatedAssessment`— 此参数在您的中设置 AWS Audit Manager 自动评估 AWS 账户。如果您有单独的管理帐户和生产帐户，则应仅为生产帐户选择此选项，而不应为管理帐户选择此选项。
   - `CreateVendorInsightsIAMRoles`— 此参数提供了一个IAM角色，允许 Vendo AWS Marketplace r Insights 读取您的中的评估数据 AWS 账户。
   - `PrimaryRegion` – 此参数为您的 SaaS 部署设置主参数 AWS 区域 。这是在您的中创建 S3 存储桶的区域 AWS 账户。如果您的 SaaS 产品仅部署到一个区域，则该区域为主要区域。
8. 根据需要配置 StackSet 选项。将执行配置保持为非活动，然后选择下一步。
9. 配置部署选项。如果您有多账户解决方案，则可以将堆栈集配置为通过单个操作跨多个账户和区域进行部署。选择下一步。

   > **Note**
   >
   > 如果您有多账户解决方案，我们不建议将解决方案作为单个堆栈集部署到所有账户。请密切注意步骤 7 中定义的参数。您可能需要启用或禁用某些参数，具体取决于您要部署到的账户的类型。 StackSets 在单个部署中将相同的参数应用于所有指定帐户。您可以通过对堆栈集中的账户进行分组来缩短部署时间，但是对于多账户解决方案，您仍然需要多次部署。

   > **Important**
   >
   > 如果您要部署到多个区域，则列出的第一个区域必须是 `PrimaryRegion`。将区域并发选项保留为顺序的默认设置。

10. 在审核页面上，审核您的选择。要进行更改，请在要更改的区域中选择编辑。在创建堆栈集之前，必须选中功能区域中的确认复选框。

    选择提交。

    每个区域的堆栈集大约需要 5 分钟才能完成。

查看您的 AWS Marketplace 供应商见解档案

您在 AWS Marketplace 供应商见解中的个人资料提供了重要信息，供买家在评估您的产品时使用。出于数据保护目的，我们建议您保护您的 AWS 账户 凭证并使用 AWS Identity and Access Management (IAM) 设置个人用户。通过该方法，每个用户只获得履行其工作职责所需的权限。有关使用 IAM 创建用户的更多信息，请参阅the section called “创建或使用组”。

买家可以使用 “ AWS Marketplace 供应商见解” 控制面板评估您的产品。在那里，买家可以看到由您添加到配置文件中的数据来源定义的产品概述。安全配置文件由 10 个类别的多个安全控制措施定义。

用于定义数据的 10 个安全类别如下：

- 访问管理
- 应用程序安全性
- 审计、合规性和安全性策略
- 业务弹性
- 数据安全性和隐私
- 最终用户设备和移动安全
- 人力资源
- 基础设施安全性
- 风险管理和事件响应
- 安全和配置策略

有关更多信息，请参阅《AWS Marketplace买家指南》中的了解控制类别。

设置和使用 Vendo AWS Marketplace r Insights，即表示您同意遵守 AWS 服务 条款和数据隐私规则，以保护用户信息的私密性和安全性。有关 AWS 数据隐私条款的更多信息，请参阅数据隐私常见问题解答。有关服务条款的更多信息，请参阅 AWS 服务 条款。

以卖家身份查看您的安全配置文件

完成自我评估并添加其他真实证据后，请务必查看您作为卖家的配置文件。您将需要查看添加到您的配置文件中的信息。

> **Note**
>
> 在您请求 Vendor Insights 支持团队更新其可 AWS Marketplace 见性之前，买家无法看到此个人资料。支持团队完成更新后，签署保密协议 (NDA) 的买家可以访问安全配置文件。
> 如果您想从 V AWS Marketplace endor Insights 个人资料中删除订阅者的个人身份信息 (PII) 数据，请通过联系AWS Support开始支持案例。

要在买家查看您的安全配置文件时查看它，请执行以下操作：

1. 登录到 AWS Management Console。
2. 转到门户中的 SaaS 产品页面。
3. 选择具有关联安全配置文件的产品。
4. 选择 Vendor Insights 选项卡，然后选择查看最新发布的快照。
5. 在概述选项卡上，将显示您上传的所有证书徽章。
6. 选择安全性与合规性选项卡，您可以在其中查看从多个控制措施收集的数据。要查看更多详细信息，可选择每个控制措施集。

在 “ AWS Marketplace 供应商见解” 中管理快照

快照是安全配置文件的一种 point-in-time 状态。在 V AWS Marketplace endor Insights 中，您可以使用快照在任何给定时间评估卖家的产品。作为卖家，您可以比较不同时间配置文件的安全状况或不同安全配置文件的最新快照，以支持您的决策。除了提供有关数据新鲜度和来源的透明度外，快照还提供了必要的安全信息。

在 AWS Marketplace 控制台的 Vendo AWS Marketplace r Insig hts 快照摘要部分，您可以查看创建和发布计划的以下快照详细信息：

- 上次创建的快照 – 上次为此配置文件创建的快照。
- 下一次计划创建 - 计划接下来创建的快照。
- 创建频率 - 创建快照之间的时间长度或创建快照的频率。
- 下一次计划发布 - 计划接下来发布的快照。
- 暂存时间 – 快照至少暂存此时间长度，然后有资格在快照发布事件期间发布。
- 发布频率 - 发布事件之间的时间长度。

在快照列表部分，快照状态如下所示：

- 已发布 – 快照是公开的，可供拥有该产品权限的用户查看。
- 待发布 – 快照已完成或强制进入最低暂存期，并计划在下一个版本中发布。
- 私有 – 在激活安全配置文件之前创建的快照，或者存在验证错误且对公众不可见。私有快照仅在卖家可见性中保留。

主题

- 创建快照
- 查看快照
- 导出快照
- 查看最新发布的快照
- 推迟快照发布
- 更改快照列表的首选项

创建快照

要为您的配置文件创建快照，请按照以下步骤操作。每天最多可以创建 20 个快照。

1. 登录 AWS Management Console 并打开AWS Marketplace 控制台。
2. 选择 Vendor Insights。
3. 从 Vendor Insights 中选择产品。
4. 在产品配置文件页面上，转到快照列表，然后选择创建新快照。
5. 将显示一条消息，通知您快照计划将发生变化。选择创建。

> **Note**
>
> 创建新快照后，快照计划会发生变化。新快照的计划时间与您手动创建的快照的时间相同。此消息包括新的计划。

新快照将在 30 分钟内创建并添加到快照列表中。新快照的创建状态为待发布。在状态更改为已发布之前，任何人都无法查看新快照。

查看快照

要查看配置文件的快照，请按照以下步骤操作。

1. 登录 AWS Management Console 并打开AWS Marketplace 控制台。
2. 选择 Vendor Insights。
3. 从 Vendor Insights 中选择产品。
4. 在产品配置文件页面上，转到快照列表，然后选择要查看的快照的快照 ID。
5. 完成后，选择返回退出快照视图。

导出快照

您可以导出为 JSON 或 CSV 格式。要导出快照，请按照以下步骤操作。

1. 登录 AWS Management Console 并打开AWS Marketplace 控制台。
2. 选择 Vendor Insights。
3. 从 Vendor Insights 中选择产品。
4. 在产品配置文件页面上，转到快照列表，然后选择要导出的快照的快照 ID。
5. 选择导出。
6. 从下拉列表中选择下载 (JSON) 或下载 (CSV)。

查看最新发布的快照

最新发布的快照是用户用来查看和评估产品运行状况的快照。了解最新发布的快照中的内容非常重要，这样才能确保用准确的信息描绘您的产品。要查看配置文件的最新快照，请按照以下步骤操作。

1. 登录 AWS Management Console 并打开AWS Marketplace 控制台。
2. 选择 Vendor Insights。
3. 从 Vendor Insights 中选择产品。
4. 在产品配置文件页面上，转到快照列表，然后选择要查看的快照的快照 ID。
5. 选择查看最新发布的快照。
6. 完成后，选择返回退出快照视图。

推迟快照发布

要推迟发布配置文件快照，您可以推迟特定快照 ID 的快照发布。

1. 登录 AWS Management Console 并打开AWS Marketplace 控制台。
2. 选择 Vendor Insights。
3. 从 Vendor Insights 中选择产品。
4. 在产品配置文件页面上，转到快照列表，然后选择要推迟发布的快照的快照 ID。
5. 在快照摘要中，选择推迟快照发布。
6. 将显示一条消息，通知您快照计划将发生变化。选择推迟。

将显示一条成功消息，表示您已成功推迟此产品的快照发布。

更改快照列表的首选项

创建快照后，您可以在快照列表中更改快照查看方式的首选项。

1. 登录 AWS Management Console 并打开AWS Marketplace 控制台。
2. 选择 Vendor Insights。
3. 从 Vendor Insights 中选择产品。
4. 在产品配置文件页面上，转到快照列表，然后选择要更改的快照的快照 ID。
5. 选择首选项图标。您可以为快照自定义以下首选项：
   - 页面大小 - 选择要在每页上列出多少快照：每页10 个资源、20 个资源或 50 个资源。
   - 换行 - 选择换行选项以查看整条记录。
   - 时间格式 - 选择想要绝对、相对还是 ISO 快照。
   - 可见列 - 选择要显示快照详细信息的选项：快照 ID、状态和创建日期。

通过 AWS Marketplace Vendor Insights 控制访问权限

AWS Identity and Access Management (IAM) AWS 服务 可以帮助您控制对 AWS 资源的访问权限。IAM无需支付额外费用即可使用。 AWS 服务 如果您是管理员，则可以控制谁可以进行身份验证（登录）和授权（拥有权限）使用 AWS Marketplace 资源。 AWS Marketplace Vendor Insights 用于IAM控制对卖家数据、评估、卖家自我认证和行业标准审计报告的访问权限。

控制谁能在中做什么的推荐方法 AWS Marketplace 管理门户 是使用IAM创建用户和群组。然后，您将用户添加到组并管理组。您可以为该群组分配提供只读权限的策略或权限。如果您有其他用户需要只读访问权限，您可以将其添加到您创建的组，而不是为用户添加权限。

策略 是定义应用于用户、组或角色的权限的文档。权限确定用户可以在 AWS中执行的操作。策略通常允许访问特定操作，并且可以选择授予允许对特定资源（例如 Amazon EC2 实例、Amazon S3 存储桶等）执行这些操作。策略还可以显式拒绝访问。权限 是策略中允许或拒绝访问特定资源的语句。

