판매자 설명서

AWS Marketplace

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

AWS Marketplace: 판매자 설명서

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

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS Marketplace란 무엇인가요? ... 1
판매자 AWS Marketplace 로 사용 ... 1
에 대한 계약 구조 AWS Marketplace ... 3
의 제품 요금 AWS Marketplace ... 4
시작 ... 6
무료 소프트웨어 제품을 게시하기 위한 판매자 요구 사항 ... 6
유료 제품에 대한 추가 판매자 요구 사항 ... 7
유료 상품에 대한 적격 관할 구역 ... 8
AWS Marketplace Management Portal ... 13
AWS Marketplace 판매자로 등록 ... 14
1단계: 공개 프로필 만들기 ... 15
2단계: 세금 정보 제공 ... 16
3단계: 은행 계좌 정보 제공 ... 17
4단계: 지급 기본 설정 지정 ... 19
5단계: 고객 파악(KYC) 프로세스 완료 ... 19
6단계: 은행 계좌 확인 프로세스 완료 ... 23
이미 AWS Marketplace 판매자이신가요? ... 24
지급 기본 설정 ... 24
지급 기본 설정 지정 ... 25
AWS Marketplace 구독 요금 청구 ... 25
판매자가 지급을 받는 방법 ... 26
사용 가능한 통화 ... 26
등록 수수료 ... 27
의 표준 제안 목록에 대한 요금 이해 AWS Marketplace ... 27
의 리전 제안 목록에 대한 요금 이해 AWS Marketplace ... 28
Amazon Payments Europe 불만 ... 28
불만 사항 제출 ... 29
Amazon Payments Europe 불만 사항 해결 기간 ... 29
불만 사항 에스컬레이션 ... 29
AWS Marketplace 판매자를 위한 추가 도구 ... 30
AWS Marketplace 상거래 분석 서비스 ... 31
AWS Marketplace 필드 데모 프로그램 ... 49
AWS 파트너 어시스턴트 ... 50
제품 준비 ... 55

제품 제공 55
에서 실행되는 제품의 지정 AWS 59
제품 요금 60
요금 모델 60
요금 모델 변경 65
요금 변경 65
비공개 제안 65
에서 제품 환불 66
리전 및 국가 70
AWS 리전 70
국가 71
번역 및 언어 71
시작 72
번역 옵트아웃 73
표준 계약 73
에 대한 표준 계약 AWS Marketplace 74
에 대한 리셀러 계약 AWS Marketplace 75
범주 및 메타데이터 76
제품 이름 지정 및 설명 77
범주 및 키워드 선택 78
AMI 및 컨테이너 제품 사용 지침 79
요구 사항 80
릴리스 정보 작성 80
사용 지침 작성 80
업그레이드 지침 작성 81
CloudFormation 전송 지침 작성 81
검색 엔진 최적화 82
검색 엔진 최적화 82
AWS Marketplace 검색 83
홍보 미디어 85
홍보 미디어 모범 사례 86
프로모션 미디어 추가 및 관리 팁 87
비공개 제안 준비 89
비공개 제안 작동 방식 90
비공개 제안 고려 사항 90
구매자를 위한 비공개 제안 경험 91

비공개 제안 보고 ... 92
지원되는 제품 유형 ... 92
AMI 제품에 대한 비공개 제안 ... 93
컨테이너 제품에 대한 비공개 제안 ... 93
전문 서비스 제품에 대한 비공개 제안 ... 94
SaaS 제품에 대한 비공개 제안 ... 94
ML 제품에 대한 비공개 제안 ... 95
비공개 제안 만들기 및 관리 ... 95
새 비공개 제안 시작 ... 96
제안 상태 이해 ... 97
비공개 제안 초안 작성 및 게시 ... 97
비공개 제안 및 데모 요청 버튼 추가 ... 99
구매자에게 비공개 제안 보내기 ... 102
비공개 제안 복제 ... 102
제안 세부 정보 다운로드 ... 103
비공개 제안 진행 상황 저장 ... 103
비공개 제안 만료 업데이트 ... 103
비공개 제안 취소 ... 104
채널 파트너 비공개 제안 ... 104
추가 정보 ... 106
ISV로서 판매 권한 만들기 ... 106
할부 플랜 만들기 ... 111
비공개 제안에 대한 할부 플랜 만들기 ... 111
할부 플랜 보고 ... 112
미래 날짜의 계약 ... 112
미래 날짜의 계약에 대한 고려 사항 ... 113
미래 날짜의 계약서 작성 ... 114
미래 날짜의 계약과 함께 할부 플랜 사용 ... 114
미래 날짜의 계약에 대한 알림 받기 ... 114
채널 파트너 비공개 제안에 대한 재판매와 함께 미래 날짜의 계약 사용 ... 115
비공개 제안 FAQ ... 116
비공개 제안 성공 팀(POST)이란 무엇이며 판매자는 어떻게 연락할 수 있습니까? ... 116
판매자는 제안 탭에 액세스할 때 오류를 어떻게 해결할 수 있습니까? ... 116
할부 플랜 또는 유연한 결제 일정이란 무엇입니까? ... 117
비공개 제안에 액세스할 때 404 오류를 받는 구매자를 어떻게 도울 수 있습니까? ... 118

비공개 제안을 수락하려고 할 때 고객에게 "이미 활성 계약이 있습니다"라는 오류가 표시되는 이유는 무엇입니까? ... 120
판매자 또는 구매자가 비공개 제안을 취소할 수 있나요? ... 122
환불 또는 계약 취소를 요청하려면 어떻게 해야 하나요? ... 122
구매자는 언제 인보이스를 발행하나요? ... 123
제안이 수락되면 판매자는 어떤 조치를 취해야 합니까? ... 123
는 판매자와 파트너에게 어떻게 AWS 지불하나요? ... 123
는 세금을 어떻게 AWS 평가하나요? ... 123
리소스 및 지원 ... 124
계약 사용 ... 125
계약 유형 ... 125
계약 작업 ... 125
계약 찾기 ... 126
계약 정렬 ... 126
계약 세부 정보 보기 ... 126
계약 세부 정보 다운로드 ... 126
계약 수정 ... 127
공개 및 비공개 제안 수정에 지원되는 제품 유형 ... 127
공개 및 비공개 제안 업그레이드, 갱신 및 수정 생성 ... 128
업그레이드, 갱신 및 수정에 대한 보고 ... 129
AMI 기반 제품 ... 131
AMI 기반 제품 제공 방법 ... 131
추가 리소스 ... 131
AMI 기반 제품의 이해 ... 132
제품 수명 주기 ... 132
AMI 제품 코드 ... 134
변경 요청 ... 135
제품 로드 양식 ... 136
연간 계약 수정 ... 137
AMI 기반 제품 생성 ... 137
사전 조건 ... 138
셀프 서비스 경험 이해 ... 138
리스팅 만들기 ... 139
추가 리소스 ... 141
제품에 CloudFormation 템플릿 추가 ... 142
CloudFormation 템플릿 준비 ... 142

아키텍처 다이어그램 ... 150
기존 제품의 CloudFormation 템플릿 변환 ... 150
서버리스 애플리케이션 구성 요소 추가 ... 155
AMI 기반 제품 관리 ... 164
변경 요청 만들기 ... 165
제품 표시 여부 업데이트 ... 168
인스턴스 추가 및 제한 ... 170
버전 관리 ... 172
제품 정보 업데이트 ... 178
가용성 관리 ... 179
EULA 업데이트 ... 183
환불 정책 업데이트 ... 184
AMI에 대한 AWS Marketplace 액세스 권한 부여 ... 184
제품 제거 ... 185
문제 해결 ... 186
AMI 빌드 모범 사례 ... 188
재판매 권한 확보 ... 188
AMI 작성 ... 188
용 AMI 준비 및 보안 AWS Marketplace ... 189
AMI를 스캔하여 게시 요구 사항 확인 ... 190
AWS Marketplace AMI에서 소프트웨어가 실행되고 있는지 확인 ... 191
AMI 제품 요금 ... 193
AMI 요금 모델 ... 194
AWS 요금 및 소프트웨어 요금 ... 198
AWS Marketplace Metering Service를 통한 사용자 지정 측정 ... 199
AMI 제품의 계약 요금 ... 214
사용 AWS License Manager ... 217
Amazon SNS 알림 수신 ... 230
Amazon SNS 주제: aws-mp-subscription-notification ... 231
Amazon SQS 대기열을 Amazon SNS 주제에 구독 ... 232
AMI 제품 체크리스트 ... 233
제품 사용 ... 233
AMI 준비 ... 233
Windows AMI ... 233
Linux AMI ... 234
제품 로드 양식 또는 제품 탭 ... 234

AMI 기반 제품 요구 사항 234
AMI 제품 판매자 정책 235
보안 정책 235
아키텍처 정책 237
AMI 제품 사용 지침 238
AMI 제품 버전 정책 238
고객 정보 정책 238
제품 사용 정책 239
Image Builder 구성 요소 기반 제품 241
변경 요청 만들기 241
셀프 서비스를 사용하여 변경 요청 생성 242
변경 요청 생성 243
변경 요청의 상태 가져오기 243
추가 리소스 244
제품 표시 여부 업데이트 244
제품 표시 여부 업데이트 244
허용 목록 업데이트(계정 미리 보기) 245
인스턴스 추가 및 제한 245
인스턴스 추가 246
인스턴스 제한 247
버전 관리 247
제품 정보 업데이트 248
가용성 관리 249
추가 AWS 리전 249
제한 AWS 리전 250
향후 AWS 리전지원 업데이트 251
국가별 가용성 업데이트 251
EULA 업데이트 252
환불 정책 업데이트 253
제품 제거 253
컨테이너 기반 제품 256
도움말 가져오기 257
컨테이너 제품 시작하기 257
제품 수명 주기 257
사전 조건 258
개요: 컨테이너 제품 만들기 259

1단계: 컨테이너 제품의 제품 ID 및 제품 코드 만들기 ... 259
2단계: 초기 리스팅 만들기 ... 260
3단계: 제품의 초기 버전 추가 ... 260
4단계: (유료 제품만 해당) 측정 또는 계약 요금 통합 ... 260
5단계: 제품 가시성 업데이트 ... 261
다음 단계 ... 262
컨테이너 제품을 스캔하여 보안 문제 확인 ... 263
제품 정보 업데이트 ... 263
제품 버전 추가 ... 264
제품 요금 관리 ... 273
국가별 가용성 업데이트 ... 275
EULA 업데이트 ... 276
제품 테스트 및 릴리스 ... 277
컨테이너 기반 제품 요구 사항 ... 279
보안 정책 ... 279
고객 정보 요구 사항 ... 280
제품 사용 요구 사항 ... 280
아키텍처 요구 사항 ... 281
컨테이너 제품 사용 지침 ... 282
Amazon EKS 추가 기능 제품에 대한 요구 사항 ... 282
컨테이너 제품 요금 ... 299
컨테이너 요금 모델 ... 300
컨테이너 제품의 계약 요금 ... 303
컨테이너 제품 결제, 측정 및 라이선스 통합 ... 306
AWS Marketplace Metering Service로 시간당 및 사용자 지정 측정 ... 307
를 사용한 계약 요금 AWS License Manager ... 309
AWS Marketplace Metering Service를 사용한 측정 구성 ... 310
AWS Marketplace Metering Service를 사용하여 사용자 지정 측정 구성 ... 322
를 사용한 계약 요금 AWS License Manager ... 334
컨테이너 제품에 대한 Amazon SNS 알림 ... 367
Amazon SNS 주제: `aws-mp-subscription-notification` ... 367
Amazon SQS 대기열을 Amazon SNS 주제에 구독 ... 368
기계 학습 제품 ... 370
기계 학습 제품 이해 ... 370
SageMaker AI 모델 패키지 ... 370
SageMaker AI 알고리즘 ... 371

추론 모델 배포 371
제품 수명 주기 372
기계 학습 제품 요금 373
서비스 제한 및 할당량 375
보안 및 지적 재산 379
의 기계 학습 보고서 AWS Marketplace 380
SageMaker AI에서 제품 준비 381
코드를 이미지로 패키징 381
이미지 업로드 404
Amazon SageMaker AI 리소스 생성 407
에 제품 나열 AWS Marketplace 414
사전 조건 414
1단계: 새 목록 생성 417
2단계: 제품 정보 제공 418
3단계: 초기 제품 버전 추가 418
4단계: 요금 모델 구성 419
5단계: 환불 정책 구성 420
6단계: EULA 구성 420
7단계: 허용 목록 구성 420
제품 관리 421
..... 421
..... 422
..... 422
..... 422
..... 424
..... 425
..... 425
제품 제거 425
비공개 제안 생성 426
요구 사항 및 모범 사례 428
ML 제품의 일반 모범 사례 428
사용 정보 요구 사항 429
입력 및 출력 요구 사항 429
Jupyter Notebook 요구 사항 430
ML 제품 목록의 요구 사항 및 권장 사항 요약 430
문제 해결 434

제품 ARN을 추가할 때 400 오류 발생 ... 435
제품 ARN을 추가할 때 404 오류 발생 ... 435
제품 생성 시 SageMaker AI 액세스 거부 오류 ... 436
제품 생성 시 SageMaker AI 실패 ... 436
SaaS 제품 ... 438
SaaS 제품 시작하기 ... 438
사전 조건 ... 439
다음 단계 ... 439
SaaS 제품 수명 주기 ... 439
SaaS 제품 생성 ... 440
초기 SaaS 제품 페이지 만들기 ... 443
SaaS 제품 설정 구성 ... 444
SaaS 구독 제품 통합 ... 455
SaaS 계약 제품 통합 ... 458
SaaS 계약을 종량제 제품과 통합 ... 462
서버리스 SaaS 통합 배포 ... 465
SaaS 제품 계획 ... 466
가격 책정 계획 ... 466
결제 통합 계획 ... 467
Amazon SNS 통합 계획 ... 467
고객이 제품에 액세스하는 방법 계획 ... 467
SaaS 제품 가이드라인 ... 468
제품 설정 가이드라인 ... 468
고객 정보 요구 사항 ... 469
제품 사용 가이드라인 ... 469
아키텍처 가이드라인 ... 470
SaaS 제품 요금 ... 475
SaaS 구독 요금 ... 476
SaaS 계약 요금 ... 478
SaaS 무료 평가판 만들기 ... 481
SaaS 무료 평가판 제안 생성 ... 482
SaaS 무료 평가판 제안 취소 ... 483
고객 온보딩 ... 484
새 구매자를 적용하도록 SaaS 제품 구성 ... 484
보안 및 주문 ... 486
SaaS 제품에 대한 Amazon SNS 알림 ... 486

Amazon SNS 주제: aws-mp-entitlement-notification 487
Amazon SNS 주제: aws-mp-subscription-notification 488
SNS 주제에 대한 SQS 대기열 구독 490
AWS Marketplace 측정 및 권한 부여 서비스 API 액세스 490
사용량 측정 491
권한 부여 확인 497
SaaS 제품 통합 체크리스트 498
보고 501
SaaS 코드 예제 502
ResolveCustomer 코드 예제 502
GetEntitlement 코드 예제 503
BatchMeterUsage 코드 예제 505
사용량 할당 태그 지정을 사용하는 BatchMeterUsage 코드 예제(선택 사항) 507
사용 AWS PrivateLink 510
소개 510
제품 구성 511
에 제품 제출 AWS Marketplace 512
VPC 엔드포인트에 대한 구매자 액세스 512
부록: 체크리스트 514
의 전문 서비스 제품 AWS Marketplace 516
도움말 가져오기 516
전문 서비스 제품 시작 517
사전 조건 517
전문 서비스 제품 만들기 518
비공개 제안 생성 519
제품 정보 편집 521
제품 요금 편집 521
제품 가시성 편집 522
전문 서비스 제품 제거 523
제품 세부 정보 제공 523
제품 설명 524
추가 리소스 526
지원 정보 526
차원 사용 526
제품 표시 여부 526
제품 요구 사항 526

제품 설정 가이드라인 527
고객 정보 요구 사항 528
제품 사용 가이드라인 528
아키텍처 가이드라인 528
전문 서비스 제품 요금 529
비공개 제안 작동 방식 529
데이터 제품 531
제품 제출 532
제품 탭 사용 534
회사 및 제품 로고 요구 사항 535
유료 리패키징 소프트웨어 제출 요구 사항 535
하드웨어 구성 요소가 있는 제품에 대한 요구 사항 536
AWS CloudFormation에서 시작되는 제품(무료 또는 유료) 또는 사용량 기반 유료 AMI 제품 536
제품 업데이트 540
제품 변경 및 업데이트 541
시간 및 기대 542
AMIs 제출 AWS Marketplace 542
AMI 셀프 서비스 검사 542
AMI 복제 및 제품 코드 배정 543
최종 체크리스트 543
제품 마케팅 546
180일 GTM 아카데미 546
제품 출시 발표 547
AWS Marketplace 메시징 547
에 대한 검토 AWS Marketplace 548
에 연결 AWS Marketplace 549
AWS Marketplace 브랜딩 사용 549
판매자 AWS 로를 사용한 구매 이해 550
AWS 사전 조건을 사용하여 구매 551
..... 551
브랜딩 변경 사항 검토 551
AWS 보고로 구매 552
에서 제품에 직접 연결 AWS Marketplace 552
보도 자료 553
AWS Marketplace 상표 사용 지침 554
알림 556

이메일 알림 ... 556
이벤트 유형 ... 557
필드 설명 ... 565
알림 관리 ... 569
Amazon EventBridge 알림 ... 570
새 제안에 대한 이벤트 ... 571
변경 세트에 대한 이벤트 ... 575
보안 요약 보고서 이벤트 ... 577
지급 이벤트 ... 578
판매자 보고서, 데이터 피드 및 대시보드 ... 580
판매자 제공 데이터 피드 ... 581
데이터 피드의 저장 및 구조 ... 581
데이터 피드 액세스 ... 584
데이터 피드 사용 ... 588
데이터 피드 테이블 개요 ... 589
데이터 피드 쿼리 예제 ... 597
데이터 피드 ... 720
판매자 보고서 ... 760
보고서 액세스 ... 761
사용 가능한 AWS Marketplace 판매자 보고서 ... 761
일별 비즈니스 보고서 ... 761
일별 고객 구독자 보고서 ... 772
지출 보고서 ... 775
월별 청구 수익 보고서 ... 783
영업 보상 보고서 ... 792
보충 보고서 ... 795
계약 세부 정보 보고서 ... 796
판매자 대시보드 ... 797
대시보드 액세스 ... 761
재무 운영용 대시보드 ... 800
영업 운영을 위한 대시보드 ... 829
마케팅용 대시보드 ... 845
AWS Marketplace 공급업체 인사이트 ... 862
AWS Marketplace Vendor Insights 이해 ... 863
판매자로 설정 ... 864
보안 프로필 생성 ... 864

인증서 업로드 ... 865
자체 평가 업로드 ... 866
AWS Audit Manager 자동 평가 활성화 ... 868
프로필 보기 ... 873
판매자로 보안 프로필 보기 ... 873
스냅샷 태깅 ... 874
스냅샷 생성 ... 875
스냅샷 보기 ... 876
스냅샷 내보내기 ... 876
최근에 릴리스된 스냅샷 보기 ... 876
스냅샷 릴리스 연기하기 ... 877
스냅샷 목록의 기본 설정 변경 ... 877
액세스 제어 ... 878
AWS Marketplace Vendor Insights 판매자에 대한 권한 ... 878
`CreateDataSource` ... 879
`DeleteDataSource` ... 879
`GetDataSource` ... 879
`UpdateDataSource` ... 880
`ListDataSources` ... 880
`CreateSecurityProfile` ... 880
`ListSecurityProfiles` ... 880
`GetSecurityProfile` ... 880
`AssociateDataSource` ... 881
`DisassociateDataSource` ... 881
`UpdateSecurityProfile` ... 881
`ActivateSecurityProfile` ... 881
`DeactivateSecurityProfile` ... 882
`UpdateSecurityProfileSnapshotCreationConfiguration` ... 882
`UpdateSecurityProfileSnapshotReleaseConfiguration` ... 882
`ListSecurityProfileSnapshots` ... 882
`GetSecurityProfileSnapshot` ... 883
`TagResource` ... 883
`UntagResource` ... 883
`ListTagsForResource` ... 883
추가 리소스 ... 883
보안 ... 235

에 대한 IAM AWS Marketplace 886
사용자 생성 887
그룹 생성 또는 사용 889
사용자로 로그인 890
AWS Marketplace 판매자를 위한 정책 및 권한 891
AWS Marketplace 판매자를 위한 정책 892
AWS Marketplace 판매자에 대한 권한 892
예제 1: KYC 상태를 볼 수 있는 권한 896
예제 2: 비공개 제안에 대한 업그레이드 및 갱신을 생성할 수 있는 권한 897
예제 3: 제안 페이지에 액세스하고 새 비공개 제안을 생성할 수 있는 권한 897
예제 4: 설정 페이지에 액세스할 수 있는 권한 898
예제 5: 파일 업로드 페이지에 액세스할 수 있는 권한 899
IAM 그룹 사용 899
AWS 관리형 정책 899
AWSMarketplaceAmiIngestion 900
AWSMarketplaceFullAccess 901
AWSMarketplaceGetEntitlements 901
AWSMarketplaceMeteringFullAccess 901
AWSMarketplaceMeteringRegisterUsage 901
AWSMarketplaceSellerFullAccess 902
AWSMarketplaceSellerProductsFullAccess 905
AWSMarketplaceSellerProductsReadOnly 908
AWSMarketplaceSellerOfferManagement 909
AWSVendorInsightsVendorFullAccess 909
AWSVendorInsightsVendorReadOnly 909
정책 업데이트 910
AWS Marketplace 상거래 분석 서비스 계정 권한 915
Amazon SQS 권한 916
AWS Marketplace 측정 및 권한 부여 API 권한 918
SaaS 제품에 적용되는 IAM 정책 918
AMI 제품에 적용되는 IAM 정책 919
컨테이너 제품에 적용되는 IAM 정책 919
재판매 권한 부여를 위한 서비스 연결 역할 920
에 대한 서비스 연결 역할 권한 AWS Marketplace 921
에 대한 서비스 연결 역할 생성 AWS Marketplace 923
에 대한 서비스 연결 역할 편집 AWS Marketplace 924

에 대한 서비스 연결 역할 삭제 AWS Marketplace 924
AWS Marketplace 서비스 연결 역할에 지원되는 리전 924
를 사용하여 AWS Marketplace API 호출 로깅 AWS CloudTrail 925
AWS Marketplace 측정 API 로그 파일 항목 예제 925
문제 해결 931
일반 판매자 등록 오류 메시지 931
문서 기록 934
AWS 용어집 956
..... cmlvii

AWS Marketplace란 무엇인가요?

AWS Marketplace 는 고객이 솔루션을 구축하고 비즈니스를 운영하기 위해 타사 소프트웨어, 데이터 및 서비스를 검색, 구매, 배포 및 관리하는 데 사용할 수 있는 엄선된 디지털 카탈로그입니다. AWS Marketplace 에는 보안, 비즈니스 애플리케이션, 기계 학습 및 의료, 금융 서비스, 통신과 같은 특정 산업 전반의 데이터 제품과 같은 인기 있는 범주의 수천 가지 소프트웨어 목록이 포함되어 있습니다. 고객은 사전 구성된 소프트웨어를 빠르게 시작할 수 있으며 Amazon Machine Image(AMI), 서비스형 소프트웨어(SaaS) 및 기타 형식의 소프트웨어 솔루션을 선택할 수 있습니다. 고객이 타사 소프트웨어를 구성, 배포 및 관리할 수 있도록 도와하는 전문 서비스도 제공됩니다. 전체 제공 방법 목록은 제품 제공을 참조하세요.

구매자(구독자), 판매자(제공자) 또는 둘 다 AWS Marketplace 로를 사용할 수 있습니다. 를 보유한 사람은 누구나 구매자 AWS Marketplace 로를 사용할 AWS 계정 수 있으며 판매자로 등록할 수 있습니다. 판매자는 독립 소프트웨어 공급업체(ISV), 채널 파트너, 관리형 서비스 제공업체(MSP) 또는 AWS 제품 및 서비스와 함께 사용할 수 있는 제품을 제공하는 개인일 수 있습니다.

> **Note**
>
> 데이터 제품 공급자는 AWS Data Exchange 자격 요건을 충족해야 합니다. 자세한 내용은 AWS Data Exchange 사용 설명서의 AWS Data Exchange에서 데이터 제품 제공을 참조하세요.

적격 파트너는 프로그래밍 방식으로 AWS Marketplace 제품을 외부에 나열할 수 있습니다 AWS Marketplace. 적격 파트너가 되는 방법에 대한 자세한 내용은 AWS Marketplace 비즈니스 개발 파트너에게 문의하십시오.

다음 동영상에서는 판매에 대해 자세히 설명합니다 AWS Marketplace.

소개 AWS Marketplace

판매자 AWS Marketplace 로 사용

에서 소프트웨어 제품을 판매하는 프로세스에는 다음 7단계가 AWS Marketplace 포함됩니다.

판매자 프로세스

단계	작업	설명
1	등록	판매자는 먼저에 등록합니다 AWS Marketplace Management Portal. 기존 AWS 조직과 쉽게 연결할 수 AWS 계정 있는 새 전용를 구현하는 것이 좋습니다. AWS 파트너의 세금 정보가 관할 구역 자격 기준을 충족하는지 확인합니다. 지급을 받으려면 은행 계좌를 제공해야 합니다.
2	제품 유형 결정	판매하려는 제품 유형을 결정합니다. 에서 제품 유형을 생성하는 방법에 대한 자세한 내용은 다음을 AWS Marketplace참조하세요. • 의 AMI 기반 제품 AWS Marketplace • 의 컨테이너 기반 제품 AWS Marketplace • 의 기계 학습 제품 AWS Marketplace • 의 SaaS 기반 제품 AWS Marketplace • 의 전문 서비스 제품 AWS Marketplace • 데이터 제품(데이터 제품에 대한 자세한 내용은 AWS Data Exchange 사용 설명서의 AWS Data Exchange란? 참조)
3	제품 준비	패키지를 구성하고, 요금 체계를 설정하고, 제품을 전시할 범주를 결정하고, 제품이 관련 검색에서 검색되도록 키워드를 추가합니다. 공개 제품 리스팅과 비공개 제안에 대해 표준화된 라이선스 조건을 사용하여 조달 프로세스를 간소화할 수 있습니다.
4	제품 제출	제품 제출 프로세스를 사용하여 제품을 AWS Marketplace에서 사용할 수 있게 만듭니다. 요금 구조가 하나인 단일 Amazon Machine Image(AMI)처럼 제품이 단순할 수 있습니다. 또는 제품이 여러 AMI, AWS CloudFormation 템플릿, 복잡한 요금 옵션 및 결제 일정을 사용하므로 복잡할 수도 있습니다.
5	제품 마케팅	에 대한 인식을 높이고의 제품 페이지로 직접 트래픽을 유도 AWS Marketplace 하여 제품의 성공에 기여합니다 AWS Marketplace.
6	보고서 및 데이터 피드 보기	판매자로 등록한 후 AWS Marketplace Management Portal 를 사용하여 제품의 사용 보고서에 액세스합니다.는 제품 판매에 대한 정보를 수집하고 분석하는 도구를 AWS Marketplace 제공합니다.

단계	작업	설명
7	제품 관리	AWS Marketplace Management Portal을 사용하여 계정 및 제품 페이지를 관리합니다.

판매자는 AWS Marketplace Management Portal로 이동하여 등록합니다. 제품 사용 요금을 청구하려면 판매자로 등록할 때 세금 및 금융 정보도 입력해야 합니다. 또한 AWS Marketplace에서 검색이 가능하도록 회사 또는 자신을 나타내는 프로필을 생성합니다. 또한 AWS Marketplace Management Portal을 사용하여 제품에 대한 제품 페이지를 생성하고 관리합니다.

에 대한 계약 구조 AWS Marketplace

에서 판매되는 소프트웨어, 서비스 및 데이터 제품의 사용 AWS Marketplace 에는 구매자와 판매자 간의 계약이 적용됩니다. AWS 는 이러한 계약의 당사자가 아닙니다.

판매자의 계약에는 다음과 같은 내용이 포함됩니다.

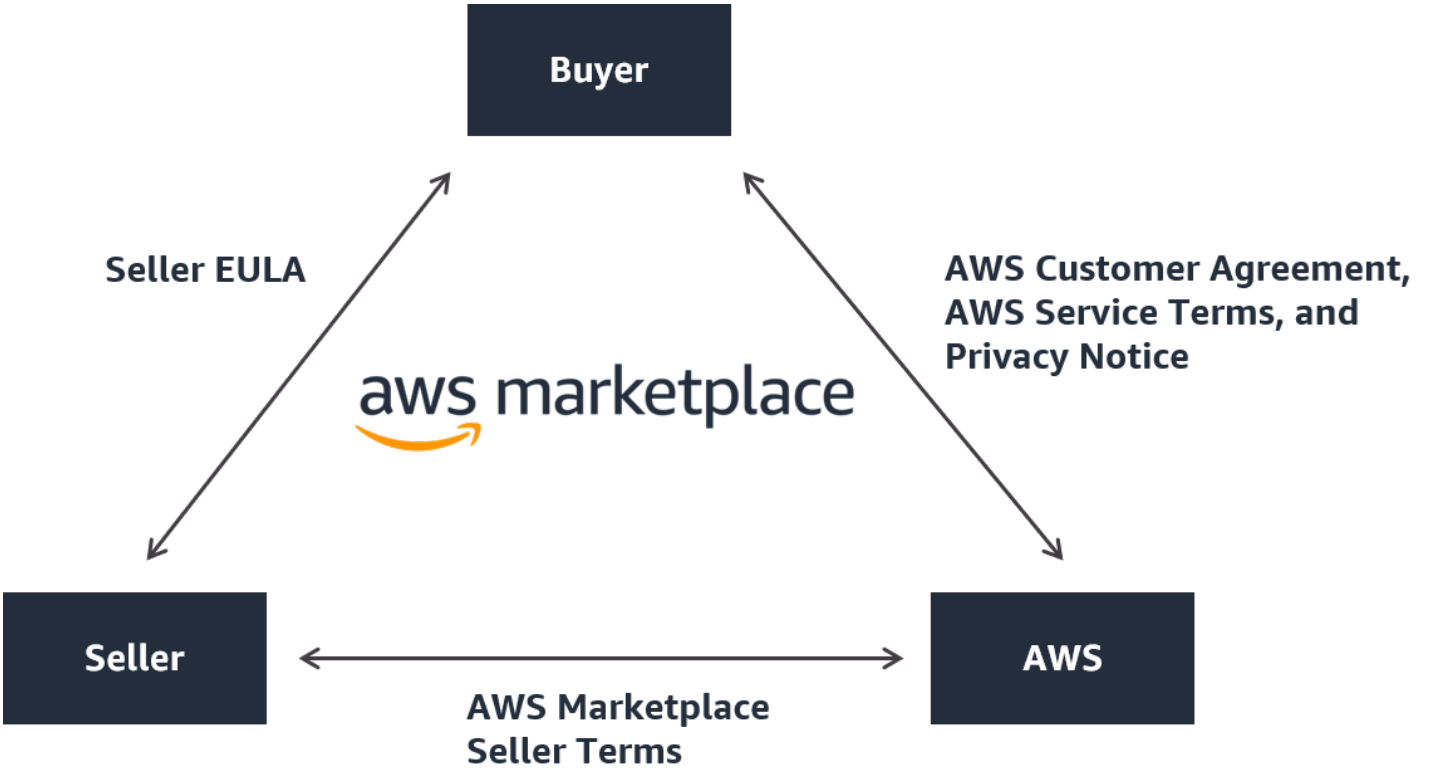
- 구매자와 체결하는 최종 사용자 라이선스 계약(EULA). AWS Marketplace의 공개 소프트웨어 목록에 대한 제품 목록 페이지에 있습니다. 많은 판매자가 표준 계약 AWS Marketplace (SCMP)을 기본 EULA로 사용합니다. SCMP를 비공개 제안 협상의 근거로 사용하고 수정 템플릿을 사용하여 SCMP를 수정할 수도 있습니다. 비공개 제안에는 당사자 간에 협상된 맞춤형 계약 조건도 포함될 수 있습니다.

  > **Note**
  >
  > 제안 유형 및 가격 모델에 따른 EULA 업데이트 시점에 대한 자세한 내용은 AWS Marketplace 구매자 안내서의 EULA 업데이트를 참조하세요.

- AWS Marketplace에서 이루어지는 활동을 규율하는 AWS Marketplace 판매자 약관.

구매자의 사용 AWS Marketplace 에는 AWS 서비스 약관, AWS 고객 계약 및 개인 정보 보호 고지가 적용됩니다.

다음 그림은 계약 구조를 보여줍니다 AWS Marketplace.



의 제품 요금 AWS Marketplace

에서는 AWS Marketplace제품을 무료로 사용하거나 관련 요금을 부과할 수 있습니다. 요금은 구매자 AWS 청구서의 일부가 되며, 구매자가 지불한 후에는가 판매자에게 AWS 지불합니다. 제품은 여러 가지 형태로 제공됩니다. 예를 들어 구매자의 AWS 계정을 사용하여 인스턴스화되는 Amazon Machine Image(AMI)로 제품을 제공할 수 있습니다. CloudFormation 템플릿을 사용하여 구매자에게 제공하도록 제품을 구성할 수도 있습니다. 제품은 ISV의 SaaS, 웹 액세스 제어 목록(웹 ACL), 규칙 세트 또는 AWS WAF에 대한 조건일 수도 있습니다. 제품은 ISV, 채널 파트너 또는 MSP의 전문 서비스일 수도 있습니다.

유연한 요금 옵션에는 무료 평가판, 시간당, 월별, 연간, 다년 및 기존 보유 라이선스 사용 모델(BYOL)이 포함되며 한 소스에서 청구됩니다.는 결제 및 결제를 AWS 처리하고 요금은 고객의 AWS 청구서에 표시됩니다.

소프트웨어 제품은 ISV의 표준 최종 사용자 라이선스 계약(EULA)을 사용하여 정가로 구매할 수 있습니다. 비공개 제안을 통해 맞춤형 요금 및 EULA를 사용하여 소프트웨어 제품을 제공하기도 합니다. 또한 계약에 따라 시간 또는 사용량 경계를 지정하여 구매하는 것도 가능합니다. 제품을 구독한 후 구매자는 AWS Service Catalog 를 사용하여 제품을 복사하고 구매자 조직에서 제품에 액세스하고 사용하는 방법을 관리할 수 있습니다. 구매자 경험에 대한 자세한 내용은 https://docs.aws.amazon.com/

marketplace/latest/buyerguide/service-catalog.html 섹션을 참조하세요. 요금에 대한 자세한 내용은 the section called “제품 요금” 단원을 참조하세요.

AWS Marketplace 판매자로 시작하기

독립 소프트웨어 공급업체(ISV), 채널 파트너, 관리형 서비스 제공업체(MSP) 또는 AWS 제품 및 서비스와 호환되는 제품을 제공하는 개인인 경우 판매자로 등록할 수 있습니다 AWS Marketplace. 판매자로 등록하는 것은 판매 대상 제품을 게시하기 위한 사전 조건입니다 AWS Marketplace. 거주지와 판매 중인 제품 유형에 따라 등록 요구 AWS Marketplace 사항이 다릅니다. 다음 섹션에서는 판매자 등록 프로세스, 요구 사항 및 관련 도구에 대한 개요를 제공합니다.

소프트웨어를 판매하려면 다음 단계를 AWS Marketplace따르세요.

- 무료 제품에 대한 판매자 요구 사항과 유료 제품에 대한 요구 사항을 검토합니다.
- 판매자 등록 프로세스를 완료합니다.
- 다음 단계 및 판매자 도구에 대해 자세히 알아보세요.

> **Notes**
>
> - AWS Marketplace 판매자로 등록하는 것은 AWS Data Exchange에 데이터 제품을 등록하고 에서 사용할 수 있도록 하기 위한 사전 조건입니다 AWS Marketplace. 이러한 요구 사항에 대한 자세한 내용은 AWS Data Exchange 사용 설명서의 AWS 데이터 교환에서 데이터 제품 제공을 참조하세요.
> - AWS Marketplace 판매자에게 필요한 권한에 대한 자세한 내용은 섹션을 참조하세요AWS Marketplace 판매자를 위한 정책 및 권한.
> - 제품 등록 수수료에 대한 자세한 내용은 AWS Marketplace Management Portal에서 AWS Marketplace 판매자 약관을 참조하세요.
> - 자주 묻는 질문에 대한 답변은 AWS Marketplace 판매자 FAQ를 참조하세요.

무료 소프트웨어 제품을 게시하기 위한 판매자 요구 사항

제품을 제공할 때 요금을 부과하는지 여부에 관계없이 해당 제품을 판매하고 AWS Marketplace있습니다. 고객에게 발생하는 비용은 0.00 USD이지만 귀사와 고객은 제품 사용에 대한 상호 계약에 동의하게 됩니다. 무료 제품만 제공하는 경우, AWS Marketplace에 은행 정보를 제공하지 않아도 됩니다.

에서 무료 제품을 생성하고 제공하려면 다음을 AWS Marketplace수행해야 합니다.

- 공개적으로 사용할 수 있는 전체 기능 프로덕션 지원 소프트웨어를 판매합니다.

- 정의된 고객 지원 프로세스 및 지원 조직을 보유합니다.
- 소프트웨어를 정기적으로 업데이트하고 취약성을 극복할 수 있는 수단을 제공합니다.
- AWS Marketplace에서 제품을 판매할 때 모범 사례와 지침을 준수합니다.
- 선의의 AWS 고객이 되고 AWS Marketplace 판매자를 위한 이용 약관의 요구 사항을 충족합니다.

유료 제품에 대한 추가 판매자 요구 사항

제품에 요금을 부과하거나 기존 보유 라이선스 사용(BYOL) 제품을 제공하는 경우 다음과 같은 요구 사항을 충족하고 추가 정보도 제공해야 합니다.

- 귀사는 적격 관할 구역 또는 그러한 지역 중 하나에서 설립되거나 법인 인가를 받은 사업체여야 합니다.
- 세금 및 은행 계좌 정보를 제공해야 합니다. 미국에 있는 기업의 경우 미국 기반 은행에서 개설한 W-9 양식 및 은행 계좌가 필요합니다. 공개 또는 비공개 제안을 생성하려면 은행 계좌 정보를 지급 기본 설정과 연결해야 합니다.
- 미국 이외 지역의 판매자는 (i) W-8 양식, 부가가치세(VAT) 또는 상품 및 서비스세(GST) 등록 번호, (ii) 적격 관할 구역의 SWIFT 코드가 있는 은행 계좌를 제공해야 합니다. 필요한 경우 Hyperwallet에서 가상 미국 은행 계좌에 등록할 수 있습니다.
- 데이터 제품을 제공하려면 사례 생성 마법사를 통해 온보딩도 요청해야 합니다 지원.
- Amazon Web Services EMEA SARL을 통해 유럽, 중동 및 아프리카(EMEA)(터키 및 남아프리카 제외)의 국가 및 지역에 AWS 계정 기반을 둔 고객에게 제품을 판매하려면 고객 파악 프로세스를 완료해야 합니다. 또한 다음과 같습니다.
  - 최대 2개의 대금이 지급됩니다(AWS Inc. 및 Amazon Web Services EMEA SARL을 통한 트랜잭션의 경우).
  - 지역에 따라 특정 거래의 리스팅 수수료에 세금이 부과될 수 있습니다. 세금에 대한 자세한 내용은 AWS Marketplace 판매자 세금 도움말 페이지를 참조하세요. 리스팅 수수료에 부가가치세(VAT)가 부과되는 경우 AWS Marketplace 에서 세금 신고가 된 인보이스를 제공합니다.
  - Amazon Web Services EMEA SARL에 대한 자세한 내용은 Amazon Web Services Europe FAQ 웹 사이트의 AWS EMEA Marketplace - 판매자를 참조하세요.

에 판매하려면 AWS GovCloud (US) Region판매자에게 AWS GovCloud (US) 계정이 있어야 합니다. ITAR 요구 사항에 대한 자세한 내용은 AWS GovCloud (US) 사용 설명서를 참조하세요.

AWS Marketplace 판매자 요구 사항 또는 등록 프로세스에 대한 질문은 AWS Marketplace 판매자 운영 팀에 문의하세요.

유료 상품에 대한 적격 관할 구역

에서 유료 소프트웨어를 판매하려면 다음 국가 또는 SARs 중 하나의 영주권자 또는 시민이거나 해당 국가에서 조직되거나 설립된 사업체여야 AWS Marketplace합니다.

- 호주[1]
- 바레인[1 2]
- 콜롬비아1 2
- 유럽 연합(EU) 회원국[1]
- 홍콩 SAR
- 이스라엘[1 2]
- 일본3
- 뉴질랜드[1]
- 노르웨이[1 2]
- 카타르
- 대한민국
- 스위스[1 2]
- 아랍 에미리트 연합국(UAE)[1 2]
- 영국(UK)[1]
- 미국(US)

[1] 이러한 국가에서 유료 제품을 판매하는 판매자는 설립 국가의 VAT 등록 정보를 제공해야 합니다.

[2] 판매자가 구매자와 같은 국가에 거주하는 경우 판매자가 세금 계산서 발행, 수금 및 송금을 책임져야 할 수도 있습니다. 세무사에 문의하세요.

3 2025년 4월 1일부터 10% 일본 소비세(JCT)의 징수 및 송금과 AWS 일본에서 판매한 제품에 대한 세금 적격 인보이스(TQI) 발행에 AWS Marketplace대해 다음 절차가 적용됩니다.

판매자 및 고객 시나리오	세금 징수 및 인보이스 절차
판매자 - 주소가 일본 외부인 독립 소프트웨어 공급업체(ISV)입니다. 고객 - AWS 일본에 주소가 있는 계정입니다.	지정된 플랫폼 과세 규칙에 따라 AWS Japan은 다음을 수행합니다. • 일본 소비세(JCT) 10%를 징수합니다.

판매자 및 고객 시나리오	세금 징수 및 인보이스 절차
	• AWS 일본 JCT 번호 T6011001106696으로 고객에게 세금 적격 인보이스(TQI)를 발행합니다. • 수집된 일본 소비세(JCT)를 일본 조세청(JTA)으로 송금합니다. 중복을 방지하려면 ISV는 일본 소비세(JCT) 징수 또는 완화와 세금 적격 인보이스(TQIs.
판매자 - 주소가 일본인 독립 소프트웨어 공급업체(ISV)입니다. 고객 - AWS 일본에 주소가 있는 계정입니다.	특별 기관 규칙에 따라 AWS Japan은 다음을 수행합니다. • 일본 소비세(JCT) 10%를 징수합니다. • AWS 일본 JCT 번호 T6011001106696으로 고객에게 세금 적격 인보이스(TQI)를 발행합니다. • 수집된 JCT를 ISV에 지급합니다. ISV는 다음을 수행해야 합니다. • Amazon Marketplace 관리 포털에 일본 기업 번호와 일본 소비세(JCT) 번호를 입력합니다. • Amazon Marketplace 관리 포털에서 사용할 수 있는 세금 적격 인보이스(TQI) 정보를 사용하여 수집된 JCT를 일본 조세당국(JTA)으로 전송합니다.

판매자 및 고객 시나리오	세금 징수 및 인보이스 절차

채널 파트너 비공개 제안(CPPO), 일본 외 ISV
 	
1. 첫 번째 트랜잭션
• 판매자 - 주소가 일본 외부인 독립 소프트웨어 공급업체(ISV)입니다.
• 고객 - AWS Marketplace 주소가 일본인 채널 파트너.
2. 두 번째 트랜잭션
• 판매자 - 동일한 AWS Marketplace 채널 파트너.
• 고객 - AWS 일본에 주소가 있는 계정입니다.
	첫 번째 트랜잭션의 경우 채널 파트너는 역방향 요금 세금 메커니즘에 따라 일본 소비세(JCT)를 보고해야 합니다.
	
	두 번째 트랜잭션의 경우:
	
	• 특별 기관 규칙에 따라 AWS Japan은 다음을 수행합니다.
	  • 일본 소비세(JCT) 10%를 징수합니다.
	  • AWS 일본 JCT 번호 T6011001106696으로 고객에게 세금 적격 인보이스(TQI)를 발행합니다.
	  • 수금된 일본 소비세(JCT)를 채널 파트너에게 지불합니다.
	• 채널 파트너는 다음을 수행해야 합니다.
	  • Amazon Marketplace 관리 포털에 일본 기업 번호와 일본 소비세(JCT) 번호를 입력합니다.
	  • Amazon Marketplace 관리 포털에서 사용할 수 있는 세금 적격 인보이스(TQI) 정보를 사용하여 수집된 JCT를 일본 조세당국(JTA)으로 전송합니다.

판매자 및 고객 시나리오	세금 징수 및 인보이스 절차

채널 파트너 비공개 제안(CPPO), 일본의 모든 판매자 및 고객
 	
1. 첫 번째 트랜잭션
• 판매자 - 주소가 일본인 독립 소프트웨어 공급업체(ISV)입니다.
• 고객 - AWS Marketplace 주소가 일본인 채널 파트너.
2. 두 번째 트랜잭션
• 판매자 - 동일한 AWS Marketplace 채널 파트너.
• 고객 - AWS 일본에 주소가 있는 계정입니다.
	첫 번째 트랜잭션의 경우:
	
	• 채널 파트너는 ISV로부터 세금 적격 인보이스(TQI)를 받습니다.
	• AWS 일본은 인보이스를 발행하거나 일본 소비세(JCT)를 징수하지 않습니다.
	
	두 번째 트랜잭션의 경우:
	
	• 특별 기관 규칙에 따라 AWS Japan은 다음을 수행합니다.
	  • 일본 소비세(JCT) 10%를 징수합니다.
	  • AWS 일본 JCT 번호 T6011001106696을 사용하여 고객에게 세금 적격 송장(TQI)을 발행합니다.
	  • 수금된 일본 소비세(JCT)를 채널 파트너에게 지불합니다.
	• 채널 파트너는 다음을 수행해야 합니다.
	  • Amazon Marketplace 관리 포털에 일본 기업 번호와 일본 소비세(JCT) 번호를 입력합니다.
	  • Amazon Marketplace 관리 포털에서 사용할 수 있는 세금 적격 인보이스(TQI) 정보를 사용하여 수집된 JCT를 일본 조세당국(JTA)으로 전송합니다.

판매자 및 고객 시나리오	세금 징수 및 인보이스 절차
일본 이외의 채널 파트너 비공개 제안(CPPO), ISV 및 채널 파트너 1. 첫 번째 트랜잭션 • 판매자 - 주소가 일본 외부인 독립 소프트웨어 공급업체(ISV)입니다. • 고객 - AWS Marketplace 주소가 일본 외부인 채널 파트너. 2. 두 번째 트랜잭션 • 판매자 - 동일한 AWS Marketplace 채널 파트너. • 고객 - AWS 일본에 주소가 있는 계정입니다.	첫 번째 거래의 경우 AWS 일본은 JCT 범위를 벗어나기 때문에 인보이스를 발행하거나 일본 소비세(JCT)를 징수하지 않습니다. 두 번째 트랜잭션의 경우: • 지정된 플랫폼 비즈니스 규칙에 따라 AWS Japan은 다음을 수행합니다. • 일본 소비세(JCT) 10%를 징수합니다. • AWS 일본 JCT 번호 T6011001106696으로 고객에게 세금 적격 인보이스(TQI)를 발행합니다. • 수집된 일본 소비세(JCT)를 일본 조세청(JTA)으로 송금합니다. 중복을 방지하려면 채널 파트너는 일본 소비세(JCT) 징수 또는 완화와 세금 적격 인보이스(TQIs.

판매자 및 고객 시나리오	세금 징수 및 인보이스 절차
채널 파트너 비공개 제안(CPPO), 일본 외 채널 파트너 1. 첫 번째 트랜잭션 • 판매자 - 주소가 일본인 독립 소프트웨어 공급업체(ISV)입니다. • 고객 - AWS Marketplace 주소가 일본 외부인 채널 파트너. 2. 두 번째 트랜잭션 • 판매자 - 동일한 AWS Marketplace 채널 파트너. • 고객 - AWS 주소가 일본인 계정.	첫 번째 거래의 경우 AWS 일본은 JCT 범위를 벗어나기 때문에 인보이스를 발행하거나 일본 소비세(JCT)를 징수하지 않습니다. 두 번째 트랜잭션의 경우: • 지정된 플랫폼 비즈니스 규칙에 따라 AWS Japan은 다음을 수행합니다. • 일본 소비세(JCT) 10%를 징수합니다. • AWS 일본 JCT 번호 T6011001106696으로 고객에게 세금 적격 인보이스(TQI)를 발행합니다. • 수집된 일본 소비세(JCT)를 일본 조세청(JTA)으로 송금합니다. 중복을 방지하려면 채널 파트너는 일본 소비세(JCT) 징수 또는 완화와 세금 적격 인보이스(TQIs.

VAT, 인보이스 발행 및 판매자로서의 세금 의무에 대한 자세한 내용은 Amazon Web Service 세금 도움말AWS Marketplace 의 판매자를 참조하세요.

기업체가 위에 나열된 국가 또는 SARs 중 하나에 설립되지 않은 경우 AWS Marketplace 관할 구역 외부의 기업을 위한 리소스를 참조하세요.

> **Note**
>
> AWS 이전에는 Amazon Internet Services Private Limited라고 불렸던 India Private Limited(AWS 인도) 계정은에 등록하는 데 사용할 수 없습니다 AWS Marketplace.

AWS Marketplace Management Portal

AWS Marketplace Management Portal을 사용하여 AWS Marketplace에서 판매 중인 제품을 관리할 수 있습니다. 등록된 모든 판매자는 제품을 생성하는 데 사용한 계정의 AWS 자격 증명을 AWS

Marketplace Management Portal 사용하여에 액세스할 수 있습니다. 사용 계정은 고객이 귀사의 제품을 구독하는 시점에 기록되어 있는 판매자로 정의됩니다. 제품의 판매자 기록에 해당하는 특정 계정을 확인하기 위해 도움이 필요하면 AWS Marketplace 판매자 작업 팀에 문의하세요.

포털에서 다음 작업을 수행할 수 있습니다.

- AWS Marketplace 판매자로 등록합니다.
- 제품 페이지에서 새 소프트웨어 제품을 제출하고 기존 소프트웨어 제품을 업데이트합니다.
- 요청의 상태를 모니터링합니다.
- 새 소프트웨어 제품을 생성하고 관리하는 데 필요한 파일을 업로드합니다.
- 시장 진출 활동을 적극 활용하여 채널 수익 증대로 이어질 수 있도록 소프트웨어 제품을 관리합니다.
- 캠페인으로 인한 사용 및 수익을 포함하여 출시 시간 이내의 마케팅 활동 결과를 측정합니다.
- 고객 서비스 담당자가 실시간으로 고객 데이터를 검색할 수 있습니다.
- 자동 Amazon Machine Image(AMI) 스캔을 시작하여 취약성을 감지합니다.

> **Note**
>
> 데이터 제품은 AWS Data Exchange 콘솔에서 게시하고 관리합니다. AWS Data Exchange 공급자는 AWS Marketplace Management Portal 를 사용하여 판매자로 등록하고, AWS Data Exchange 온보딩을 요청하고, 판매자 보고서에 액세스하고, 환불 요청을 제출할 수 있습니다.

AWS Marketplace 판매자로 등록

독립 소프트웨어 공급업체(ISV), 채널 파트너, 관리형 서비스 제공업체(MSP) 또는 AWS 제품 및 서비스와 호환되는 제품을 제공하는 개인인 경우 판매자로 등록할 수 있습니다 AWS Marketplace. 판매자로 등록하는 것은 판매 대상 제품을 게시하기 위한 사전 조건입니다 AWS Marketplace. 다음 섹션에서는 판매자로 성공적으로 등록하는 단계를 안내합니다 AWS Marketplace.

에서 판매자로 등록하려면 기존를 사용하거나 새 계정을 AWS 계정 생성할 AWS Marketplace수 있습니다. 모든 AWS Marketplace 상호 작용은 선택한 계정에 연결됩니다. AWS Marketplace 에서는 루트 계정 자격 증명을 사용하는 AWS Marketplace Management Portal 대신 AWS Identity and Access Management (IAM) 역할을 사용하여에 로그인할 것을 강력히 권장합니다. 자세한 내용은 AWS Marketplace 보안을 참조하세요. IAM을 사용하여 다양한 권한을 가진 여러 사용자가에 액세스할 수 AWS 계정 있도록 기본를 구성할 수도 있습니다 AWS Marketplace Management Portal. 자세한 내용은 the section called “에 대한 IAM AWS Marketplace” 단원을 참조하십시오.

> **Note**
>
> 중요한 알림 및 계정 업데이트를 받으려면 조직 내 적절한 연락처가 액세스할 수 있는 이메일 주소를 제공해야 합니다. 일부 AWS 팀은 등록된 이메일 주소에서만 이메일을 수신할 수 있으므로 별칭을 이메일 주소로 대체할 수 없습니다.

판매자로 등록하려면 다음 단계를 완료하세요.

주제

- 1단계: 공개 프로필 만들기
- 2단계: 세금 정보 제공
- 3단계: 은행 계좌 정보 제공
- 4단계: 지급 기본 설정 지정
- 5단계: 고객 파악(KYC) 프로세스 완료
- 6단계: 은행 계좌 확인 프로세스 완료
- 이미 AWS Marketplace 판매자이신가요?

1단계: 공개 프로필 만들기

등록할 첫 번째 단계는 기본 AWS Marketplace 계정으로 AWS 계정 사용할를 선택하고 콘솔에서 잠재적 구매 AWS Marketplace 자에게 표시되는 정보를 제공하는 것입니다. 이 계정은의 제품에 대한 레코드 판매자 AWS Marketplace 가 되며의 보고, 지급 및 통신 AWS Marketplace 에 사용됩니다.

AWS 계정 를 사용하여 판매자로 등록하고 제품을 나열 AWS Marketplace한 후에는 제품과 연결된 계정을 변경할 수 없습니다. 새 계정을 사용하여 AWS Marketplace 판매자로 등록하는 것이 좋습니다. 하지만 기존 계정이 2017년 9월 27일 이후에 생성되었다면 사용해도 됩니다.

공개 프로필을 생성하는 방법

1. AWS Marketplace Management Portal (AMMP)에서 지금 등록을 선택한 다음 판매자에 로그인합니다 AWS 계정.
2. 공개 프로필 추가를 선택하여 판매자 정보를 제공합니다.

공개 프로필을 작성한 후에는 무료 제품을 게시하고 판매할 수 있습니다. 유료 제품을 판매하려면 세금 및 은행 정보를 제공해야 합니다.

2단계: 세금 정보 제공

가 제품 판매에 대한 세금을 정확하게 보고하고 원천징수 AWS Marketplace 할 수 있도록 세금 정보 및 해당하는 경우 부가가치세(VAT) 정보를 제공해야 합니다.

세금 정보를 제공하는 방법

1. AWS Marketplace Management Portal에 로그인하고 설정을 선택합니다.
2. 결제 정보 섹션에서 세금 대시보드로 이동을 선택합니다.
3. 미국 세금 인터뷰를 완료합니다.
   - 전문 서비스를 판매하려면 DAC7에 대한 세금 설문지를 작성해야 AWS Marketplace합니다.
   - "납세 인터뷰 위치가 비즈니스 위치와 일치하지 않습니다."라는 오류 메시지가 표시되면 Billing and Cost Management에 제공된 은행 및 세금 정보가에 입력된 내용과 일치하는지 확인합니다 AWS Marketplace Management Portal. 세금 인터뷰 위치는 사업장 위치와 일치해야 합니다.
4. 세금 정보를 입력한 후 설정 페이지로 돌아가서 사용 가능한 경우 VAT 정보 완료를 선택합니다. 이 옵션을 선택하면 AWS Billing 콘솔의 세금 설정 페이지로 리디렉션됩니다.

> **Note**
>
> VAT 정보 섹션은 VAT를 지원하는에 AWS 리전 있는 경우에만 사용할 수 있습니다.

추가 세금 등록

특정 지역의 판매에 현지 사업자 등록 번호(TRN)를 사용하는 경우 AWS Marketplace 계정에 보충 TRN으로 추가할 수 있습니다. 예를 들어 보충 스위스 TRN은 스위스 구매자의 세금 징수 및 인보이스 발행을 관리하는 데 사용됩니다. 보충 TRN과 관련된 비즈니스 주소는 독립 소프트웨어 공급업체, 채널 파트너 및 구매자와 공유할 수 있습니다.

보충 TRN이 없으면는 기본 TRN을 사용하여 세금 관할 구역을 AWS Marketplace 결정합니다. 자세한 내용은 가 계정의 위치를 AWS 어떻게 결정하나요?를 참조하세요.

보충 세금 등록을 추가하려면

1. AWS Marketplace Management Portal에 로그인합니다.
2. 설정을 선택합니다.

3. 설정 페이지에서 세금 설정 탭을 선택합니다.
4. 보충 세금 정보 추가를 선택합니다.
5. 보충 세금 등록 세부 정보를 입력합니다. TRN은 부가가치세(VAT) 번호, VAT ID, VAT 등록 번호 또는 사업자 등록 번호일 수 있습니다.
6. 제출을 선택합니다.

세금 문서 액세스

에서 1099 양식과 같은 세금 문서에 액세스할 수 있습니다 AWS Marketplace Management Portal.

세금 문서에 액세스하려면

1. AWS Marketplace Management Portal에 로그인하고 설정을 선택합니다.
2. 결제 정보 섹션으로 이동합니다.
3. 관련 세금 양식(1099K 또는 DAC7)을 선택합니다.
4. 세금 양식을 사용할 수 있는 경우 세금 대시보드 페이지에서 다운로드할 수 있습니다.

3단계: 은행 계좌 정보 제공

AWS Marketplace에서 유료 제품을 판매하려는 모든 판매자는 적격 관할 구역의 은행 계좌가 필요합니다. 은행 계좌에서 USD 지불을 수락할 수 있어야 합니다.

> **Note**
>
> 유료 제품을 제공할 수 있는 국가 목록은 섹션을 AWS Marketplace참조하세요유료 상품에 대한 적격 관할 구역.

은행 정보를 제공하려면

1. AWS Marketplace Management Portal에 로그인하고 설정을 선택합니다.
2. 결제 정보 섹션에서 전체 은행 정보를 선택합니다.
3. 은행 계좌에 대한 필수 정보를 제공합니다. 하나 이상의 은행 계좌를 제공할 수 있습니다. 이러한 은행 계좌는 미국 ACH 계좌, 적격 관할 구역의 SWIFT 은행 계좌 또는 Hyperwallet 계좌일 수 있습니다.

> **Note**
>
> 세금 정보(및 해당하는 경우 부가가치세 정보)를 아직 제공하지 않은 경우 은행 정보를 제공할 수 없습니다.

미국 기반 은행 계좌를 사용하여 USD 지불을 수락하려면에서 미국 계좌를 제공할 Hyperwallet 수 있습니다.

Hyperwallet은 지원되는 통화로 다른 은행 계좌로 자금을 이체할 수 있는 독립 서비스 공급자입니다. 한정된 기간 동안은 AWS Marketplace 지급금과 관련하여 특정한 Hyperwallet 서비스 요금을 지불할 필요가 없습니다.

- AWS Marketplace 판매자 Hyperwallet 계정에 계정 세부 정보를 추가하면 AWS Marketplace 가 판매자로서의 AWS Marketplace 상태를 확인하기 Hyperwallet 위해와 이름, 이메일 주소 및 계정 번호를 공유하는 데 동의하고 승인합니다.
- 환율은 물론 Hyperwallet 서비스 사용(자국 통화로 자금을 이체하는 데 필요한 이체 수수료 및 환전 수수료 포함)에 별도의 요금이 적용될 수 있습니다. Hyperwallet 서비스 요금은 제한된 기간 동안 면제되며 유료 제품의 수익금을 Hyperwallet 계정으로 AWS Marketplace 지급하는 경우에만 면제됩니다. 자세한 내용은 Hyperwallet 사이트의 요금 섹션을 참조하거나 Hyperwallet에 문의하여 자세한 내용을 확인하고 해당 요금을 검토하세요. Hyperwallet의 서비스에 대한 자세한 내용은 Hyperwallet 지원 사이트를 참조하세요.

Hyperwallet에 등록을 시작하고 미국 은행 계좌를 개설하는 방법

1. AWS Marketplace Management Portal에 로그인하고 설정을 선택한 다음, 결제 정보 섹션에서 전체 은행 정보를 선택합니다.
2. 현재 Hyperwallet 계정이 없고 AWS Marketplace에서 사용할 계정이 필요한 경우 미국 은행 계좌가 있나요? 질문과 Hyperwallet에 등록하셨나요? 질문에 아니오를 선택합니다. 개인 식별 번호(PIN)와 Hyperwallet에 가입할 수 있는 링크가 제공됩니다.
3. Hyperwallet 계정을 활성화한 후, Hyperwallet 등록 포털에서 설명하는 단계에 따라 등록을 완료하고 예금 계좌 정보를 받습니다.
4. 에서 계정을 얻은 경우 Hyperwallet에 AWS 계정 로그인하여에 Hyperwallet 계정 정보를 추가합니다AWS Marketplace Management Portal. 설정을 선택한 다음, 결제 정보 섹션에서 전체 은행 정보를 선택합니다.

> **Note**
>
> Hyperwallet 계정의 정보를 업데이트하려면 Hyperwallet 포털의 Hyperwallet 지원 팀에 문의하세요. 시간 및 연락처 정보는 지원 탭을 참조하세요.

4단계: 지급 기본 설정 지정

일별 또는 월별 지급 옵션을 선택하고, 지급을 받을 날짜를 선택하고, 은행 계좌에 통화를 할당합니다. 자세한 정보는 AWS Marketplace 판매자에 대한 지급 기본 설정 섹션을 참조하세요.

5단계: 고객 파악(KYC) 프로세스 완료

판매자가 다음을 수행하려면 고객 파악(KYC) 프로세스를 완료해야 합니다.

- 대한민국 거래에 대한 지불 받기
- 영국 기반 은행 계좌를 사용하여 거래
- EMEA 고객에게 판매 간소화

이 단계는 선택 사항이지만 다른 리전에 권장됩니다. KYC에는 회사, 주요 연락처, 수익 소유권 및 지원 문서에 대한 추가 정보 제공이 포함됩니다.

> **Note**
>
> 확인된 사용자만 환불을 처리하고, KYC 정보를 수정하고, 은행 계좌 세부 정보와 같은 금융 정보를 변경할 수 있습니다. 확인된 사용자는 KYC 확인 사용자 및 확인을 완료한 보조 사용자입니다.

KYC는 금융 기관 및 온라인 기업이 고객의 신원을 확인하기 위해 사용하는 규정 준수 요구 사항입니다. 이 요구 사항은 다음과 같습니다.

- 대한민국에서 지정된 금융 거래 정보의 보고 및 사용에 관한 법률 제5조의 2
- Payment Services Directive(PSD 2)
- 금융 기관을 관리하는 유럽 연합 돈세탁 방지 지침

AWS Marketplace Amazon Web Services EMEA SARL을 통한 트랜잭션은 결제 서비스를 사용하기 위해 자격 증명 확인이 필요한 룩셈부르크의 라이선스 전자화폐 기관인 Amazon Payments Europe, S.C.A.(APE)를 통해 처리됩니다.

> **Note**
>
> 거주지가 영국인 은행 계좌는 EUR 및 GBP로만 지급을 받을 수 있습니다. 다른 통화로 지급하려면 다른 관할 구역의 계좌가 필요합니다. 또한 Amazon Web Services EMEA SARL을 통한 판매만 가능합니다.

KYC 프로세스를 완료하는 방법

1. AWS Marketplace 관리 포털에서 설정을 선택합니다.
2. 계정 요약 섹션에서 표시된 국가가 올바른지 확인합니다.

   > **Note**
   >
   > 정보 링크를 선택하여 국가를 변경하는 방법을 알아보세요.

3. KYC 정보로 이동을 선택하거나 고객확인제도(KYC) 탭을 선택하고 KYC 규정 준수 시작을 선택하면 KYC 등록 포털로 리디렉션됩니다.

   에서 정보를 사용하고 공유하는 방법에 대한 자세한 내용은 Amazon Payments Europe 개인 정보 보호 공지를 AWS Marketplace참조하세요.
4. KYC 개요로 이동을 선택합니다.
5. 고객확인제도(KYC) 개요에서 필수 정보 및 문서 목록을 끝까지 읽고, 필수 문서를 아직 수집하지 않았으면 지금 수집합니다. KYC 규정 준수 계속을 선택합니다.
6. 지침에 따라 기본 세부 정보를 입력합니다. Amazon Payments Europe 이용 약관을 검토한 후 동의 후 계속을 선택합니다.

   KYC 프로세스의 다음 페이지 또는 다음 단계를 진행하면 Amazon Payments Europe 이용 약관에 동의한 것으로 간주됩니다.

   궁금한 점이 있으면 콘솔 오른쪽에 있는 자주 묻는 질문(FAQ) 을 참조하세요.
7. 지침에 따라 필수 비즈니스 정보를 입력하고 다음을 선택합니다.

> **Note**
>
> 다음 단계로 이동하려면 다음을 선택할 때마다 정보가 저장됩니다.

8. 지침에 따라 필수 연락 담당자 정보를 입력하고 다음을 선택합니다.
9. 수익 소유자가 연락 담당자와 동일 인물인지 선택하고, 필요하다면 수익 소유자를 추가하고(최대 4명), 추가 정보를 확인한 후 다음을 선택합니다.
10. 법적 대리인이 연락 담당자 또는 수익 소유자와 동일한지 여부를 선택합니다. 법적 대리인이 다른 엔터티인 경우 필요한 정보를 제공하고 항목을 저장한 후 다음을 선택합니다.
11. 추가 문서의 경우 사업자 등록증, 신분증 및 위임장(해당하는 경우)을 업로드합니다.

> **Note**
>
> 승인서가 필요한 경우 다음 샘플을 사용할 수 있습니다.
>
> ```
> *Letterhead of the company*
>
> POWER TO ACT ON BEHALF OF THE COMPANY
> The undersigned **name of Company here** (herein after, the "Company"), duly
>  represented by
> (name, date of birth, and function) **add full name, date of birth, and
>  function of the signatory here**,
> confirms that **add full name of the Person of Contact here** is authorized
>  to open an Amazon Web
> Services account with Amazon Payments, accept the User Agreement and other
>  Policies, have access
> to the Amazon Web Services account, and initiate transactions in the name
>  and on behalf of the Company.
>
>     Dated this:
>
>     Signed by:
> ```

12. 검토 및 제출에서 입력한 모든 정보를 검토하고 확인합니다.

    필요한 경우 편집을 선택하여 이전 섹션으로 돌아갈 수 있습니다.
13. 확인을 위해 제출을 선택합니다.

그러면 KYC 규정 준수 상태가 검토됩니다(일반적으로 24시간 이내). 검토가 완료되면 이메일 메시지를 통해 알림을 받게 됩니다. 전체 KYC 프로세스는 일반적으로 약 2주가 소요됩니다.

설정 탭으로 돌아가서 계정 요약 카드에서 KYC 규정 준수 상태를 확인할 수 있습니다. KYC 상태에 대한 자세한 내용을 보려면 계정 요약 카드에서 고객확인제도(KYC) 탭을 선택합니다. 검토가 완료될 때까지 검토 중으로 표시됩니다.

KYC가 확인된 후, 결제 정보 탭에서 은행 거래 내역서를 제공해야 합니다. 그래야만 APE를 통해 대금을 받을 수 있습니다.

(선택 사항) 고객확인제도의 보조 사용자 추가

> **Note**
>
> 사용자는 결제 정보를 업데이트하려면 다중 인증(MFA)을 활성화해야 합니다. MFA에 대한 자세한 내용은 IAM 다중 인증(MFA)을 참조하세요.

보조 사용자는 KYC 정보를 수정하고, 대금 또는 환불 흐름을 관리하고, 은행 계좌 정보와 같은 금융 정보를 변경할 수 있는 개인입니다.

KYC 확인을 거친 보조 사용자만이 앞에서 언급한 업데이트를 수행할 수 있습니다. 이러한 보조 사용자에게는 루트 계정 소유자와 동일한 지속적 심사 컨트롤이 적용됩니다.

KYC 확인을 받으려는 보조 사용자는 5단계: 고객 파악(KYC) 프로세스 완료의 절차를 완료해야 합니다.

고객확인제도의 보조 사용자를 추가하는 절차

1. 사용자에게 AWS Marketplace Management Portal에 로그인하라고 요청합니다.
2. 설정 탭으로 이동합니다.
3. 고객확인제도(KYC) 탭을 선택하고 보조 사용자 정보 섹션을 봅니다.
4. 전체 보조 사용자 정보를 선택합니다.

   그러면 보조 사용자 등록 포털로 리디렉션됩니다.
5. 보조 사용자 등록 포털에서 필수 필드를 작성하고 다음을 선택합니다.
6. 검토 및 제출 페이지에서 신분증 사본(여권 업로드) 및 거주지 증빙 서류(문서 업로드)를 업로드합니다.

7. 확인을 위해 제출을 선택합니다.

그러면 KYC 규정 준수 상태가 검토됩니다(일반적으로 24시간 이내). 검토가 완료되면 이메일 메시지를 통해 알림을 받게 됩니다. 전체 KYC 프로세스는 일반적으로 약 2주가 소요됩니다.

6단계: 은행 계좌 확인 프로세스 완료

Amazon Payments Europe(APE)에서 대금을 받으려면 AWS Marketplace Management Portal의 결제 정보 탭에 나열된 지급 은행 계좌를 확인하기 위한 추가 정보를 제공해야 합니다.