> **Important**
>
> 您创建的所有用户使用其凭证进行身份验证。但是，他们使用相同的 AWS 账户。用户进行的任何更改可能影响整个账户。

AWS Marketplace 已定义权限以控制具有这些权限的用户在中可以执行的操作 AWS Marketplace 管理门户。还有一些策略可以 AWS Marketplace 创建和管理这些策略，这些策略结合了多种权限。`AWSMarketplaceSellerProductsFullAccess` 策略授予用户对 AWS Marketplace 管理门户中产品的完全访问权限。

有关可用操作、资源和条件键的更多信息，请参阅《服务授权参考》中的 Vendo AWS Marketplace r Insights 的操作、资源和条件密钥。

AWS Marketplace 供应商洞察卖家的权限

您可以在 AWS Marketplace 供应商见解IAM策略中使用以下权限。您可以将权限合并到单个IAM策略中，以授予所需的权限。

`CreateDataSource`

`CreateDataSource` 允许用户创建新的数据来源资源。支持的数据来源包括：

- SOC2Type2AuditReport
- ISO27001AuditReport
- AWSAuditManagerSecurityAutomatedAssessment
- FedRAMPCertification
- GDPRComplianceReport
- HIPAAComplianceReport
- PCIDSSAuditReport

- SecuritySelfAssessment

操作组：读写

所需资源：无

创建资源：DataSource

DeleteDataSource

DeleteDataSource 允许用户删除他们拥有的数据来源。数据来源必须与任何要删除的配置文件断开关联。有关更多信息，请参阅 the section called “AssociateDataSource”。

操作组：读写

所需资源：DataSource

GetDataSource

GetDataSource 允许用户检索数据来源的详细信息。数据来源的详细信息包括元数据信息，例如关联的时间戳、原始创建参数和处理信息（如有）。

操作组：只读、读写

所需资源：DataSource

UpdateDataSource

UpdateDataSource 允许用户更新数据来源的详细信息。详细信息包括元数据信息，例如名称和来源信息（例如，角色、来源 Amazon 资源名称 (ARN) 和源内容）。

操作组：只读、读写

所需资源：DataSource

ListDataSources

ListDataSources 允许用户列出他们拥有的数据来源。

操作组：只读、读写、仅列出

所需资源：无

CreateSecurityProfile

CreateSecurityProfile 允许用户创建新的安全配置文件。安全配置文件是一种用于管理快照生成方式和时间的资源。用户还可以通过控制配置文件的状态和适用条款来控制买家访问快照的方式。

操作组：只读、读写、仅列出

所需资源：无

创建资源：SecurityProfile

ListSecurityProfiles

ListSecurityProfiles 允许用户列出他们拥有的安全配置文件。

操作组：只读、读写、仅列出

所需资源：无

GetSecurityProfile

CreateSecurityProfile 允许用户获取安全配置文件的详细信息。

操作组：只读和读写

所需资源：SecurityProfile

AssociateDataSource

AssociateDataSource允许用户将现有配置文件DataSource与 AWS Marketplace 供应商见解配置文件相关联。用户可以通过将数据来源与配置文件关联或取消关联来控制快照的内容。

操作组：读写

所需资源：SecurityProfile 和 DataSource

DisassociateDataSource

DisassociateDataSource允许用户取消现有DataSource与 AWS Marketplace 供应商见解配置文件的关联。用户可以通过将数据来源与配置文件关联或取消关联来控制快照的内容。

操作组：读写

所需资源：SecurityProfile 和 DataSource

UpdateSecurityProfile

UpdateSecurityProfile 允许用户修改安全配置文件属性，例如名称和描述。

操作组：读写

所需资源：SecurityProfile

ActivateSecurityProfile

ActivateSecurityProfile 允许用户为安全配置文件设置 Active 状态。激活安全配置文件后，可以 Staged 状态下创建新快照，这样如果满足其他条件就可以发布它们。有关更多信息，请参阅 the section called “UpdateSecurityProfileSnapshotReleaseConfiguration”。

具有至少一个Released快照Active的安全配置文件有资格为最终用户发现 Vendo AWS Marketplace r Insights。

操作组：读写

所需资源：SecurityProfile

DeactivateSecurityProfile

DeactivateSecurityProfile 允许用户为安全配置文件设置 Inactive 状态。安全配置文件的这种终端状态等同于从共享状态中删除配置文件。只有当安全配置文件没有活跃订阅用户时，用户才能停用该配置文件。

操作组：读写

所需资源：SecurityProfile

UpdateSecurityProfileSnapshotCreationConfiguration

UpdateSecurityProfileSnapshotCreationConfiguration 允许用户为快照创建配置定义自定义计划。使用此操作可以覆盖每周创建的默认创建配置。

用户可以使用此操作来更改计划，包括取消计划、将计划推迟到将来的某个日期，或者提前开始创建新的快照。

操作组：读写

所需资源：SecurityProfile

UpdateSecurityProfileSnapshotReleaseConfiguration

UpdateSecurityProfileSnapshotReleaseConfiguration 允许用户为快照发布配置定义自定义计划。此操作可以覆盖每周发布的默认创建配置，其中有两天的暂存期可供审核。

用户可以使用此操作来更改计划，包括取消计划或将计划推迟到将来的某个日期。

操作组：读写

所需资源：SecurityProfile

ListSecurityProfileSnapshots

ListSecurityProfileSnapshots 允许用户列出他们拥有的安全配置文件的快照。

操作组：只读、仅列出和读写

所需资源：SecurityProfile

GetSecurityProfileSnapshot

GetSecurityProfileSnapshot 允许用户获取他们拥有的安全配置文件的快照。

操作组：只读和读写

所需资源：SecurityProfile

TagResource

TagResource 允许用户为资源添加新标签。支持的资源包括 SecurityProfile 和 DataSource。

操作组：标记

可选资源：SecurityProfile 和 DataSource

UntagResource

UntagResource 允许用户从资源中删除标签。支持的资源包括 SecurityProfile 和 DataSource。

操作组：标记

可选资源：SecurityProfile 和 DataSource

ListTagsForResource

ListTagsForResource 允许用户列出资源的资源标签。支持的资源包括 SecurityProfile 和 DataSource。

操作组：只读

可选资源：SecurityProfile 和 DataSource

其他 资源

《IAM用户指南》中的以下资源提供了有关入门和使用的更多信息IAM：

- IAM 中的安全最佳实践
- 管理IAM策略
- 将策略附加到IAM用户组
- IAM身份（用户、用户组和角色）
- 在您的 AWS 账户中创建IAM用户
- 创建IAM用户组
- 使用策略控制对 AWS 资源的访问权限

AWS Marketplace 安全

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的 安全性和云中 的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。 AWS 还为您提供可以安全使用的服务。作为 AWS 合规性计划的一部分，我们的安全措施的有效性定期由第三方审计员进行测试和验证。要了解适用的合规计划 AWS Marketplace，请参阅按合规计划划分的范围内的AWS 服务。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用时如何应用分担责任模型 AWS Marketplace。以下主题向您介绍如何配置 AWS Identity and Access Management 以管理访问权限， AWS Marketplace 以实现您的安全和合规性目标。您还可以学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS Marketplace 资源。

要详细了解与您在中提供的产品相关的安全政策和其他政策 AWS Marketplace，请参阅以下主题：

- 基于 AMI 的产品要求
- 基于容器的产品要求
- SaaS 产品准则
- 对专业服务产品的要求

> **Note**
>
> 要了解AWS数据产品的 Data Exchange 安全性，请参阅《AWS数据交换用户指南》中的安全性。
> 要了解买家的安全性 AWS Marketplace，请参阅《买家指南》 AWS Marketplace中的 “AWS Marketplace 安全性”。

主题

- 控制对 AWS Marketplace 管理门户的访问
- AWS Marketplace 卖家策略和权限

- AWSAWS Marketplace 卖家托管政策
- AWS Marketplace 商务分析服务账号权限
- Amazon SQS 权限
- AWS Marketplace 计量和授权 API 权限
- 将服务相关角色用于 AWS Marketplace
- 使用记录 AWS Marketplace 计量API呼叫 AWS CloudTrail

控制对 AWS Marketplace 管理门户的访问

AWS Identity and Access Management (IAM) AWS 服务 可以帮助您控制对 AWS 资源的访问权限。如果您是管理员，则可以控制谁可以进行身份验证（登录）和授权（拥有权限）使用 AWS Marketplace 资源。IAM无需支付额外费用即可使用。 AWS 服务

控制谁能在中执行的操作的推荐方法 AWS Marketplace 管理门户 是使用IAM创建用户和群组。然后，您将用户添加到组并管理组。例如，如果应允许 John 查看您的产品，则为他创建用户并将其用户添加到您为进行只读访问而创建的组。您可以为该群组分配提供只读权限的策略或权限。如果您有其他用户需要只读访问权限，您可以将其添加到您创建的组，而不是为用户添加权限。如果 John 的角色发生变化而不再需要只读访问权限，您可以从组中删除 John。

策略 是定义应用于用户、组或角色的权限的文档。反过来，权限确定用户可以在 AWS中执行的操作。策略通常允许访问特定操作，并且可以选择授予允许对特定资源（例如 Amazon EC2 实例、Amazon S3 存储桶等）执行这些操作。策略还可以显式拒绝访问。权限 是策略中允许或拒绝访问特定资源的语句。您可以通过以下方式声明任意权限：“A has permission to do B to C.”例如，Jane (A) 有权从 John 的 Amazon Simple Queue Service 队列 (C) 读取消息 (B)。每当 Jane 向 Amazon SQS 发送使用 John 队列的请求时，该服务都会检查她是否获得许可。它还进一步检查请求是否满足 John 在权限中指定的条件。

> **Important**
>
> 您创建的所有用户使用其凭证进行身份验证。但是，他们使用相同的 AWS 账户。用户进行的任何更改可能影响整个账户。

AWS Marketplace 已定义权限以控制拥有这些权限的用户可以执行的操作 AWS Marketplace 管理门户。还有一些策略可以 AWS Marketplace 创建和管理这些策略，这些策略结合了多种权限。

以下资源提供了有关入门和使用的更多信息IAM。

- 创建管理用户
- IAM 中的安全最佳实践
- 管理IAM策略
- 将策略附加到IAM用户组
- IAM身份（用户、群组和角色）
- 使用策略控制对 AWS 资源的访问权限

下面主题提供一些用于创建用户和组以及以用户身份登录的高级指导原则。

主题

- 创建用户
- 创建或使用组
- 作为用户登录

创建用户

要允许贵公司的人员登录 AWS Marketplace 管理门户，请为每个需要访问权限的人员创建一个用户。

创建用户

1. 登录 AWS Management Console 并打开IAM控制台，网址为https://console.aws.amazon.com/iam/。
2. 在导航窗格中的访问管理下，选择用户，然后选择创建用户。
3. 在带编号的文本框中，为每个要创建的用户输入名称。
4. 清除为每个用户生成访问密钥旁的复选框，然后选择创建。

为刚创建的每个用户分配密码

1. 在用户列表中，选择新用户的名称。
2. 选择安全凭证选项卡，然后选择管理密码。
3. 为自动生成的密码或自定义密码选择一个选项。或者，若要求用户在下次登录时选择新密码，请选中要求用户在下次登录时创建新密码的复选框。选择 应用。
4. 选择 “下载凭据”，将登录凭证和账户特定的登录信息保存到计算机上URL以逗号分隔的值 () CSV 文件中。然后选择关闭。

> **Note**
>
> 要使用您刚刚创建的登录凭据登录，用户必须导航到您的账户专属登录。URLURL它位于您刚刚下载的凭据文件中，也可以在IAM控制台上找到。有关更多信息，请参阅《IAM用户指南》中的IAM用户如何登录到您 AWS 账户的。

> **Tip**
>
> 即使您是 AWS 账户 所有者，也可以为自己创建登录凭证。对于在 AWS Marketplace 中作为用户工作的所有人（甚至包括账户所有者），这是建议的最佳实践。有关如何为自己创建具有管理权限的用户的说明，请参阅《用户指南》中的创建管理IAM用户。

创建或使用组

创建用户后，请创建组，创建用于访问 AWS Marketplace 管理门户中页面的权限，将这些权限添加到组，然后将用户添加到组。

当您将权限分配到组时，就允许该组的任何成员执行特定操作。当您向组中添加新用户时，该用户自动获得分配给该组的权限。组可以拥有执行多个操作的权限。我们建议使用 AWS Marketplace 托管策略，而不是创建您自己的策略。

为组分配托管策略 AWS Marketplace

1. 打开IAM控制台，网址为https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择组，然后选择要向其附加策略的组。
3. 在组的摘要页面上，在权限选项卡下，选择附加策略。
4. 在附加策略页上的筛选条件: 旁边，输入 awsmarketplace。
5. 选择要附加的一个或多个策略，然后选择附加策略。

创建具有 AWS Marketplace 管理门户 权限的策略

1. 打开IAM控制台，网址为https://console.aws.amazon.com/iam/。
2. 在导航窗格中选择策略，然后选择创建策略。
3. 在策略生成器旁，选中选择。

4. 在编辑权限页面上，执行以下操作：
   a. 对于效果，选择允许。
   b. 对于 AWS 服务，选择 AWS Marketplace 管理门户。
   c. 对于操作，选择要允许的一个或多个权限。
   d. 选择添加声明。
   e. 选择下一步。
5. 在查看策略页面上，执行以下操作：
   a. 对于策略名称，输入此策略的名称。请记录此策略名称，因为您在后面的步骤中需要使用它。
   b. （可选）对于描述，输入此策略的描述。
   c. 请选择创建策略。

创建具有适当权限的IAM群组并将用户添加到群组中

1. 打开IAM控制台，网址为https://console.aws.amazon.com/iam/。
2. 在导航窗格中，选择组，然后选择创建新组。
3. 对于组名称：，键入组的名称。然后选择下一步。
4. 在附加策略页面上，执行以下操作：
   a. 对于筛选条件：，选择客户管理型策略。
   b. 选中要附加到此组的策略名称旁边的复选框。这通常是您刚刚创建的策略。
   c. 选择下一步。
5. 选择创建组。
6. 在组列表中找到您的新组，然后选中它旁边的复选框。选择组操作，然后单击将用户添加到组。
7. 选中要添加到组的每个用户旁边的复选框，然后选择添加用户。

作为用户登录

在中创建用户后IAM，用户可以使用自己的登录凭据登录。为此，他们需要使用与您关联URL的唯一值 AWS 账户。您可以获取登录信息并将其分发URL给您的用户。

要获得您账户的唯一登录信息 URL

1. 打开IAM控制台，网址为https://console.aws.amazon.com/iam/。

2. 在导航窗格中，选择控制面板。
3. 在内容窗格顶部附近，找到IAM用户的登录链接：并记下登录链接，其格式如下所示：

```
https://AWS_account_ID.signin.aws.amazon.com/console/
```

> **Note**
>
> 如果您希望登录页面包含您的公司名称（或其他友好标识符）而不是您的 AWS 账户 ID，则可以通过选择 “自定义” 为您的账户创建别名。URL有关更多信息，请参阅《IAM用户指南》中的您的 AWS 账户 ID 及其别名。

4. 将此URL信息以及您为每个人创建的登录凭据分发给贵公司中可以与之合作的人。 AWS Marketplace指导他们在访问之前使用您账户的唯一登录URL信息进行登录。 AWS Marketplace

AWS Marketplace 卖家策略和权限

AWS Marketplace 有多个托管策略可供您使用 AWS Marketplace 管理门户。此外，您还可以使用个人权限来创建自己的 AWS Identity and Access Management (IAM) 策略。

您还可以提供对 “设置”、“联系我们”、“文件上传” 和 “见解” 选项卡的精细访问权限。 AWS Marketplace 管理门户 精细访问权限使您可以执行以下操作：

- 向其他人员授予管理和使用您 AWS 账户 中的资源的权限，而无需共享您的密码或访问密钥。
- 向多人授予对各种资源的精细权限。例如，您可以允许某些用户查看 AWS Marketplace 管理门户中的设置选项卡。对于其他用户，您可以允许在设置和联系我们选项卡中进行编辑。

> **Note**
>
> 有关 Data Exchange 中针对AWS数据产品的策略和权限的更多信息，请参阅《AWSAWS数据交换用户指南》中的 Data Exchang e 中的身份和访问管理。
> 有关买家政策和权限的更多信息，请参阅《 AWS Marketplace 买家指南》AWS Marketplace 中的控制 AWS Marketplace 订阅访问权限。

AWS Marketplace 卖家政策

您可以使用以下托管策略向用户提供对 AWS Marketplace 管理门户的受控访问权限：

AWSMarketplaceSellerFullAccess

允许完全访问 AWS Marketplace 管理门户 和其他 AWS 服务（例如 Amazon 系统映像 (AMI) 管理）中的所有页面。

AWSMarketplaceSellerProductsFullAccess

允许完全访问 AWS Marketplace 管理门户中的产品页面。

AWSMarketplaceSellerProductsReadOnly

允许只读访问 AWS Marketplace 管理门户中的产品页面。

> **Important**
>
> AWS Marketplace 买家可以使用托管政策来管理他们购买的订阅。您使用的托管策略的名称以 AWS Marketplace 管理门户 开头`AWSMarketplaceSeller`。在中搜索策略时IAM，请务必搜索以开头的策略名称`AWSMarketplaceSeller`。

AWS Marketplace 还为特定场景提供了专门的托管策略。有关 AWS Marketplace 卖家AWS托管政策的完整列表及其提供的权限说明，请参阅AWSAWS Marketplace 卖家托管政策。

AWS Marketplace 卖家权限

您可以在IAM策略中使用以下权限 AWS Marketplace 管理门户：

aws-marketplace-management:PutSellerVerificationDetails

允许访问以启动 “了解您的客户” (KYC) 流程。

aws-marketplace-management:GetSellerVerificationDetails

允许访问查看中的KYC状态 AWS Marketplace 管理门户。

aws-marketplace-management:PutBankAccountVerificationDetails

允许访问以启动银行账户验证流程。

aws-marketplace-management:GetBankAccountVerificationDetails

允许访问以在 AWS Marketplace 管理门户中查看银行账户验证状态。

aws-marketplace-management:PutSecondaryUserVerificationDetails

允许在中添加次要用户 AWS Marketplace 管理门户。

aws-marketplace-management:GetSecondaryUserVerificationDetails

允许访问以在 AWS Marketplace 管理门户中查看次要用户状态。

aws-marketplace-management:GetAdditionalSellerNotificationRecipients

允许访问电子邮件联系人以获取 AWS Marketplace 通知。

aws-marketplace-management:PutAdditionalSellerNotificationRecipients

允许访问更新电子邮件联系人以获取 AWS Marketplace 通知。

tax:PutTaxInterview

允许访问以在 AWS Marketplace 管理门户中进行税务调查。

tax:GetTaxInterview

允许访问以在 AWS Marketplace 管理门户中查看税务调查状态。

tax:GetTaxInfoReportingDocument

允许 AWS Marketplace 卖家从税务控制面板查看和下载税务文件（例如 1099-K 表格）

payments:CreatePaymentInstrument

允许向添加银行账户的权限 AWS Marketplace 管理门户。

payments:GetPaymentInstrument

允许访问查看中的现有银行账户 AWS Marketplace 管理门户。

aws-marketplace:ListTasks

允许访问以查看待卖家操作的任务列表。

aws-marketplace:DescribeTask

允许访问以查看任何待卖家操作的任务详细信息。

aws-marketplace:UpdateTask

允许访问以编辑待卖家操作的任务。

aws-marketplace:CompleteTask

允许访问以提交对待卖家操作的任务所做的编辑。

support:CreateCase

允许访问在中创建 AWS Marketplace 案例 AWS Marketplace 管理门户。

`aws-marketplace-management:viewSupport`

允许访问 AWS Marketplace 管理门户中的客户支持资格页面。

`aws-marketplace-management:viewReports`

允许访问 AWS Marketplace 管理门户中的报告页面。

`aws-marketplace:ListEntities`

允许访问列出中的对象 AWS Marketplace 管理门户。需要访问 AWS Marketplace 管理门户中的文件上传、优惠和合作伙伴页面。

> **Note**
>
> 要允许访问权限查看 “设置” 选项卡，您可以使用此权限、`ListEntity`权限和以下 Amazon 资源名称 (ARN)：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`。

`aws-marketplace:DescribeEntity`

允许访问中对象的详细信息 AWS Marketplace 管理门户。需要访问 AWS Marketplace 管理门户中的文件上传、优惠、合作伙伴和协议页面。

> **Note**
>
> 要允许访问查看 “设置” 选项卡，您可以使用此权限、`DescribeEntity`权限和以下内容ARN：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/*`。

`aws-marketplace:StartChangeSet`

允许在中创建产品变更的权限 AWS Marketplace 管理门户。需要在 AWS Marketplace 管理门户中的文件上传、优惠、合作伙伴和协议页面上做出更改。

> **Note**
>
> 要允许访问注册为卖家 AWS Marketplace，您可以使用此权限、`catalog:ChangeType:"CreateSeller"`条件键和以下内容ARN：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`。