추가 은행 정보 제공

추가 은행 정보를 제공하는 방법

1. 에 로그인 AWS Marketplace Management Portal한 다음 설정을 선택합니다.
2. 결제 정보 섹션에서 은행 정보 업데이트를 선택합니다.
3. 적절한 지급 계좌를 선택합니다.

   확인 상태가 확인되지 않음으로 표시됩니다.
4. 확인를 선택합니다.
5. 은행 거래 내역서를 업로드하고 제출할 수 있는 은행 계좌 확인 등록 포털로 리디렉션됩니다.

   Hyperwallet 가상 은행 계좌 솔루션을 사용하는 경우 the section called “Hyperwallet에서 은행 계좌 확인 명세서를 다운로드합니다.” 섹션을 참조하세요.
6. 포털에서 은행 문서 업로드를 선택하고 제출을 선택합니다.

Hyperwallet에서 은행 계좌 확인 명세서를 다운로드합니다.

Hyperwallet 가상 은행 계좌 솔루션을 사용하는 판매자는 다음 절차에 따라 Hyperwallet 은행 거래 명세서를 다운로드할 수 있습니다. 그 후 6단계: 은행 계좌 확인 프로세스 완료의 지침에 따라 은행 문서를 업로드할 수 있습니다.

Hyperwallet에서 은행 거래 내역서를 다운로드하는 방법

1. Hyperwallet 계좌에 로그인합니다.
2. 예금 계좌 정보 페이지로 이동합니다.
3. 은행 계좌 확인 명세서를 다운로드합니다.

이미 AWS Marketplace 판매자이신가요?

판매자로 등록한 후 다음 주제에서 AWS Marketplace 및 다음 단계에 대한 자세한 내용을 검토하세요.

- 지급 - AWS Marketplace 판매자는 지급 기본 설정을 지정하여 선택한 통화로 미결제 잔액을 받을 수 있습니다. 자세한 내용은 AWS Marketplace 판매자에 대한 지급 기본 설정 단원을 참조하십시오.
- 등록 수수료 - AWS Marketplace 판매자 등록 수수료에 대한 자세한 내용은 섹션을 참조하세요AWS Marketplace 판매자의 등록 수수료 이해.
- Amazon Payments Europe(APE) AWS Marketplace 에서에 대한 불만 처리 정책 - Amazon Payments Europe(APE)에서 제공하는 서비스에 문제가 있는 경우 AWS Marketplace알려주십시오. 여러분의 피드백은 여러분과 모든 AWS 구매자 및 판매자에게 보다 나은 경험을 제공하는 데 도움이 됩니다. 자세한 내용은 Amazon Payments Europe(APE)에 대한 AWS Marketplace 판매자 불만 제출 단원을 참조하십시오.
- 추가 판매자 도구 -는 고객 기반에 대한 더 많은 인사이트를 얻고 판매를 더 잘 이해하는 데 사용할 수 있는 추가 판매자 도구를 AWS Marketplace 제공합니다. 자세한 내용은 AWS Marketplace 판매자를 위한 추가 도구 단원을 참조하십시오.
- 제품 준비 - 계정을 판매자로 등록을 완료한 후 구매자에게 판매할 제품을 만들 수 있습니다 AWS Marketplace. 자세한 내용은 용 제품 준비 AWS Marketplace 단원을 참조하십시오.

AWS Marketplace 판매자에 대한 지급 기본 설정

AWS Marketplace 독립 소프트웨어 공급업체(ISVs) 및 채널 파트너를 포함한 판매자는 미결제 잔액을 받기 위해 지급 기본 설정을 지정할 수 있습니다. 판매자는 일일 또는 월별 지급 옵션을 선택할 수 있고, 지급을 받을 날짜를 선택하고, 은행 계좌에 통화를 지정할 수 있습니다. 은행 계좌에 여러 통화를 할당할 수 있습니다. 이 섹션에서는 지급 기본 설정을 지정하는 방법을 보여줍니다. 또한 AWS Marketplace 구독에 대한 결제 및에서 결제를 AWS 지급하는 방법에 대한 정보도 제공합니다.

> **Note**
>
> 미국 기반 ACH 계정 및 Hyperwallet 계정은 USD로만 결제를 받을 수 있습니다. USD가 아닌 지급의 경우 SWIFT 은행 계좌를 제공해야 합니다.

주제

- 지급 기본 설정 지정
- AWS Marketplace 구독 요금 청구

- 판매자가 지급을 받는 방법
- 사용 가능한 통화

지급 기본 설정 지정

지급 기본 설정을 지정하려면 다음 단계를 따르세요. 최근에 등록한 경우 퍼블릭 프로필을 생성하고 지급 방법을 추가하는 사이에 영업일 기준 2일을 기다려야 할 수 있습니다.

1. 에 로그인 AWS Marketplace Management Portal하고 설정을 선택합니다.
2. 결제 정보 탭을 선택합니다.
3. 지급 방법 섹션에서 지급 방법 추가를 선택합니다. 또는 지급 방법을 선택하고 편집을 선택하여 설정을 업데이트할 수 있습니다.
4. 통화에서 원하는 지급 통화를 선택합니다. 그런 다음 은행 계좌에서 선택한 통화로 결제를 받을 수 있는 계정을 선택합니다.
5. 일별 지급에서 월별 지급으로 변경하려면 월별을 선택하고, 지급을 처리할 날짜를 1~28 사이의 숫자로 선택합니다.
6. 지급 방법 추가를 선택합니다.

> **Note**
>
> 판매자는 세분화된 IAM 권한에 온보딩해야만 지급 기본 설정 옵션에 액세스할 수 있습니다. 세분화된 IAM 권한에 온보딩하려면 the section called “AWS Marketplace 판매자를 위한 정책 및 권한” 섹션을 참조하세요.

AWS Marketplace 구독 요금 청구

AWS 는 사용자를 대신하여 결제 메커니즘 역할을 합니다. 구매자가 이용할 수 있는 가장 일반적인 두 가지 결제 옵션은 신용카드와 인보이스 발행입니다.

에 대한 자세한 내용은 다음을 AWS Marketplace참조하세요.

- 선결제 구매는 구독 즉시 요금이 청구됩니다.
- 비공개 제안의 결제 일정은 구매자와 판매자가 합의합니다.
- 인보이스 결제 조건(결제 기한 포함)은 구매자와 AWS가 합의합니다. 약관은 공급업체에 공개되지 않습니다.

- 유연한 결제 스케줄러를 사용하는 비공개 제안은 결제 옵션으로 인보이스를 발행해야 합니다.
- 청구된 수익 대시보드를 사용하여 인보이스를 검증할 수 있습니다. 이 대시보드는 AWS 사용자를 대신하여의 인보이스를 요약하고 인보이스 생성 날짜와 인보이스 기한을 표시합니다.

판매자가 지급을 받는 방법

- 유효한 결제 방법, 등록된 은행 계좌 및 제출된 W9 양식이 있어야 지급이 이루어집니다.
- 유료 상품 판매자는 W-8 양식, 부가가치세(VAT) 또는 상품 및 서비스세(GST) 등록 번호 및 적격 관할권의 은행 계좌를 제공해야 합니다.
- AWS 는 다음과 같은 방법으로 결제를 지불합니다.
  - 일별 - 대금 지급이 가능한 시점에 일일 지급이 이루어집니다. 판매자는 대금을 받으려면 잔액이 양수여야 합니다.
  - 월별 - 판매자는 대금을 받을 날짜를 1일~28일 사이에서 선택합니다. 수집 및 지급 대시보드에는 지급 날짜가 표시됩니다.
- AWS 는 구매자가 인보이스를 지불한 후 ACH(Automated Clearing House) 이체 또는 SWIFT 이체를 사용하여 결제를 지불합니다.
- 고객에게 징수한 후에만 자금이 지출됩니다.
- 대금 지급일 이후 판매자의 은행 계좌에 입금될 때까지 약 1~2영업일이 소요됩니다. 정확한 시간은 은행과 표준 시간대에 따라 다릅니다.
- 지급 대시보드는 지급 후 AWS Marketplace Management Portal 3~5일 이내에 업데이트됩니다.
- 모든 계좌 미수금을 포함하여 지급된 대금과 비결제 대금에 대한 세부 정보는 지급 보고서에서 확인할 수 있습니다.

> **Note**
>
> 구매자가 비공개 제안에서 합의된 통화를 지원하지 않는 관할권으로 재배치하는 경우 지원되는 통화로 새 비공개 제안을 협상해야 합니다. 자세한 내용은 the section called “비공개 제안” 단원을 참조하세요.

사용 가능한 통화

판매자는 계약 요금으로 비공개 제안을 만들고 다음 통화로 지급받을 수 있습니다.

- 미국 달러(USD)
- 유로(EUR)
- 영국 파운드(GBP)
- 호주 달러(AUD)
- 일본 엔(JPY)

> **Note**
> 모든 공개 제안 및 소비 요금이 포함된 비공개 제안은 USD로만 생성할 수 있습니다.

AWS Marketplace 판매자의 등록 수수료 이해

AWS Marketplace 는 AWS Marketplace 판매자에게 표준 및 리전 등록 요금을 제공합니다. 등록 수수료는에서 발행한 등록 수수료 인보이스에 단일 품목으로 표시됩니다 AWS.

> **Note**
> 이 등록 수수료는 2024년 1월 5일 자정(UTC)부터 적용됩니다.

의 표준 제안 목록에 대한 요금 이해 AWS Marketplace

표준 공개 제안, 비공개 제안, 채널 파트너 비공개 제안(CPPOs) 및 전문 서비스에 대한 등록 수수료에 대해 알아봅니다.는 제품에 대해 다음과 같은 등록 수수료를 AWS Marketplace 제공합니다.

공개 제안 등록 수수료

소프트웨어 및 데이터 공개 제안에 대한 등록 수수료는 배포 방법에 따라 결정됩니다.

- 서비스형 소프트웨어(SaaS) - 3%
- 서버(Amazon Machine Image(AMI), 컨테이너, 머신 러닝) - 20%
- AWS Data Exchange - 3%

비공개 제안 등록 수수료

비공개 제안에 대한 등록 수수료는 총 계약 금액 및 비공개 제안이 이전 비공개 제안에서 갱신되는지 아니면 AWS Marketplace이외의 이전 계약에서 갱신되는지에 따라 결정됩니다.

- 1백만 달러 미만 - 3%
- 1백만 달러에서 1천만 달러 미만 - 2%
- 1천만 달러 이상 - 1.5%
- 모든 갱신 - 1.5%

채널 파트너 비공개 제안(CPPO) 등록 수수료

CPPO 제품은 오퍼 유형이나 배포 방법에 관계없이 등록 수수료가 0.5% 인상됩니다. 예를 들어, 상품이 총 계약 금액이 1백만 달러 미만인 SaaS 비공개 제안인 경우 등록 수수료는 3.5%입니다.

전문 서비스 등록 수수료

모든 전문 서비스 제품에는 비공개 제안에 2.5%의 리스팅 수수료가 부과됩니다.

의 리전 제안 목록에 대한 요금 이해 AWS Marketplace

다음 관할 구역(들)의 구매자와의 거래에 대한 표준 등록 요금에는 추가 리전 등록 요금이 적용됩니다.

> **Note**
>
> 리전 등록 수수료는 표준 등록 수수료와 함께 추가됩니다. 예를 들어, 제품이 한국의 구매자에게 판매된 총 계약 금액이 1백만 달러 미만인 SaaS 비공개 제안인 경우 등록 수수료는 4%(3% 표준 비공개 제안 등록 수수료 + 1% 리전 등록 수수료)입니다.

리전	추가 리전 등록 요금	발효일
대한민국	1%	04/01/2025

Amazon Payments Europe(APE)에 대한 AWS Marketplace 판매자 불만 제출

AWS Marketplace 판매자는 Amazon Payments Europe S.C.A(APE)에서 제공하는 서비스에 문제가 있는 경우 불만을 제출할 수 있습니다. Amazon Payments Europe(APE)에서 제공하는 서비스와

관련하여 문제가 있으면 AWS에 알려주세요. 여러분의 피드백은 여러분과 모든 AWS 구매자 및 판매자에게 보다 나은 경험을 제공하는 데 도움이 됩니다. 다음 섹션에서는 불만 정책에 따라 Amazon Payments Europe(APE)과 관련된 불만을 제출하는 데 필요한 특정 단계의 세부 정보를 제공합니다. 또한 이 주제에서는 Amazon Payments Europe(APE)에 대한 해결 기간 및 불만 에스컬레이션 등 불만 정책에 대한 정보를 제공합니다.

> **Note**
>
> 오직 AWS Marketplace와 관련된 불만 사항만 다음 절차를 통해 해결됩니다. Amazon Payments Europe S.C.A.에서 제공하는 서비스는 결제 거래 처리, 요금 부과에서 발생할 수 있는 오류 확인, 대금 지급 등입니다.

불만 사항 제출

Amazon Payments Europe S.C.A.를 이용하는 AWS Marketplace 계정이 있는 경우 불만 사항은 Amazon Payments Europe S.C.A.에서 처리합니다.

불만 사항을 제출하는 방법

1. AWS Marketplace 판매자 계정에 로그인합니다.
2. AWS에 문의로 이동합니다.
3. 상업용 마켓플레이스, 판매자 계정, 등록을 선택합니다.
4. 불만 사항에 대한 세부 정보를 입력하고 제출을 선택합니다.

Amazon Payments Europe 불만 사항 해결 기간

Amazon Payments Europe S.C.A.(APE)는 불만 사항을 접수한 날로부터 15영업일 이내에 불만 사항에 대한 업데이트를 제공합니다. APE의 통제를 벗어나는 예외적인 상황에서는 APE가 불만 사항을 최초로 접수한 날로부터 최대 35영업일까지 불만 사항 해결 기간이 연장될 수 있습니다.

불만 사항 에스컬레이션

대응이 만족스럽지 않으면 다음 연락처로 연락하여 불만 사항을 에스컬레이션할 수 있습니다.

- Amazon Payments Europe 고위 경영진

고위 경영진에게 <ape-management@amazon.lu>으로 이메일 메시지를 보내 불만 사항을 제출하세요. AWS는 귀하의 의견을 신중하게 검토하고, 고위 경영진이 귀하의 불만 사항을 접수한 날로부터 15영업일 이내에 응답합니다. Amazon Payment Europe의 통제를 벗어나는 예외적인 상황에서는 고위 경영진이 불만 사항을 최초로 접수한 날로부터 최대 35영업일까지 불만 사항 해결 기간이 연장될 수 있습니다.

- Commission de Surveillance du Secteur Financier (CSSF)

  CSSF는 룩셈부르크 금융 부문의 기업을 감독하는 기관입니다. 110 Route d'Arlon L-2991 Luxembourg로 문의하거나 https://www.cssf.lu/contacts/의 연락처 페이지를 사용할 수 있습니다. CSSF에 관한 추가 정보와 연락 방법은 CSSF 웹 사이트의 고객 불만을 참조하세요.
- 온라인 분쟁 해결

  EU에서 온라인으로 계정을 개설한 경우 온라인 분쟁 해결 플랫폼을 사용하여 CSSF에 불만 사항을 접수할 수도 있습니다. Amazon Payments Europe S.C.A.에서 금융 서비스를 제공하고 CSSF는 라이선스를 담당하는 기관이기 때문에 이 옵션을 사용할 수 있습니다. 자세한 내용은 유럽 위원회 웹 사이트의 온라인 분쟁 해결 플랫폼을 참조하세요.

AWS Marketplace 판매자를 위한 추가 도구

AWS Marketplace 는 제품을 판매하고 관리하는 데 사용할 수 있는 판매자 도구를 제공합니다. 이러한 도구를 사용하면 고객 기반에 대한 인사이트를 얻고 판매를 더 잘 이해하는 데 도움이 됩니다. 이 주제에서는 추가 리소스에 대한 링크와 함께 이러한 도구에 대한 정보를 제공합니다.

AWS Marketplace Management Portal는 제품을 판매하기 위한 기본 도구입니다 AWS Marketplace. AWS Marketplace Management Portal에서 제공하는 시장 진출 활동을 활용해 제품을 관리하여 점진적인 채널 수익을 창출할 수 있습니다. 자세한 내용은 AWS Marketplace Management Portal 단원을 참조하십시오.

AWS Marketplace 는 다음과 같은 추가 판매자 도구를 제공합니다.

- 추가 리소스 AWS Marketplace Management Portal- AWS Marketplace Management Portal를 열고 로그인하면 홈 페이지의 Marketplace 리소스 섹션에서 추가 리소스에 대한 링크를 볼 수 있습니다. 예를 들어 출시 전 90일 이내와 출시 후 90일 이내에 제품 마케팅에 대한 지원을 받으려면 AWS Marketplace Management Portal 홈페이지의 Marketplace 리소스에서 연결된 180일 GTM 아카데미를 확인할 수 있습니다.
- AWS Marketplace 상거래 분석 서비스 AWS Marketplace - 상거래 분석 서비스를 사용하면를 통해 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다 AWS Marketplace. 서비스에

등록한 후 AWS SDK를 통해 사용량, 구독 및 결제 보고서에 액세스할 수 있습니다. 자세한 내용은 AWS Marketplace Commerce Analytics Service를 사용하면 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다. 단원을 참조하십시오.

- AWS Marketplace 필드 데모 프로그램(FDP) - FDP는 AWS 필드 팀(내부에서 승인된 AWS 직원)이 AWS Marketplace 를 통해 일부 제품 및 솔루션을 무료로 사용할 수 있도록 허용합니다. 자세한 내용은 AWS Marketplace 필드 데모 프로그램 단원을 참조하십시오.
- 판매자 보고서, 데이터 피드 및 대시보드 - AWS Marketplace 는 제품 판매에 대한 정보를 수집하고 분석하는 도구를 제공합니다. 자세한 내용은 의 판매자 보고서, 데이터 피드 및 대시보드 AWS Marketplace 단원을 참조하세요.

자세한 내용은 다음 항목을 참조하세요.

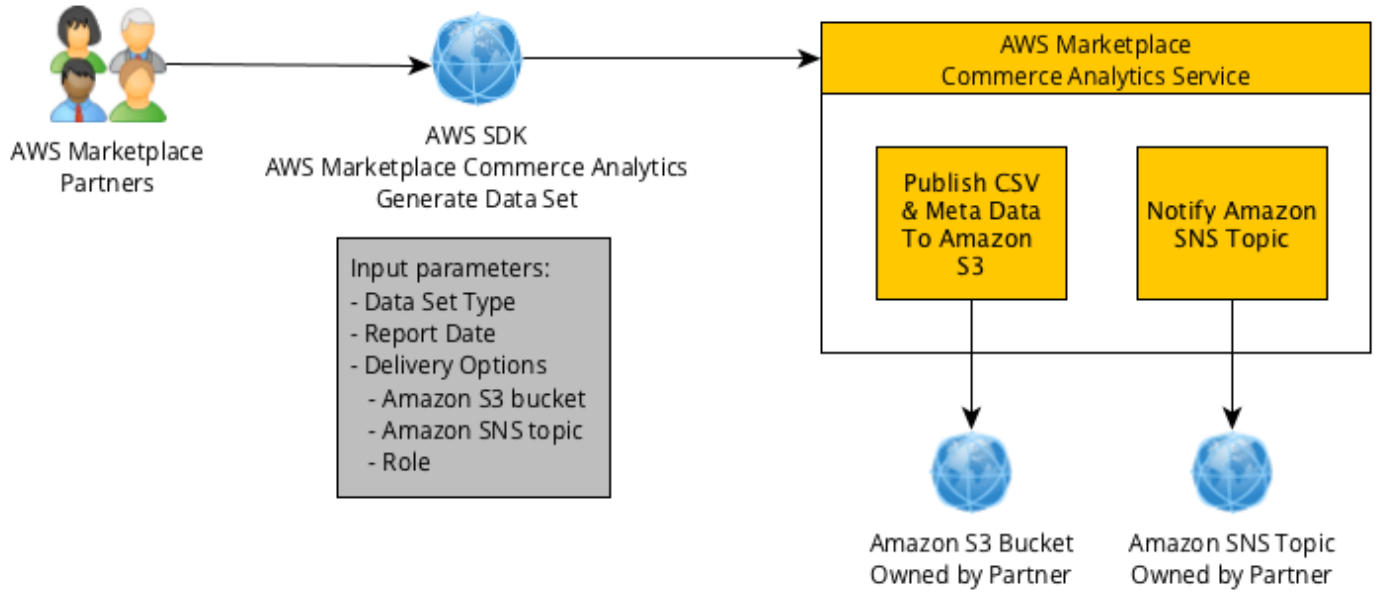
주제

- AWS Marketplace Commerce Analytics Service를 사용하면 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다.
- AWS Marketplace 필드 데모 프로그램
- AWS 파트너 어시스턴트

AWS Marketplace Commerce Analytics Service를 사용하면 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다.

상거래 분석 서비스를 사용하면 AWS Marketplace 를 통해 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다 AWS Marketplace. 서비스에 등록한 후 AWS SDKs. SDK 도구를 사용하여 요청한 데이터는 AWS 계정 에 데이터 세트로 제공됩니다. 대부분의 데이터 세트는 AWS Marketplace Management Portal에서 사용할 수 있는 텍스트 기반 보고서와 동일한 형태의 데이터에 해당합니다. 특정 날짜에 대한 데이터 세트를 요청할 수 있으며 데이터는 제공된 Amazon S3 버킷으로 전달됩니다. Amazon Simple Notification Service(Amazon SNS)를 통해 데이터 제공 알림을 받습니다. 이 주제에서는 AWS Marketplace 상거래 분석 서비스 사용에 대한 이용 약관을 제공합니다.

다음 시각화는 Commerce Analytics Service가 AWS Marketplace 에서 제품 및 고객 데이터에 액세스하고 이를 데이터 세트로 Amazon S3 버킷에 전달하여 Amazon SNS를 통해 알림을 시작하는 방법을 보여줍니다.



이용 약관

이 AWS Marketplace 상거래 분석 서비스 이용 약관(이 'CAS 약관')에는 AWS Marketplace 상거래 분석 서비스('CA 서비스')의 사용 및 액세스와 관련된 이용 약관이 포함되어 있으며,이 이용 약관과 함께 제공된 '수락함' 버튼 또는 확인란을 클릭하거나 이전에는 CA 서비스 제품을 사용할 때부터 유효합니다. 본 CAS 약관은 귀하와 Amazon Web Services, Inc. 간의 AWS Marketplace 판매자 이용 약관(“AWS Marketplace 판매자 약관”)의 부록입니다. ('AWS', '저희', '저희' 또는 '저희의')이며, 그 약관은 여기에 통합되어 있습니다. 본 CAS 약관과 AWS Marketplace 판매자 약관이 상충되는 경우 본 CAS 약관의 조건이 적용되지만 그렇게 상충되는 범위에만 그리고 CA 서비스 사용과 관련하여 전적으로 적용됩니다. 여기에 사용되었지만 정의되지 않은 대문자로 된 용어는 AWS Marketplace 판매자 약관에 명시된 의미를 갖습니다.

1. CA 서비스 및 CAS 데이터. CA 서비스에 액세스할 수 있는 자격을 갖추려면 기존 AWS Marketplace 판매자 약관의 적용을 받는 AWS Marketplace 판매자여야 합니다. CA 서비스와 관련하여 수신하거나 액세스할 수 있는 정보 및 데이터(“CAS 데이터”)는 구독자 정보를 구성하며 AWS Marketplace 판매자 약관에 명시된 제한 및 의무의 적용을 받습니다. (a) CAS 데이터를 제3자에게 공개하지 않는 한, CAS 데이터를 비공개로 사용하여 AWS Marketplace 콘텐츠와 관련된 마케팅 및 기타 홍보 활동을 개선하고 대상으로 지정할 수 있습니다. (b) 관련 개인 정보 보호 정책 또는 법률과 일치하지 않는 방식으로 CAS 데이터를 사용하는 행위 (c) 구독자에게 연락하여 외부에서 대체 구매를 하도록 영향을 미칩니다 AWS Marketplace. (d) 당사를 비난합니다. 제휴사, 또는 해당 제품 또는 (e) AWS Marketplace 의도된 수신자를 구독자로 하여 모든 종류의 통신을 대상으로 합니다.

2. CA 서비스 제한 및 보안. CA 서비스 문서에 설명된 방법으로만 CA 서비스를 액세스하거나 액세스를 시도할 수 있습니다. CA 서비스를 사용할 때 귀하의 자격 증명이나 클라이언트의 자격 증명을 허위 진술하거나 숨길 수 없습니다. 당사는 재량에 따라 일정 기간 동안 CA 서비스를 액세스하도록 허용된 호출 또는 서버 수를 포함하되 이에 국한되지 않고 CA 서비스 사용에 대한 제한을 설정하고 시행할 수 있는 권리를 보유합니다. 귀하는 그러한 제한에 동의하고 회피를 시도하지 않습니다. 당사는 귀하가 이러한 CAS 약관을 위반하거나 CA 서비스를 오용한다고 판단되는 경우 귀하의 CA 서비스 액세스 권한을 제한, 중지 또는 종료할 수 있습니다.
3. CA 서비스 자격 증명 비밀 유지 및 보안. CA 서비스 자격 증명(예: 암호, 키 또는 클라이언트 ID)은 API 클라이언트를 식별할 목적으로 사용됩니다. 자격 증명의 기밀을 유지하는 것은 전적으로 귀하의 책임이며 유사한 성격의 기밀 정보를 보호하기 위해 취해진 조치를 포함하여 해당 자격 증명의 공개, 전파 또는 무단 사용을 피하기 위해 합당한 모든 조치가 취해집니다. CA 서비스 자격 증명은 오픈 소스 프로젝트에 포함되지 않을 수 있습니다. 자격 증명을 사용한 모든 CA 서비스 액세스는 전적으로 귀하의 책임입니다.
4. 수정. 당사는 AWS 사이트에 수정된 버전을 게시하거나 AWS Marketplace 판매자 약관에 따라 사용자에게 알림을 제공하여 언제든지 이러한 CAS 약관을 수정할 수 있습니다. 수정된 약관은 게시 또는 이메일 메시지에 명시된 대로 귀하에게 이메일로 통보하는 시점에서 효력을 발휘합니다. 본 CAS 약관을 수정한 효력일 이후에도 CA 서비스를 계속 사용하거나 액세스하면 수정된 약관을 준수할 것에 동의하는 것입니다.
5. 해지. 본 CAS 약관 및 여기에서 허용된 CAS 데이터 사용 권한은 어떠한 이유로 AWS Marketplace 판매자 약관이 해지되는 경우 통보 여부에 관계 없이 종료됩니다. 또한 언제든 이유에 관계 없이 당사는 CA 서비스 제공을 중지하거나 CA 서비스 액세스를 종료할 수 있습니다.

시작

온보딩, 기술 구현 및 문제 해결 정보를 포함하여 AWS Marketplace 상거래 분석 서비스에 대한 자세한 내용은 다음 주제를 참조하세요.

주제

- AWS Marketplace 상거래 분석 서비스에 온보딩
- AWS CLI 및 AWS Marketplace 에서 상거래 분석 서비스 사용 AWS SDK for Java
- AWS Marketplace 상거래 분석 서비스를 사용하여 데이터 세트 생성
- AWS Marketplace 상거래 분석 서비스 문제 해결

AWS Marketplace 상거래 분석 서비스에 온보딩

AWS Marketplace 상거래 분석 서비스를 사용하면를 통해 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다 AWS Marketplace. AWS Marketplace 상거래 분석 서비스를 시작하려면 상거래 분석 서비스를 AWS 서비스 사용하도록 AWS 계정 및 AWS Marketplace 를 구성해야 합니다. 이 섹션에서는 상거래 분석 서비스를 AWS 서비스 사용하도록 AWS 계정 및 AWS Marketplace 를 구성하는 방법을 보여줍니다.

AWS Marketplace 상거래 분석 서비스를 사용하려면

- 1단계: 권한을 AWS 계정 사용하여 설정
- 2단계: 대상 Amazon S3 버킷 만들기
- 3단계: 응답 알림에 대한 Amazon SNS 주제 구성
- 4단계: Commerce Analytics Service 프로그램 등록
- 5단계: 구성 확인

1단계: 권한을 AWS 계정 사용하여 설정

에서는 루트 계정 자격 증명을 사용하는 AWS Marketplace Management Portal 대신 AWS Identity and Access Management (IAM) 역할을 사용하여에 로그인할 것을AWS Marketplace 강력히 권장합니다. AWS Marketplace 상거래 분석 서비스 권한의 특정 IAM 권한은 the section called “AWS Marketplace 판매자를 위한 정책 및 권한” 섹션을 참조하세요. 계정에 액세스하는 사용자에 대해 개별 사용자를 만들면 각 사용자에 따라 서로 다른 보안 자격 증명 조합을 부여할 수 있습니다. 또한 각 사용자에게 다양한 권한을 부여하고, 필요할 경우 언제든지 사용자의 권한을 변경 또는 취소할 수 있습니다.

2단계: 대상 Amazon S3 버킷 만들기

상거래 분석 서비스는 판매자가 요청한 데이터를 판매자가 지정한 Amazon S3 버킷에 제공합니다. 사용할 Amazon S3 버킷이 이미 있는 경우 다음 단계를 진행합니다.

Amazon S3 버킷이 없거나이 데이터 전용 Amazon S3 버킷을 생성하려면 Amazon S3 버킷을 생성하려면 어떻게 해야 하나요?를 참조하세요.

3단계: 응답 알림에 대한 Amazon SNS 주제 구성

상거래 분석 서비스는 Amazon SNS를 사용하여 응답 알림을 제공합니다. 서비스는 이 주제에 메시지를 게시하여 데이터 세트가 이용 가능하거나 오류가 발생할 때 알림을 보냅니다. 이 목적을 위한 Amazon SNS 주제가 이미 있는 경우 다음 단계를 진행합니다.

이 서비스에 대한 Amazon SNS 주제를 구성하지 않은 경우 지금 구성합니다. 지침은 주제 생성을 참조하세요.

생성한 주제의 주제 Amazon 리소스 이름(ARN)을 기록해 둡니다. 서비스를 호출할 때 ARN이 필요합니다.

4단계: Commerce Analytics Service 프로그램 등록

판매자가 주제 및 버킷 이름에 ARN을 사용하여 서비스를 구성하면 상거래 분석 서비스는 Amazon S3 버킷 및 Amazon SNS 주제에 액세스합니다.

액세스를 활성화하려면

1. AWS Marketplace 제품을 관리하는 데 AWS 계정 사용하는 AWS Marketplace Management Portal로에 로그인합니다.
2. AWS Marketplace 상거래 분석 서비스에 등록하는 데 필요한 IAM 권한이 있는지 확인합니다.
3. 상거래 분석 서비스 등록 페이지로 이동합니다.
4. Amazon S3 버킷 이름과 Amazon SNS 주제 ARN을 입력하고 등록을 선택합니다.
5. 권한 페이지에서 허용을 선택합니다.
6. 에서 성공 메시지에 역할 이름 ARN을 AWS Marketplace Management Portal기록합니다. 서비스 호출 시 ARN이 필요합니다.

> **Note**
>
> 상거래 분석 서비스에 온보딩하면에 IAM 역할이 생성됩니다 AWS 계정. IAM 역할을 사용하면 가 Amazon S3 버킷에 AWS Marketplace 쓰고 Amazon SNS 주제에 알림을 게시할 수 있습니다.는 계정 452565589796을 AWS Marketplace 사용하여이 IAM 역할과 관련된 작업을 수행합니다.

5단계: 구성 확인

마지막 단계는 구성이 예상대로 작동하는지 확인하는 것입니다.

구성을 테스트하려면

1. AWS 명령줄 인터페이스(AWS CLI)를 다운로드, 설치 및 구성합니다.
2. 를 사용하여이 명령을 AWS CLI실행합니다.

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

- `--data-set-publication-date`에서 `{TODAY'S DATE}`를 ISO-8601 형식의 현재 날짜(`YYYY-MM-DDT00:00:00Z`)로 바꿉니다. 여기서 `YYYY`는 4자리 연도, `MM`은 2자리 월, `DD`는 2자리 일입니다.
- `--role-name-arn`의 경우 `{YOUR-ROLE-NAME-ARN}`을 4단계: Commerce Analytics Service 프로그램 등록의 등록 프로세스에서 받은 역할의 ARN으로 바꿉니다.
- --destination-s3-bucket-name의 경우 {amzn-s3-demo-bucket}을 2단계: 대상 Amazon S3 버킷 만들기에서 생성한 Amazon S3 버킷의 이름으로 바꿉니다.
- –sns-topic-arn의 경우 {YOUR-SNS-TOPIC-ARN}을 3단계: 응답 알림에 대한 Amazon SNS 주제 구성에서 생성한 Amazon SNS 주제로 바꿉니다.

dataSetRequestId가 포함된 응답을 받으면 온보딩 프로세스가 완료된 것입니다. 성공적인 응답은 다음과 같습니다.

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

AWS CLI 및 AWS Marketplace 에서 상거래 분석 서비스 사용 AWS SDK for Java

AWS Marketplace 상거래 분석 서비스를 사용하면를 통해 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다 AWS Marketplace. AWS Marketplace 상거래 분석 서비스는 AWS SDK를 통해 제공됩니다. AWS CLI 및 AWS SDK for Java를 사용하여 Commerce Analytics Service와 상호 작용합니다. 이 섹션에서는 Java용 AWS CLI 및 SDK를 사용하여 상거래 분석 서비스를 구현하는 방법을 보여줍니다.

주제

- 상거래 분석 서비스에 대한 IAM 정책
- 를 사용하여 요청 AWS CLI
- AWS SDK for Java를 사용하여 요청 생성

상거래 분석 서비스에 대한 IAM 정책

사용자가 상거래 분석 서비스를 사용할 수 있게 하려면 다음 권한이 필요합니다.

다음 IAM 권한 정책을 사용하여 AWS Marketplace 상거래 분석 서비스에 등록합니다.

JSON

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

다음 IAM 권한 정책을 사용하여 사용자가 AWS Marketplace 상거래 분석 서비스에 대한 요청을 생성할 수 있도록 허용합니다.

JSON

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

```
            "Action": "marketplacecommerceanalytics:GenerateDataSet",
            "Resource": "*"
        }
    ]
}
```

자세한 내용은 IAM 사용 설명서의 IAM 콘솔에서 정책 생성을 참조하세요.

를 사용하여 요청 AWS CLI

시작하려면 AWS CLI를 다운로드합니다. 다음 AWS CLI 예제에서는 2017년 10월 1일에 대한 시간별/월별 구독 데이터 세트를 요청합니다. 이 데이터 세트는 demo-prefix 접두사를 사용하여 Amazon S3 버킷 demo-bucket에 게시되며, 알림 메시지는 Amazon SNS 주제 demo-topic에 전달됩니다.

```
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"
```

이 요청은 각 요청에 대해 고유한 식별자를 반환합니다. 이 식별자를 사용하여 요청을 Amazon SNS 주제에 게시된 알림과 연결할 수 있습니다. 다음은 이 식별자의 예제입니다.