要允许访问更新中的卖家资料 AWS Marketplace，您可以使用此权限、`catalog:ChangeType: "UpdateInformation"`条件键和以下内容ARN：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`。
要允许访问更新 Amazon Web Services 的付款首选项，您可以使用此权限、`catalog:ChangeType: "UpdateDisbursementPreferences"`条件密钥和以下内容ARN：`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`。

`aws-marketplace:SearchAgreements`

允许在 “协议” 页面上查看协议的高级列表，以及在 “合作伙伴” 页面上查看与渠道合作伙伴之间的ISVs机会。

`aws-marketplace:DescribeAgreement`

允许在 “协议” 页面上查看高级协议详细信息，以及在 “合作伙伴” 页面上查看与渠道合作伙伴之间的ISVs机会。

`aws-marketplace:GetAgreementTerms`

允许在 “协议” 页面上查看所有协议条款的详细信息，以及在 “合作伙伴” 页面上查看与渠道合作伙伴之间的ISVs机会。

`aws-marketplace:GetSellerDashboard`

允许访问 AWS Marketplace 管理门户中的见解页面上的控制面板。

> **Note**
>
> 要使用户能够访问管理产品页面，您必须使用`AWSMarketplaceSellerProductsFullAccess` 或`AWSMarketplaceSellerProductsReadOnly` 托管权限。

您可以将上述权限合并到一个IAM策略中，以授予所需的权限。请见以下 示例。

示例 1：查看KYC状态的权限

要授予查看中KYC状态的权限 AWS Marketplace 管理门户，请使用与以下示例类似的策略。

要授予查看中KYC状态的权限 AWS Marketplace 管理门户，请使用与以下示例类似的策略。

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

示例 2：为专属优惠创建升级和续订的权限

要授予查看和使用协议页面为专属优惠创建升级和续订的权限，请使用与以下示例类似的策略。

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

示例 3：访问优惠页面和创建新的专属优惠的权限

要授予查看和使用优惠页面权限以查看现有专属优惠并创建专属优惠，请使用与以下示例类似的策略。

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

示例 4：访问设置页面的权限

要授予查看和使用设置页面的权限，请使用与以下示例类似的策略。

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

示例 1：访问文件上传页面的权限

要授予查看和使用文件上传页面的权限，请使用与以下示例类似的策略：

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

使用IAM群组

或者，您可以创建单独的IAM群组来授予对中每个页面的访问权限 AWS Marketplace 管理门户。一个用户可能属于多个组。因此，如果用户需要访问多个页面，则您可以将用户添加到所有适当的组。例如，创建一个IAM群组并授予该群组访问 Insights 页面的权限，创建另一个群组并授予该群组访问文件上传页面的权限，依此类推。如果用户需要访问见解页面和文件上传页面的权限，则可以将用户添加到这两个组。

有关用户和群组的更多信息，请参阅《用户指南》中的IAM身份（用户、群组和角色）。IAM

AWSAWS Marketplace 卖家托管政策

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的客户托管式策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 当新服务启动或现有服务 AWS 服务 有新API操作可用时，最有可能更新 AWS 托管策略。

有关更多信息，请参阅《IAM用户指南》中的AWS 托管策略。

本部分列出了用于管理卖家对 AWS Marketplace的访问的每项策略。有关买家政策的信息，请参阅《买家指南》中的 AWS Marketplace AWS Marketplace 买家AWS 托管政策。

主题

- AWS 托管策略：AWSMarketplaceAmiIngestion
- AWS 托管策略：AWSMarketplaceFullAccess
- AWS 托管策略：AWSMarketplaceGetEntitlements
- AWS 托管策略：AWSMarketplaceMeteringFullAccess
- AWS 托管策略：AWSMarketplaceMeteringRegisterUsage
- AWS 托管策略：AWSMarketplaceSellerFullAccess
- AWS 托管策略：AWSMarketplaceSellerProductsFullAccess
- AWS 托管策略：AWSMarketplaceSellerProductsReadOnly
- AWS 托管策略：AWSVendorInsightsVendorFullAccess
- AWS 托管策略：AWSVendorInsightsVendorReadOnly
- AWS MarketplaceAWS 托管策略的更新

AWS 托管策略：`AWSMarketplaceAmiIngestion`

您可以使用此策略创建服务角色，然后使用该角色代表您执行操作。 AWS Marketplace 有关使用 `AWSMarketplaceAmiIngestion` 的更多信息，请参阅允许 AWS Marketplace 访问您的 AMI。

此政策用于向贡献者授予权限， AWS Marketplace 允许他们复制您的 Amazon 系统映像 (AMIs) 以便在上架这些映像 AWS Marketplace。

权限详细信息

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:ModifySnapshotAttribute"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/snap-*"
        },
        {
            "Action": [
                "ec2:DescribeImageAttribute",
                "ec2:DescribeImages",
```

```
                "ec2:DescribeSnapshotAttribute",
                "ec2:ModifyImageAttribute"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AWS 托管策略：**AWSMarketplaceFullAccess**

您可以将该AWSMarketplaceFullAccess策略附加到您的IAM身份。

该政策授予管理权限，允许卖家 AWS Marketplace 和买家完全访问和相关服务。这些权限包括以下能力：

- 订阅和取消订阅 AWS Marketplace 软件。
- 从中管理 AWS Marketplace 软件实例 AWS Marketplace。
- 在您的账户中创建和管理 Private Marketplace。
- 提供对亚马逊和亚马逊 S EC2 EC2 ystem AWS CloudFormation s Manager 的访问权限。

权限详细信息

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:*",
                "cloudformation:CreateStack",
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:List*",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
```

```
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:DescribeVpcs",
                "ec2:RunInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CopyImage",
                "ec2:DeregisterImage",
                "ec2:DescribeSnapshots",
                "ec2:DeleteSnapshot",
                "ec2:CreateImage",
                "ec2:DescribeInstanceStatus",
                "ssm:GetAutomationExecution",
                "ssm:UpdateDocumentDefaultVersion",
                "ssm:CreateDocument",
                "ssm:StartAutomationExecution",
                "ssm:ListDocuments",
                "ssm:UpdateDocument",
                "ssm:DescribeDocument",
                "sns:ListTopics",
                "sns:GetTopicAttributes",
                "sns:CreateTopic",
                "iam:GetRole",
                "iam:GetInstanceProfile",
                "iam:ListRoles",
                "iam:ListInstanceProfiles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:ListBucket",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::*image-build*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish",
                "sns:setTopicAttributes"
            ],
            "Resource": "arn:aws:sns:*:*:*image-build*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

AWS 托管策略：**AWSMarketplaceGetEntitlements**

您可以将该`AWSMarketplaceGetEntitlements`策略附加到您的IAM身份。

此政策授予只读权限，允许软件即服务 (SaaS) 产品销售商检查客户是否已订阅其 AWS Marketplace SaaS 产品。

权限详细信息

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

AWS 托管策略：**AWSMarketplaceMeteringFullAccess**

您可以将该AWSMarketplaceMeteringFullAccess策略附加到您的IAM身份。

此政策向贡献者授予权限，允许他们报告与之对应的计量使用量AMI和开启灵活消费定价的容器产品。 AWS Marketplace

权限详细信息

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

AWS 托管策略：**AWSMarketplaceMeteringRegisterUsage**

您可以将该AWSMarketplaceMeteringRegisterUsage策略附加到您的IAM身份。

此政策向贡献者授予权限，允许他们报告与按小时定价的容器产品相对应的计量使用量。 AWS Marketplace

权限详细信息

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

AWS 托管策略：**AWSMarketplaceSellerFullAccess**

您可以将该AWSMarketplaceSellerFullAccess策略附加到您的IAM身份。

该政策授予管理权限，允许他们完全访问卖家对基于商品的所有操作 AWS Marketplace AWS Marketplace 管理门户，包括和管理EC2AMIAMI基于亚马逊的商品。

权限详细信息

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MarketplaceManagement",
            "Effect": "Allow",
            "Action": [
                "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:ListTasks",
                "aws-marketplace:DescribeTask",
                "aws-marketplace:UpdateTask",
                "aws-marketplace:CompleteTask",
```

```
                "aws-marketplace:GetSellerDashboard",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots",
                "ec2:ModifyImageAttribute",
                "ec2:ModifySnapshotAttribute"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AgreementAccess",
            "Action": [
                "aws-marketplace:SearchAgreements",
                "aws-marketplace:DescribeAgreement",
                "aws-marketplace:GetAgreementTerms"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws-marketplace:PartyType": "Proposer"
                },
                "ForAllValues:StringEquals": {
                    "aws-marketplace:AgreementType": [
                        "PurchaseAgreement"
                    ]
                }
            }
        },
        {
            "Sid": "IAMGetRole",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Sid": "AssetScanning",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
```

```
                "StringEquals": {
                    "iam:PassedToService": "assets.marketplace.amazonaws.com"
                }
            }
        },
        {
            "Sid": "VendorInsights",
            "Effect": "Allow",
            "Action": [
                "vendor-insights:GetDataSource",
                "vendor-insights:ListDataSources",
                "vendor-insights:ListSecurityProfiles",
                "vendor-insights:GetSecurityProfile",
                "vendor-insights:GetSecurityProfileSnapshot",
                "vendor-insights:ListSecurityProfileSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TagManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:TagResource",
                "aws-marketplace:UntagResource",
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Sid": "SellerSettings",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace-management:GetSellerVerificationDetails",
                "aws-marketplace-management:PutSellerVerificationDetails",
                "aws-marketplace-management:GetBankAccountVerificationDetails",
                "aws-marketplace-management:PutBankAccountVerificationDetails",
                "aws-marketplace-management:GetSecondaryUserVerificationDetails",
                "aws-marketplace-management:PutSecondaryUserVerificationDetails",
                "aws-marketplace-management:GetAdditionalSellerNotificationRecipients",
                "aws-marketplace-management:PutAdditionalSellerNotificationRecipients",
                "payments:GetPaymentInstrument",
                "payments:CreatePaymentInstrument",
                "tax:GetTaxInterview",
                "tax:PutTaxInterview",
```

```
                "tax:GetTaxInfoReportingDocument"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Support",
            "Effect": "Allow",
            "Action": [
                "support:CreateCase"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ResourcePolicyManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:GetResourcePolicy",
                "aws-marketplace:PutResourcePolicy",
                "aws-marketplace:DeleteResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "resale-
authorization.marketplace.amazonaws.com"
                }
            }
        }
    ]
}
```

AWS 托管策略：**AWSMarketplaceSellerProductsFullAccess**

您可以将该AWSMarketplaceSellerProductsFullAccess策略附加到您的IAM身份。

该政策向贡献者授予权限，允许他们完全访问管理商品 AWS Marketplace 管理门户、以及管理AMI基于商品的亚马逊EC2AMI产品。

权限详细信息

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:StartChangeSet",
                "aws-marketplace:CancelChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:ListTasks",
                "aws-marketplace:DescribeTask",
                "aws-marketplace:UpdateTask",
                "aws-marketplace:CompleteTask",
                "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/*"
        }
    ]
}
```