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

AWS SDK for Java를 사용하여 요청 생성

시작하려면 AWS Java SDK를 다운로드합니다. 다음 AWS SDK for Java 예제에서는 2015년 10월 1일에 시간별/월별 구독 데이터 세트를 요청합니다. 이 데이터 세트는 demo-prefix 접두사를 사용하여 Amazon S3 버킷 demo-bucket에 게시되며, 알림 메시지는 Amazon SNS 주제 demo-topic에 전달됩니다.

```
/*
* 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);
}
}
```

이 예에서는 다음과 비슷한 결과가 나올 것입니다.

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

AWS Marketplace 상거래 분석 서비스를 사용하여 데이터 세트 생성

상거래 분석 서비스에 등록한 후 AWS CLI 및 AWS Marketplace 를 통해 사용량, 구독 및 결제 보고서에 액세스할 수 있습니다 AWS SDK for Java. SDK 도구를 사용하여 요청하는 데이터는 데이터 세트 AWS 계정 로에 전달됩니다. 대부분의 데이터 세트는 AWS Marketplace Management Portal에서 사용할 수 있는 텍스트 기반 보고서와 동일한 형태의 데이터에 해당합니다. 자세한 내용은 AWS Marketplace Commerce Analytics Service를 사용하면 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다. 단원을 참조하세요. 이 섹션에서는 Commerce Analytics Service에서 생성한 데이터세트와 관련된 파라미터, 응답 및 출력에 대한 정보를 제공합니다.

주제

- GenerateDataSet 파라미터
- 응답
- 결과

`GenerateDataSet` 파라미터

AWS Marketplace 상거래 분석 서비스는 Amazon S3 버킷에 데이터 세트를 게시하도록 요청하는 데 사용할 수 `GenerateDataSet`있는 하나의 메서드인를 제공합니다. 다음 표는 `GenerateDataSet`에 대한 파라미터를 나열합니다.

데이터 세트 파라미터

필드	설명
Data Set Type	이 데이터 세트는 요청의 결과로 반환됩니다.
Data Set Publication Date	데이터 세트가 게시된 날짜입니다. 일별 데이터 세트의 경우 원하는 날의 일 수준으로 세분화된 날짜를 입력합니다. 월별 데이터 세트의 경우 원하는 달의 월 수준으로 세분화된 날짜를 입력합니다. 일 값은 무시됩니다.
Role Name ARN	연결된 권한 정책이 포함된 역할의 ARN은 리소스에 대한 액세스를 포함한 서비스를 제공합니다.

필드	설명
Destination Amazon S3 Bucket Name	대상 Amazon S3 버킷의 이름(ARN이 아닌 익숙한 이름)입니다. 데이터 세트가 이 위치에 게시됩니다.
Destination Amazon S3 Prefix	(선택 사항) 게시된 데이터 세트의 Amazon S3 접두사이며 표준 파일 시스템의 디렉터리 경로와 유사합니다. 예를 들어 해당 버킷 이름이 `mybucket`이고 접두사가 `myprefix/mydatasets` 인 경우 출력 파일은 `s3://amzn-s3-demo-bucket/myprefix/mydatasets/outputfile` 에 게시됩니다. 접두사 구조 디렉터리가 존재하지 않으면 새로 생성됩니다. 접두사가 제공되지 않으면 데이터 세트가 Amazon S3 버킷 루트에 게시됩니다.
SNS 주제 ARN	데이터 세트가 게시되거나 오류가 발생할 때 알림을 받는 Amazon SNS 주제의 ARN입니다.

응답

AWS Marketplace 상거래 분석 서비스는 두 개의 응답을 반환합니다. 첫 번째는 즉시 반환되는 동기식이고, 두 번째는 Amazon SNS를 사용하여 반환되는 비동기식입니다. 동기식 응답은 다음 예제와 유사합니다.

데이터 세트 파라미터

필드	설명
Data Set Request ID	서비스에 대한 특정 요청을 나타내는 고유 식별자입니다. 이 식별자를 사용하여 요청에 Amazon SNS 주제에 대한 알림을 연결할 수 있습니다.

비동기식 응답은 Amazon SNS 주제에 JSON 형식의 문서로 게시되며 다음 예제와 유사합니다.

데이터 세트 파라미터

필드	설명
Data Set S3 Location	전달된 데이터 세트의 버킷 이름 및 키입니다.
Data Set Meta Data S3 Location	전달된 데이터 세트 메타데이터 파일의 버킷 이름 및 키입니다.
Data Set Request ID	서비스에 대한 특정 요청을 나타내는 고유 식별자입니다. 이 식별자를 사용하여 요청에 Amazon SNS 주제에 대한 알림을 연결할 수 있습니다.
Success	작업이 성공하면 "true"이고, 성공하지 않으면 "false”입니다.
메시지	(선택 사항) 오류가 발생한 경우(예: “Success”가 “false”인 경우) 이 메시지에는 결함에 대한 정보가 포함됩니다.

JSON 형식 비동기식 응답 예제:

```
 {
   "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
 }
```

결과

성공적인 요청 이후 요청한 데이터 세트가 .csv 파일로 Amazon S3 버킷에 제공됩니다. JSON 형식 메타데이터 파일이 데이터 세트 파일과 동일한 위치에 게시됩니다. 메타데이터 파일은 데이터 세트 및 원래 요청 파라미터에 관한 유용한 정보를 제공합니다. 메타데이터 파일은 데이터 세트 파일과 동일한 이름이 있지만 .meta.json 확장자로 끝납니다. 다음 표에 .csv 파일의 메타데이터 필드가 나열되어 있습니다.

메타데이터 필드

필드	설명
Data Set Request ID	서비스에 대한 특정 요청을 나타내는 고유 식별자입니다. 이 식별자를 사용하여 요청에 Amazon SNS 주제에 대한 알림을 연결할 수 있습니다.
Data Set Coverage Range	데이터 범위에 대한 시작 날짜/시간 및 종료 날짜/시간을 정의합니다. 이 날짜는 ISO 8601 형식입니다.
Data Set Request Parameters	`GenerateDataSet` 메서드에 대한 원래 요청 파라미터입니다.
Data Set S3 Location	전달된 데이터 세트의 버킷 이름 및 키입니다.
Data Set Meta Data S3 Location	전달된 데이터 세트 메타데이터 파일의 버킷 이름 및 키입니다.

다음은 JSON 형식 메타데이터 콘텐츠의 예입니다.

```
{
"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"
}
}
```

가용성 날짜를 포함하여 사용 가능한 데이터 세트의 전체 목록은 AWS SDK 설명서를 참조하세요.

AWS Marketplace 상거래 분석 서비스 문제 해결

제품 및 고객 데이터를 프로그래밍 방식으로 제공하는 AWS Marketplace 상거래 분석 서비스의 문제를 해결할 수 있습니다 AWS Marketplace. 오류 또는 기타 구성 문제가 발생하는 경우 Commerce Analytics Service 문제를 해결해야 할 수 있습니다. 다음 섹션에서는 Commerce Analytics Service의 일반적인 문제를 진단 및 해결하는 단계를 다루는 문제 해결 프로세스를 안내합니다.

허용 목록 문제로 인해 서비스에 액세스 할 수 없습니다.

에 아직 판매자로 등록하지 않은 경우 AWS Marketplace를 방문하여 등록AWS Marketplace Management Portal합니다. 판매자로 이미 등록한 경우 AWS Marketplace 판매자 운영 팀에 AWS Marketplace문의하세요.

과거 날짜에 있는 데이터 세트를 요청할 수 없습니다. 그런데 SDK 설명서에서는 이러한 날짜에 대해 이용할 수 있다고 나와 있습니다.

데이터 세트가 과거의 특정 날짜에 대해 이용 가능한 것으로 나열되어도 AWS Marketplace에 가입한 이후의 데이터만 보유합니다. 오류라고 생각되면 AWS Marketplace 판매자 작업 팀에 문의하세요.

서비스를 호출할 때 다음 오류 메시지를 받았습니다. “엔드포인트 URL: https://marketplacecommerceanalytics.eu-central-1.amazonaws.com/에 연결할 수 없음”

AWS Marketplace 상거래 분석 서비스는 미국 동부(버지니아 북부) 리전에서만 사용할 수 있습니다. 상거래 분석 서비스에 대한 모든 호출은 `us-east-1` 엔드포인트에서 이루어져야 합니다.

를 사용하는 경우 다음 예제와 `us-east-1`같이 각 호출에 `"--region` 플래그를 AWS CLI추가하고를 AWS 리전 로 지정합니다.

```
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
```

온보딩 프로세스를 거쳤을 때 선택한 것과 다른 Amazon S3 버킷 또는 Amazon SNS 주제를 사용하고자 합니다.

AWS Marketplace 상거래 분석 서비스에 등록할 때 Amazon S3 버킷 및 Amazon SNS 주제를 지정했습니다. 온보딩 프로세스는 이 서비스가 특정 리소스에만 액세스할 수 있도록 IAM 권한을 구성합니다. 다른 리소스를 사용하려면 IAM 정책을 수정해야 합니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. IAM 콘솔의 왼쪽에서 역할을 선택합니다.
3. MarketplaceCommerceAnalyticsRole을 선택합니다.
4. 이미 펼쳐지지 않은 경우 Inline Roles(인라인 역할) 섹션을 펼칩니다.
5. 이름이 oneClick_MarketplaceCommerceAnalyticsRole로 시작하는 정책을 찾아 정책 편집을 선택합니다.
6. 이 정책 문서에서 수정하고자 하는 서비스와 관련이 있는 작업을 지정하는 섹션을 찾습니다. 예를 들어 Amazon S3 버킷을 변경하려면 s3:으로 시작하는 작업이 포함된 섹션을 찾은 다음, 해당 리소스 선택 내용을 변경하여 새 Amazon S3 버킷을 지정합니다.

IAM 정책에 대한 자세한 내용은 다음 가이드를 참조하세요. https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

`GenerateDataSet` 작업을 호출할 때 `AccessDeniedException` 오류가 발생하는 경우

이 문제는 사용자가 `GenerateDataSet`를 호출하는 데 필요한 권한이 없을 때 발생할 수 있습니다. 다음 절차에서는 IAM 콘솔을 사용하여 이러한 권한이 있는 IAM 정책을 생성하고 사용자, 그룹 또는 역할에 이 권한을 추가하는 단계를 설명합니다.

JSON 정책 편집기를 사용하여 정책을 생성하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 정책을 선택합니다.

   정책을 처음으로 선택하는 경우 관리형 정책 소개 페이지가 나타납니다. 시작을 선택합니다.
3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. 다음 JSON 정책 문서를 입력합니다.

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

6. 다음을 선택합니다.

> **Note**
>
> 언제든지 시각적 편집기 옵션과 JSON 편집기 옵션 간에 전환할 수 있습니다. 그러나 변경을 적용하거나 시각적 편집기에서 다음을 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 정책 재구성을 참조하세요.

7. 검토 및 생성 페이지에서 생성하는 정책에 대한 정책 이름과 설명(선택 사항)을 입력합니다. 이 정책에 정의된 권한을 검토하여 정책이 부여한 권한을 확인합니다.
8. 정책 생성을 선택하고 새로운 정책을 저장합니다.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.
- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

문제가 이곳에 나열되어 있지 않습니다.

AWS Marketplace 판매자 작업 팀에 문의하세요.

AWS Marketplace 필드 데모 프로그램

AWS Marketplace 필드 데모 프로그램(FDP)을 사용하면 AWS 필드 팀(내부에서 승인된 AWS 직원)이 AWS Marketplace 를 통해 일부 제품과 솔루션을 무료로 사용할 수 있습니다.

승인된 AWS 직원의 예로는 솔루션 아키텍트와 영업 및 마케팅 전문가가 포함될 수 있습니다. FDP를 통해 이러한 직원들은 교육을 위해 고객 워크로드에 포함시킬 제품 기능을 시연할 수 있습니다.

다음 제품 유형이 지원됩니다.

- Amazon 머신 이미지(AMI)
- 컨테이너
- 기계 학습 알고리즘 및 모델 패키지(SageMaker AI)
- 데이터 세트(AWS Data Exchange)

> **Note**
>
> AWS Data Exchange 제품의 경우 공개 제안이 0 USD(무료)인 제품에만 FDP가 적용됩니다. 구독 확인이 활성화된 AWS Data Exchange 제품의 경우 공급자가 구독 요청을 승인해야 합니다. 구독 확인에 대한 자세한 내용은 AWS Data Exchange 사용 설명서의 구독자의 구독 확인을 참조하세요.

AWS Marketplace 판매자로 가입하면 FDP 프로그램에 자동으로 등록됩니다. 옵트아웃하려면 AWS Marketplace 판매자 운영 팀에 지원 요청을 제출하세요.

이 프로그램에 따른 제품 사용에 대해 알아보려면 월별 청구 수익 보고서의 AWS 필드 데모 사용량 섹션을 참조하세요.

AWS 파트너 어시스턴트

AWS Partner Assistant는 AWS Partner를 위한 생성형 AI 기반 챗봇입니다. Partner Central 및 AWS Marketplace Management Portal (AMMP) 모두에서 액세스할 수 있으며 등록된 파트너 전용 설명서에 따라 훈련된 Amazon Q Business를 사용합니다.

> **Note**
>
> AWS Partner Assistant에 액세스하려면 파트너는 AWS Partner Central 계정을 자신의 AWS 계정과 연결해야 합니다. 자세한 내용은 AWS Partner Central 시작 안내서의 Partner Central 계정과 AWS Marketplace 판매자 계정 연결을 참조하세요. AWS

챗봇은 AWS 서비스, AWS Marketplace 판매자 지침 및 AWS 파트너 네트워크 인센티브, 혜택 및 프로그램과 관련된 다양한 주제에 대한 실시간 답변과 문서 요약을 제공합니다. 챗봇이 응답을 제공할 수 없는 경우 지원 팀에 티켓을 제출하는 방법을 알려줍니다. 또한 챗봇의 정확도와 품질을 개선하는 데 도움이 되는 썸업 및 썸다운 버튼을 사용하여 챗봇 내에서 피드백을 제공할 수 있습니다.

다음은 AWS Partner Assistant와의 일반적인 상호 작용입니다.

1. 질문하기: AWS Partner Central 또는 AMMP에서 챗봇을 열고 일상적인 언어를 사용하여 질문합니다.
2. AWS Partner Assistant는 요청을 처리합니다. 챗봇은 자연어 처리(NLP)를 사용하여 쿼리를 해석합니다.

3. AWS Partner Assistant는 응답을 생성합니다. 챗봇은 추가 리소스가 포함된 관련 응답을 반환합니다.
4. 피드백 제공: 챗봇이 피드백을 묻는 메시지를 표시합니다.

이점 및 특징

AWS Partner Assistant는 정보에 대한 액세스를 개선하는 셀프 서비스 옵션을 제공합니다.

- 즉각적인 답변: 파트너는 일반적으로에 AWS 직접 문의하여 도움을 받아야 하지만 AWS Partner Assistant는 항상 사용할 수 있습니다.
- 정확한 답변: 파트너는 응답 변동성을 최소화하면서 사실적이고 시기 적절한 정보를 받습니다.
- 셀프 서비스 지원: 관련 리소스를 찾는 데 걸리는 시간을 줄입니다.
- 고급 응답: 자연스럽고 직관적이며 사람과 유사한 응답을 제공합니다.
- 컨텍스트 이해: 복잡한 주제에 대한 추가 정보를 제공합니다.
- 자체 학습 시스템: 상호 작용에서 학습하고, 시간이 지남에 따라 응답을 개선하고, 지식 기반을 확장하여 지속적으로 개선합니다.
- 확장 가능한 지원: 대량의 쿼리를 동시에 처리할 수 있습니다.

AWS Partner Assistant 사용 방법

1. AWS Partner Central 또는 AMMP에 로그인합니다.
2. 채팅에 액세스합니다.
   a. 오른쪽 하단 모서리에서 보라색 부동 아이콘을 선택합니다.
   b. 제안된 프롬프트 중 하나를 선택하거나 텍스트 필드에 질문을 입력합니다.
   c. 챗봇은 소스와 링크로 응답합니다.
3. Partner Central에서 챗봇에 액세스합니다.
   a. 지원 메뉴를 선택합니다.
   b. 드롭다운에서 AWS Partner Assistant와의 인스턴트 채팅을 선택합니다.
   c. (선택 사항) 지원 티켓에서 챗봇에 액세스합니다. 지원 티켓을 제출하기 전에 AWS 파트너 어시스턴트에게 문의를 선택하여 즉각적인 답변을 받으세요.
4. AMMP에서 챗봇에 액세스합니다.
   a. 문의 페이지에서 AWS 파트너 어시스턴트에게 문의를 선택합니다.

b. (선택 사항) 지원 티켓에서 챗봇에 액세스합니다. 지원 티켓을 제출하기 전에 문의 페이지에서 지원 요청 제출을 선택합니다. 즉시 답변을 받으려면 AWS 파트너 어시스턴트에게 문의를 선택합니다.

5. (선택 사항) 피드백 제공:
   a. 각 응답 후 피드백을 제공할 수 있는 옵션이 있습니다. 응답이 요구 사항을 충족하는 경우 썸업을 선택합니다.
   b. 응답이 도움이 되지 않으면 축소를 선택합니다.
   c. 축소를 선택하는 경우 텍스트 상자에 자세한 내용을 제공하거나 선택 목록에서 응답이 도움이 되지 않는 이유를 가장 잘 설명하는 옵션을 선택합니다.
   d. 챗봇 창의 왼쪽 하단 모서리에서 피드백을 선택하여 경험 평가: 피드백을 선택하여 평가 척도가 있는 팝업을 엽니다. 추가 의견이나 피드백이 있으면 텍스트 필드를 사용합니다.

파트너 어시스턴트 자주 묻는 질문

다음은 AWS Partner Assistant에 대해 자주 묻는 질문입니다.

AWS Partner Assistant란 무엇입니까?

AWS Partner Assistant는 파트너가 정보를 찾고 질문에 답변하는 데 도움이 되는 생성형 AI 기반 챗봇입니다.

AWS Partner Assistant에 액세스하려면 어떻게 해야 하나요?

AWS Partner Assistant는 AWS 파트너와 Marketplace 판매자 모두에게 제공되지만 Marketplace 구매자에게는 제공되지 않습니다. AWS Partner Central 또는 AWS Marketplace Management Portal(AMMP)을 통해 모든 페이지에서 챗봇에 액세스할 수 있습니다.

AWS Partner Assistant를 항상 사용할 수 있나요?

예.

Partner Assistant는 AWS 어떤 언어를 지원하나요?

현재 AWS Partner Assistant는 영어만 지원하지만 향후 릴리스에서는 다른 언어를 지원합니다.

AWS Partner Assistant는 무엇을 할 수 있나요?

챗봇은 AWS 서비스, AWS Marketplace 판매자 지침, AWS Partner 프로그램, 인센티브 및 이점에 대한 실시간 답변과 문서 요약을 제공합니다. 챗봇은 파트너만 사용할 수 있는 게시된 콘텐츠 라이브러리를 사용합니다.

AWS Partner Assistant에게 어떤 질문을 할 수 있나요?

챗봇은 AWS 서비스, AWS Marketplace 판매자 지침 및 AWS 파트너 네트워크 프로그램, 인센티브 및 이점에 대한 실시간 답변과 설명서 요약을 제공합니다. 챗봇은 파트너만 사용할 수 있는 게시된 콘텐츠 라이브러리를 사용합니다.

AWS Partner Assistant에게 물어보면 안 되는 문제는 무엇인가요?

심층적인 문제 해결, 계정 확인 또는 민감한 데이터에 대한 액세스가 필요한 문제는 AWS 고객 지원 센터에 직접 문의하세요.

AWS Partner Assistant는 개인 맞춤형 응답을 제공할 수 있나요?

현재 AWS Partner Assistant는 일반적인 정보와 지침만 제공합니다. 크리에이티브 콘텐츠를 생성하거나 작업을 수행할 수 없습니다. AWS Partner Assistant가 질문에 적절하게 답변할 수 없는 경우 지원 티켓을 제출하거나 피드백을 제출합니다. 다음은 추가 예제입니다.

수행	하지 말 것
티어 갱신 날짜는 어디에서 찾을 수 있나요?	티어 갱신 날짜는 언제인가요?
경로에 등록하려면 어떻게 해야 하나요?	소프트웨어 경로에 등록합니다.
MSP 프로그램을 신청하려면 어떻게 해야 하나요?	MSP 프로그램을 신청할 수 있나요?

> **Note**
> 계정 관련 질문의 경우 파트너는 기존 AWS 지원 채널을 사용해야 합니다.

파트너가 AWS Partner Assistant를 사용하여 계정 또는 서비스를 변경할 수 있나요?

아니요. 하지만 이러한 변경을 수행하는 방법을 안내하거나 할 수 있는 지원 팀을 안내할 수 있습니다.

Partner Assistant는 AWS 기밀 정보를 어떻게 처리하나요?

AWS Partner Assistant는 사용자 개인 정보 및 데이터를 보호하도록 설계되었지만 민감한 정보를 공유하는 것은 권장하지 않습니다.

AWS Partner Assistant가 내 질문을 이해하지 못하면 어떻게 되나요?

AWS Partner Assistant가 질문에 답할 수 없는 경우 질문을 다시 표현해 보세요. 또는 지원 팀에 도움을 요청할 수 있습니다. 이 시나리오에서는 챗봇의 성능을 개선하는 데 도움이 되는 추가 컨텍스트와 함께 썸다운 피드백을 제출합니다. 선택적으로 공통 범주의 선택 목록에서 구조화된 피드백을 제공할 수 있습니다.

AWS Partner Assistant가 응답하지 않는 경우 어떻게 해야 합니까?

페이지를 새로 고치거나 애플리케이션을 다시 시작해 보세요. 문제가 지속되면 지원 팀에 문제를 보고하세요.

AWS Partner Assistant가 잘못된 정보를 제공하는 경우 어떻게 해야 하나요?

AWS Partner Assistant가 잘못된 정보를 제공했다고 생각되면 세션이 끝날 때 축소 옵션을 사용하여 피드백을 제공합니다.

AWS Partner Assistant를 사용하지 않으려면 끌 수 있나요?

AWS Partner Assistant는 선택적 기능입니다. 사용하지 않으려면 지원 팀에 문의하세요.

용 제품 준비 AWS Marketplace

다음 주제에서는를 통해 판매할 소프트웨어 제품을 준비하는 방법을 설명합니다 AWS Marketplace. 제품을 준비하려면 패키지를 구성하고, 요금 체계를 설정하고, 제품을 나열할 범주를 결정하고, 제품이 관련 검색에 표시되도록 키워드를 추가해야 합니다. 독립 소프트웨어 개발 판매 회사(ISV), 채널 파트너, 관리형 서비스 제공업체(MSP)이거나 AWS 제품 및 서비스와 호환되는 제품을 제안할 수 있는 개인인 경우 이 프로세스를 사용하여 AWS Marketplace에 제품을 게시할 준비를 할 수 있습니다.

주제

- 에 대한 제품 제공 AWS Marketplace
- 에서 실행되는 제품의 지정 AWS
- 에 대한 제품 요금 AWS Marketplace
- AWS Marketplace 제품의 리전 및 국가
- 번역 및 언어
- 에서 표준화된 계약 사용 AWS Marketplace
- AWS Marketplace 제품에 대한 메타데이터 제공
- 에 대한 AMI 및 컨테이너 제품 사용 지침 생성 AWS Marketplace
- 검색을 위한 AWS Marketplace 제품 최적화
- 프로모션 미디어로 AWS Marketplace 제품 개선

에 대한 제품 제공 AWS Marketplace

제품 제공 방법마다 패키지, 요금 및 제공을 위한 몇 가지 옵션이 있습니다. 일부 메서드는 지원하는 프로그램에 등록할 AWS Marketplace 때까지에 판매자로 사용할 수 없습니다.

표준 정가 및 EULA(최종 사용자 라이선스 계약)를 사용하여 제품을 생성할 수 있습니다. 또한 사용자 지정 가격 및 EULA를 사용하여 개별 고객에 비공개 제안을 생성할 수 있습니다. 계약 조건에 추가 변경 사항이 필요한 경우 AWS Marketplace 팀과 협력하여 사용자 지정 비공개 제안을 생성할 수 있습니다. 공개 제품 리스팅과 비공개 제안에 대해 표준화된 라이선스 조건을 사용하여 조달 프로세스를 간소화할 수 있습니다.

> **Note**
>
> AWS 는 특정 판매자에게 안내 데모를 제공할 수 있는 옵션을 제공합니다 AWS Marketplace. APN Customer Engagements(ACE)를 사용할 수 있는 AWS 파트너 네트워크(APN) 파트너이

고 구매자에게이 옵션을 제공하려면 비공개 제안 및 데모 요청 버튼 추가 섹션을 참조하세요. https://aws.amazon.com/partners/programs/ace/

다음 표에는 소프트웨어 제품을 제공하는 데 사용할 수 있는 방법과 AWS Marketplace 구매자가 콘솔에서 각 유형의 제공 방법을 찾는 데 사용하는 필터가 AWS Marketplace 나와 있습니다.

제품 제공 방법	콘솔의 전송 방법 필터	설명
단일 AMI	Amazon Machine Image(AMI)	판매자가 제품의 단일 사용자 지정 Amazon Machine Image(AMI)를 제공합니다. AMI는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 구매자는 단일 AMI를 사용하여 제품이 이미 설치되어 사용 가능한 상태로 Amazon EC2 인스턴스를 생성할 수 있습니다. 자세한 내용은 의 AMI 기반 제품 AWS Marketplace 단원을 참조하십시오.
AWS CloudFormation 템플릿을 사용하여 제공되는 AMIs	CloudFormation 템플릿	CloudFormation 템플릿을 사용하여 AWS Marketplace 구매자에게 제공되는 AMI 기반 제품을 나열할 수 있습니다. AMIs CloudFormation 템플릿으로 제공하는 방법에 대한 자세한 내용은를 사용한 AMI 기반 전송 AWS CloudFormation을 참조하세요.

제품 제공 방법	콘솔의 전송 방법 필터	설명
		CloudFormation 템플릿에 대한 자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 개념을 참조하세요.
EC2 Image Builder 구성 요소 기반 제품	EC2 Image Builder 구성 요소	EC2 Image Builder 구성 요소를 사용하여 AWS Marketplace 구매자에게 배송된 제품을 나열할 수 있습니다. EC2 Image Builder 구성 요소 사용에 대한 자세한 내용은 API 참조의 EC2 Image Builder 구성 요소 제품 작업을 참조하세요. AWS Marketplace
컨테이너 기반 제품 또는 애플리케이션	컨테이너	컨테이너 이미지에 패키징된 제품을 제공합니다. 컨테이너 제품은 함께 작동하는 컨테이너 이미지와 배포 템플릿 세트인 옵션으로 구성됩니다. 자세한 내용은 의 컨테이너 기반 제품 AWS Marketplace 단원을 참조하세요.

제품 제공 방법	콘솔의 전송 방법 필터	설명
데이터 제품	AWS Data Exchange	판매자가 AWS Data Exchange를 사용하여 데이터 제품을 생성합니다. AWS Data Exchange를 통해 데이터 제품 및 제안을 게시하고 관리하는 방법에 대한 자세한 내용은 AWS Data Exchange 사용 설명서의 AWS Data Exchange에서 데이터 제품 제공을 참조하세요.
기계 학습 알고리즘 및 모델 패키지	SageMaker 모델	Amazon SageMaker AI를 사용하여 알고리즘 또는 모델 패키지를 생성한 다음 게시합니다 AWS Marketplace. 기계 학습 알고리즘 및 모델 패키지 제공에 대한 자세한 내용은 기계 학습 제품 섹션을 참조하세요. SageMaker AI에 대한 자세한 내용은 Amazon SageMaker AI 개발자 안내서의 SageMaker AI란 무엇인가요?를 참조하세요. Amazon SageMaker

제품 제공 방법	콘솔의 전송 방법 필터	설명
서비스형 소프트웨어(SaaS)	SaaS	구독 기반, 계약 기반 또는 소비 요금 모델과의 계약이 포함된 SaaS 제품을 제공할 수 있습니다. 자세한 내용은 의 SaaS 기반 제품 AWS Marketplace 단원을 참조하십시오.
전문 서비스	전문 서비스	다른 AWS Marketplace 제품을 지원하거나 사용하는 전문 서비스를 제공할 수 있습니다.

에서 실행되는 제품의 지정 AWS

> **Note**
>
> 다음 지정은 2025년 5월 1일부터 적용됩니다.

에 배포된 제품은 검색 결과 및 제품 세부 정보 페이지에서 특별한 지정 AWS Marketplace 을 AWS 받습니다. 다음 표에는 지정을 받을 수 있는 제품 제공 유형이 나열되어 있습니다.

제품 제공 방법	에 배포된 것으로 지정됨 AWS
AMI 기반	예
컨테이너 기반	예. Amazon EKS Anywhere에 배포된 제품은 예외입니다. 이 기능을 활성화하는 컨테이너 제품에 대한 자세한 내용은이 가이드 AWS Marketplace for Containers Anywhere와 License Manager 통합뒷부분의 섹션을 참조하세요.
SageMaker AI 알고리즘 및 모델 패키지 기반	예

제품 제공 방법	에 배포된 것으로 지정됨 AWS
SaaS 기반	완전히 실행되는 제품에만 해당됩니다 AWS. 자세한 내용은이 가이드의 아키텍처 가이드라인뒷부분을 참조하세요.
데이터 기반	예
전문 서비스 기반	아니요

에 대한 제품 요금 AWS Marketplace

AWS Marketplace 판매자는 제품의 요금 모델을 선택할 수 있습니다. 유료 제품의 경우 AWS Marketplace 가 고객에게 소프트웨어 수수료를 수취합니다. 모든 요금은 미국 달러(USD) 기준입니다. 고객이 무료로 사용할 수 있는 무료 또는 오픈 소스 소프트웨어에 대해서는 서비스 수수료가 없습니다. 이 주제에서는 AWS Marketplace의 소프트웨어 제품의 요금 모델 및 요금에 대한 일반적인 정보를 제공합니다.

환불에 대한 자세한 내용은 에서 제품 환불 AWS Marketplace 섹션을 참조하세요.

주제

- 요금 모델
- 요금 모델 변경
- 요금 변경
- 비공개 제안
- 에서 제품 환불 AWS Marketplace

요금 모델

다음 주제에서는 AWS Marketplace에서 제공하는 요금 모델에 대한 일반 정보를 제공합니다.

주제

- 연간 요금
- 사용 요금
- 계약 요금

- 기존 보유 라이선스 사용 요금

특정 제품 제공 방법의 요금 모델에 대한 자세한 내용은 다음을 참조하세요.

- 에 대한 AMI 제품 요금 AWS Marketplace
- 에 대한 컨테이너 제품 요금 AWS Marketplace
- 에 대한 기계 학습 제품 요금 AWS Marketplace
- 의 SaaS 제품 요금 AWS Marketplace
- 에서 전문 서비스 제품 요금 책정 AWS Marketplace

연간 요금

연간 요금 모델을 사용하면 12개월 구독을 구매할 수 있는 고객에게 제품을 제공할 수 있습니다. 예를 들어 구독 요금은 동일한 제품을 오랫동안 시간당 요금으로 실행할 때와 비교하여 최대 40%를 절감할 수 있습니다. 구독 시 고객에게 계약의 전체 금액에 대한 인보이스가 발행됩니다. 고객에게 연간 구독이 제시되는 방식에 대한 자세한 내용은 AMI 구독 또는 유료 컨테이너 제품에 적용되는 요금 모델을 참조하세요.

연간 구독을 작업할 때 다음 사항을 고려해야 합니다.

- 연간 요금은 인스턴스 유형마다 정의됩니다. 모든 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 유형을 똑같이 정의할 수도 있고 인스턴스 유형마다 다르게 할 수도 있습니다.
- 모든 연간 인스턴스 유형에는 시간당 인스턴스 유형도 정의되어 있어야 합니다. AWS Marketplace는 동일한 제품에 연간 전용 요금이나 연간 요금이 없는 시간당 요금을 제공하지 않습니다. 연간 요금을 제공하는 제품의 경우 시간당 요금도 지정되어야 합니다.
- 시간당 요금도 0 USD이며 다른 0 USD가 아닌 연간 인스턴스 유형이 정의되지 않은 경우 특정 인스턴스 유형에 대해 0 USD의 연간 요금이 허용됩니다.
- 연간 구독 기간이 끝날 때 고객에게 시간당 요금이 청구되기 시작합니다.
- 고객이 X 연간 구독을 구매했지만 Y 인스턴스에 Y 소프트웨어를 실행하고 있는 경우 고객에게는 연간 구독이 적용되지 않는 시간당 소프트웨어 가격으로 (Y-X) 인스턴스에 대한 요금이 부과됩니다. 따라서, 모든 연간 요금 인스턴스 유형에 시간당 요금이 포함되어야 합니다.
- 판매자 비공개 제안을 사용하여 선결제 또는 유연한 결제 일정으로 다년(최대 3년) 또는 사용자 지정 기간 AMI를 제안할 수 있습니다. 다년 및 사용자 지정 기간 계약에 대한 자세한 내용은 AWS Marketplace 제품에 대한 비공개 제안 준비 및 비공개 제안 할부 플랜 단원을 참조하세요.

에서 연간 제품을 제공하는 경우 AWS Marketplace의 파일 업로더 문서 섹션에 있는 연간 제품에 대한 특정 환불 정책에 동의하는 것입니다AWS Marketplace Management Portal.

가격 변경

90일마다 연간 요금(USD 가격. 예: 연간 1,000 USD에서 연간 1,200 USD로 인상)을 변경할 수 있습니다. 하지만 기존 연간 요금 고객에게는 90일 전에 알려야 합니다. 새로운 가격이 새 구독에 적용되지만, 기존 구독에는 영향을 주지 않습니다.

> **Note**
>
> 90일의 알림 기간에는 지원되는 인스턴스 유형을 업데이트할 수 없습니다.

자동 갱신일로부터 최소 90일 전에 가격이 변경된 경우에만 가격 변경이 자동 갱신에 적용됩니다. 고객은 자동 갱신 전에 새 가격이 포함된 이메일 메시지를 받게 됩니다.

최종 사용자 라이선스 계약

연간 구독에 따라 AWS 고객이 12개월 동안 소프트웨어를 사용하는 경우 제품 세부 정보 페이지에서 제공하는 EULA가 적용됩니다 AWS Marketplace.

사용 요금

사용량에 따른 요금이라고도 하는 사용량 요금 모델을 통해 사용한 만큼만 요금을 지불하는 고객에게 제품을 제안할 수 있습니다.

판매자는 다음 사용량 범주 중 하나를 선택할 수 있습니다.

- Users
- 호스트
- 대역폭
- 데이터
- 티어
- 단위(사용자 지정 범주인 경우)

또한 제품에 대해 최대 24개의 차원을 정의할 수 있습니다. 요금은 소프트웨어에서 API를 호출할 때 측정하여 보고됩니다. 사용 사례에 따라 다르지만, 판매자는 시간당 한 번 호출되도록 API를 구성하는 것

이 가장 좋습니다. 모든 사용량은 매월 계산되며 기존 AWS Marketplace 소프트웨어와 동일한 메커니즘을 사용하여 매월 청구됩니다.

AWS Marketplace 측정 서비스를 사용하면 여러 가지 새로운 요금 시나리오를 처리할 수 있습니다.

Example 호스트당 요금

소프트웨어에서 호스트를 모니터링하는 경우 모니터링된 각 호스트에 대해 요금을 부과하고 호스트 크기에 따라 다른 요금을 설정할 수 있습니다.

Example 사용자당 요금

소프트웨어가 조직 전체에서 여러 사용자를 허용하는 경우 사용자별로 요금을 부과할 수 있습니다. 시간마다 고객에게 프로비저닝된 사용자의 총 수에 대해 요금이 부과됩니다.

> **Note**
>
> 제품 로드 양식(PLF)에서 관련 열 앞에 Flexible Consumption Pricing(FCP)이 붙습니다. 예: FCP 범주(사용자 지정 요금 범주).

AWS Marketplace Metering Service 제품의 경우 다음 사항에 유의하세요.

- 소프트웨어가 이미 켜져 있는 경우 대체 사용 차원을 활성화하려면 제품을 생성 AWS Marketplace 해야 합니다. AWS Marketplace 측정 서비스를 사용하도록 표준 제품을 전환할 수는 없습니다. 새 제품이 게시된 후 이전 제품을 제거할 수도 있고 둘 다 웹 사이트에 유지할 수도 있습니다.
- AWS Marketplace 측정 서비스에서는 소프트웨어가 매 시간 사용을 보고하여 해당 시간에 대한 고객 사용을 기록해야 합니다. 측정 서비스 레코드의 전송 또는 수신에 장애가 있는 경우는 이러한 사용량에 대해 요금을 청구할 AWS 수 없습니다. 측정 레코드의 수신 성공을 확인하는 것은 사용자의 책임입니다.
- AWS Marketplace 측정 서비스를 사용하는 제품은 1-Click을 지원하지 않습니다. 구매자는 특정 권한이 있고 인터넷 게이트웨이가 있는 AWS Identity and Access Management (IAM) 역할로 소프트웨어를 시작해야 합니다.
- 무료 평가판 및 연간 요금은 AWS Marketplace 측정 서비스와 호환되지 않습니다.
- 차원(사용자, 호스트, 대역폭 및 데이터) 또는 차원 이름을 변경할 수 없습니다. 새 제품을 생성해야 합니다.

계약 요금

계약 요금 모델을 사용하면 고객에게 1개월, 12개월, 24개월 또는 36개월 라이선스를 구매할 수 있는 선결제 요금을 제안할 수 있습니다.

계약 요금을 사용할 수 있는 제품은 다음과 같습니다.

- 단일 AMI 기반 제품 및 AWS CloudFormation 템플릿 기반 제품이 포함된 AMI. 자세한 내용은 의 AMI 제품에 대한 계약 요금 AWS Marketplace 섹션을 참조하세요.
- 컨테이너 기반 제품 자세한 내용은 컨테이너 제품의 계약 요금 단원을 참조하세요.
- 서비스형 소프트웨어(SaaS) 기반 제품 자세한 내용은 SaaS 계약 요금 단원을 참조하세요.

> **Note**
>
> AMI 및 컨테이너 기반 제품의 계약 요금은 새 제품에만 적용됩니다.
> 기존 AMI 또는 컨테이너 기반 제품이 있고 계약 요금을 사용하려는 경우 새 목록을 생성한 다음 제품 로드 양식(PLF)을 사용하여 다른 차원을 추가하고 AMI 또는 컨테이너 기반 제품을와 통합 AWS License Manager한 다음 AMI 또는 컨테이너 기반 제품을 게시하여 계약 요금 모델을 적용합니다.
> 고객이 계약 요금이 적용되는 제품을 구매하면에서 고객 AWS Marketplace 에 소프트웨어가 License Manager API를 사용하여 확인할 수 AWS 계정 있는 라이선스가 생성됩니다. 고객은 AMI 또는 컨테이너 기반 제품의 인스턴스를 시작하려면 IAM 역할이 필요합니다.

기존 보유 라이선스 사용 요금

AWS Marketplace에 등록된 BYOL(Bring Your Own License) 제품에 대한 서비스 수수료가 없습니다.

고객의 선택 약속을 전달하기 위해 모든 BYOL 제품에도 유료 옵션이 있어야 합니다. 그렇게 해야 기존 라이선스가 없는 고객이 제품을 구매하고 사용할 수 있습니다.

BYOL 제품의 경우, 온라인 소프트웨어 구매가 일부 회사의 비즈니스 수행 방식을 벗어납니다. 따라서 시작 후 첫 90일 동안 이 소프트웨어에 AWS Marketplace에서 구매할 수 있는 버전이 포함되어야 한다는 요구 사항을 완화합니다. 이 기간 동안 AWS Marketplace 계정 관리 팀은 고객과 협력하여 문제를 해결합니다. 이 팀은 소프트웨어를 AWS Marketplace에서 판매할 수 있는지 여부와 판매 방법을 결정하는 데 도움을 줄 수 있습니다.

요금 모델 변경

긍정적인 고객 경험을 보장하고 모든 당사자에 대한 위험을 줄이기 AWS Marketplace 위해 요금 모델에 대한 변경 사항을에서 검토하고 승인해야 합니다. AWS Marketplace 판매자 작업 팀에 연락하여 변경하고 싶은 요금 모델에 대해 문의하십시오.

요금 모델 변경에 대한 모든 요청은 처리 및 검토하는 데 30~90일이 걸릴 수 있습니다.

> **Note**
>
> SaaS 제품에는 요금 모델 변경이 지원되지 않습니다. SaaS 제품 요금에 대한 자세한 내용은 의 SaaS 제품 요금을 AWS Marketplace 참조하세요.

요금 변경

AWS Marketplace Management Portal를 통해 요금 및 메타데이터를 업데이트할 수 있습니다.

요금을 변경하려면

1. AWS Marketplace Management Portal에 로그인합니다.
2. 제품 탭을 보면, 판매자가 생성한 현재 제품 목록이 표시됩니다. 여기에서 제품 목록을 편집하거나 변경 사항을 요청합니다.

> **Note**
>
> 신규 구독자의 경우 가격 변경이 즉시 적용됩니다. 공개 제안의 기존 구독자의 경우 가격 변경은 90일 기간 이후 매월 1일에 적용됩니다. 예를 들어 3월 16일에 가격 변경 알림을 보내는 경우 가격 변경은 90일 기간인 7월 1일 이후 매월 1일에 발생합니다.

비공개 제안

AWS Marketplace 판매자 비공개 제안 프로그램에서 AWS Marketplace 판매자는 개별 AWS Marketplace 고객(구매자)과 사용자 지정 요금 및 EULAs 협상할 수 있습니다. 자세한 내용은 AWS Marketplace 제품에 대한 비공개 제안 준비 단원을 참조하십시오.

에서 제품 환불 AWS Marketplace

> **Note**
>
> Amazon Web Services EMEA 운영자를 통해 거래하는 경우 확인된 사용자만 환급을 처리하고, 고객 파악(KYC) 정보를 수정하고, 은행 계좌 세부 정보와 같은 금융 정보를 변경할 수 있습니다. 확인된 사용자는 KYC 확인 사용자 및 확인을 완료하는 보조 사용자입니다. KYC 프로세스에 대한 자세한 내용은 고객 파악 프로세스 완료를 참조하세요.

요금 모델에 AWS Marketplace관계없이의 모든 유료 제품에는 소프트웨어 요금에 대해 명시된 환불 정책이 있어야 합니다. 환불 정책에는 환불 조건과 환불을 요청하기 위해 판매자에게 문의하는 방법이 포함되어야 합니다.

환불 정책의 세부 사항은 판매자가 결정합니다. 그러나 구매자에게 제품 사용에 대한 환불 방법을 제공하는 것이 좋습니다. 게시된 환급 정책을 준수해야 합니다. 이 주제에서는 AWS Marketplace 제품 환불 요청 유형, 관련 정책 및 승인 프로세스, 구매자에 대한 환불 요청을 제출하는 방법에 대한 정보를 제공합니다.

주제

- AWS Marketplace 제품에 대한 환불 요청 유형
- AWS Marketplace 제품 환불 정책 및 승인
- 제품 환불 요청
- 대량 환불 요청

AWS Marketplace 제품에 대한 환불 요청 유형

구매자는 AWS Marketplace 제품에 대해 다양한 유형의 환급을 요청할 수 있습니다. 에서 판매하는 AWS Marketplace 제품의 경우 환불 정책 페이지를 AWS참조한 다음를 사용하여 고객 지원 양식을 제출합니다 AWS Support Center Console. 구매자가 직접 소프트웨어 환불을 요청하는 경우 해당 제품에 대해 게시된 지원 연락처 정보를 사용하여 판매자에게 문의하도록 AWS지시합니다. AWS 인프라 요금에 대한 환급은의 재량에 따르 AWS 며 소프트웨어 환급과 관계없이 처리됩니다.

타사에서 판매하는 제품의 경우 구매자는 제품 세부 정보 페이지에서 환불 정책을 확인해야 합니다. AWS Marketplace 구독에 대한 소프트웨어 요금은 제품 판매자에게 지불되며 판매자로부터 직접 환불을 요청해야 합니다. 각 AWS Marketplace 판매자는 AWS Marketplace 페이지에 환불 정책을 포함해야 합니다.

AWS Marketplace 제품 환불 정책 및 승인

다음 목록에서는 AWS Marketplace 환불 정책과 승인 필요 여부를 설명합니다.

- 무료 평가판

  소프트웨어를 무료 평가판 제품으로 등록하면 AWS 는 무료 평가판에서 유료 구독으로 전환한 후 7일 이내에 발생한 소프트웨어 요금을 사용자를 대신하여 환불할 수 있습니다. 무료 체험판 변환과 관련하여 지급된 환급금에는 사용자의 별도 작업이 필요하지 않습니다. 제품에 대한 무료 평가판을 활성화하면이 정책에 동의하는 것입니다.

- 비공개 제안

  비공개 제안에 대한 모든 환불은에서 AWS 처리하기 전에 사용자의 승인을 받아야 합니다.

- 소프트웨어 측정 환불

  AWS Marketplace Metering Service 를 사용하여 소프트웨어 사용량을 측정하는 경우는 소프트웨어 측정 오류로 인한 소프트웨어 요금을 사용자 대신 환급할 AWS 수 있습니다. 이러한 오류가 여러 구매자에게 공통적인 경우는 각 구매자에게 적절한 환불을 결정하고 각 구매자에게 직접 적용할 수 있는 권리를 AWS 보유합니다. 와 관련하여 발행된 환불 AWS Marketplace Metering Service 은 판매자와 한 번 확인해야 하지만 판매자가 각 개별 환불을 확인할 필요는 없습니다. 제품과 AWS Marketplace Metering Service 함께를 사용하면이 정책에 동의하는 것입니다.

- 구매 후 48시간 내에 구독 취소

  구매자가 비공개 제안 구매 후 48시간 이내에 구독을 취소하는 경우는 전체 환불(100% 환불로 취소)을 AWS 발행합니다. 판매자가 구매 후 48시간 내에 취소하여 환불된 경우 판매자는 아무 것도 할 필요가 없습니다. 48시간 후에는 판매자의 재량에 따라 구매자 요청을 처리할 수 있습니다. 에 제품을 등록하면이 정책에 AWS Marketplace동의하는 것입니다.

- 구독 업그레이드

  구매자가 기존 비프라이빗 제안 구독을 더 비싼 구독 또는 동일한 가격의 구독으로 대체하는 경우는 하위 계층 구독에 대해 사용자를 대신하여 환불을 발행할 AWS 수 있습니다. 구매자는 새 구독을 구매한 후 기존 구독의 취소와 환불을 요청하는 2단계 절차를 거치게 됩니다.

- 구독 다운그레이드

  모든 다운그레이드 구독 환불 요청은 사용자가 승인해야에서 처리할 AWS 수 있습니다.

AWS 승인된 모든 환불은 자동으로 처리되며 사용자가 조치를 취할 필요가 없습니다.

제품 환불 요청

다음 절차에 따라 외부 구매자 또는 내부 테스트 계정에 대한 소프트웨어 환불을 요청합니다.

소프트웨어 환불을 요청하려면

1. 다음 정보를 수집합니다.
   - 제품을 구독하는 데 사용되는 구매자 AWS 계정 ID(12자리)입니다. 구매자가 조직의 지급인인 경우 구독에 사용된 연결된 계정의 AWS 계정 ID를 가져옵니다.
   - 구매자와 연결된 구매자의 이메일 주소입니다 AWS 계정.
   - 판매자 AWS 계정 ID(12자리).
   - 제품 ID 계약 및 갱신 대시보드 또는 청구 수익 대시보드에서 제품 ID를 찾을 수 있습니다.
   - 제안 ID입니다. 계약 및 갱신 대시보드 또는 청구 수익 대시보드, 제안 탭 또는의 계약 탭에서 제안 ID를 찾을 수 있습니다AWS Marketplace Management Portal.
   - 결제 기간.
   - 청구 수익 대시보드의 인보이스 번호입니다.
   - 세금 전 환급 금액입니다.
2. AWS Marketplace Management Portal에 로그인합니다.
3. 지원을 선택합니다.
4. 환불 요청을 선택합니다.
5. 구독자의 AWS 계정 번호, 제품 ID 및 결제 기간을 입력합니다.
6. 추가 설명(선택 사항) 필드에 제품 ID, 환불 유형 및 인보이스 번호(사용 가능한 경우)를 포함하여 요청에 대한 간략한 설명을 입력합니다. 예시:

   Cancel the buyer's subscription to *<product-ID>* and *<offer-ID>* and process a full refund for invoice *<invoice-number>*.
7. 연간 구독의 환불, 업그레이드 또는 다운그레이드 요청을 제출하려면 다음을 수행합니다.
   a. 계약 및 갱신 대시보드 또는 청구 수익 대시보드를 사용하여 구매자가 연간 구독을 구매했는지 확인합니다.
   b. 추가 설명(선택 사항) 필드에 구독 취소 날짜를 입력하고 구매자가 환불, 업그레이드 또는 다운그레이드를 요청하는지 여부를 지정합니다.
8. 양식을 제출합니다. 알림을 받게 되며 환불을 처리하고 구매자에게 발급하기 시작합니다.

9. 환급 요청을 관리하기 위해 AWS Marketplace 구매자 지원 팀은 지원 센터 콘솔에서 해당 지원 사례를 생성합니다. 환불 상태는 지원 사례의 제목 줄에 표시됩니다. 상태는 다음 중 하나일 수 있습니다.

   - 완료됨 - 환불이 처리되었으며 추가적인 조치가 필요하지 않습니다.
   - 보류 중 - 현재 청구 주기가 종료되면 환불이 처리됩니다.
   - 조치 필요 - 요청을 처리할 수 없으며 추가 정보가 필요합니다. 지원 사례에 직접 응답할 수 있지만 새 환급 요청 양식도 제출해야 합니다.

   환불은 24~48시간 이내에 구매자의 계정에 표시됩니다. 그러나 자금이 구매자의 금융 계좌에 표시되려면 영업일 기준 최대 5일이 걸릴 수 있습니다.

대량 환불 요청

다음 단계에서는 20개 이상의 인보이스 또는 20개 이상의 계정과 관련된 환불 요청을 생성하는 방법을 설명합니다.이 프로세스는 대규모 환불 요청을 간소화하고 필요한 정보를 제공합니다.

대량 환불을 요청하려면

1. 다음과 같은 필수 정보를 수집합니다.

   - 판매자의 AWS 계정 ID(12자리)
   - 모든 AWS 계정 IDs 목록(각각 12자리). 여러 구매자를 입력하거나 여러 결제 기간 동안 단일 구매자를 입력할 수 있습니다.
   - 모든 제품 IDs
   - 해당하는 경우 제안 IDs
   - 모든 인보IDs
   - 각 인보이스의 청구 기간
   - AWS Marketplace 환불 티켓 참조 ID
2. 다음 열을 사용하여 필요한 정보를 CSV 스프레드시트에 컴파일합니다.

   - **`Seller Account ID`**
   - **`Subscriber Account ID`**
   - **`Payer Account ID`**
   - **`Billing Period`**

- **Invoice ID**
- **Targeted Amount**
- **Product ID**

3. 루트 사용자로 지원 Center Console에 로그인합니다.

> **Note**
> 루트 사용자로 로그인하지 않으면이 단계를 완료할 수 없습니다.

4. 계정 및 결제에 대한 지원 사례를 생성하고를 선택합니다AWS Marketplace.
5. 제목 줄에를 입력합니다**Request Bulk Refund**.
6. 추가 설명에 다음을 입력합니다.
   - 영향을 받는 총 계정 수
   - 총 인보이스 수
   - 총 환급 금액
   - 환급에 대한 비즈니스 근거
   - AWS Marketplace 환불 티켓 참조 ID
7. CSV 파일을 연결하고 지원 사례를 생성합니다.
8. 지원 사례는 지원을 위해 사용 가능한 다음 에이전트로 라우팅됩니다.

AWS Marketplace 제품의 리전 및 국가

에서 제품을 생성할 때 제품을 사용할 수 있는 AWS 리전을 AWS Marketplace선택합니다. 구매자가 제품을 구매할 수 있는 국가도 선택할 수 있습니다. 이 두 속성은 비슷하지만 동일하지는 않습니다. 예를 들어 구매자가 미국에 살면서 미국에서 제품을 구매하지만 유럽(프랑크푸르트) 리전에 제품을 설치할 수 있습니다. 이 구매자가 제품을 구매할 수 있으려면 국가 목록에 미국을 포함시키고 리전 목록에 유럽(프랑크푸르트) 리전을 포함시켜야 합니다. 이 주제를 사용하여 AWS Marketplace 제품의 리전 및 국가에 대해 자세히 알아볼 수 있습니다.

AWS 리전

서버 또는 기계 학습 제품 정보를 생성하거나 편집할 때 사용자가 제품을 설치하고 사용할 수 있는 특정 AWS 리전으로 제품을 제한할 수 있습니다.

Amazon Machine Image(AMI), 컨테이너 및 AWS CloudFormation기반 제품을 포함한 서버 제품의 경우 제품을 사용할 수 있는 특정 리전을 선택할 수 있습니다. 제품을 사용할 수 있게 되면 새로운 미국 리전, 미국 외 리전 또는 모든 리전에서 자동으로 제공하도록 선택할 수도 있습니다.

기계 학습 제품의 경우 판매자가 특정 리전을 선택할 수도 있고 향후 제공할 리전을 포함한 모든 리전을 선택할 수도 있습니다.

AWS 리전에 대한 자세한 내용은 AWS 일반 참조의 AWS 서비스 엔드포인트를 참조하세요.

국가

기본적으로를 사용할 수 있는 모든 국가의 구매자 AWS Marketplace 는 제품을 사용할 수 있습니다. 신규 및 기존 서버 제품과 서비스형 소프트웨어(SaaS) 제품의 경우 세금, 규정 준수, 지원 또는 마케팅 목적으로 국가별 제품 가용성을 판매자가 제어할 수 있습니다.

이 기능에는 예외가 있습니다.

- 이전 구매 - 제품을 새 국가 목록으로 업데이트하더라도 이전에 제품을 구독한 구매자는 구독이 활성 상태인 동안에는 계속 제품을 이용할 수 있습니다.
- 비공개 제안 - 제품을 특정 국가의 구매자로 제한해도 비공개 제안은 제한되지 않습니다. 특정 구매자에게 비공개 제안을 생성하면 해당 구매자가 판매자가 지정한 국가 이외의 국가에 거주하더라도 해당 구매자는 비공개 제안을 이용할 수 있습니다.

> **Note**
>
> 고객 자격은 AWS 연결된 계정 수준에서 결정됩니다. 자세한 내용은 에서 계정의 위치를 어떻게 확인하나요 AWS ?를 참조하세요.
> 권한을 공유하는 고객은 판매자가 허용한 리전에서만 권한을 활성화할 수 있습니다. 권한 관리에 대한 자세한 내용은 AWS Marketplace 구매자 설명서의 조직 내에서 구독 공유를 참조하세요.

번역 및 언어

기본적으로는 제품 목록 및 제안을 다음 언어로 AWS Marketplace 번역합니다.

- 프랑스어(fr-FR)
- 일본어(ja-JP)

- 한국어(ko-KR)
- 스페인어(es-ES)

AWS Marketplace 는 참조용으로만 해당 언어로 된 일련의 예제 표준 계약도 제공합니다. 계약은 법적 구속력이 없습니다. 계약에 대한 자세한 내용은 AWS Marketplace 구매자 안내서의 지원되는 언어를 참조하세요.

다음 주제에서는 번역 작동 방식, 번역된 표준 계약을 다운로드하는 방법, 번역을 옵트아웃하는 방법을 설명합니다.

주제

- 시작
- 번역 옵트아웃

시작

AWS Marketplace 번역에 대해 다음 사항을 기억하세요.

- 글로벌 시장으로의 도달 범위를 넓히는 데 도움이 되도록는 추가 노력 없이 제공하는 영어 콘텐츠를 AWS Marketplace 번역합니다. 번역에 문제가 있는 경우 AWS Marketplace 관리 포털의 문의처 페이지를 사용합니다. 제품 목록 및 현지 언어 옵트아웃 범주를 선택하고 문제에 대한 피드백을 제공합니다.

  AWS Marketplace 는 기술 용어나 제품 이름을 번역하지 않는 등 제한된 변경을 수행할 수 있습니다.
- 목록 번역을 옵트아웃할 수 있습니다. 이렇게 하면 구매자는 영어로 된 콘텐츠와 원하는 언어로 된 관련 AWS Marketplace 콘텐츠를 볼 수 있습니다. 옵트아웃에 대한 자세한 내용은 번역 옵트아웃 아래를 참조하세요.
- 판매자 등록 및 제품 목록 정보를 영어로 제출해야 합니다.
- 지원되는 언어로 EULAs를 제출할 수 있습니다. 관련 제품 목록은 지리적 대상이어야 하며 언어는 해당 국가의 기본 언어여야 합니다.

  AWS Marketplace 는 참조용으로만 위에 나열된 언어로 번역된 표준 예제 계약 세트를 제공합니다. AWS Marketplace 구매자 안내서의 지원되는 언어에서 계약을 다운로드할 수 있습니다.
- 새 제품을 게시할 때 제품이 제한된 상태에서 번역을 검토하는 것이 가장 좋습니다. 제품을 공개하기 전에 문제가 있으면 문의하세요.

- 제품을 제한된 상태로 이동하면 번역 프로세스가 시작됩니다. 구매자는 번역을 사용할 수 있을 때까지 영어 텍스트를 볼 수 있습니다.

번역 옵트아웃

다음 단계에서는 제품 목록 번역을 옵트아웃하는 방법을 설명합니다. 일부 또는 모든 언어를 옵트아웃할 수 있습니다. 예를 들어 목록을 프랑스어와 일본어로만 번역할 수 있습니다.

옵트아웃이 적용되려면 3~5일이 걸립니다.

옵트아웃하려면

1. AWS Marketplace 관리 포털로 이동하여 문의를 선택합니다.
2. 나열된 순서대로 다음 범주를 선택합니다.
   - 상용 마켓플레이스
   - 제품 목록
   - 로컬 언어 옵트아웃
3. 목록 ID와 옵트아웃하려는 언어를 포함합니다.
4. 가장 좋은 방법은 구매자가 영어 텍스트를 보는 이유를 이해하는 데 도움을 주는 것입니다. 예시:
   a. 모든 번역을 옵트아웃**English**할 때 제품 이름 끝에를 추가합니다.
   b. 하나 이상의 번역을 옵트아웃할 때 제품 설명 끝에 설명 텍스트를 추가합니다. 예를 들어 **We have declined automatic translation services for this product**입니다.
5. 요청을 제출합니다.

에서 표준화된 계약 사용 AWS Marketplace

AWS Marketplace 판매자로서 제품을 준비할 때, 제품 사용에 적용할 최종 사용자 라이선스 계약(EULA)을 결정해야 합니다. 다음을 사용할 수 있습니다.

- 사용자 고유의 EULA.
- 표준 계약 AWS Marketplace (SCMP)은 조달 워크플로를 간소화하고 트랜잭션 속도를 높이는 데 도움이 될 수 있는 계약 템플릿입니다. SCMP에 대한 연방 부록을 사용하여 미국 정부 조달 속도를 높일 수도 있습니다.
- ISVs가 사용할 수 있는 표준화된 리셀러 계약 템플릿인 리셀러 계약 AWS Marketplace (RCMP).

다음 주제에서는 표준화된 AWS Marketplace 계약 템플릿을 사용하는 방법을 설명합니다.

주제

- 에 대한 표준 계약 AWS Marketplace
- 에 대한 리셀러 계약 AWS Marketplace

에 대한 표준 계약 AWS Marketplace

AWS Marketplace 는 구매자 및 판매자 커뮤니티와 협력하여 표준 계약 AWS Marketplace (SCMP)을 개발했습니다. SCMP는 디지털 솔루션의 사용을 규제하고 디지털 솔루션과 관련된 구매자와 판매자의 의무를 정의합니다. 디지털 솔루션의 예로는 서버 소프트웨어, 서비스형 소프트웨어(SaaS), 인공 지능 및 기계 학습(AI/ML) 알고리즘 등이 있습니다.

SCMP는 사용, 보증, 배상 및 준거법과 같은 주요 계약 조항 전반의 공통 기반을 사전에 정의합니다. 판매자는 SCMP 조건을 셀프 서비스 거래의 EULA로 제공하며, 구매자는 솔루션을 검색, 구매하고 신속하게 배포할 수 있습니다. 비공개 제안의 경우 구매자가 판매자에게 SCMP 템플릿을 요청할 수 있으며, 계약 당사자들이 합의한 대로 사용자 지정 거래 요구 사항을 충족하도록 조건을 수정할 수 있습니다.

셀프 서비스 또는 비공개 제안의 경우 SCMP와 함께 다음과 같은 선택적 부록을 사용할 수도 있습니다.

- 보안 강화 부록 - 데이터 보안 요구 사항이 강화된 거래를 지원합니다.
- HIPAA 비즈니스 제휴 부록 - Health Insurance Portability and Accountability Act of 1996(HIPAA) 규정 준수 요구 사항이 적용되는 거래를 지원합니다.
- 연방 부록 - 미국 정부와 관련된 소프트웨어 구매를 지원합니다.

> **면책 조항**
>
> EULA는 판매자와 구매자 간에 체결됩니다. SCMP를 EULA로 사용하는 것은 판매자의 재량입니다. 제품 목록에 SCMP를 적용하면 SCMP 프로그램에 참여하는 것으로 간주됩니다. 이 프로그램에서 AWS 는 SCMP 템플릿을 주기적으로 업데이트할 수 있으며 현재 버전의 용어가 포함된 제품 목록을 업데이트할 수 있습니다. 판매자는 언제든지 SCMP 템플릿을 판매자 고유의 EULA로 대체하여 SCMP 프로그램에서 탈퇴할 수 있습니다.

SCMP 시작하기

EULA를 SCMP로 업데이트하고 신규 및 기존 단일 Amazon Machine Image(AMI) 및 서비스형 소프트웨어(SaaS) 제품 구매자에게 제공할 수 있습니다. 업데이트를 요청하는 절차는 제품이 셀프 서비스 목록(SSL)을 통해 리스팅되었는지 여부에 따라 다릅니다.

질문이 있는 경우의 AWS Marketplace 표준 계약 팀에 이메일 메시지를 보냅니다aws-mp-standardcontract@amazon.com.

SSL을 통해 등록된 AMI 및 SaaS 제품의 EULA를 SCMP로 업데이트하는 방법

1. 표준 계약 AWS Marketplace 조건을 검토합니다.
2. AWS Marketplace Management Portal에 로그인합니다.
3. 제품에서 제품 유형을 선택하고, 계약을 업데이트하려는 제품 목록을 선택합니다.
4. 변경 요청을 선택하고 리전 및 요금 업데이트를 선택합니다.
5. EULA가 아직 선택되지 않은 경우 표준 계약을 AWS Marketplace 선택합니다.
6. 메모 및 알림에 **`Please update this product to SCMP V2`**를 입력합니다.
7. 변경 사항을 검토하고, 검토를 위해 제출을 선택합니다.

SSL을 통해 등록되지 않은 AMI 및 SaaS 제품의 EULA를 SCMP로 업데이트하는 방법

1. AWS Marketplace Management Portal에서 AWS에 문의 페이지를 엽니다.
2. AWS Marketplace 판매자 계정에 로그인합니다.
3. 다음과 같이 양식을 작성합니다.
   - 질문에 대한 주제에서 Commercial Marketplace(상업용 마켓플레이스)를 선택합니다.
   - 범주에서 Product Listing(제품 리스팅)을 선택합니다.
   - 하위 범주에서 Standard Contract Request(표준 계약 요청)을 선택합니다.
   - 설명에 **`Please update these products to SCMP V2`**를 입력하고 SCMP로 업데이트하려는 제품 제목과 ID를 입력합니다.

에 대한 리셀러 계약 AWS Marketplace

리셀러 계약 AWS Marketplace (RCMP)은 채널 파트너가 ISVs 제품을 AWS Marketplace 구매자에게 재판매하도록 승인할 때 ISV가 사용할 수 있는 표준화된 리셀러 계약 템플릿입니다. 이 계약은 ISV

와 채널 파트너가 리셀러 관계를 맺거나 채널 파트너 비공개 제안(CPPO)에 리셀러 조건을 사용할 때 법적 계약 검토의 중복을 줄이고 출시 시간을 단축하는 데 도움이 됩니다. CPPO를 생성할 때 ISVs 계약을 기회에 업로드(재판매 승인)한 다음 채널 파트너는 계약을 보고 수락할 수 있습니다. AWS Marketplace 구매자는 RCMP를 볼 수 없습니다.

> **면책 조항**
>
> RCMP는 ISV를 위한 선택적 계약입니다. ISV는 채널 파트너를 통해 제품을 재판매하기로 결정하는 경우 영업 기회를 생성할 때 RCMP 또는 자체적인 (기존 또는 사전 협상된) 맞춤형 계약 조건을 첨부할 수 있습니다.

RCMP 시작하기

이 섹션에서는 조건을 검토하고 RCMP를 사용하는 방법을 설명합니다.

AWS Marketplace 기회를 생성하는 동안 RCMP를 사용하려면

1. 리셀러 계약 AWS Marketplace 조건을 검토합니다.
2. RCMP 가이드를 참조하면서 채널 파트너를 위한 재판매 기회 만들기의 단계를 따릅니다.

AWS Marketplace 제품에 대한 메타데이터 제공

판매자는 제품을 추가할 때 제품 메타데이터를 AWS Marketplace지정합니다. 제품 메타데이터에는 고객을 위한 제품을 설명하는 데 사용하는 이름, 설명, 범주 및 키워드가 포함됩니다.는 품질 보증 및 오류 수정을 위해서만 제품 메타데이터를 AWS Marketplace 수정합니다. 이 주제에서는 AWS Marketplace 판매자가 제품의 이름을 효과적으로 지정하고, 설명하고, 분류하는 데 사용할 수 있는 정보와 모범 사례를 제공합니다.

> **Note**
>
> 요금, 사용량, 유사 제품 또는 제품 비교와 같은 모든 제품 세부 정보 페이지 탭 및 섹션은 동적이며에서 관리합니다 AWS Marketplace. 판매자는 섹션을 추가하거나 제거하도록 제품 세부 정보 페이지를 사용자 지정할 수 없습니다.

제품 이름 지정 및 설명

귀사 제품에 대해 제공한 정보는 구매자에게 표시됩니다. 잠재적인 구매자가 제품 구매 시 정보에 기반한 결정을 내릴 수 있는 충분한 정보를 확보할 수 있게 해야 합니다.

제품 이름 생성

제품 이름을 만들 때 다음 지침을 유념하세요:

- 단어의 첫 글자를 대문자로(각 중요 단어의 첫 번째 글자는 대문자로)
- 구매자가 이름만으로 제품을 식별할 수 있어야 함
- 브랜드 또는 제조업체의 이름 사용
- 서술적 데이터 또는 과장법이 포함되지 않아야 함

제품 이름 예: Smart Solution Load Balancer - Premium Edition

지원되는 문자

메타데이터 필드에 지원되는 문자는 다음과 같습니다.

- ASCII 문자 코드 0~126
  - 자세한 내용은 표준 ASCII 테이블을 참조하세요.
- 저작권 ©
- 등록된 ®
- 상표 ™
- 통화 기호 " £ " ¥

제품 설명 작성

제품 설명에는 제품의 기능, 장점 및 사용법을 나열합니다. 또한 관련 있는 특정 제품 정보를 제공할 수 있습니다. 설명의 길이는 최대 350자입니다.

제품 설명을 쓸 때 다음 지침을 유념하세요.

- 불필요한 대문자 자제
- 불필요한 구두점 자제

- 리디렉션 정보가 포함되지 않아야 함
- 맞춤법 및 문법 확인
- 중요하고 유용한 정보만 포함
- 서술적 데이터와 과장을 피할 것

제품 솔루션 예: 스마트 솔루션은 수신되는 애플리케이션 트래픽을 여러 Amazon EC2 인스턴스에 자동으로 분산합니다. 따라서 애플리케이션의 내결함성을 크게 높이고, 수신되는 애플리케이션 트래픽에 응답하는 데 필요한 로드 밸런싱 용량을 제공할 수 있습니다. 스마트 솔루션은 풀 내에서 비정상 인스턴스를 검색하고 비정상 인스턴스가 복원될 때까지 자동으로 트래픽을 정상 인스턴스로 재라우팅합니다. 단일 AWS 가용 영역 또는 여러 가용 영역에서 스마트 솔루션을 활성화하여 보다 일관된 애플리케이션 성능을 보장할 수 있습니다.

제품 주요 내용 작성

제품 정보 페이지에는 최대 3개의 제품 주요 내용 글머리표가 표시됩니다. 이러한 주요 내용 글머리표를 사용하여 제품의 주요 장점을 간략하게 설명할 수 있습니다.

제품 주요 내용 예: 비용 예측 - 스마트 솔루션은 사용한 만큼만 지불하면 됩니다. 스마트 솔루션을 실행하는 시간별로 또는 부분 시간에 대해 요금이 청구됩니다.

지원 정보 작성

고객은 서비스 사용, 문제 해결 및 환불 요청(해당하는 경우)과 같은 문제에 대한 도움을 쉽게 받을 수 있어야 합니다. 이행 랜딩 페이지에서 이메일, 전화번호 또는 지원 웹 양식 링크와 같은 지원 연락처 옵션을 지정해야 합니다.

범주 및 키워드 선택

제품을 나열할 때 제품에 대한 최대 3개의 소프트웨어 범주와 해당하는 하위 범주를 선택할 수 있습니다. 이렇게 하면 구매자가 제품을 검색하거나 검색할 때 제품을 검색할 수 있습니다 AWS Marketplace. 귀사의 제품과 관련된 범주만 선택합니다. 대부분의 경우 하나의 범주만 적용됩니다. 제품 로드 양식과 제품 탭 모두 범주의 전체 목록이 포함되어 있습니다.