AWS 托管策略：**AWSMarketplaceSellerProductsReadOnly**

您可以将该AWSMarketplaceSellerProductsReadOnly策略附加到您的IAM身份。

此政策授予只读权限，允许用户查看基于商品的亚马逊产品 AWS Marketplace 管理门户，以及查看AMI基于亚马逊的产品中EC2AMI使用的商品。

权限详细信息

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:ListTasks",
```

```
                "aws-marketplace:DescribeTask",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        }
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        }
    ]
}
```

AWS 托管策略：**AWSVendorInsightsVendorFullAccess**

您可以将该AWSVendorInsightsVendorFullAccess策略附加到您的IAM身份。

此政策授予在 AWS Marketplace 供应商洞察上创建和管理所有资源的完全访问权限。 AWS Marketplace 在本指南中，Vendor Insights将评估人确定为买方和供应商等同于卖方。 AWS Marketplace 更新AWSVendorInsightsVendorFullAccess以添加协议搜索、更新配置文件快照、供应商标记，并允许对 AWS Artifact 第三方报告进行只读访问。

权限详细信息

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:DescribeEntity",
      "Resource": "arn:aws:aws-marketplace:*:*:*/SaaSProduct/*"
    },
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:ListEntities",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "vendor-insights:CreateDataSource",
        "vendor-insights:UpdateDataSource",
        "vendor-insights:DeleteDataSource",
        "vendor-insights:GetDataSource",
        "vendor-insights:ListDataSources",
        "vendor-insights:CreateSecurityProfile",
        "vendor-insights:ListSecurityProfiles",
        "vendor-insights:GetSecurityProfile",
        "vendor-insights:AssociateDataSource",
        "vendor-insights:DisassociateDataSource",
        "vendor-insights:UpdateSecurityProfile",
        "vendor-insights:ActivateSecurityProfile",
        "vendor-insights:DeactivateSecurityProfile",
        "vendor-insights:UpdateSecurityProfileSnapshotCreationConfiguration",
        "vendor-insights:UpdateSecurityProfileSnapshotReleaseConfiguration",
        "vendor-insights:GetSecurityProfileSnapshot",
        "vendor-insights:ListSecurityProfileSnapshots"
        "vendor-insights:TagResource",
        "vendor-insights:UntagResource",
        "vendor-insights:ListTagsForResource",
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:AcceptAgreementApprovalRequest",
        "aws-marketplace:RejectAgreementApprovalRequest",
        "aws-marketplace:GetAgreementApprovalRequest",
        "aws-marketplace:ListAgreementApprovalRequests"
        "aws-marketplace:CancelAgreement",
        "aws-marketplace:SearchAgreements"
      ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws-marketplace:AgreementType": "VendorInsightsAgreement"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
         "artifact:GetReport",
```

```
          "artifact:GetReportMetadata",
          "artifact:GetTermForReport",
          "artifact:ListReports",

      ],
      "Resource": "arn:aws:artifact:*::report/*"
    }
  ]
  }
```

AWS 托管策略：**AWSVendorInsightsVendorReadOnly**

您可以将该`AWSVendorInsightsVendorReadOnly`策略附加到您的IAM身份。

此政策授予查看 AWS Marketplace 供应商洞察资料和相关资源的只读访问权限。 AWS Marketplace 在本指南中，Vendor Insights将评估人确定为买方和供应商等同于卖方。 AWS Marketplace 更新`AWSVendorInsightsVendorReadOnly`为添加列出标签的权限并允许对 AWS Artifact 第三方报告进行只读访问。

权限详细信息

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:DescribeEntity",
      "Resource": "arn:aws:aws-marketplace:*:*:*/SaaSProduct/*"
    },
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:ListEntities",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "vendor-insights:GetDataSource",
        "vendor-insights:ListDataSources",
        "vendor-insights:ListSecurityProfiles",
        "vendor-insights:GetSecurityProfile",
        "vendor-insights:GetSecurityProfileSnapshot",
        "vendor-insights:ListSecurityProfileSnapshots"
```

```
        "vendor-insights:ListTagsForResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "artifact:GetReport",
         "artifact:GetReportMetadata",
         "artifact:GetTermForReport",
         "artifact:ListReports"
      ],
      "Resource": "arn:aws:artifact:*::report/*"
    }
  ]
}
```

AWS MarketplaceAWS 托管策略的更新

查看 AWS Marketplace 自该服务开始跟踪这些更改以来 AWS 托管策略更新的详细信息。要获得有关此页面变更的自动提醒，请订RSS阅该 AWS Marketplace 文档历史记录页面上的订阅源。

更改	描述	日期
AWSMarketplaceSellerFullAccess - 更新的策略	更新了`AWSMarketplaceSellerFullAccess` 文档，以反映已删除以下操作：`aws-marketplace-management:viewMarketing` `aws-marketplace-management:viewSettings` 、和`aws-marketplace-management:uploadFiles` 。此更新还包括删除 “使用细粒度权限” 部分。	2024 年 6 月 4 日
AWSMarketplaceGetEntitlements - 更新的策略	AWS Marketplace 已更新`AWSMarketplaceGetE`	2024 年 3 月 22 日

更改	描述	日期
	ntitlements sid为在政策声明中添加。	
AWSMarketplaceSellerFullAccess - 更新的策略	AWS Marketplace 更新AWSMarketplaceSellerFullAccess 为添加了创建服务相关角色的权限。	2024 年 3 月 15 日
AWSMarketplaceSellerFullAccess - 更新的策略	AWS Marketplace 更新AWSMarketplaceSellerFullAccess 为添加了访问税务信息的权限。	2024 年 2 月 8 日
AWSVendorInsightsVendorFullAccess-更新了政策	AWS Marketplace 已更新AWSVendorInsightsVendorFullAccess 以添加更新数据源的权限。	2023 年 10 月 18 日
AWSMarketplaceSellerFullAccess - 更新的策略	AWS Marketplace 更新AWSMarketplaceSellerFullAccess 为添加共享实体的权限。	2023 年 6 月 1 日
AWSMarketplaceSellerFullAccess - 更新的策略	AWS Marketplace 已更新AWSMarketplaceSellerFullAccess ，添加了与账户验证、银行账户验证、案例管理和卖家通知详情相关的权限。	2023 年 6 月 1 日
AWSMarketplaceSellerFullAccess - 更新的策略	AWS Marketplace 已更新AWSMarketplaceSellerFullAccess ，添加了访问卖家控制面板的权限。	2022 年 12 月 23 日

更改	描述	日期
AWSMarketplaceSellerFullAccess, AWSMarketplaceSellerProductsFullAccess, AWSMarketplaceSellerProductsReadOnly - 对现有策略的更新	AWS Marketplace 更新了新的基于标签的授权功能的策略。	2022 年 12 月 9 日
AWS Marketplace 已更新 AWSVendorInsightsVendorFullAccess	AWS Marketplace 更新`AWSMarketplaceSellerProductsFullAccess` 以添加协议搜索、更新配置文件快照、供应商标记，并允许对 AWS Artifact 第三方报告进行只读访问（预览）。	2022 年 11 月 30 日
AWS Marketplace 已更新 AWSVendorInsightsVendorReadOnly	AWS Marketplace 更新`AWSVendorInsightsVendorReadOnly` 为添加列出标签的权限并允许对 AWS Artifact 第三方报告进行只读访问（预览）。	2022 年 11 月 30 日
AWSVendorInsightsVendorFullAccess 和 AWSVendorInsightsVendorReadOnly— 添加了新政策	AWS Marketplace 为新功能 “ AWS Marketplace 供应商见解” 添加了政策：`AWSMarketplaceSellerProductsFullAccess` 和`AWSVendorInsightsVendorReadOnly` 。	2022 年 7 月 26 日

更改	描述	日期
AWSMarketplaceSellerProductsFullAccess和AWSMarketplaceSellerFullAccess— 更新了政策	AWS Marketplace 更新了新功能 “ AWS Marketplace 供应商见解” 的政策：`AWSMarketplaceSellerProductsFullAccess` 和`AWSMarketplaceSellerFullAccess` 。	2022 年 7 月 26 日
AWSMarketplaceSellerFullAccess 和 AWSMarketplaceSellerProductsFullAccess – 对现有策略的更新	AWS Marketplace 更新了政策，使`iam:PassedToService` 条件仅适用于`iam:PassRole` 。	2021 年 11 月 22 日
AWSMarketplaceFullAccess – 现有策略更新	AWS Marketplace 已从`AWSMarketplaceFullAccess` 策略中移除重复的`ec2:DescribeAccountAttributes` 权限。	2021 年 7 月 20 日
AWS Marketplace 已开始跟踪更改	AWS Marketplace 开始跟踪其 AWS 托管策略的更改。	2021 年 4 月 20 日

AWS Marketplace 商务分析服务账号权限

使用以下 IAM 权限策略注册 AWS Marketplace 商务分析服务。

有关如何注册的说明，请按照载入指南进行操作。

```
{
    "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 权限策略允许用户向 AWS Marketplace Commerce Analytics Service 提出请求。

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

有关此特征的更多信息，请参阅 AWS Marketplace 商务分析服务。

Amazon SQS 权限

作为 SaaS 产品发布流程的一部分， AWS Marketplace 为您提供一个 Amazon SNS 主题，当客户的订阅或授权状态发生变化时，您可以使用该主题接收通知。您可以为该主题配置一个或多个 Amazon SQS 队列，以便队列可以对通知执行操作。例如，如果客户为其对 SaaS 产品的订阅添加了更多存储空间，则 Amazon SNS 主题可以向 Amazon SQS 队列发送消息，该队列启动流程以自动增加该客户可用的存储容量。