범주는 키워드와 동일하지 않습니다. 사용 가능한 범주 및 하위 범주는 미리 정의되어 있으며 AWS Marketplace제품 요청 프로세스 중에 목록에서 선택하여 제품에 적용할 범주 및 하위 범주를 결정합니다. 키워드는 사전 정의되어 있지 않지만, 프로세스 도중 생성됩니다. 범주를 키워드로 추가할 필요가 없습니다.

검색 키워드 생성

제품 요청 프로세스 중 구매자가 사이트 검색을 통해 제품을 검색할 수 있도록 최대 3개의 키워드(한 단어 또는 구)를 입력할 수 있습니다. 키워드 필드에는 최대 250자가 포함될 수 있습니다.

다음 팁은 관련된 검색 키워드 집합을 생성하는 데 도움이 될 수 있습니다:

- 관련 약관을 사용합니다.
- 다른 판매자가 게시한 제품의 이름 또는 다른 판매자의 이름이 포함되어서는 안 됩니다.
- 구매자의 어휘에서 키워드를 선택합니다. 즉, 구매자가 이 제품 유형에 대해 생각할 때 사용할 것 같은 단어와 구를 선택합니다.
- 제품의 특정 기능을 기준으로 키워드를 생성합니다.
- 제품 제목을 키워드로 사용하지 않습니다. 제품 제목은 이미 검색에서 인덱싱됩니다.

> **Note**
>
> 키워드는 소프트웨어 범주와 동일하지 않습니다. 키워드는 제품과 관련된 더 구체적인 용어입니다.

에 대한 AMI 및 컨테이너 제품 사용 지침 생성 AWS Marketplace

AWS Marketplace 판매자는 다양한 방식으로 구매자에게 제품을 제공합니다. Amazon Machine Image(AMIs) 및 컨테이너 이미지와 함께 제공할 때 해당 이미지에 대한 사용 지침을 작성하는 것은 사용자의 책임입니다. 다음 섹션에서는 이러한 제품 유형에 대한 사용 지침 작성에 대한 요구 사항 및 권장 사항을 제공합니다.

자세한 내용은 의 AMI 기반 제품 AWS Marketplace 단원을 참조하십시오.

주제

- 요구 사항
- 릴리스 정보 작성
- 사용 지침 작성
- 업그레이드 지침 작성
- CloudFormation 전송 지침 작성

요구 사항

제품 사용 지침을 생성할 때, 다음 정보를 포함해야 합니다.

- 고객이 저장한 모든 민감한 정보의 위치입니다.
- 모든 데이터 암호화 구성을 설명합니다. 애플리케이션이 암호화 기술을 사용하는 경우 사용자가 애플리케이션과 상호 작용하여 필요한 데이터를 해독하는 방법에 대한 자세한 지침을 제공합니다.
- 제품에 암호화 구성 요소가 포함된 경우 사용 지침에 교체 요구 사항을 포함해야 합니다. 자격 증명 및 암호화 키를 사용하는 목록에 대한 the section called “AMI 기반 제품 요구 사항” 기본 요구 사항은 섹션을 참조하세요.
- 제품의 데이터 스토어가 독점적인 경우 구성, 백업 및 복구에 대한 step-by-step 지침을 제공합니다.
- 애플리케이션의 상태와 적절한 기능을 평가하고 모니터링하는 방법에 대한 단계별 지침. 예시:
  - Amazon EC2 콘솔로 이동하여 올바른 리전에 있는지 확인합니다.
  - 인스턴스를 선택하고, 시작된 인스턴스를 선택합니다.
  - 메타데이터 페이지를 표시할 서버를 선택하고, 페이지 하단에서 상태 확인 탭을 선택하여 상태 확인을 통과했는지 아니면 실패했는지 검토합니다.
- AWS 서비스 할당량 관리에 대한 권장 지침입니다. 자세한 내용은 AWS 일반 참조 가이드를 참조하세요.
- 표준 할당량 이상으로 추가된 AWS 리소스 실행 비용을 포함한 요금 분류입니다. 이 정보를 제품 사용 지침에 포함해도 되고, 서비스 할당량 증가 관리 및 요청에 대한 자세한 정보가 포함된 문서에 링크해도 됩니다.

릴리스 정보 작성

제품을 업데이트할 때마다 변경 사항에 대한 설명을 출시 정보에 제공해야 합니다. 출시 정보에는 사용자가 업데이트를 설치할지 여부를 결정하는 데 도움이 되는 특정 정보가 포함되어야 합니다. 업데이트에 대한 명확한 레이블을 사용합니다. 예를 들어, 보안 업데이트의 경우 "심각"을, 기타 업데이트 유형의 경우 "중요" 또는 "선택 사항"을 사용합니다.

사용 지침 작성

구매자가 소프트웨어를 성공적으로 구성하고 실행할 수 있도록 사용 지침을 제공합니다. 제공한 사용 지침은 구성 프로세스 중에 표시됩니다.

효과적인 사용 지침을 작성하려면 다음 지침을 따르세요.

- 사용자가 관심이 있지만 정보가 없다고 가정합니다.
- 사용자에게 구성 설정 및 특별 단계를 포함하여 제품을 시작하고 사용하는 데 필요한 모든 정보를 제공합니다.

사용 지침의 예:

1. 1-Click을 사용하여 제품을 시작합니다.
2. 웹 브라우저를 사용하여에서 애플리케이션에 액세스합니다`https://<EC2_Instance_Public_DNS>/index.html`.
3. 다음 자격 증명을 사용하여 로그인합니다.
   - 사용자 이름: `user`
   - 암호: 인스턴스 ID(`instance_id`)

업그레이드 지침 작성

구매자가 이전 버전의 제품에서 업그레이드할 수 있는 방법에 대한 자세한 정보를 제공합니다. 다른 인스턴스 생성 시 데이터 및 설정을 보호하는 방법에 대한 정보를 포함합니다. 업그레이드 경로가 없는 경우 이 필드를 편집하여 업그레이드 경로를 자세히 설명합니다.

업그레이드 지침의 예:

1. ****을(를) 수행한 다음 ****을(를) 수행합니다.
2. ***를 실행하여 프로젝트에서 사용하는 모든 플러그인이 버전 *.*와 호환되는지 확인합니다. 호환되지 않는 경우 ***을(를) 수행합니다.
3. ***을(를) 수행하여 데이터 백업을 만듭니다.

CloudFormation 전송 지침 작성

CloudFormation 제공 방법을 사용하는 경우 다음 정보도 포함해야 합니다.

- 템플릿에서 생성한 AWS CloudFormation 각 AWS Identity and Access Management (IAM) 역할 및 IAM 정책의 목적
- AWS CloudFormation 템플릿에서 생성한 각 키의 용도 및 위치
- 두 가지 이상의 요소가 포함된 배포의 네트워크 구성 세부 정보

- 애플리케이션이 시작되는 방법과 배포에 여러 AWS 리소스가 포함된 경우 애플리케이션이 통신하도록 구성된 방법에 대한 자세한 가이드
- 표준 한도를 초과하여 추가된 AWS 리소스 실행 비용을 포함하는 요금 분류입니다. AWS 서비스 제한 관리에 대한 규범적 지침을 제공합니다.
- 모든 데이터 암호화 구성. 예: Amazon S3 서버 측 암호화, Amazon Elastic Block Store(Amazon EBS) 암호화, Linux Unified Key Setup(LUKS) 등

검색을 위한 AWS Marketplace 제품 최적화

AWS Marketplace 판매자는 구매자가 제품을 더 쉽게 찾을 수 있도록 검색에 맞게 제품을 최적화할 수 있습니다. 검색은 구매자 여정의 중요한 도구로, 고객이 고유한 요구 사항에 맞는 최상의 제품을 찾을 수 있도록 해줍니다. AWS Marketplace 고객의 경우 검색은 검색 엔진(예: Google 또는 Bing)과 검색 함수의 두 가지 기본 위치에서 수행됩니다 AWS Marketplace. 이 주제에서는 두 위치에 대한 리스팅을 최적화하는 방법에 대한 정보를 제공합니다.

주제

- 검색 엔진 최적화
- AWS Marketplace 검색

검색 엔진 최적화

검색 엔진을 통해 제품 세부 정보 페이지를 보다 고유하게 방문하도록 유도하려면 관련 키워드에 대해 제품 세부 정보 페이지의 순위를 높게 지정하는 것이 중요합니다.

제품 세부 정보 페이지, 키워드, 제목 태그, H1 제목 태그를 포함하여 모든 웹 페이지에 대한 오가닉 검색을 개선하는 데 효과적인 세 가지 기본 페이지 요소가 있습니다.

키워드

키워드는 주제를 검색 엔진 결과를 유도하는 집중 쿼리로 추출하므로 검색 엔진 최적화의 핵심 요소입니다. 페이지에 가장 관련성이 높은 키워드를 식별하는 프로세스에는 키워드 조사가 포함됩니다. 검색 엔진 최적화 도구는 키워드 검색 볼륨(Google에서 키워드를 매월 검색하는 횟수), 현재 순위, 검색 추세, 키워드 경쟁력 및 관련 키워드와 같은 중요한 정보를 제공할 수 있습니다. 이 조사에서 기본 및 보조 키워드를 식별할 수 있습니다.

기본 검색 엔진 최적화 키워드는 페이지의 주요 주제를 나타내는 고유한 단일 단어 또는 구문이어야 합니다. 이 기본 키워드는 제품 제목, 간단한 설명 및 강조 표시 섹션의 사본에 자연스럽게 포함되어야 합니다. 보조 키워드는 나머지 페이지 콘텐츠에서 찾을 수 있고 관련성이 높은 용어여야 합니다.

제목 태그

제목 태그는 검색 엔진 결과 페이지에 나타나고 브라우저 창 또는 탭에 있는 페이지의 제목으로 표시되며, 리더와 검색 엔진 웹 크롤러에 페이지의 내용을 알려줍니다. AWS Marketplace 제품 세부 정보 페이지의 경우 제품 제목이 제목 태그 역할을 하므로 검색 엔진 최적화 키워드로 제품 제목을 최적화하여 순위 가능성을 개선하는 것이 중요합니다. 검색 엔진 결과 페이지에서 높은 순위를 달성할 가능성을 높이려면 브랜드 이름, 제품 이름 및 관련 키워드를 제목 태그에 통합합니다.

H1 제목 태그

H1 제목 태그에는 세 가지 역할이 있습니다.

- 방문자가 필요한 정보를 찾기 위해 페이지 콘텐츠를 스캔하는 데 도움이 됩니다.
- 시각적 장애가 있는 방문자가 화면 리더를 사용하여 페이지의 콘텐츠를 이해하도록 하여 접근성을 높입니다.
- 페이지 제목에 있는 키워드를 제공하며, 이어지는 페이지 콘텐츠가 이를 뒷받침할 경우 검색 엔진 최적화 관련성 가중치를 추가로 받습니다.

AWS Marketplace 검색

이 AWS Marketplace 웹 사이트는 업계 전체에서 사용되는 것과 유사한 검색 최적화 기법을 사용하여 검색 쿼리의 결과를 순위로 매깁니다. 가 검색 결과의 AWS Marketplace 순위를 매기고 검색 결과를 반환하는 방법을 이해하면 AWS Marketplace 검색 엔진에 최적화된 제품 세부 정보를 생성할 수 있습니다. 제품 세부 정보 페이지 생성 시 이 지침을 고려하는 것이 좋습니다.

키워드

제품 생성 프로세스 중 고객이 사이트 검색을 통해 제품을 검색할 수 있도록 최대 3개의 키워드(한 단어 또는 구)를 제출할 수 있습니다. 키워드 텍스트 상자에는 최대 250자가 들어갈 수 있습니다.

다음 팁을 따라 검색 키워드를 생성합니다.

- 고객이 제품을 쉽게 찾을 수 있도록 관련된 용어를 사용하세요.
- 고객의 어휘에서 키워드를 선택합니다. 즉, 고객이 제품 유형에 대해 생각할 때 사용할 것 같은 단어와 구를 선택합니다.

- 제품의 특정 기능을 기준으로 키워드를 생성합니다.
- 제출하는 용어에 제품 제목을 포함시키지 마세요. 제품 제목은 이미 검색에서 인덱싱됩니다.

> **Note**
> 키워드는 소프트웨어 범주와 동일하지 않습니다. 키워드는 제품과 관련된 더 구체적인 용어입니다.

제품을 생성한 후 제품의 메타데이터를 편집하여 키워드를 편집할 수 있습니다. AWS Marketplace Management Portal에서 제품 탭을 사용하여 생성한 제품의 경우 제품 탭을 사용하여 변경할 수도 있습니다. 자세한 내용은 Product changes and updates 섹션을 참조하세요.

AWS Marketplace 판매자 운영 팀은 비슷한 의미의 단어 또는 비슷한 의미의 단어로 쿼리를 리디렉션하는 데 도움을 줍니다. 판매자는 고객이 차를 검색할 것으로 예상하는데 고객이 자동차를 검색하는 경우를 예로 들 수 있습니다.

소프트웨어 범주

제품을 나열할 때 제품에 대한 최대 3개의 소프트웨어 범주와 해당하는 하위 범주를 선택할 수 있습니다. 이를 통해 고객은 제품을 검색하거나 검색할 때 제품을 검색할 수 있습니다 AWS Marketplace. 제품과 관련된 범주만 선택하세요. 대부분의 경우 범주 하나만 적용됩니다. 제품 로드 양식과 제품 페이지 모두 범주의 전체 목록이 포함되어 있습니다.

> **Note**
> 범주는 키워드와 동일하지 않습니다. 사용 가능한 범주 및 하위 범주는 미리 정의되어 있습니다 AWS Marketplace. 목록에서 선택하여 제품에 어떤 범주를 적용할지 결정합니다. 키워드는 사전 정의되어 있지 않지만, 프로세스 도중에 생성됩니다.

주요 내용 섹션

제품 세부 정보 페이지에는 최대 3개의 제품 주요 내용 글머리표가 표시됩니다. 고객은 주요 내용으로 제품을 검색할 수 있습니다. 따라서 제품 생성 시 주요 내용을 포함시키세요. 주요 내용은 제품의 주요 판매 요소를 간략하고, 유익한 언어로 설명해야 합니다.

Example 하이라이트

- 예상 비용: AnyCompany의 제품은 사용한 만큼만 지불하면 됩니다. 제품이 실행되는 시간당 요금 또는 부분 시간에 대해 요금이 청구됩니다.

간단한 설명

제품 설명은 제품의 기능, 이점, 사용 지침을 나열하며 기타 관련 및 특정 제품 정보를 제공합니다. 제품 설명을 작성할 때 다음 지침을 유념하세요:

- 불필요한 대문자 및 구두점 자제
- 리디렉션 정보가 포함되지 않아야 함
- 맞춤법 및 문법 확인
- 중요하고 유용한 정보만 포함

Example 간단한 설명

AnyCompany의 제품은 수신되는 애플리케이션 트래픽을 여러 Amazon EC2 인스턴스에 자동으로 분산합니다. 따라서 애플리케이션의 내결함성을 개선하고, 수신되는 애플리케이션 트래픽에 응답하는 데 필요한 로드 밸런싱 용량을 원활하게 제공할 수 있습니다. AnyCompany의 제품은 풀 내에서 비정상 인스턴스를 검색하고 비정상 인스턴스가 복원될 때까지 자동으로 트래픽을 정상 인스턴스로 재라우팅합니다. 고객은 단일 AWS 가용 영역 또는 여러 가용 영역에서 이를 활성화하여 보다 일관된 애플리케이션 성능을 제공할 수 있습니다.

프로모션 미디어로 AWS Marketplace 제품 개선

AWS Marketplace 판매자는 프로모션 미디어로 제품 목록을 개선하여 고객이 제품을 더 잘 찾고 평가할 수 있도록 도울 수 있습니다. 프로모션 미디어는 제품 페이지에 눈에 띄게 표시되는 비디오 및 이미지로, 고객에게 제품에 대해 쉽게 알아볼 수 있는 방법을 제공합니다. 다음 섹션에서는 프로모션 미디어에 대한 모범 사례와 ()에서 프로모션 미디어를 AWS Marketplace Management Portal 추가하고 관리하기 위한 팁을 제공합니다AMMP.

주제

- 홍보 미디어 모범 사례
- 프로모션 미디어 추가 및 관리 팁

홍보 미디어 모범 사례

최대 5개의 비디오와 10개의 이미지가 지원됩니다. 각 홍보 미디어 항목에는 필수 제목과 선택적 설명이 포함되어야 합니다. 설명은 미디어의 대체 텍스트로 사용되며 시각적 접근성과 검색 엔진 최적화를 개선하는 것이 좋습니다.

> **Note**
>
> 프로모션 미디어를 업로드하거나 추가하려면 적절한 권한과 권한이 있어야 합니다. 제품에 추가된 미디어는 모든 사용자가 공개적으로 사용할 수 있습니다 AWS Marketplace.

비디오

이 자산 유형은 간결한 개요를 통해 제품과 회사를 소개할 수 있는 기회를 제공합니다. 고객 인터뷰, 인용문, 관련 이점 및 데이터 포인트와 같은 콘텐츠를 포함할 수도 있습니다. 또한 비디오는 특히 주요 기능 또는 사용 사례에 대해 녹화된 데모 또는 연습을 통해 제품을 보여주는 좋은 방법입니다. 짧은 비디오는 더 큰 영향을 미치므로 비디오의 지속 시간은 2~5분인 것이 좋습니다.

사양:

- 공개적으로 액세스할 수 있는 S3 링크를 통해 비디오를 직접 업로드AMMP할 수 있습니다.
- 권장 비디오 해상도는 1080p(1920x1080픽셀)이며 선호하는 종횡비는 16:9입니다.
- 비디오 형식은 최대 파일 크기가 50MB인 .mp4여야 합니다.
- (선택 사항) 자동 생성된 커버 대신 커버 이미지를 추가할 수 있습니다.
  - 커버 이미지의 해상도는 500x281픽셀(16:9 종횡비)이어야 합니다.
  - 이미지 형식은 투명도 없이 .png(기본 설정), .jpg 또는 .svg여야 합니다.

> **Note**
>
> 외부에서 호스팅되는 비디오는 직접 지원되지 않습니다. 퍼블릭 S3 링크는 추가 중에에서 AWS Marketplace 캐시되며 추가 후 변경 사항은 미디어가 다시 추가될 때까지 반영되지 않습니다.

YouTube 비디오는 제한된 기능으로 지원됩니다.

- 제품 비디오 URL 필드(프로모션 미디어와 별도의 필드)에 YouTube 비디오에 대한 직접 링크가 포함된 경우 비디오에 임베딩이 활성화된 경우 비디오가 마지막 프로모션 미디어 항목으로 직접 임베딩됩니다.
- 제품 비디오 URL은 하나만 지원되며(예: YouTube 비디오 하나만 지원) YouTube 임베디드 미디어의 재정렬은 지원되지 않습니다.

이미지

이 자산 유형은 스크린샷과 주요 개념 콜아웃을 통해 제품을 홍보할 수 있는 기회를 제공합니다. 이미지는 간결한 다이어그램 또는 흐름도를 통해 복잡한 제품 기능을 설명하는 데 적합합니다. 또한 이미지를 사용하여 요금 차원 설명으로 해결할 수 없는 요금 계층 및 그룹을 설명할 수 있습니다.

사양:

- 공개적으로 액세스할 수 있는 S3 링크를 통해 이미지를 직접 업로드AMMP할 수 있습니다.
- 권장 이미지 해상도는 780x439픽셀이며 선호하는 종횡비는 16:9입니다. 최대 3480x3480픽셀.
- 이미지 형식은 투명도 없이 .png(기본 설정), .jpg 또는 .svg여야 합니다.
- 썸네일은 원본 소스 이미지에서 자동으로 생성됩니다.

프로모션 미디어 추가 및 관리 팁

미디어 추가

- 프로모션 미디어는에 로그인AWS Marketplace Management Portal하고 새 제품을 생성하거나 기존 제품을 업데이트하여 추가할 수 있습니다. 이미지 및 비디오 섹션은 프로모션 미디어를 추가하는 데 사용되며 제품 정보 영역 내에 있습니다.
- 를 사용하여 프로모션 미디어를 추가할 API 수도 있지만 직접 업로드는 AMMP를 통해서만 지원됩니다. 퍼블릭 S3 링크는 API 및 모두에서 지원됩니다AMMP.
- 프로모션 미디어를 표시할 순서대로 추가합니다. 첫 번째 항목은 제품 페이지의 추천 미디어입니다.
- 이미지 자산 유형을 사용하는 경우 선택적 커버 이미지를 사용할 수 없습니다. 이미지는 추가되면 썸네일이 자동으로 생성됩니다.

미디어 업데이트 또는 제거

- 프로모션 미디어가 제품에 추가되면 제품을 편집하여 필요에 따라 제목, 설명 및 커버(사용 가능한 경우)를 조정할 수 있습니다.

- 제출 전에 AMMP에서 해당 자산에 대한 삭제를 클릭하여 프로모션 미디어를 추가한 후 제거할 수 있습니다.

미디어 주문

- 기본적으로 새 프로모션 미디어 자산은 자산 주문 끝에 추가됩니다.
- 미디어를 재정렬하려면 순서가 맞지 않는 자산을 삭제하고 올바른 순서로 다시 추가합니다.

AWS Marketplace 제품에 대한 비공개 제안 준비

비공개 제안은 제품을 구매하는 데 사용되는 협상된 용어입니다 AWS Marketplace. 여기에는 사용자 지정 요금제, 최종 사용자 라이선스 계약(EULA) 또는 사용자 지정 솔루션이 포함될 수 있습니다. 판매자와 구매자는 공개 제안과 다른 비공개 제안을 하기 전에 협상합니다. 비공개 제안을 여러 개 생성하여 단일 구매자에게 제공할 수 있습니다. 비공개 제안을 받은 구매자는 비공개 제안과 공개 제안 중에서 선택할 수 있습니다. 구매자는 특정 시점에 하나의 제안만 구독할 수 있습니다. 비공개 제안과 공개 제안을 동시에 구독할 수는 없습니다. 이 주제에서는 특별 고려 사항, 구매자 경험 및 판매자 보고서를 포함하여 비공개 제안의 작동 방식에 대한 정보를 제공합니다.

> **Note**
>
> AWS 는 제품 세부 정보 페이지에서 직접 제품에 대한 비공개 제안을 요청할 AWS Marketplace 수 있도록에 고유한 사용 사례 또는 엔터프라이즈 사용 사례가 있는 구매자에게 제공합니다. APN Customer Engagements(ACE)를 사용할 수 있는 AWS 파트너 네트워크(APN) 파트너이고 구매자에게이 옵션을 제공하려는 경우 자세한 내용은 비공개 제안 및 데모 요청 버튼 추가 섹션을 참조하세요. https://aws.amazon.com/partners/programs/ace/

주제

- 비공개 제안 작동 방식
- 비공개 제안 고려 사항
- 구매자를 위한 비공개 제안 경험
- 비공개 제안 보고
- AWS Marketplace 비공개 제안에 지원되는 제품 유형
- 비공개 제안 만들기 및 관리
- AWS Marketplace 채널 파트너로서 비공개 제안 생성
- 비공개 제안 할부 플랜
- 비공개 제안에 대한 미래 날짜의 계약 만들기
- 비공개 제안 FAQ

비공개 제안 작동 방식

AWS Marketplace Management Portal의 제안 페이지에서 비공개 제안을 만들고 관리할 수 있습니다. 고유한 ID 및 URL을 생성하기 위해 제안의 제품을 지정합니다. 비공개 제안에 대한 요금제를 만들고, 법적 조건과 판매 문서를 추가하고, 특정 구매자 AWS 계정에 제안을 확장합니다. 제안은 해당 제안을 만든 계정에만 표시됩니다.

비공개 제안을 만들고 잠재적 구매자에게 알리면 잠재적 구매자는 비공개 제안을 살펴보고 수락할 수 있습니다. 제안을 보려면 구매자는 제안을 받은 AWS 계정 에 로그인해야 합니다.

> **Note**
>
> 해당 제안을 연결 계정 또는 관리 계정에 제시하지 않으면 구매자가 제안을 볼 수 없습니다. 제안에 서비스 제한을 제공할 수 없습니다. 따라서 구매자는 제품에서 제한이 설정되지 않은 한 협상된 가격으로 원하는 만큼 제품을 사용할 수 있습니다.

비공개 제안 생성에 대한 자세한 내용은 Creating and managing private offers를 참조하세요.

비공개 제안은 판매자 보고서에서 추적됩니다. 자세한 내용은 Reporting for private offers 및 Seller reports guide를 참조하세요.

비공개 제안 고려 사항

비공개 제안 작업 시 다음 사항을 고려합니다.

- 새 인스턴스 유형에 대한 지원을 추가하거나 제품에 대한 비공개 제안을 이미 구독한 AWS 리전고객은 새로 추가된 인스턴스 또는 리전에 자동으로 액세스할 수 없습니다. 고객이 액세스하려는 인스턴스 및 리전을 사용하여 다른 비공개 제안을 만들어야 합니다. 고객은 새 제안을 수락한 후 새로 추가된 인스턴스 및 리전에 액세스할 수 있습니다. 이러한 인스턴트와 리전이 비공개 제안에 포함되어 있는 한 향후 제품을 구독하는 고객은 여기에 액세스할 수 있습니다. 새 비공개 제안을 생성하는 방법에 대한 자세한 내용은 섹션을 참조하세요에서 계약 수정 AWS Marketplace.
- AWS CloudFormation 제품을 사용하거나 고객 사용을 제한하기 위해 타사, Amazon Machine Image(AMI) 월간 또는 multi-AMI-based 전송에 대한 비공개 제안을 생성할 수 없습니다.
- 할부 플랜이 포함된 비공개 제안의 경우 선결제를 여러 번의 결제로 나눌 수 있습니다. 자세한 내용은 비공개 제안 할부 플랜 단원을 참조하십시오.
- 비공개 제안의 구매자 계정이 비공개 마켓플레이스를 통해 관리되는 경우 구매자의 계정과 구매자의 비공개 마켓플레이스 관리자가 포함된 계정을 모두 제안에 포함해야 합니다.

- 비공개 제안은 Bring Your Own License(BYOL) 모델을 지원하지 않습니다.
- 비공개 제안을 만들 때 사용자 지정 EULA 옵션을 사용하여 협상된 고유의 계약 조건을 비공개 제안에 포함하세요. 최대 5개 문서를 첨부할 수 있습니다.
- 서비스형 소프트웨어(SaaS) 계약과 소비 제품이 포함된 SaaS 계약의 경우 판매자는 구매자가 비공개 제안을 수락했을 때 체결된 계약에 대한 업그레이드 및 갱신을 제공할 수 있습니다. 예를 들어 새로운 권한을 부여하거나, 요금 할인을 적용하거나, 결제 일정을 조정하거나, 표준화된 라이선스 조건을 사용하도록 최종 사용자 라이선스 계약(EULA)을 변경하기 위해 이렇게 할 수 있습니다. 자세한 내용은 에서 계약 수정 AWS Marketplace 단원을 참조하십시오.

구매자를 위한 비공개 제안 경험

구매자가 제품의 구독 페이지로 이동하면 배너에 비공개 제안을 사용할 수 있음이 표시됩니다. 구매자가 제안을 수락하면 모든 AWS Marketplace 거래에 사용된 것과 동일한 포털 도구를 사용하여 구매에 대한 인보이스가 발행됩니다. 수락된 제안은 계약이 됩니다. 구매자는의 구독 관리 섹션에서 계약 세부 정보를 찾을 수 있으며 AWS Management Console, 판매자는의 계약 탭에서 세부 정보를 찾을 수 있습니다 AWS Marketplace Management Portal.

AWS Marketplace 구매자는 비공개 제안을 위해 타사 파이낸싱에 액세스할 수 있습니다. 자세한 내용은 이제에서 고객 파이낸싱을 참조 AWS Marketplace하세요.

> **Note**
> 제안은 만료 날짜 전까지만 수락할 수 있습니다. 제안이 만료되면 수락 및 만료된 제안 탭으로 이동됩니다.

비공개 제안을 살펴보고 수락하는 방법	구매자가 할 수 있는 일
AWS Marketplace 콘솔에서	AWS Marketplace 콘솔에서 비공개 제안으로 이동하여 사용 가능한 제안 탭에서 제안 ID를 선택합니다. 비공개 제안의 구매자 경험에 대한 자세한 내용은 AWS Marketplace 구매자 설명서의 비공개 제안을 참조하세요.
판매자가 제공한 링크 사용	판매자가 보낸 링크를 따라 비공개 제안에 직접 액세스합니다.

비공개 제안을 살펴보고 수락하는 방법	구매자가 할 수 있는 일
	자세한 내용은 구매자에게 비공개 제안 보내기를 참조하세요.
제품 페이지에서	해당 제품의 제품 페이지로 이동한 다음, 배너의 링크를 선택하여 비공개 제안을 봅니다. 비공개 제안의 구매자 경험에 대한 자세한 내용은 AWS Marketplace 구매자 설명서의 비공개 제안을 참조하세요.

비공개 제안 보고

비공개 제안은 기존 판매자 보고서 및 제안 관련 보고서에 표시됩니다. 월별 청구 수익 보고서는 매달 생성되며 제안 표시 여부 및 제안 ID 정보를 제공합니다. 구매자에게 보낼 인보이스가 생성되면 해당 청구 기간의 보고서에 표시됩니다. 자세한 내용은 판매자 대시보드를 참조하세요.

Offer ID 필드에는 비공개 제안에 대해 생성된 고유한 제안 ID가 있습니다. 보고 항목이 비공개 제안인 경우가 아닌 한 빈 칸입니다. Offer Visibility 필드는 보고 항목이 공개 또는 비공개인지 여부를 나타냅니다. 모든 비공개 제안의 경우 이 항목은 비공개로 표시됩니다.

AWS Marketplace 비공개 제안에 지원되는 제품 유형

AWS Marketplace 판매자는 비공개 제안을 사용하여 Amazon Machine Image(AMIs), 컨테이너, 전문 서비스, 기계 학습(ML), 서비스형 소프트웨어(SaaS) 제품 등의 제품 유형을 판매할 수 있습니다. 비공개 제안은 제품을 구매하는 데 사용되는 협상된 조건입니다 AWS Marketplace. 이 주제에서는 AMI, 컨테이너, SaaS, ML 및 전문 서비스 제품의 비공개 제안에 대한 정보를 제공합니다.

비공개 제안에 대한 자세한 내용은 AWS Marketplace 제품에 대한 비공개 제안 준비 섹션을 참조하세요.

주제

- AMI 제품에 대한 비공개 제안
- 컨테이너 제품에 대한 비공개 제안
- 전문 서비스 제품에 대한 비공개 제안
- SaaS 제품에 대한 비공개 제안
- ML 제품에 대한 비공개 제안

AMI 제품에 대한 비공개 제안

AMI 제품에 비공개 제안을 제공할 수 있습니다.

제안의 기간은 다음과 같이 원하는 기간으로 지정할 수 있습니다.

- AMI 시간당 또는 AMI 연간 시간당 비공개 제안: 최대 3년(1,095일). AMI 연간 시간당 비공개 제안 또는 AMI 계약만 유연한 결제 일정을 지원합니다.
- AMI 계약 비공개 제안: 최대 5년(60개월)

  AMI 계약의 경우 비공개 제안은 사용량을 모니터링하지 않습니다.

  구매자는 언제든지 수동으로 새 계약 수준으로 업그레이드할 수 있습니다. 그러나 계약 등급을 정의하고, 서비스 제한을 적용하고, 구매자에게 계약의 단위 수를 수동으로 업그레이드하도록 조언하는 것은 독립 소프트웨어 개발 판매 회사(ISV)의 몫입니다. 현재는 비계층형 요금 기반 계약만 업그레이드할 수 있습니다. 비공개 제안의 계약 기간은 공개 제품 목록과 일치하거나 월 단위로 사용자 지정(최대 60개월)할 수 있습니다.

라이선스 권한 부여는 구매자가 비공개 제안을 수락하는 날부터 시작됩니다.

유연한 결제 일정이 적용되는 AMI 비공개 제안의 경우 계약 기간 동안 계약서에 합의된 연간 인스턴스 유형의 수를 설정할 수 있습니다.

> **Note**
>
> 비공개 제안은 월간 청구 계약에 사용할 수 없습니다.

컨테이너 제품에 대한 비공개 제안

컨테이너 기반 제품 계약에 대한 비공개 제안 요금을 제공할 수 있습니다.

제안의 기간은 다음과 같이 원하는 기간으로 지정할 수 있습니다.

- 시간당 컨테이너 또는 시간당 요금과 장기 비공개 제안이 함께 적용되는 컨테이너 - 최대 3년(1,095일) 시간당 요금과 장기 비공개 제안이 함께 적용되는 컨테이너 또는 컨테이너 계약만 유연한 결제 일정을 지원합니다.
- AMI 계약 비공개 제안 - 최대 5년(60개월)

컨테이너 계약의 경우 비공개 제안은 사용량을 모니터링하지 않습니다. 비계층형 요금을 사용하는 경우에만 컨테이너 계약을 업그레이드할 수 있습니다.

구매자는 언제든지 수동으로 새 계약 수준으로 업그레이드할 수 있습니다. 그러나 계약 등급을 정의하고, 서비스 제한을 적용하고, 구매자에게 계약의 단위 수를 수동으로 업그레이드하도록 조언하는 것은 독립 소프트웨어 개발 판매 회사(ISV)입니다. 현재는 비계층형 요금 기반 계약만 업그레이드할 수 있습니다. 비공개 제안의 계약 기간은 공개 제품 목록과 일치하거나 월 단위로 사용자 지정(최대 60개월)할 수 있습니다.

라이선스 권한 부여는 구매자가 비공개 제안을 수락하는 날부터 시작됩니다. 유연한 결제 일정이 적용되는 컨테이너 비공개 제안의 경우 계약 기간 동안 계약서에 합의된 단위 수를 설정할 수 있습니다. 또한 구매자가 동일한 단위를 추가로 사용할 경우 적용할 사용자 지정 시간당 가격을 정의할 수 있습니다.

> **Note**
>
> 비공개 제안은 월간 청구 계약에 사용할 수 없습니다.

전문 서비스 제품에 대한 비공개 제안

모든 전문 서비스 제품에 대한 제안은 비공개 제안으로 이루어집니다. 자세한 내용은 비공개 제안 생성 단원을 참조하세요.

SaaS 제품에 대한 비공개 제안

서비스형 소프트웨어(SaaS) 비공개 제안 제품은 시기에 따라 특정 요금 등급의 요금 수준을 변경할 수 없습니다. 예를 들어 3개월 동안 제안의 요금을 시간당 0.80 USD로 청구하다가 3개월 이후부터 동일한 등급의 요금을 시간당 0.60 USD로 변경할 수 없습니다. SaaS 계약의 경우 비공개 제안은 사용량을 모니터링하지 않습니다.