当您为 Amazon Simple Queue Service (Amazon SQS) 队列订阅所提供的 Amazon SNS 主题时，系统将自动添加权限，以允许该主题向队列发布消息。但是，您仍然需要一个 IAM 策略来授予 AWS Marketplace 计量和授权服务 API 用户访问队列的权限。如果服务使用相同的凭证运行，则上述这种情况可应用于同一用户。使用以下内容创建策略，并将其附加到您的用户或角色。

```
{
    "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` 字段是您的 Amazon SQS 队列的 Amazon 资源名称 (ARN)。

有关您的 SaaS 产品的消息通知和排队的更多信息，请参阅the section called “为SQS队列订阅主题SNS”和the section called “访问 AWS Marketplace 计量和授权服务 APIs”。

AWS Marketplace 计量和授权 API 权限

软件即服务 (SaaS) 产品、亚马逊系统映像 (AMI) 产品和容器产品可以使用 AWS Marketplace Metering Service 和 AWS Marketplace Entitlement Service API。每种类型都需要不同的 AWS Identity and Access Management (IAM) 权限。对于您的一个或多个产品，您可以按所有使用量进行计费，并 AWS 根据您提供的计量记录向客户计费。要启用提供 AWS Marketplace 计量记录所需的集成，集成使用的服务账户需要受限的 IAM 策略才能启用访问权限。将为其发送计量信息的产品类型的策略附加到用于集成的用户或角色。

主题

- SaaS 产品的 IAM 策略
- AMI 产品的 IAM 策略
- 容器产品的 IAM 策略

SaaS 产品的 IAM 策略

在以下策略中，所有 SaaS 集成都需要第一个权限 `aws-marketplace:ResolveCustomer`。 AWS Marketplace Metering Service API 需要第二个权限 `aws-marketplace:BatchMeterUsage`。 AWS Marketplace Entitlement Service API 需要第三个权限 `aws-marketplace:GetEntitlements`。

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

有关 SaaS 产品的更多信息，请参阅基于 SaaS 的产品。

AMI 产品的 IAM 策略

请对 AMI 产品使用以下 IAM 策略。

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

有关 AMI 产品的更多信息，请参阅AMI基于产品。

容器产品的 IAM 策略

请对容器产品使用以下 IAM 策略。

```
{
```

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

有关容器产品的更多信息，请参阅基于容器的产品。

有关添加用户的更多信息，请参阅《IAM 用户指南》中的在您的 AWS 账户中创建用户。有关创建和分配策略的更多信息，请参阅更改 IAM 用户的权限。

此策略向策略所附加到的 IAM 角色或用户授予对 API 的访问权限。有关如何为这些 API 调用启用其他账户担任角色的更多信息，请参阅 AWS Partner Network (APN) 博客 AWS 账户上的 “如何跨多个账户最好地架构 AWS Marketplace SaaS 订阅”。

将服务相关角色用于 AWS Marketplace

AWS Marketplace 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种与之直接关联的 IAM 角色的独特类型。 AWS Marketplace服务相关角色由服务预定义 AWS Marketplace ，包括该服务代表您调用其他 AWS 服务所需的所有权限。

使用角色进行转售授权 AWS Marketplace

AWS Marketplace 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种与之直接关联的 IAM 角色的独特类型。 AWS Marketplace服务相关角色由服务预定义 AWS Marketplace ，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色使设置变得 AWS Marketplace 更加容易，因为您不必手动添加必要的权限。 AWS Marketplace 定义其服务相关角色的权限，除非另有定义，否则 AWS Marketplace 只能担任其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其他 IAM 实体的权限策略。

只有在首先删除相关资源后，您才能删除服务相关角色。这样可以保护您的 AWS Marketplace 资源，因为您不会无意中删除访问资源的权限。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 配合使用的AWS 服务，并查找服务相关角色列中显示为是的服务。选择是和链接，查看该服务的服务相关角色文档。

的服务相关角色权限 AWS Marketplace

AWS Marketplace 使用名为的服务相关角色 AWSServiceRoleForMarketplaceResaleAuthorization，该角色允许访问转售授权所使用或管理 AWS Marketplace 的 AWS 服务和资源。

AWSServiceRoleForMarketplaceResaleAuthorization 服务相关角色信任以下服务来代入该角色：

- `resale-authorization.marketplace.amazonaws.com`

名为的角色权限策略AWSMarketplaceResaleAuthorizationServiceRolePolicy AWS Marketplace 允许对指定资源完成以下操作。

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

您必须配置允许用户、组或角色创建、编辑或删除服务相关角色的权限。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为创建服务相关角色 AWS Marketplace

您无需手动创建服务相关角色。当您在中创建服务相关角色时 AWS Marketplace 管理门户， AWS Marketplace 会为您创建服务相关角色。

要创建服务相关角色

1. 在中 AWS Marketplace 管理门户，登录管理账户并选择设置。
2. 在 “设置” 部分中，选择 “服务相关角色” 选项卡。
3. 在服务相关角色页面上，为转售授权或转售授权集成选择服务相关角色，然后选择创建服务相关角色或配置集成。
4. 在 “转售授权的服务关联角色” 或 “创建转售授权集成” 页面上，查看信息，然后选择 “创建服务相关角色” 或 “创建集成” 进行确认。

服务关联角色页面上会显示一条消息，表示已成功创建转售授权服务相关角色。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您在中创建服务相关角色时 AWS Marketplace 管理门户， AWS Marketplace 会再次为您创建服务相关角色。

编辑的服务相关角色 AWS Marketplace

AWS Marketplace 不允许您编辑 AWSServiceRoleForMarketplaceResaleAuthorization 服务相关角色。创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除的服务相关角色 AWS Marketplace

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样就没有未被主动监控或维护的未使用实体。

> **Note**
>
> 如果独立软件供应商 (ISV) 没有该角色，则 AWS Resource Access Manager 不会自动与目标渠道合作伙伴共享新的转售授权。如果渠道合作伙伴没有该角色，则 AWS Resource Access Manager 不会自动接受针对他们的转售授权。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除 AWSServiceRoleForMarketplaceResaleAuthorization 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

AWS Marketplace 服务相关角色支持的区域

AWS Marketplace 支持在提供服务的所有区域中使用服务相关角色。有关更多信息，请参阅 AWS 区域和端点。

使用记录 AWS Marketplace 计量API呼叫 AWS CloudTrail

AWS Marketplace 与 AWS CloudTrail一项服务集成，该服务提供用户、角色或角色所执行操作 AWS 服务 的记录 AWS Marketplace。 CloudTrail 将API呼叫捕获 AWS Marketplace 为事件。捕获的调用包括来自 AWS Marketplace 控制台的调用和对 AWS Marketplace API操作的代码调用。

CloudTrail 在您创建账户 AWS 账户 时已在您的账户上启用。当支持的事件活动发生在中时 AWS Marketplace，该活动会与其他 CloudTrail 事件一起记录在 AWS 服务 事件历史记录中。您可以在 账户中查看、搜索和下载最新事件。

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根凭证还是 AWS Identity and Access Management 用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

AWS Marketplace 支持将`BatchMeterUsage`操作作为事件记录在 CloudTrail日志文件中。

AWS Marketplace 计量API日志文件条目示例

例如：`BatchMeterUsage`

以下示例显示了一个 CloudTrail 日志条目，该条目演示了中的`BatchMeterUsage`操作 AWS Marketplace Metering Service。当卖家发送计量记录以报告其客户对所列软件即服务 (SaaS) 产品的使用情况时 AWS Marketplace，此 CloudTrail 日志条目将记录在卖家的日志条目中 AWS 账户。

```
{
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2018-04-19T16:32:51Z",
            "eventSource": "metering-marketplace.amazonaws.com",
            "eventName": "BatchMeterUsage",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "192.0.0.2/24",
            "userAgent": "Coral/Netty14",
            "requestParameters": {
                "usageRecords": [
                    {
                        "dimension": "Dimension1",
                        "timestamp": "Apr 19, 2018 4:32:50 PM",
```

```
                        "customerIdentifier": "customer1",
                        "quantity": 1
                    }
                ],
                "productCode": "EXAMPLE_proCode"
            },
            "responseElements": {
                "results": [
                    {
                        "usageRecord": {
                            "dimension": "Dimension1",
                            "timestamp": "Apr 19, 2018 4:32:50 PM",
                            "customerIdentifier": "customer1",
                            "quantity": 1
                        },
                        "meteringRecordId": "bEXAMPLE-98f0-4e90-8bd2-bf0EXAMPLE1e",
                        "status": "Success"
                    }
                ],
                "unprocessedRecords": [ ]
            },
            "requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
            "eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
            "readOnly": false,
            "eventType": "AwsApiCall",
            "recipientAccountId": "123456789012"
        }
    ]
}
```

示例：容器的 RegisterUsage

以下示例显示了一个 CloudTrail 日志条目，该条目演示了中的RegisterUsage操作 AWS Marketplace Metering Service。在买家部署按小时定价的容器产品时，容器中的软件会在买家RegisterUsage内部调用 AWS 账户，为该亚马逊弹性容器服务 (亚马逊ECS) 任务或亚马逊弹性 Kubernetes Service (Amazon) Pod 启动每小时计量。 AWS Marketplace AWS 账户 EKS此 CloudTrail 日志条目记录在买家的日志中 AWS 账户。

```
{
    "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"
}
```

示例：**MeterUsage**适用于 Amazon 上的容器 EKS

以下示例显示了一个 CloudTrail 日志条目，该条目演示了在 Amazon 上 AWS Marketplace Metering Service 对容器执行的`MeterUsage`操作EKS。当在买方部署具有自定义计量的集装箱产品时 AWS 账户，集装箱中的软件会在买方`MeterUsage`内部调用每小时 AWS 账户 进行一次报告。 AWS Marketplace 此 CloudTrail 日志条目记录在买家的日志中 AWS 账户。

```
{
    "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"
}
```

示例：`MeterUsage`开启 AMIs

以下示例显示了一个 CloudTrail 日志条目，该条目演示了 Amazon 系统映像 (AMIs) 中的`MeterUsage`操作。 AWS Marketplace Metering Service 当在买方部署 AWS Marketplace 具有自定义计量的AMI产品时 AWS 账户，买方`MeterUsage`内部AMI的 AWS 账户 软件会每小时报告使用情况。此 CloudTrail 日志条目记录在买家的日志中 AWS 账户。