구매자는 언제든지 수동으로 새 계약 수준으로 업그레이드할 수 있습니다. 그러나 계약 등급을 정의하고, 서비스 제한을 적용하고, 구매자에게 필요할 때 계약 등급을 수동으로 업그레이드하도록 조언하는 것은 독립 소프트웨어 개발 판매 회사(ISV)입니다. 비공개 제안의 계약 기간은 공개 제품 목록과 일치하거나 월 단위로 사용자 지정(최대 60개월)할 수 있습니다.

ML 제품에 대한 비공개 제안

기계 학습(ML) 비공개 제안 제품은 특정 구매자에게 공개적으로 표시된 가격과 다른 가격을 제공합니다. 비공개 제안에서 판매자와 구매자 간에 체결되는 조건 및 계약은 공개 제안 또는 기타 비공개 제안의 조건 및 계약과 다를 수 있습니다.

비공개 제안은 다음 방법 중 하나로 작동합니다.

- 시간당 - 비공개 제안의 시간당 요금은 공개적으로 표시되는 시간당 요금과 다를 수 있습니다. 이 시간당 요금은 기계 학습 제품에 대한 비공개 제안이 만료되지 않기 때문에 영구적입니다. 향후 가격 변경이 필요한 경우 구매자는 새로운 비공개 제안으로 전환해야 합니다. 제품의 기존 실행 인스턴스 또는 엔드포인트에는 승인된 새 제안에 설정된 시간당 요금이 자동으로 청구됩니다. 비공개 제안 내의 계약 구성 요소가 만료된 후 제품의 시간당 요금으로 설정해야 합니다. 이 시간당 요금을 0 USD로 설정하면 구매자가 소프트웨어 요금 없이 제품을 무기한 사용할 수 있습니다.
- 추론당 - 제품을 엔드포인트로 배포한 시점을 기준으로 추론 요금을 구성한 경우 비공개 제안의 추론 요금은 공개적으로 표시되는 추론 요금과 다를 수 있습니다.
- 계약 - 지정된 일수 동안 고정 선결제 요금이 적용되는 비공개 제안일 수 있습니다. 구매자는 전체 계약 기간 동안 인스턴스를 무제한으로 사용할 수 있습니다. 계약이 종료된 후에도 계속 실행되는 인스턴스에 대해 판매자가 비공개 제안에서 설정한 시간당 요금이 청구됩니다. 예를 들어 365일 동안 무제한으로 사용할 수 있는 고정 선결제 요금이 적용되는 계약을 생성할 수 있습니다. 비공개 제안의 시간당 요금도 설정할 수 있습니다. 구매자는 이 비공개 제안을 수락할 때 선결제 요금을 결제합니다. 계약이 종료되면 실행 중인 모든 인스턴스에 해당하는 시간당 요금이 청구됩니다. 무료 비공개 평가판을 제공하는 경우 무료 영구 라이선스를 제공하는 일이 없도면 무료 평가판 기간이 끝난 후에 적용되는 시간당 요금을 올바르게 설정해야 합니다.

비공개 제안을 여러 개 생성하여 단일 구매자에게 제공할 수 있습니다. 비공개 제안을 받은 구매자는 비공개 제안과 공개 제안 중에 선택할 수 있습니다. 구매자는 특정 시점에 하나의 제안만 구독할 수 있습니다. 비공개 제안과 공개 제안을 동시에 구독할 수는 없습니다.

SageMaker 제품의 특정 구매자에 대한 비공개 제안을 생성하려면 섹션을 참조하세요the section called “비공개 제안 생성”.

비공개 제안 만들기 및 관리

AWS Marketplace 판매자는 비공개 제안을 생성하고 관리할 수 있습니다. 비공개 제안은 제품을 구매하는 데 사용되는 협상된 용어입니다 AWS Marketplace. 여기에는 사용자 지정 요금제, 최종 사용자 라

이선스 계약(EULA) 또는 사용자 지정 솔루션이 포함될 수 있습니다. 다음 섹션에서는 비공개 제안을 만들고 관리하는 방법에 대해 설명합니다.

> **Note**
>
> 비공개 제안을 발행하려면 활성 공개 등록이 하나 이상 있어야 합니다. 공개 등록이 있고 비공개 제안 탭에 액세스할 수 없는 경우 IAM 권한을 참조하거나 지원팀에 문의하세요. AWS Marketplace

주제

- 새 비공개 제안 시작
- 제안 상태 이해
- 비공개 제안 초안 작성 및 게시
- 비공개 제안 및 데모 요청 버튼 추가
- 구매자에게 비공개 제안 보내기
- 비공개 제안 복제
- 제안 세부 정보 다운로드
- 비공개 제안 진행 상황 저장
- 비공개 제안 만료 업데이트
- 비공개 제안 취소

새 비공개 제안 시작

다음 프로세스를 사용하여 CreateOffer API 변경 요청을 사용하여 제안을 만들고 제안 ID를 생성합니다. 초안 상태의 빈 제안을 만듭니다.

새 비공개 제안을 만들기 시작하려면

1. AWS Marketplace Management Portal에 로그인하고 제안을 선택합니다.
2. 제안 페이지에서 제안 생성을 선택합니다.
3. 제안 생성 페이지에서 비공개 제안을 만들 제품 유형과 제품을 선택합니다. 처리에는 최대 30초가 걸립니다. 처리가 완료될 때까지 페이지를 닫거나 새로 고치지 마세요.

> **Note**
>
> 제안을 생성한 후에는 제품 유형과 제품을 변경할 수 없습니다. 제품 유형별 비공개 제안에 대한 자세한 내용은 지원되는 제품 유형 섹션을 참조하세요.
> AWS Marketplace 채널 파트너인 경우 재판매 권한 부여에서 자체 제품에 대한 제안을 생성할지 아니면 AWS Marketplace 채널 파트너 비공개 제안(CPPO)을 생성할지 선택합니다. CPPO인 경우 독립 소프트웨어 개발 판매 회사(ISV), 제품 및 권한을 선택합니다.

4. 계속하여 세부 정보 제공을 선택합니다. 비공개 제안을 계속 만들 수 있는 단계별 경험이 열립니다.

제안 상태 이해

제안에는 수명 주기에 따라 다음 세 가지 상태 중 하나가 있습니다.

- 초안 - 제안이 완료되지 않았으며 아직 준비 중입니다. 초안 상태의 비공개 제안에는 보존 일정이 적용되지 않습니다. 제안을 게시하고 구매자에게 확장하려면 필요한 모든 세부 정보를 작성하고 제출해야 합니다.
- 활성 - 제안이 게시되고 구매자에게 확장되었습니다. 제안이 만료되지 않았으므로 구매자는 제안을 구독할 수 있습니다.
- 만료됨 - 제안이 게시되고 구매자에게 확장되었습니다. 제안이 만료되었으므로 구매자는 제안을 구독할 수 없습니다. 만료 날짜를 업데이트하여 구매자에게 제안을 수락할 시간을 더 많이 제공할 수 있습니다. 제안 만료를 업데이트하려면 비공개 제안 만료 업데이트를 참조하세요.

> **Note**
>
> 제안이 수락되면 계약 탭에 계약으로 표시됩니다. 제안의 상태는 변경되지 않습니다.

비공개 제안 초안 작성 및 게시

다음 프로세스를 사용하여 비공개 제안의 초안을 작성하고 게시합니다.

비공개 제안 초안을 작성 및 게시하려면

1. 제안 정보 제공 페이지에서 제안 이름, 제안 세부 정보, 갱신 유형 및 제안 만료 날짜를 입력합니다. 갱신 제안인 경우 에서 생성된 기존 AWS Marketplace 계약을 갱신하려는 갱신의 경우의 기존 고객을 선택하고, AWS Marketplace기존 고객을 마이그레이션하려는 갱신의 경우 로 이전 AWS Marketplace을 선택해야 합니다 AWS Marketplace.

   > **Note**
   >
   > 제안 만료일은 제안이 무효가 되는 날짜입니다. 이 날짜의 23:59:59 UTC 이후에는 구매자가 이 비공개 제안을 보고 수락할 수 없습니다.

2. 다음을 선택합니다.
3. 제안 요금 및 기간 구성 페이지에서 요금 모델, 계약 또는 사용 기간, 요금, 통화 및 결제 일정을 선택합니다. 할부 플랜이 있는 요금 모델은 할부 플랜을 참조하세요.

   > **Note**
   >
   > USD가 아닌 비공개 제안은 계약 요금 상품으로 제한됩니다. 또한 USD가 아닌 지급 기본 설정을 구성했는지 확인합니다. 자세한 내용은 the section called “지급 기본 설정” 단원을 참조하세요.
   > 모든 공개 제안 및 소비 요금이 포함된 비공개 제안은 USD로만 생성할 수 있습니다.

4. 구매자 추가 페이지에서 비공개 제안을 제시하려는 각 AWS Marketplace 구매자의 AWS 계정 ID를 제공합니다. 선택한 각 구매자는 선택한 제안 통화가 지원되는 AWS 계정 AWS 리전 에가 있어야 합니다. 다른 AWS 계정 ID를 추가하려면 다른 구매자 추가를 선택합니다. 각 비공개 제안에 최대 24명의 구매자를 추가할 수 있습니다.
5. 다음을 선택합니다.
6. 법률 용어 및 제안 문서 구성 페이지에서 다음 옵션 중 하나를 선택합니다.
   - 공개 제안 최종 사용자 라이선스 계약(EULA) - 공개 제안의 EULA를 사용합니다.
   - 표준 계약 AWS Marketplace (SCMP) -에서 제공하는 표준 계약을 사용합니다 AWS Marketplace.
   - 사용자 지정 법률 용어 - 법률 용어, 작업 명세서, 청구서, 요금표 또는 기타 부록을 포함하여 비공개 제안과 관련된 파일을 최대 5개까지 업로드합니다. 제안이 생성되면 이러한 파일이 하나의 문서로 병합됩니다.

7. 검토 및 생성 페이지에서 비공개 제안 의 세부 정보를 검토합니다. 검토 및 확인한 후 제안 생성을 선택하여 제안을 게시하고 선택한 구매자에게 확장합니다. 제안 게시에는 AWS Marketplace 카탈로그 API에 대한 요청이 포함되므로 제안을 검증하고 처리하는 데 최대 1시간이 걸릴 수 있습니다. 이 요청은 요청 페이지에서 볼 수 있습니다.

> **Note**
>
> 요청이 성공한 경우에만 제안이 게시되고 확장됩니다. 요청이 실패하면 제안이 고객에게 확장되지 않습니다. 실패는 다시 제출하기 전에 수정해야 하는 오류 또는 시스템 오류가 있음을 의미합니다.

비공개 제안 및 데모 요청 버튼 추가

판매자는 제품 세부 정보 페이지에 call-to-action 버튼을 추가할 수 있습니다. 버튼을 사용하면 구매자가 비공개 제안 및 안내 제품 데모를 요청할 수 있습니다. 제품 세부 정보 페이지에 버튼 하나 또는 둘 다를 추가할 수 있습니다.

버튼은 다음 제품 유형과 함께 사용할 수 있습니다.

- Amazon Machine Image
- 서비스형 소프트웨어(SaaS)
- 컨테이너
- CloudFormation 템플릿

버튼을 사용하려면 APN Customer Engagements Program(ACE)에 속해야 합니다. 구매자가 제안 또는 데모를 요청하면 연락처 데이터를 입력하고 양식에 세부 정보를 요청합니다. 그런 다음 AWS 수요 생성 팀은 요청을 검증하고 Partner Central의 ACE를 통해 시작된 AWS 기회로 해당 적격 요청을 전송합니다. 그런 다음 고객에게 후속 조치를 취하여 제안 세부 정보를 논의하거나 안내 데모를 예약합니다. ACE에 대한 자세한 내용은 APN 고객 참여 프로그램 웹 사이트와 APN 고객 참여(ACE) FAQ의 리드 및 기회를 참조하세요. FAQs

다음 주제의 단계에서는 제품 세부 정보 페이지에 버튼을 추가하는 방법을 설명합니다.

주제

- 버튼 사전 조건
- 버튼 활성화

버튼 사전 조건

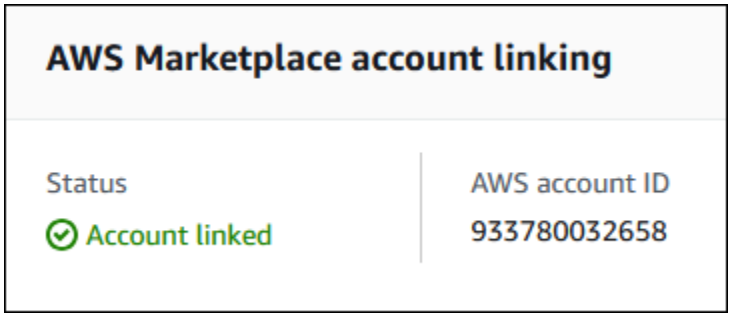
제품 세부 정보 페이지에 call-to-action 버튼을 추가하려면 먼저 다음 사전 조건이 있어야 합니다.

- AWS Partner Central에서 AWS 추천된 리드와 기회를 받을 수 있는지 확인합니다. 자세한 내용은 APN Customer Engagements Program 웹 사이트를 참조하세요.

> **Note**
>
> ACE 프로그램에 등록하면 2주마다 상태 업데이트가 수행됩니다. 비공개 제안 및 데모 요청 버튼은 상태 업데이트가 완료된 후에만 나타납니다. 이러한 옵션을 선택하고 메시지를 수신하면 다음 격주 업데이트에서 액세스가 보류됩니다.

- AWS Partner Central과 AWS Marketplace 계정을 연결합니다. 이렇게 하려면 다음을 수행해야 합니다.
  - `CreatePartnerCentralCloudAdminRole` IAM: 정책을 생성합니다. 자세한 내용은 AWS Partner Central 시작 안내서의 계정 연결을 위한 사전 조건을 참조하세요.
  - AWS Partner Central과 AWS Marketplace 계정을 연결합니다. 자세한 내용은 AWS Partner Central 시작 안내서의 Partner Central 계정을 AWS Marketplace 계정에 연결을 참조하세요. AWS

    AWS Partner Central과 AWS Marketplace 계정을 연결하면 Partner Central 홈 페이지에 다음 상태 메시지가 표시됩니다.



자세한 내용은 Partner Central에 로그인하여 다음을 참조하세요.

- AWS Partner Central & Marketplace 계정 연결 가이드
- AWS Partner 및 Marketplace 계정 연결 데모 비디오

> **Note**
> 이러한 리소스를 사용하려면 로그인해야 합니다.

버튼 활성화

AWS 추천을 받을 수 있는 ACE 자격이 되면 AWS Marketplace Management Portal을 사용하여 하나 또는 두 call-to-action 버튼을 활성화합니다.

새 제품 목록을 생성할지 아니면 현재 목록을 업데이트할지에 따라 별도의 프로세스에 따라 버튼을 활성화합니다.

새 제품에 대해 버튼을 활성화하려면

1. AWS Marketplace Management Portal를 사용하여 다음 유형의 제품을 생성하고 공개합니다.
   - AMI
   - SaaS
   - 컨테이너
   - 클라우드 프런트 템플릿
2. 제품 생성의 일환으로 가이드 데모 및 비공개 제안 요청에서 구매자를 위한 가이드 데모 요청 활성화와 구매자를 위한 비공개 제안 요청 활성화의 조합을 선택합니다.

> **Note**
> 이 버튼은 제품을 공개한 후에만 비공개 제안의 제품 세부 정보 페이지에 표시됩니다.

기존 제품에 대해 버튼을 활성화하려면

1. 의 제품 AWS Marketplace Management Portal탭에서 변경하려는 제품을 선택합니다.
2. 변경 요청 목록을 열고 제품 정보 업데이트를 선택합니다.
3. 구매자를 위한 안내 데모 요청 활성화와 구매자를 위한 비공개 제안 요청 활성화의 조합을 선택합니다.

버튼은 변경 사항을 저장한 후에만 제품 세부 정보 페이지에 표시됩니다.

구매자에게 비공개 제안 보내기

비공개 제안이 게시된 후, 구매자는 AWS Marketplace Management Portal콘솔의 비공개 제안 페이지에서 이용 가능한 비공개 제안 탭으로 이동하여 계정에 제시된 모든 제안을 볼 수 있습니다. 구매자는 사용 가능한 비공개 제안 탭에서 AWS Marketplace 채널 파트너가 제공하는 제안을 레코드 판매자 열에서 볼 수 있습니다. 독립 소프트웨어 개발 판매 회사(ISV)가 게시자 열에 표시됩니다. 구매자는 제안 목록에서 적절한 제안 ID를 선택하여 비공개 제안으로 이동할 수 있습니다.

구매자는 수락되었거나 만료된 제안 탭에서 수락되었거나 만료된 제안 ID를 볼 수 있습니다.

또한 비공개 제안이 게시된 후에는 구매자에게 해당 제안의 이행 페이지 URL을 보낼 수 있습니다.

구매자에게 비공개 제안을 보내는 방법

1. AWS Marketplace Management Portal에 로그인하고 제안을 선택합니다.
2. 제안 옆의 라디오 버튼을 선택합니다.
3. 작업을 선택하고 제안 URL 복사를 선택합니다.
4. 구매자에게 URL을 보냅니다.

비공개 제안 복제

AWS Marketplace 채널 파트너 비공개 제안을 포함하여 비공개 제안을 복제할 수 있습니다. 복제를 사용하여 템플릿을 사용하여 새 제안을 생성하거나 기존 제안을 업데이트하고 대체합니다.

비공개 제안을 복제하려면

1. AWS Marketplace Management Portal에 로그인하고 제안을 선택합니다.
2. 제안 테이블에서 복제하려는 제안 옆의 옵션을 선택합니다.
3. 제안 복제를 선택합니다.
4. 선택한 제안의 미리 채워진 정보가 포함된 새로운 제안 생성 환경이 열립니다. 필요에 따라 제안 세부 정보를 검토하고 수정합니다.
5. (선택 사항) 기존 제안을 교체하기 위해 복제하는 경우 기존 제안 취소를 선택합니다. 선택하면 원래 제안이 자동으로 만료되며이 새 제안이 게시될 때 구매자가 액세스할 수 없습니다. 이는 제안의 접근성에만 영향을 미치며 구매자가 이미 원래 제안을 수락한 경우 기존 구독에는 영향을 미치지 않습니다.
6. 비공개 제안 복제를 선택합니다. 그러면 제안이 게시되고 이전에 선택한 구매자에게 확장됩니다.

제안 세부 정보 다운로드

다음 절차에 따라 제안 세부 정보를 .pdf 파일로 다운로드합니다.

제안 세부 정보를 다운로드하려면

1. AWS Marketplace Management Portal에 로그인하고 제안을 선택합니다.
2. 제안 테이블에서 제안 옆의 옵션을 선택하고 세부 정보 보기를 선택합니다. 또는 제안 ID 열에서 제안에 대한 링크를 선택할 수 있습니다.
3. 제안 세부 정보 페이지에서 PDF 다운로드를 선택합니다.

비공개 제안 진행 상황 저장

다음 프로세스를 사용하여 진행 상황을 저장하고 나중에 다시 시작합니다.

작업을 저장하고 재개하려면

1. 완료된 단계에서 저장 및 종료를 선택합니다. 대화 상자에서 콘텐츠를 초안 상태로 저장하고 있는지 확인하고 검증 오류를 검토합니다. 검증 오류 또는 누락된 세부 정보가 있는 경우 수정을 선택하여 단계로 이동하여 문제를 해결할 수 있습니다. 준비가 되면 저장 후 종료를 선택하여 변경 내용을 저장합니다.

   저장 후 종료하면 요청이 처리되는 동안 검토됩니다. 처리가 완료될 때까지 몇 분 또는 몇 시간이 걸릴 수 있습니다. 요청이 성공하기 전에는 단계를 계속하거나 변경할 수 없습니다. 요청이 성공하면 저장이 완료된 것입니다. 요청이 실패하면 다시 제출하기 전에 수정해야 하는 오류 또는 시스템 오류가 있음을 의미합니다.
2. 제안 작업을 재개하려면 제안 페이지를 열고 제안을 선택한 다음 제안 생성 재개를 선택합니다.
3. 완료되면 저장 후 종료를 선택하여 진행 상황을 저장하거나 제안 생성을 선택하여 비공개 제안을 게시하고 선택한 구매자에게 확장할 수 있습니다.

비공개 제안 만료 업데이트

다음 프로세스를 사용하여 비공개 제안의 만료 날짜를 업데이트합니다.

비공개 제안의 만료 날짜를 업데이트하려면

1. AWS Marketplace Management Portal에 로그인하고 제안을 선택합니다.

2. 제안 페이지에서 업데이트하려는 제안을 선택합니다.
3. 편집을 선택합니다.
4. 새 제안 만료 날짜를 입력합니다.
5. 제출을 선택합니다.

   업데이트가 완료되면 제안이 활성 상태로 변경되고 구매자가 제안을 수락할 수 있습니다.

비공개 제안 취소

다음 프로세스를 사용하여 비공개 제안을 취소합니다.

1. AWS Marketplace Management Portal에 로그인하고 제안을 선택합니다.
2. 제안 페이지에서 업데이트하려는 제안을 선택합니다.

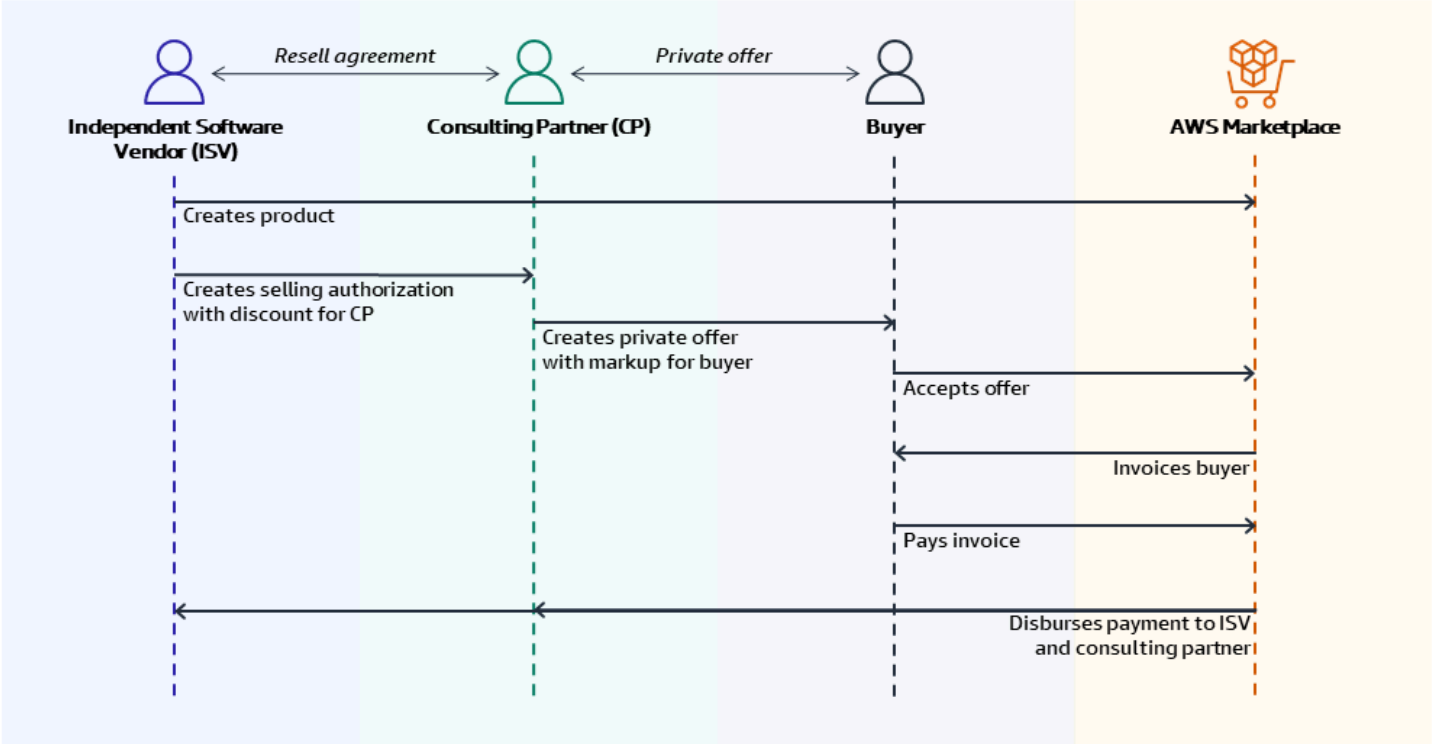
   > **Note**
   > 제안을 취소하면 제안 만료 날짜가 수정되므로 이 제안을 연장한 구매자의 경우 제안이 만료된 것으로 표시됩니다.

3. 작업을 선택한 다음 제안 취소를 선택합니다.

AWS Marketplace 채널 파트너로서 비공개 제안 생성

AWS Marketplace 채널 파트너 비공개 제안은 채널 파트너에게 독립 소프트웨어 공급업체(ISVs) 제품을 재판매할 수 있는 기회를 제공합니다 AWS Marketplace. AWS Marketplace 채널 파트너와 ISV는 하나 이상의 ISV 제품을 재판매하는 계약을 체결하고 채널 파트너는 해당 제품의 구매자에게 비공개 제안을 제공합니다.

다음 다이어그램은 ISV, 채널 파트너, 구매자 간의 이 관계를 보여줍니다.



> **Note**
>
> ISV로서 채널 파트너의 판매 권한을 만드는 방법에 대한 자세한 내용은 ISV로서 AWS Marketplace 채널 파트너에 대한 판매 권한 생성 섹션을 참조하세요.

각 AWS Marketplace 채널 파트너 비공개 제안은 한 명의 구매자에게만 표시되며, 구매자의 요구 사항에 맞는 사용자 지정 요금과 고유한 상업 조건이 적용됩니다. 비공개 제안을 생성할 때, ISV가 설정한 도매 가격에서 시작합니다. 그 후 해당 가격을 인상하여 구매자의 제안 가격을 생성합니다.

> **Note**
>
> 비공개 제안을 생성할 때 채널 파트너는 ISV가 판매 권한 부여에서 정의하는 통화를 사용해야 합니다.

다음 방법 중 하나로 도매 비용을 결정합니다.

- 반복성 할인 - ISV는 AWS Marketplace 채널 파트너가 제품을 정가에서 약정 할인하여 반복 판매 권한을 통해 재판매할 수 있도록 승인합니다. AWS Marketplace 채널 파트너는이 할인을 활용하여 ISV

와의 추가 가격 협상 없이 제품을 계속 재판매할 수 있습니다. 이 할인은 지정된 날짜까지 또는 ISV나 채널 파트너가 종료할 때까지 무기한으로 지속되도록 설정할 수 있습니다.

- 비반복 할인 - ISV가 AWS Marketplace 채널 파트너에게 제공하는 판매 권한 부여는 특정 구매자만 사용할 수 있는 일회성 할인입니다.

두 경우 모두 구매자가 비공개 제안에 대한 비용을 지불한 후는 표준 프로세스를 AWS Marketplace 사용하여 합의된 요금에 따라 AWS Marketplace 채널 파트너와 ISV에 자금을 분배합니다.

> **Tip**
>
> ISVs와 채널 파트너는의 파트너 메뉴를 사용하여 판매 승인을 AWS Marketplace Management Portal 볼 수 있습니다.

비공개 제안 생성에 대한 자세한 지침은 AWS Marketplace 채널 파트너 비공개 제안 - 제안 생성을 참조하세요.

비공개 제안을 위한 제3자 파이낸싱에 대한 자세한 내용은 이제 AWS Marketplace에서 고객 파이낸싱 이용 가능을 참조하세요.

추가 정보

추가 정보 및 질문이 있는 경우 ISVs 및 채널 파트너가 AWS Marketplace 채널 팀과 연결할 것을 권장합니다. 구체적으로 누구에게 연락해야 하는지 모르는 경우 aws-mp-channel@amazon.com으로 이메일을 보내면 팀원 중 한 명이 1영업일 이내에 답변을 드립니다.

ISV로서 AWS Marketplace 채널 파트너에 대한 판매 권한 생성

독립 소프트웨어 공급업체(ISV)는 AWS Marketplace 채널 파트너가 해당 파트너에 대한 판매 승인을 생성하여 제품을 재판매할 수 있도록 권한을 부여할 수 있습니다. 채널 파트너는 판매 권한을 사용하여 최종 구매자에게 채널 파트너 비공개 제안(CPPO)을 생성할 수 있습니다. 지원되는 제품 유형은 다음과 같습니다.

- AMI 기반 제품
- 컨테이너 기반 제품
- SaaS 기반 제품
- 전문 서비스 제품

다음 절차에서는 ISVs AWS Marketplace 채널 파트너에 대한 판매 권한을 생성하는 방법을 간략하게 설명합니다. 이 기능을 사용하려면의 판매 권한 부여 탭을 사용할 수 있는 권한이 있어야 합니다 AWS Marketplace Management Portal. 자세한 내용은 AWS Marketplace 판매자를 위한 정책 단원을 참조하십시오.

판매 권한 부여 생성

1. AWS Marketplace 판매자 계정으로 AWS Marketplace Management Portal에 로그인합니다.

   > **Tip**
   > AWS Marketplace 판매자 계정으로 로그인하기 AWS 계정 전에 다른에서 로그아웃했는지 확인합니다.

2. 판매 권한 부여 탭을 선택한 다음 판매 권한 부여 생성을 선택합니다.
3. 판매 권한 부여 생성 페이지에서 다음 세부 정보를 입력합니다.

   - 판매 권한 부여 이름에 권한 부여 이름을 입력합니다.

     > **Note**
     > 판매 권한 부여 이름에 입력한 정보는 채널 파트너가 판매자 보고서에서 볼 수 있습니다.

   - 리셀러의 경우 드롭다운 목록에서 권한을 부여하려는 AWS Marketplace 채널 파트너(리셀러)를 선택합니다. 이름 또는 계정 ID로 리셀러를 선택할 수 있습니다.
   - 제품 유형에서 제품 유형을 선택한 다음 판매 권한 부여를 생성할 제품 중 하나를 선택합니다.
   - 권한 부여 세부 정보로 계속을 선택합니다.
4. 세부 정보 지정 페이지에서 다음 세부 정보를 입력합니다.

   - 판매 권한 설명에 설명을 입력합니다.

     > **Note**
     > 판매 권한 설명에 입력한 정보는 채널 파트너가 판매자 보고서에서 볼 수 있습니다.

   - 갱신의 경우이 권한이 동일한 제품에 대해 기존 고객과 기존 유료 구독을 갱신하려는 것인지 여부를 지정합니다.

- (선택 사항) 하나 이상의 구매자 계정 ID를 설정하여 해당 구매자에게만 판매 권한이 제공되도록 지정합니다.
- 저장 및 종료를 선택하거나 다음을 선택할 수 있습니다. 프로세스의 어느 단계에서든 저장 및 종료를 선택하면 판매 권한이 초안으로 저장됩니다. 다음을 선택하면 기간 및 요금 선택 페이지로 이동합니다.

5. 기간 및 요금 선택 페이지에서 다음 세부 정보를 입력합니다.

- 요금 모델에서 다음 옵션 중 하나를 선택합니다.
  - 할부 플랜이 포함된 계약 요금 - 각 차원 유형에 고정 수량, 초과에 대한 시간당 요금 및 할부 플랜을 지정합니다. 초과 요금은 시간당 요금으로 청구되며 별도로 청구됩니다.
  - 선결제 계약 요금 - 각 차원 유형에 대한 계약 요금과 추가 사용에 대한 시간당 요금을 지정합니다. 구매자는 각 차원 유형에 대해 계약 가격으로 약정할 수량을 선택하고 수락 시 전체 금액에 대한 인보이스를 받을 수 있습니다. 추가 사용량은 시간당 요금으로 청구되며 별도로 청구됩니다.
  - 사용 요금 - 각 차원 유형에 시간당 요금을 지정합니다. 구매자에게는 사용량에 대한 시간당 요금을 기준으로 요금이 부과됩니다.
- 통화에서 판매 권한 부여의 통화를 선택합니다.

> **Note**
>
> USD 외 통화는 계약 요금 제안에만 사용할 수 있습니다.

- 기간에서 판매 권한 부여 기간을 선택합니다.

> **Note**
>
> 리셀러의 시작일은 제조업체가 재판매 승인서에 기재한 날짜보다 이전이어야 합니다.

- 제품 차원 섹션에는 현재 제품에서 사용 가능한 계약 차원이 표시됩니다. 판매 권한 부여에 포함할 차원을 선택하고 수량 또는 가격을 제공할 수 있습니다. 할부 플랜을 사용한 계약 요금은 차원당 수량을 입력할 수 있고 선결제 금액을 사용한 계약 요금은 차원당 가격을 입력할 수 있습니다.
- 사용자 지정 차원 관리를 선택하여 향후 제안 및 판매 권한 부여를 위해이 제품에서 사용할 수 있는 차원을 제품에 추가합니다. 제품에 추가된 사용자 지정 차원은 제거할 수 없습니다. 하나의 제품에 최대 100개의 차원을 가질 수 있습니다.

- 사용량 차원당 요금에서 각 차원 유형에 시간당 요금을 지정합니다. 구매자에게는 사용량에 대한 시간당 요금을 기준으로 요금이 부과됩니다. 공개 제안 가격은 기본적으로 여기에 채워집니다.
- 구매자 할부 플랜의 경우 계약 합계를 입력하고 원하는 빈도에 따라 할부 플랜을 생성합니다. 이는 할부 플랜이 포함된 계약 요금에 필요합니다.
- 인스턴스 유형당 요금의 경우 각 인스턴스 유형에 대한 사용 요금을 설정합니다. 선택적 요금 도구를 사용하면 퍼블릭 요금에 할인을 적용하거나 모든 인스턴스에 동일한 가격을 적용하여 가격을 대량 업데이트할 수 있습니다.
- 판매 권한 부여 가용성의 경우 다음 옵션 중 하나를 선택하여 생성된 비공개 제안 수 또는이 판매 권한 부여를 사용하여 특정 비공개 제안을 생성할 수 있을 때까지의 가용성을 제한합니다.
  - 일회용 - 리셀러가 지정된 종료 날짜까지 단일 제안을 생성할 수 있도록 허용합니다.
  - 특정 기간 - 리셀러가 지정된 종료 날짜까지 여러 제안을 생성할 수 있습니다.
  - 시간 설정 없음 - 판매 권한이 수동으로 비활성화될 때까지 리셀러가 여러 제안을 생성할 수 있습니다.
- 저장 및 종료, 이전 또는 다음을 선택할 수 있습니다. 다음을 선택하면 법률 용어 구성 페이지로 이동합니다.

6. 법률 용어 구성 페이지에서 다음 세부 정보를 입력합니다.

   - 최종 사용자 라이선스 계약에서 퍼블릭 EULA 또는 표준 계약 AWS Marketplace (SCMP)을 선택하거나 사용자 지정 EULA를 업로드합니다.