```
{
    "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"
}
```

文档历史记录

下表介绍此版本的《AWS Marketplace 卖家指南》的文档。

要获得有关本文档更新的通知，您可以订阅 Feed。RSS

变更	说明	日期
已移除已弃用的权限	更新了`AWSMarketplaceSellerFullAccess` 文档，删除了三个操作。	2024 年 6 月 4 日
更新了 Amazon Machine Image (AMI) 年度协议修订选项	卖家现在可以为现有订阅添加或切换实例类型。	2024 年 5 月 30 日
更新了 “了解你的客户” (KYC) 流程	为卖家的KYC流程添加了额外的步骤。	2024 年 5 月 21 日
为 AWS Marketplace 卖家更新了私人报价体验	添加了内容，以增强创建和管理私人优惠的体验。	2024 年 5 月 20 日
更新了 Amazon EKS 附加商品的要求	更新了 “准备您的容器产品作为 AWS Marketplace 附加组件” 部分，并添加了 “附加组件提供商的插件配置要求和最佳实践”。	2024 年 5 月 8 日
更新了 AWS Marketplace 卖家权限	更新了权限示例以修复语法错误。	2024 年 4 月 2 日
更新了 SaaS 合同定价	更新了与使用 pay-as-go定价模型报告SaaS合同的超额费用相关的内容。	2024 年 4 月 2 日
开启了新的演示和私募优惠选项 AWS Marketplace	AWS Marketplace 现在支持在商品详情页面上为精选卖家提供演示和私密报价请求选项。	2024 年 4 月 1 日

新的税收仪表板	AWS Marketplace 现在支持税务仪表板，该仪表板可为交易提供有关美国和国际税收的可视化和细粒度数据。 AWS Marketplace	2024 年 3 月 29 日
更新了 `AWSMarketplaceGetEntitlements`	已`sid`添加到`AWSMarketplaceGetEntitlements` 托管策略中。	2024 年 3 月 22 日
更新了专业服务程序	更新了编辑产品可见性并删除了专业服务产品程序。	2024 年 3 月 19 日
更新了AMI访问策略	更新了本节以阐明特定于 Linux 和类似 Un AMI ix 的访问策略。	2024 年 3 月 19 日
添加了以下内容的搜索引擎优化信息 AWS Marketplace	添加了与搜索引擎优化相关的内容 AWS Marketplace。	2024 年 3 月 19 日
更新了针对以下地区的卖家的托管政策 AWS Marketplace	已更新`AWSMarketplaceSellerFullAccess` ，添加了与创建服务相关角色相关的权限。	2024 年 3 月 15 日
产品的新服务相关角色 AWS Marketplace	AWS Marketplace 现在提供了一个与服务相关的角色，允许访问转售授权所使用或管理 AWS Marketplace 的AWS服务和资源。	2024 年 3 月 15 日
更新了针对以下地区的卖家的托管政策 AWS Marketplace	已更新`AWSMarketplaceSellerFullAccess` ，添加了与访问税务信息相关的权限。	2024 年 2 月 8 日

用于安全报告的亚马逊 EventBridge 事件	AWS Marketplace 现在支持亚马逊 EventBridge 事件（以前称为 Amazon CloudWatch Events），即卖家商品有安全漏洞报告可用。	2024 年 1 月 31 日
支持 Amazon EKS 插件	添加了与将基于 AWS Marketplace 容器的产品发布到 Amazon 的EKS插件相关的内容和程序。	2024 年 1 月 29 日
增加了对专业服务卖家转售产品的支持	卖家现在可以作为独立软件供应商为渠道合作伙伴创造转售机会（ISVs）。	2024 年 1 月 18 日
Future 过期协议的正式发布日期为 AWS Marketplace	All ISVs AWS Marketplace and Channel Partners 可以在发布预付定价的私人报价时指定未来的开始日期。	2024 年 1 月 16 日
全新的使用情况控制面板	AWS Marketplace 现在支持使用情况控制面板，该仪表板为使用 SaaS 和基于服务器使用情况的产品的客户提供可视化和细粒度数据。	2024 年 1 月 10 日
面向卖家的全新“快速启动”部署选项	添加了与新的“快速启动”部署选项相关的内容和程序，该选项适用于软件即服务（SaaS）产品。	2023 年 11 月 29 日
面向专属优惠的灵活付款计划现已可用	现在，所有客户都可以使用私人优惠的灵活付款时间表 (FPS) AWS Marketplace。	2023 年 11 月 17 日
添加了容器产品的自助服务内容	添加了与容器产品自助服务操作相关的内容和过程。	2023 年 11 月 3 日

用于协议和续订的新控制面板	AWS Marketplace 现在为卖家提供了协议和续订控制面板。	2023 年 10 月 31 日
针对变更集的 Amazon EventBridge 活动	AWS Marketplace 现在支持 Amazon EventBridge 事件（以前称为 Amazon CloudWatch Events），即更改集完成时状态为成功、失败或已取消。	2023 年 10 月 31 日
在《AWS Marketplace 供应商见解》中更新了针对卖家的托管政策	AWS Marketplace 已更新`AWSVendorInsightsVendorFullAccess` 以添加更新数据源的权限。	2023 年 10 月 18 日
终止 AWS Marketplace 对桌面应用程序的支持 (AMDA)	AWS Marketplace 已于 2023 年 10 月 2 AMDA 日终止对的支持。与之相关的所有内容和程序AMDA均已从指南中删除。	2023 年 10 月 2 日
添加了 SaaS 产品的自助服务内容	添加了与软件即服务 (SaaS) 产品自助服务操作相关的内容和过程。	2023 年 9 月 12 日
卖家现在可以享受每日付款 AWS Marketplace	AWS Marketplace 管理门户 现在，卖家可以选择每天或每月接收付款。	2023 年 9 月 7 日
对 Amazon 的支持 EventBridge	AWS Marketplace 现在支持卖家收到新报价时 EventBridge 的事件。	2023 年 9 月 6 日
AWS Marketplace 供应商洞察的全新自助服务入门体验	AWS Marketplace 供应商见解现在支持自助入职体验。	2023 年 8 月 17 日

更新了针对以下地区的卖家的托管政策 AWS Marketplace	更新了 `AWSMarketplaceSellerFullAccess` 以添加共享实体相关权限。	2023 年 6 月 1 日
更新了针对以下地区的卖家的托管政策 AWS Marketplace	更新了 `AWSMarketplaceSellerFullAccess` 以添加与账户验证、银行账户验证、案例管理和卖家通知详情相关的权限。	2023 年 6 月 1 日
添加了有关细粒度访问权限的内容 AWS Marketplace	在中添加了用于细粒度访问的概述信息、权限和程序。 AWS Marketplace 管理门户	2023 年 6 月 1 日
增加了AMI产品程序	增加了AMI产品的自助服务程序。	2023 年 5 月 12 日
终止对 Amazon 免税计划和美国税收计算服务的支持 AWS Marketplace	亚马逊免税计划和美国税务计算服务已从指南中删除，因为对该服务的支持已于 2023 年 3 月 6 日终止。	2023 年 3 月 6 日
增加了针对容器产品的过程	添加了更改容器产品设置的新过程。	2023 年 2 月 13 日
终止对 Produc AWS Marketplace t Support 连接	AWS Marketplace 自 2022 年 11 月 30 日起，不再支持产品支持连接和使用商务分析服务共享客户联系方式。 AWS Marketplace 已从指南中移除 P AWS Marketplace roduct Support Connec	2023 年 1 月 27 日
咨询合作伙伴名称变更	AWS Marketplace 现在将咨询合作伙伴称为渠道合作伙伴。更新了指南，仅反映了名称的更改。	2023 年 1 月 26 日

客户服务团队名称变更	托管目录操作 (MCO) 团队更名为 AWS Marketplace 卖家运营团队。更新了指南，仅反映了名称的更改。	2023 年 1 月 24 日
专属优惠页面	经过身份验证的买家现在可以在 AWS Marketplace 私人报价页面 AWS 账户 上查看延伸到他们的私密报价。	2023 年 1 月 19 日
添加了有关 SaaS 产品设置的主题	添加了一个新主题，描述了更改 SaaS 产品设置的过程。	2023 年 1 月 6 日
更新了针对以下地区的卖家的托管政策 AWS Marketplace	AWS Marketplace 已更新`AWSMarketplaceSellerFullAccess`，添加了访问卖家控制面板的权限。	2022 年 12 月 23 日
更新了卖家电子邮件通知	现在，当专属优惠发布时，卖家会收到通知。	2022 年 12 月 22 日
AWS Marketplace支持卖家创建采用订阅模式的 SaaS 免费试用	卖家现在可以为订阅产品创建免费试用。	2022 年 12 月 16 日
更新AMI自助上架商品（版本 2）的卖家体验 AWS Marketplace	开店的卖家现在 AWS Marketplace 可以创建单一AMI商品自助服务清单。卖家可以自己进行更新，无需等待额外的处理时间。	2022 年 12 月 14 日

更新了三项基于标签的授权策略	更新了 AWS Marketplace 基于标签的授权功能的三个策略（`AWSMarketplaceSellerFullAccess` `AWSMarketplaceSellerProductsFullAccess` 、和`AWSMarketplaceSellerProductsReadOnly` ）。	2022 年 12 月 9 日
在《 AWS Marketplace 供应商见解》中更新了针对卖家的政策	更新了`AWSVendorInsightsVendorReadOnly` 针对 AWS Marketplace 供应商洞察卖家的托管政策`AWSVendorInsightsAssessorFullAccess` 。	2022 年 11 月 30 日
在 “ AWS Marketplace 供应商见解” 中控制卖家的访问权限	为 Vendo AWS Marketplace r Insights 添加了一个新主题，用于描述卖家可用的操作和权限。	2022 年 11 月 30 日
更新了《 AWS Marketplace 供应商见解》的四项托管政策	更新`AWSVendorInsightsVendorFullAccess` 并`AWSVendorInsightsVendorReadOnly` 管理了《 AWS Marketplace 供应商见解》的政策。	2022 年 11 月 28 日
卖家可以在 Amazon 上发布附加商品 EKS	与 Amazon AWS Marketplace 的集成EKS使卖家能够在亚马逊EKS控制台中向买家展示其商品。	2022 年 11 月 28 日

更新了“ AWS Marketplace 供应商见解”设置	更新了“ AWS Marketplace 供应商见解”的设置程序。	2022 年 11 月 18 日
更新了《 AWS Marketplace 供应商洞察》的两项政策	更新了两项政策`AWSMarketplaceSellerProductsFullAccess` 以及`AWSMarketplaceSellerFullAccess` AWS Marketplace 供应商洞察政策。	2022 年 7 月 26 日
为 Vendo AWS Marketplace r Insights 添加了两项政策，一项提供软件风险评估的功能。	添加了两项政策`AWSVendorInsightsVendorFullAccess` ，`AWSVendorInsightsVendorReadOnly` 并为 AWS Marketplace 供应商见解添加了一项提供软件风险评估的功能。	2022 年 7 月 26 日
AWS Marketplace 供应商见解是一项新增功能 AWS Marketplace	AWS Marketplace 供应商洞察是一项提供软件风险评估的功能。 AWS Marketplace 供应商见解供应商洞察是一项提供软件风险评估的功能。	2022 年 7 月 26 日
AWS Marketplace 商务分析服务权限更新	AWS Marketplace 商务分析服务具有其他IAM权限。	2022 年 7 月 21 日
添加了卖家传输数据源服务部分	仅限文档更新，添加了卖家传输数据源服务部分，并重新整理了数据源相关部分。	2022 年 6 月 15 日
添加了补充报告部分	为补充报告添加了一个新部分，其中 AWS Marketplace 提供了最近发布的功能。	2022 年 6 月 14 日

卖家现在可以通过 SaaS 免费试用合同 AWS Marketplace	现在，卖家可以通过定义免费试用期限、试用期内的可用维度以及客户获得的免费使用容量量来创建免费试用，而无需进行额外的开发工作。	2022 年 5 月 31 日
添加了买家和卖家交易电子邮件通知	新功能允许向买家和卖家发送电子邮件通知，以验证报价和协议 AWS Marketplace。	2022 年 5 月 23 日
添加了创建机器学习产品列表的示例	机器学习部分的仅文档更新，增加了一些示例，以比较卖家和买家在创建机器学习产品时的视图。	2022 年 4 月 22 日
更新了机器学习部分	对机器学习部分进行了仅文档更新，以明确过程。	2022 年 4 月 15 日
以色列现在是符合条件的司法管辖区	以色列居民现在有资格成为 AWS Marketplace上的卖家。	2022 年 4 月 13 日
更新了优惠数据源中的新项目	制造商现在可以收到其渠道合作伙伴创建的优惠信息。这包括其账户是记录卖家和优惠制造商的优惠。	2022 年 3 月 29 日
经销商机会通知	卖家现在可以接收有关经销商机会的通知。	2022 年 3 月 28 日
为专业服务产品添加了视频	使用详细介绍如何管理服务产品的视频更新了专业服务产品页面。	2022 年 2 月 24 日
关于部署无服务器 SaaS 集成解决方案的新主题	添加了有关集成无服务器 SaaS 部署的新信息，包括用于参考部署步骤的 AWS 快速入门链接。	2022 年 2 月 15 日

对基于容器的要求和章节进行最低限度的更新 AMI	对基于容器的要求中的政策进行了最少的更新，并删除了定AMI价合同的不正确信息	2022 年 2 月 14 日
容器版本控制更新	仅文档更新，以阐明如何将容器映像和其他构件推送到存储库。	2022 年 2 月 10 日
更新 SaaS 产品的 `ResolveCustomer` 代码示例	SaaS 产品的 `ResolveCustomer` 代码示例已更新为包括 `CustomerAWSAccountID` 。	2022 年 2 月 3 日
为随处容器产品添加 AWS Marketplace 了 AWS License Manager 与集成的文档	仅限文档的更新，通过与 License Manager 集成， AWS Marketplace 为你的 Containers Anywhere 产品添加合同定价的详细指南。	2022 年 2 月 1 日
SaaS SNS 通知更新	仅文档更新，以澄清 SaaS 通知消息。	2022 年 1 月 25 日
卖家能够通过亚马逊 Web Serv EMEA ices 与当地买家进行交易 EMEA SARL	符合条件的 AWS Marketplace 卖家现在可以通过 Amazon Web Services EMEA SARL 与居住在欧洲、中东和非洲 (EMEA) 的国家和地区的买家进行交易。 AWS 账户	2022 年 1 月 7 日

添加了有关使用 Helm 图表交付方法为基于容器的产品创建交付选项的文档	卖家现在可以使用 Helm 图表交付方式提供交付选项。买家可以在启动环境中安装卖家提供的 Helm 图表，从而使用这些交付选项来启动基于容器的应用程序。在提供 Helm chart 交付方式时，卖家可以 QuickLaunch 为买家启用。QuickLaunch是买家可以 AWS CloudFormation 用来快速创建新的 Amazon EKS 集群并在其上启动基于容器的应用程序的功能。	2021 年 11 月 29 日
对现有策略的更新	AWS Marketplace 卖家安全政策已更新。	2021 年 11 月 22 日
基于容器的产品的AMI合同定价	独立软件供应商 (ISVs) 现在可以发布AMI基于新产品或基于容器的产品，并为买家提供前期合同定价。	2021 年 11 月 17 日
供应商计量标记	供应商计量标相关仅文档更新，包括代码示例。	2021 年 11 月 11 日
针对AMI或容器商品的 Amazon 简单通知服务	当客户通过 Amazon Simple Notification 服务订阅或取消订阅AMI或取消订阅容器产品时，独立软件供应商 (ISVs) 可以收到通知。	2021 年 11 月 10 日
新的卖家权限	AWS Marketplace 添加了访问中 “优惠和合作伙伴” 选项卡的新权限 AWS Marketplace 管理门户。	2021 年 11 月 9 日

为机器学习产品部署为异步推理配置的端点	对于期望比最大有效载荷推断更高的机器学习软件，或者要求处理时间超过每次调用的最大处理时间，买家可以部署为 Amazon SageMaker 异步推理配置的终端节点。	2021 年 11 月 8 日
退款政策和批准	仅文档更新，以阐明退款政策，并将所有退款信息移至《AWS Marketplace 卖家指南》中的一个中心位置。	2021 年 8 月 20 日
选择或上传咨询合作伙伴EULA的私人报价	现在，在为咨询合作伙伴创造转售机会时，独立软件供应商可以选择或上传最终用户许可协议 (EULA)。	2021 年 8 月 17 日
SaaS 合同产品的自定义产品维度	在为咨询合作伙伴创造转售机会时，独立软件供应商 (ISVs) 现在可以自定义 SaaS 合同产品维度。	2021 年 8 月 17 日
AWS Marketplace 现场演示计划	仅限文档的更新，以阐明 AWS Marketplace 现场演示计划对 AWS Data Exchange 数据集产品的要求。	2021 年 8 月 3 日
SaaS 产品准则更新	更新了 SaaS 产品的产品准则。	2021 年 7 月 29 日
基于容器的产品要求更新	更新了基于容器的产品要求。	2021 年 7 月 29 日
AMI安全策略更新	AMI产品的安全政策已更新。	2021 年 7 月 29 日
更多符合条件的司法管辖区	以下商品现在有资格成为以下商品的卖家 AWS Marketplace：香港SAR和卡塔尔。	2021 年 6 月 23 日

数据源概述	仅文档更新，概述了为卖家提供的数据源的结构。	2021 年 6 月 23 日
更新了机器学习章节	有关创建和维护机器学习产品的信息的仅文档更新。	2021 年 5 月 27 日
容器产品自助服务更新	现在，卖家可以更简单快速地通过 AWS Marketplace 管理门户更新其基于容器的产品。	2020 年 12 月 17 日
专业服务	卖家现在可以为 AWS Marketplace 买家提供专业服务。向文档中添加了专业服务部分。	2020 年 12 月 3 日
AMI产品自助更新	现在，卖家可以通过更简单、更快捷的方式更新基于亚马逊机器图片 (AMI) 的商品 AWS Marketplace 管理门户。	2020 年 11 月 23 日
更多符合条件的司法管辖区	以下商品现在有资格成为以下国家的卖家 AWS Marketplace：巴林、挪威、瑞士和阿拉伯联合酋长国 (UAE)。	2020 年 6 月 17 日
您可以为已接受的专属优惠提供升级和续订	对于 SaaS 合同和带消费产品的 SaaS 合同，您可以使用专属优惠为以前接受的专属优惠提供升级和续订。	2020 年 5 月 28 日
数据源中提供了更多信息	报告中的更多信息细分为较小的数据源，以简化数据的查找和分析。	2020 年 5 月 21 日
标准化许可条款现已提供	您可以提供标准化的许可条款来代替自定义条款EULAs，以简化签约流程。	2020 年 4 月 28 日

澳大利亚和新西兰是符合条件的司法管辖区	以下人员现在有资格成为卖家 AWS Marketplace：(i) 澳大利亚 (AU) 新西兰 (NZ) 的永久居民和公民或 (ii) 在其中一个地区组建或注册的商业实体。	2020 年 4 月 2 日
容器产品现在支持自定义计量和定价增强功能	如果您想定义自己的定价单位，并将使用量计量给我们计费，请与 AWS Marketplace 计量服务的`meterUsage` 操作集成。	2019 年 11 月 14 日
AWS Marketplace 通过 Data Exchange 支持AWS数据产品	您现在可以在 AWS Marketplace中提供数据产品。	2019 年 11 月 13 日
介绍 AWS Marketplace 目录API服务	AWS Marketplace 目录API服务为获得批准的卖家提供了一个以编程方式管理其商品的API界面。	2019 年 11 月 12 日
AWS Marketplace 支持按小时付费的容器	AWS Marketplace 现在支持在亚马逊 Elastic Kubernetes Service（亚马逊）上运行的按小时付费容器。EKS	2019 年 9 月 25 日
更新了AMI产品功能	现在，您可以使用同时部署 AMIs和 Lambda 函数。 AWS CloudFormation	2019 年 9 月 11 日
添加了安全性部分	将安全相关内容整合到了新的安全部分下。	2019 年 5 月 7 日
更新的AMI安全政策	更新了AMI产品的安全策略	2019 年 4 月 11 日
在机器学习产品部分中增加了版本控制信息	添加了介绍机器学习产品的产品版本功能的内容。	2019 年 3 月 21 日
添加了机器学习产品部分	添加了发布机器学习产品的内容	2018 年 11 月 28 日

添加了基于容器的产品部分	添加了发布基于容器的产品的内容。	2018 年 11 月 27 日
更新了提交卖家帮助请求的链接	将电子邮件地址更改为 Web 表单地址。	2018 年 10 月 22 日
添加了具有消费内容的 SaaS 合同	新编 SaaS 内容，添加了支持发布具有消费特征的 SaaS 合同。	2018 年 10 月 18 日
添加了面向专属优惠的灵活付款计划内容	添加了支持针对专属优惠内容发布灵活付款计划程序。	2018 年 10 月 15 日
更新了IAM权限内容	添加了支持新增只AMMP读访问IAM权限的内容。	2018 年 10 月 9 日
添加了咨询合作伙伴专属优惠内容	添加了支持发布咨询合作伙伴专属优惠特征的内容。	2018 年 10 月 9 日
添加了有关私有映像生成的内容	添加了支持发布 Private Image Build AMIs 功能的内容。	2018 年 8 月 13 日
添加了面向卖家的搜索引擎优化指南。	添加了面向希望优化其产品搜索的卖家的指南。	2018 年 7 月 3 日
更新了 Mark AWS etplace 徽标链接	更新了指向新的 AWS Marketplace 徽标的链接。	2018 年 6 月 12 日
添加了卖家指南	将所有PDF卖家指南转换为在线内容。	2018 年 5 月 9 日

AWS 词汇表

有关最新 AWS 术语，请参阅《AWS 词汇表 参考资料》中的AWS 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。