     > **Note**
     >
     > 전문 서비스 판매자에게는 사용자 지정 EULA만 지원됩니다.

   - (선택 사항) 리셀러 계약 AWS Marketplace (RCMP)을 선택하거나 판매 권한 부여에 포함할 사용자 지정 계약을 업로드합니다.
   - 저장 및 종료, 이전 또는 다음을 선택할 수 있습니다. 다음을 선택하면 검토 및 생성 페이지로 이동합니다.

7. 검토 및 생성 페이지에서 모든 정보가 올바른지 확인합니다. 판매 권한이 게시되면 수정할 수 없습니다.
8. 판매 권한 생성을 선택하여 리셀러에게 판매 권한을 게시합니다.

판매 권한 부여 관리

- 판매 권한 생성 테이블이 판매 권한 이름, 제품 이름, 리셀러 이름, 생성 날짜, 만료 날짜 및 상태를 포함한 관련 판매 권한 세부 정보를 표시하도록 업데이트되었습니다.
- 판매 권한이 생성된 후에는 만료 날짜를 연장하거나 다른 세부 정보를 수정할 수 없습니다.
- 리셀러가 더 이상 판매 권한을 사용하지 않도록 하려면 판매 권한 부여를 비활성화할 수 있습니다. 판매 권한 부여를 비활성화하면 해당 판매 권한 부여를 사용하여 새 제안을 생성할 수 없습니다. 이미 생성된 제안은 영향을 받지 않습니다.
- 판매 권한을 선택하고 복제를 선택하여 판매 권한을 복제할 수도 있습니다. 이렇게 하면 값이 모든 필드에 미리 채워지고 편집이 허용됩니다.

판매 권한 부여 상태 및 작업

다음 목록은 판매 권한 부여 상태 값과 그 의미를 설명합니다.

- 초안 - 판매 권한이 생성되었지만 아직 활성화되지 않았습니다.
- 권한 있음 - 판매 권한이 활성 상태이며 채널 파트너 비공개 제안(CPPOs. 아직 생성된 CPPOs 없습니다.
- 권한 부여(재사용 가능) - 판매 권한 부여는 하나 이상의 CPPO를 생성하는 데 사용되었으며 추가 CPPOs 생성하는 데 사용할 수 있습니다. 이 상태는 특정 기간이 있거나 설정된 기간이 없는 권한 부여의 경우 일반적입니다.
- 권한 있음(소비됨) - 판매 권한이 완전히 활용되었으며 추가 CPPOs 생성하는 데 사용할 수 없습니다. 이는 일반적으로 리셀러가 비공개 제안을 생성한 후 일회용 권한 부여를 통해 발생합니다.
- 만료됨 - 판매 권한 부여의 가용성 종료일이 지났습니다. 더 이상 CPPOs 생성하는 데 사용할 수 없습니다.
- 비활성화됨 - 독립 소프트웨어 공급업체(ISV)가 권한 부여를 수동으로 비활성화했습니다. 더 이상 CPPOs 생성하는 데 사용할 수 없습니다. API에서이 상태를 "제한"이라고 합니다.

> **Note**
>
> 시스템은 판매 권한 부여의 내부 상태, 가용성 종료 날짜, 제안 연장 상태 등 여러 요인을 기반으로 상태를 결정합니다. 상태 값은 잠재적 중복을 해결하기 위해 우선 순위에 따라 할당됩니다.

비공개 제안 할부 플랜

AWS Marketplace 판매자는 할부 플랜(유연한 결제 일정이라고도 함)을 사용하여 사용자 지정 결제 일정으로 비공개 제안을 연장할 수 있습니다. 할부 플랜은 특정 제품 및 요금 유형에 대한 비공개 제안에 사용할 수 있습니다. 자세한 내용은 비공개 제안 대상인 제품 유형을 참조하세요. 결제 일정은 수락된 계약 기간 동안 분산될 수 있으며, 구매자는 정기적으로 할부 결제를 할 수 있습니다.

다년 및 사용자 지정 기간 Amazon Machine Image(AMI) 제품의 경우 제안에 포함된 각 인스턴스 유형에 대한 인스턴스 수와 추가로 시작된 인스턴스에 대한 시간당 요금을 설정합니다. 구매자가 지정된 수의 인스턴스를 시작하면 시작된 추가 인스턴스에는 비공개 제안에 지정된 시간당 요금이 부과됩니다.

구매자가 연장하고 구독한 비공개 제안에 대한 결제 일정은 수정될 수 없습니다. 변경하려면 새 제안을 생성해야 합니다.

비공개 제안에 대한 할부 플랜 만들기

비공개 제안을 생성할 때 할부 플랜을 사용하여 사용자 지정 결제 일정을 설정할 수 있습니다.

비공개 제안에 대한 할부 플랜을 생성하려면

1. 제안 요금 및 기간 구성 페이지의 제품 요금에서 할부 플랜으로 계약 요금을 선택합니다.
2. 이 제안의 계약 기간을 선택하고 제안 세부 정보를 지정합니다. 자세한 내용은 AWS Marketplace 제품에 대한 비공개 제안 준비 단원을 참조하십시오.
3. 구매자 할부 플랜에 다음 파라미터를 입력합니다.
   - 계약 합계
   - (선택 사항) 첫 번째 결제가 다른 결제와 다르도록 하려면 초기 결제를 입력합니다. 나머지 잔액은 후속 결제 간에 균등하게 분할됩니다.
   - Frequency(주파수)

     월별, 분기별, 연간 또는 사용자 지정을 선택합니다. 사용자 지정을 선택한 경우 할부 수도 입력합니다.

     최대 60개의 결제를 추가할 수 있습니다. 또한 각 결제 라인 항목을 조정할 수 있는 옵션도 있습니다. 결제 라인 항목을 조정할 때마다 구매자의 결제 총액이 업데이트됩니다.
   - 첫 번째 인보이스 날짜
4. 할부 계획 생성을 선택합니다. 인보이스 날짜가 계약 기간을 벗어나면 오류 메시지가 표시됩니다.

5. 모든 인보이스 금액과 날짜를 확인한 후 구매자가 지불해야 하는 총 금액이 비공개 제안 과정에서 구매자가 지불하려는 총 가격과 일치하는지 확인합니다. 비공개 제안 생성을 완료하려면의 나머지 단계를 완료합니다the section called “비공개 제안 초안 작성 및 게시”.

구매자가 비공개 제안을 수락하면 결제 일정에 정의한 인보이스 날짜에 00:00 UTC에 인보이스가 발행됩니다. 가 구매자로부터 결제를 받은 후 각 인보이스에 대한 결제를 AWS Marketplace 받습니다. 제안 수락 날짜 이전에는 인보이스 날짜 하나만 발행할 수 있습니다. 결제 일정의 첫 번째 인보이스 날짜 이후에 비공개 제안이 수락되면 제안이 수락된 직후 첫 번째 인보이스가 생성됩니다. 구매자는 구독한 후 일정 및 인 AWS 보이스에서 모든 결제를 볼 수 있으므로 지출을 추적할 수 있습니다.

할부 플랜 보고

할부 플랜을 통한 비공개 제안 보고는 월별 청구 수익 보고서의 섹션 4에 표시됩니다. 자세한 정보는 섹션 4: 유연한 결제 일정을 포함하여 생성된 계약 섹션을 참조하세요.

비공개 제안에 대한 미래 날짜의 계약 만들기

AWS Marketplace 판매자는 미래 날짜의 계약을 사용하여 구매자가 미리 정해진 미래 날짜에 받게 될 제품을 판매할 수 있습니다. 일반적인 AWS Marketplace 거래에서 구매자는 제안이 수락되거나 계약이 생성된 직후에 제품 라이선스 또는 자격을 받습니다. 이와 대조적으로 미래 날짜의 계약(FDA)의 경우, 구매자는 미리 정해진 미래의 날짜에 제품 라이선스 또는 권리를 받습니다. FDA는 구매자와의 기존 거래에 대한 갱신을 설정하는 데 사용할 수 있습니다. FDA는 유연한 결제 유무에 관계없이 계약 및 소비 계약(CCP) 가격 책정에 대한 서비스형 소프트웨어(SaaS) 제품을 지원합니다. 다음 섹션에서는 미래 날짜의 계약 작업에 대한 정보를 제공합니다.

FDA를 이용하면 구매자가 제품 사용을 시작하려는 시점이 아니라 구매자가 원하는 시점에 거래를 종결할 수 있습니다. FDA를 사용하여 AWS Marketplace의 트랜잭션에 대해 다음 작업을 종속적으로 수행할 수 있습니다.

- 판매 요구 사항에 따라 거래를 예약(구매자가 제안 수락)하세요.
- 재무 또는 회계 요구 사항에 따라 구매자에게 비용을 청구하세요.
- 구매자의 필요에 따라 구매자에게 제품에 대한 액세스 권한(예: 라이선스 또는 자격 활성화)을 제공합니다.

구매자는 다음과 같은 경우 현재 계약이 유효한 동안에도 FDA 제안을 수락할 수 있습니다.

- FDA 시작일은 기존 계약이 종료된 후 발생합니다.

- FDA를 수락하기 전에 자동 갱신이 비활성화됩니다.

> **Important**
>
> FDA가 수락되면 자동 갱신을 다시 활성화할 수 없습니다.

- 서비스 날짜는 다른 승인된 FDAs와 겹치지 않습니다.

주제

- 미래 날짜의 계약에 대한 고려 사항
- 미래 날짜의 계약서 작성
- 미래 날짜의 계약과 함께 할부 플랜 사용
- 미래 날짜의 계약에 대한 알림 받기
- 채널 파트너 비공개 제안에 대한 재판매와 함께 미래 날짜의 계약 사용

미래 날짜의 계약에 대한 고려 사항

미래 날짜의 계약을 사용할 때는 다음 날짜를 염두에 두세요.

계약 체결 날짜

구매자가 제안을 수락하고 계약서를 생성한 날짜입니다.

계약 시작 날짜

제품에 대한 구매자의 라이선스 또는 자격이 활성화되고 구매자가 제품 사용을 시작할 수 있는 날짜입니다.

계약 종료 날짜

계약의 종료일입니다. 계약 및 구매자의 라이선스 또는 자격은 이 날짜에 만료됩니다.

> **Note**
>
> 최종 사용자 라이선스 계약, 주문 양식 또는 구독자와의 기타 계약에 지정된 계약 기간은 구독자의 인 AWS Marketplace 보이스에 지정된 조건과 충돌하는 경우 적용됩니다.

미래 날짜의 계약서 작성

레코드 판매자는 시작일이 미래인 비공개 제안를 생성할 때 계약 시작일을 설정합니다. 구매자는 시작일을 변경할 수 없지만 AWS Marketplace에서 비공개 제안을 수락하기 전에 시작일을 검토할 수 있습니다.

시작일이 미래인 비공개 제안를 만들려면

1. 비공개 제안을 만들 때 계약 기간 에서 미래 날짜에 시작을 선택합니다.
2. 서비스 날짜 섹션에 서비스 시작일과 서비스 종료일을 입력합니다. 여기서 선택한 서비스 시작일은 구매자가 제안을 수락할 때 미래 날짜가 되는 계약의 계약 시작일이 됩니다.

> **Note**
>
> 갱신에 FDA를 사용하려면 서비스 시작 날짜가 갱신하려는 계약 종료 날짜로부터 1일 이상이어야 합니다. 예를 들어 계약 종료 날짜가 12/31/2024인 경우 서비스 시작 날짜를 1/1/2025로 설정해야 합니다.
> 판매자는 서비스 시작일을 최대 3년 후까지 선택할 수 있습니다.

미래 날짜의 계약과 함께 할부 플랜 사용

FDA의 할부 플랜을 사용하면 계약 서명일과 계약 종료일 사이에 언제든지 구매에 대한 결제가 이루어지도록 설정할 수 있습니다. 여기에는 계약 시작일 이전 및 이후의 결제가 포함됩니다.

레코드 판매자는 비공개 제안 결제 날짜 및 금액을 선택합니다. 할부 계획 설정에 대한 자세한 내용은 the section called “ 비공개 제안에 대한 할부 플랜 만들기” 섹션을 참조하세요.

미래 날짜의 계약에 대한 알림 받기

미래 날짜의 계약에 대해 취해진 다음 조치에 대해 지정된 루트 계정으로 이메일 알림을 받게 됩니다.

- 제안 수락/계약 생성(계약 서명일)
- 라이선스 또는 권한 활성화 시(계약 시작일)
- 30일, 60일 또는 90일 후에 만료되는 계약에 대한 사전 알림
- 계약 만료(계약 종료일)
- 계약 수정 또는 교체 시

> **Note**
>
> SaaS를 위한 기존의 Amazon Simple Notification Service(SNS) 알림은 모두 FDA에서도 작동합니다. FDA의 경우 두 Amazon SNS 주제 모두 계약 시작 날짜(계약 서명 날짜가 아님)에 시작됩니다. 자세한 내용은 the section called "SaaS 제품에 대한 Amazon SNS 알림" 단원을 참조하세요.

채널 파트너 비공개 제안에 대한 재판매와 함께 미래 날짜의 계약 사용

제조업체와 리셀러는 AWS Marketplace 채널 파트너 비공개 제안에 미래 날짜의 계약을 사용할 수 있습니다.

제조업체의 경우:

- 표준 AWS Marketplace 채널 파트너 비공개 제안(CPPOs과 마찬가지로 제조업체는 AWS Marketplace 채널 파트너에게 재판매 권한을 연장하여 향후 시작일이 있는 CPPOs를 생성할 수 있는 권한을 부여해야 합니다.

  재판매 승인을 만드는 방법을 배우려면 the section called "ISV로서 판매 권한 만들기" 페이지의 단계를 따르세요.
- 재판매 승인을 생성할 때 제조업체는 선택적으로 최대 허용 서비스 시작일을 지정하도록 선택할 수 있습니다. 이는 채널 파트너가 해당 채널 AWS Marketplace 파트너 비공개 제안을 생성할 때 지정할 수 있는 최대 서비스 시작 날짜 AWS Marketplace 입니다.

  > **Note**
  >
  > 제조업체가 최대 날짜를 지정하지 않는 경우 AWS Marketplace 채널 파트너는 향후 최대 3년까지 향후 서비스 날짜를 지정할 수 있습니다.

리셀러의 경우:

- 리셀러와 채널 파트너의 경우 미래 날짜의 채널 파트너 비공개 제안과 일반적인 미래형 비공개 제안을 만드는 단계는 동일하지만 한 가지 중요한 차이점이 있습니다. 리셀러가 지정할 수 있는 계약 시작일은 제조업체의 재판매 승인에 최대 허용 서비스 시작일로 지정한 날짜보다 이전이어야 합니다.
- 채널 파트너로서 구매자를 위한 비공개 제안을 생성하는 방법을 알아보려면 the section called "채널 파트너 비공개 제안" 섹션을 참조하세요.

비공개 제안 FAQ

이 FAQ는 비공개 제안 생성, 관리 및 문제 해결에 대해 자주 묻는 질문에 대한 답변을 제공합니다.

주제

- 비공개 제안 성공 팀(POST)이란 무엇이며 판매자는 어떻게 연락할 수 있습니까?
- 판매자는 제안 탭에 액세스할 때 오류를 어떻게 해결할 수 있습니까?
- 할부 플랜 또는 유연한 결제 일정이란 무엇입니까?
- 비공개 제안에 액세스할 때 404 오류를 받는 구매자를 어떻게 도울 수 있습니까?
- 비공개 제안을 수락하려고 할 때 고객에게 "이미 활성 계약이 있습니다"라는 오류가 표시되는 이유는 무엇입니까?
- 판매자 또는 구매자가 비공개 제안을 취소할 수 있나요?
- 환불 또는 계약 취소를 요청하려면 어떻게 해야 하나요?
- 구매자는 언제 인보이스를 발행하나요?
- 제안이 수락되면 판매자는 어떤 조치를 취해야 합니까?
- 는 판매자와 파트너에게 어떻게 AWS 지불하나요?
- 는 세금을 어떻게 AWS 평가하나요?
- 리소스 및 지원

비공개 제안 성공 팀(POST)이란 무엇이며 판매자는 어떻게 연락할 수 있습니까?

POST는 AWS Marketplace 비공개 제안 환경에서 외부 대상을 활성화합니다. 판매자는 AWS Marketplace 관리 포털의 지원 양식을 사용할 수 있습니다. 지원 양식에 대한 도움이 필요하면 비공개 제안 지원 양식 가이드 PDF를 다운로드하여 참조하세요.

판매자는 제안 탭에 액세스할 때 오류를 어떻게 해결할 수 있습니까?

AWS Marketplace Management Portal에서 제안 탭을 선택할 때 오류가 발생하면 다음 사전 조건을 충족하는지 확인합니다.

소프트웨어 또는 서비스를 직접 판매하기 위한 비공개 제안을 생성하기 위한 사전 조건

1. AWS 계정에 적절한 Identity and Access Management(IAM) 정책이 있는지 확인합니다. 필수 정책에 대한 자세한 내용은 비공개 제안에 대한 IAM 정책을 참조하세요.

> **Note**
>
> IAM 정책 또는 권한을 수정하는 데 도움이 필요한 경우 내부 AWS 관리자에게 문의하세요. 공동 책임 모델에 따라 고객이 액세스를 관리하므로는 IAM 정책 또는 권한을 지원할 수 AWS 없습니다.

2. 0이 아닌 제품이 하나 이상 공개되어 있어야 합니다. 에서 공급업체를 검색하여 이를 확인할 수 있습니다 AWS Marketplace. 제품이 표시되지 않으면 제품을 공개적으로 등록하지 않았거나 제한된 목록이 있을 수 있습니다.

소프트웨어 또는 서비스를 재판매하기 위한 채널 파트너 비공개 제안(CPPOs)을 생성하기 위한 사전 조건

- AWS Marketplace다음에서 "판매자"가 되기 위한 지리적 사전 조건을 충족합니다.
  - 적격 관할 구역의 법인 또는 해당 영역 중 하나에 설립된 사업체를 사용합니다.
  - 은행 계좌에 대한 필수 정보를 제공합니다. 다음을 포함하여 하나 이상의 은행 계좌를 제공할 수 있습니다.
    - 미국 ACH(Automated Clearing House) 계정
    - 적격 관할 구역의 전 세계 은행 간 금융 통신 협회(SWIFT) 은행 계좌
    - Hyperwallet 계정

할부 플랜 또는 유연한 결제 일정이란 무엇입니까?

유연한 결제 일정(FPS)이라고도 하는 할부 플랜을 사용하면 사용자 지정 결제 일정으로 비공개 제안을 연장할 수 있습니다. 이러한 플랜은 특정 제품 및 요금 유형에 대한 비공개 제안에 사용할 수 있습니다. 자세한 내용은 비공개 제안에 적합한 제품 유형을 참조하세요.

결제 일정은 수락된 계약 기간 동안 분산될 수 있으며, 구매자는 정기적으로 할부 결제를 할 수 있습니다. 구독 후 고객은 일정 및 AWS 인보이스에서 모든 결제를 볼 수 있으므로 지출을 추적할 수 있습니다.

할부 플랜 및 FPSs 통해 판매자는 다음을 지정할 수 있습니다.

- 차원당 또는 인스턴스 유형당 단위 수
- 계약의 결제 조건(선결제, 지연 또는 여러 인보이스)

할부 플랜 만들기

> **Note**
> 소프트웨어 재판매 시나리오에서 독립 소프트웨어 공급업체(ISV)가 할부 플랜을 결정합니다.

할부 플랜을 생성하려면

1. 제안 요금 및 기간 구성 페이지의 제품 요금에서 할부 플랜으로 계약 요금을 선택합니다.
2. 계약 기간을 선택하고 제안 세부 정보를 지정합니다.
3. 구매자 할부 플랜에서 원하는 파라미터를 입력합니다.
   - 수락 시 선결제 인보이스 발행의 경우 달러 금액을 입력하고 인보이스 날짜를 비공개 제안을 생성하는 날짜로 설정합니다.
   - 지연된 인보이스 발행의 경우 달러 금액을 입력하고 향후 인보이스 날짜를 설정합니다.
   - 할부 인보이스 발행의 경우 결제 추가를 선택하여 달러 금액 및 인보이스 날짜가 있는 여러 결제 행 항목을 입력합니다.

자세한 내용은 할부 플랜 생성을 참조하세요. 할부 플랜에 대한 비디오 자습서를 볼 수도 있습니다.

고정 SKU 또는 인스턴스 유형 설정

제안 생성 페이지의 차원 섹션에서 다음을 수행합니다.

1. "구매자는 제공된 옵션을 하나 이상 선택할 수 있습니다."를 선택합니다.
2. 각 차원을 수량과 함께 입력합니다.
3. "차원 추가"를 선택하여 여러 차원을 포함하거나 "새 차원 생성 및 추가"를 선택하여이 비공개 제안에 대한 사용자 지정 차원을 생성합니다.

비공개 제안에 액세스할 때 404 오류를 받는 구매자를 어떻게 도울 수 있습니까?

다음은 404 오류 및 해결 방법의 일반적인 이유입니다.

잘못된 계정 연결

1. 구매자에게 AWS 콘솔의 비공개 제안 탭에 액세스하도록 요청합니다.
2. 사용 가능한 비공개 제안 또는 수락 및 만료된 제안 탭에 제안이 표시되지 않는 경우:
   - 구매자가 올바른 계정에 로그인했는지 확인합니다. AWS 콘솔의 오른쪽 상단 모서리에서 계정 ID를 확인할 수 있습니다.
   - 올바른 계정에 로그인한 경우 비공개 제안이 Amazon Machine Image(AMI) 제한 목록에 대한 것인지 확인합니다(아래 AMI 제한 목록 섹션 참조).

해결 방법: 구매자가 올바른 계정에 로그인하거나 올바른 AWS 계정 ID에 비공개 제안을 발행해야 합니다.

만료된 비공개 제안

- 수락 및 만료된 제안 탭에 제안이 표시되면 제안이 만료된 것입니다.

해결 방법: 만료 날짜를 연장하거나 새 비공개 제안을 발행합니다.

AMI 제한 목록 제품

- 두 탭에 제안이 표시되지 않고 구매자가 올바른 계정에 로그인한 경우 제품이 제한된 목록 상태일 수 있습니다.

해결 방법:

- 단일 AMI 제한 목록의 경우: 구매자 계정 허용 목록에 대한 지침에 따라 구매자의 계정을 허용 목록에 추가합니다.
- 다른 서버 유형 제한 목록의 경우: 비공개 제안 성공 팀(POST)에 문의하여 구매자의 계정을 허용 목록에 등록합니다.

> **Note**
>
> 허용 목록은 제품당 일회성 활동입니다.

프라이빗 마켓플레이스 제한

- 구매자에게 프라이빗 마켓플레이스가 있는 경우 프라이빗 카탈로그에서 제품을 사용할 수 없다는 오류가 표시될 수 있습니다.

해결 방법: 구매자가 허용 목록에 제품을 추가해야 합니다. 프라이빗 마켓플레이스에 제품 추가를 참조하세요.

문제가 지속되면 구매자에게 다음을 시도하도록 요청합니다.

- 로그아웃했다가 다시 로그인
- 브라우저 캐시 지우기
- 쿠키 삭제
- incognito 창에 로그인
- 다른 브라우저 사용(Internet Explorer 아님)

비공개 제안을 수락하려고 할 때 고객에게 "이미 활성 계약이 있습니다"라는 오류가 표시되는 이유는 무엇입니까?

이 오류는 구매자 계정에 이미 제품에 대한 활성 구독이 있는 경우 발생합니다. 해상도는 제품 유형에 따라 다릅니다.

서비스형 소프트웨어(SaaS) 계약의 경우

각 구매자 계정은 SaaS 계약 또는 소비 제품이 포함된 계약에 대해 활성 구독을 하나만 가질 수 있습니다. 활성 구독을 업데이트하거나 확장하려면:

1. AWS Marketplace 관리 포털(AMMP)의 계약 탭에서 계약 기반 제안을 생성합니다.
2. 계약 기반 제안에는 원래 제안의 보류 중인 결제 조건이 모두 포함됩니다. 아직 청구되지 않은 나머지 결제는 재정의되기 때문입니다.

계약 기반 제안 생성

Marketplace 비공개 제안(MPPO)의 경우:

1. AMMP의 계약 탭으로 이동합니다.

2. 구매자의 계정 ID를 검색합니다.
3. 구매자의 현재 계약을 선택합니다.
4. “계약 기반 제안 생성”을 선택합니다.

채널 파트너 비공개 제안(CPPO)의 경우:

1. ISV는 채널 파트너에게 라이선스 단위 수, CP에 대한 결제 조건, 계약 기간을 포함하여 파트너 탭의 새 요금으로 판매 권한을 부여합니다.
2. 채널 파트너가 AMMP의 계약 탭으로 이동합니다.
3. 구매자의 계정 ID를 검색합니다.
4. 구매자의 현재 계약을 선택합니다.
5. 계약 기반 제안 생성을 선택합니다.
6. 판매 승인 목록을 열고 1단계에서 새 기회를 선택합니다. 이렇게 하면 채널 파트너가 계약 기반 제안에 대한 조건을 입력할 수 있습니다.

자세한 내용은 계약 기반 제안 생성을 참조하세요.

SaaS 계약 및 소비 요금(CCPs이 적용되는 계약의 기타 옵션:

- 다른 구매자 계정을 대상으로 하는 동일한 제품 ID에 대한 새 비공개 제안을 생성합니다.
- 동일한 구매자 계정에 다른 제품 ID에 대한 새 비공개 제안을 발행합니다.
- 활성 계약의 취소를 요청한 다음 동일한 제품 ID 및 구매자 ID에 대해 새 비공개 제안을 발행합니다.

AMI 시간당 및 연간의 경우

각 구매자 계정에는 활성 구독이 하나만 있을 수 있습니다. 업데이트 또는 확장:

1. 제안 탭에서 새 제안을 발행합니다.
2. 구매자가 수락하면 새 약관이 이전 계약 약관보다 우선합니다.
3. 취소하지 않는 한 이전 제안의 결제 일정에서 보류 중인 요금은 계속됩니다.

AMI 계약 및 전문 서비스의 경우

새 비공개 제안을 수락하기 전에 구매자의 현재 구독을 취소해야 합니다. 환불/취소 양식을 사용하여 취소 요청을 시작합니다.

판매자 또는 구매자가 비공개 제안을 취소할 수 있나요?

- 구매자가 구독하지 않은 경우: 판매자는 AWS Marketplace 관리 포털의 제안 탭으로 이동하여 제안을 선택하고 취소를 선택하여 취소할 수 있습니다.
- 구매자가 구독한 경우: 기록 판매자는 AWS Marketplace Management Portal의 환급/취소 양식을 사용하여 취소 요청을 시작해야 합니다. 자세한 지침은이 비디오 자습서를 참조하세요.

환불 또는 계약 취소를 요청하려면 어떻게 해야 하나요?

환불 및 계약 취소는 AWS 고객 서비스 팀에서 처리합니다.

판매자 - Marketplace 비공개 제안ISVs, 채널 파트너 비공개 제안의 채널 파트너 - 환불 또는 취소를 시작하려면 환불/취소 양식을 사용해야 합니다.

이 동영상에서는 전체 프로세스를 설명합니다.

판매자는 다음 데이터를 입력해야 합니다.

- 구매자 계정 ID:이 정보는 제안 세부 정보 또는 청구 수익 대시보드에서 확인할 수 있습니다. 구독자 계정 ID여야 합니다.
- 판매자 계정 ID: 비공개 제안을 생성하는 데 사용되는 판매자의 AWS 계정 ID입니다.
- 제품 ID: 제안 세부 정보 또는 레거시 제품 ID 열의 청구 수익 대시보드에서이 정보를 찾을 수 있습니다.
- 결제 날짜: 제안 세부 정보 또는 사용 시작 기간 열의 청구 수익 대시보드에서이 정보를 찾을 수 있습니다.
- 환불 금액: 환불이 필요하지 않은 경우 판매자는 이를 0 USD로 설정할 수 있습니다.
- 추가 세부 정보: 다음 참고 사항을 참조하세요.

> **Important**
>
> - 계약 취소가 포함된 요청의 경우 다음 텍스트를 포함합니다."
>
> 계정 X의 offer-X 구독을 취소하세요.
>
> 환불의 경우이 섹션에서 구매자의 AWS 인보이스 ID를 지정하는 것이 도움이 되지만 필수는 아닙니다.
>
> - 후속 조치가 발생할 경우 나중에 참조할 수 있도록 제출 후 제공된 참조 ID를 저장합니다.

요청을 제출한 후 AWS 지원 콘솔에서 상태 업데이트를 확인합니다.

구매자는 언제 인보이스를 발행하나요?

- 수락 시: 인보이스는 구독 즉시 결제 콘솔에서 생성됩니다.
- 유연한 결제 일정: 인보이스는 판매자와 구매자 간에 협상된 사용자 지정 결제 일정을 기반으로 합니다.
- AWS 통합 청구서(이달 2/3): 공개 제안 구매 및 측정 제품을 포함할 수 있습니다.

제안이 수락되면 판매자는 어떤 조치를 취해야 합니까?

1. 제안 수락을 추적합니다.
2. 재무 팀의 경우: 표준 관행은 판매자가 미결제 결제를 생성하고 AWS Marketplace 주문에 대해 고객에게 인보이스 생성을 금지하여 구매 추적을 처리하는 것입니다.

는 판매자와 파트너에게 어떻게 AWS 지불하나요?

- 구독자로부터 자금을 성공적으로 수금한 후에만 지급이 시작됩니다.
- 기본 지급 주기에 따라 매월 7일부터 10일까지 지급이 이루어집니다. 또는 ISV 및 채널 파트너는 매일 또는 매월 지급 일정을 선택할 수 있습니다. ISV 또는 채널 파트너가 월별 옵션을 선택하는 경우 지출을 받을 날짜를 선택할 수 있습니다.
- 지급은 ISV 또는 채널 파트너의 계정에 있는 미국 은행 계좌에서 등록 수수료를 공제하고 입금됩니다. 일반 은행 ACH SLAs를 사용하는 데 영업일 기준 1~3일이 걸릴 수 있습니다. 미국 은행 계좌가 없는 경우 Hyperwallet을 사용하여 Amazon 매출에 대한 지급을 입금 계좌로 수신하고 현지 통화로 현지 은행으로 직접 이전할 수 있습니다.

는 세금을 어떻게 AWS 평가하나요?

AWS Marketplace 는 다음을 기준으로 세금을 부과합니다.

- 제품 구독자의 세금 주소
- 제품 유형
- Marketplace 진행자 법률

Marketplace Facilitator: Marketplace 운영자가 과세 기관에 세금을 청구, 징수 및 납부해야 합니다.

마켓플레이스가 아닌 진행자: 책임은 판매자에게 있습니다.

세금 의무에 대한 자세한 내용은 AWS Marketplace - 판매자를 위한 세금 도움말을 참조하세요.

리소스 및 지원

특정 요청이 있는 경우 AWS Marketplace Management Portal을 통해 다음 AWS Marketplace 팀 중 하나에 문의하세요.

비공개 제안 성공 팀(POST): 팀은 비공개 제안 운영 활성화 및 지원을 통해 판매자(ISV 및 채널 파트너) 및 구매자를 지원합니다. 지원 양식을 통해 문의하세요.

공급업체 재무 성공 팀(VFS): 팀은 공급업체 재무 현금 애플리케이션, 조정 및 관련 보고에 영향을 미치는 프로세스를 개선하고 공개 제안이 아닌 MPPOs 및 CPPOs와 관련된 재무 관련 질문을 통해 판매자(ISV 및 채널 파트너) 온보딩을 지원합니다. 지원 양식을 통해 문의하세요.

관리형 카탈로그 운영(MCO): 팀은 타사 소프트웨어 판매자를 플랫폼에 온보딩하고 AWS Marketplace , 정책 준수 및 구매자 경험을 위해 소프트웨어 제품을 검토 및 처리하고, 판매자와의 운영 관계를 관리할 책임이 있습니다. 지원 양식을 통해 문의하세요.

비공개 제안 프로세스를 처음 사용하는 경우이 비디오 라이브러리를 사용하여 가장 일반적인 질문에 대한 소개, 개요 및 답변을 시작할 수 있습니다. 구매자 인보이스 발행, 지출 또는 AWS 세금 규칙에 대한 질문에 대한 답변은 연결된 비디오 또는 AWS Marketplace Vendor Finance Success PDF에서 확인할 수도 있습니다.

계약 사용

AWS Marketplace 판매자 계약은 판매된 제품에 대한 판매자와 구매자 간의 계약입니다 AWS Marketplace. 구매자가 제안을 수락하면 계약이 형성되며,이 제안은 공개 또는 비공개일 수 있습니다. 이 섹션의 주제에서는 계약 유형, 상태 및 계약을 변경하거나 수정하는 방법을 설명합니다.

주제

- 계약 유형
- 계약 작업
- 에서 계약 수정 AWS Marketplace

계약 유형

AWS Marketplace 는 다음과 같은 유형의 계약을 제공합니다.

- 공개 제안 계약 - 구매자가 판매자로부터 공개적으로 등록된 제품을 구매할 때 수락하는 협상되지 않은 표준 법적 계약입니다 AWS Marketplace.
- 비공개 제안 계약 - 구매자가 수락하는 비공개 제안이 계약이 됩니다. 자세한 정보는 비공개 제안 만들기 및 관리 섹션을 참조하세요.
- 미래 날짜의 계약 - 구매자는 미리 정해진 미래 날짜에 제품 라이선스 또는 권한을 받습니다. 자세한 정보는 비공개 제안에 대한 미래 날짜의 계약 만들기 섹션을 참조하세요.

계약 작업

의 계약 페이지에서 계약을 보고 관리합니다AWS Marketplace Management Portal. 계약 페이지의 계약 테이블에는 판매하는 제품에 대한 계약이 표시됩니다 AWS Marketplace. 계약 테이블에서 계약은 다음 상태 중 하나를 가질 수 있습니다.

- 활성 - 계약 조건이 활성 상태입니다.
- 만료 중 - 계약이 90일 이내에 만료됩니다.
- 만료됨 - 정의된 계약 종료 날짜에 계약이 종료되었습니다.
- 갱신됨 - 계약이 갱신되어 업데이트된 조건이 포함된 새 계약이 생성되었습니다.
- 취소됨 - 수락자가 계약을 조기에 종료했습니다.

- 대체됨 - 계약이 계약 대체 제안을 사용하여 대체되었습니다.
- 종료됨 - 결제 실패와 같은 이유로 원래 계약 종료일 AWS 이전에에 의해 계약이 종료되었습니다.
- 보관됨 - 지정된 이유 없이 계약이 종료되었습니다.

계약 찾기

계약을 찾으려면 필터를 설정합니다. 계약 찾기 검색 창을 선택한 다음 필터 속성과 값을 선택합니다. 예를 들어 특정 계약을 찾으려면 계약 ID를 선택한 다음 ID를 선택합니다. 필터를 저장하려면 필터 지우기에서 새 필터 세트로 저장을 선택합니다.

계약 정렬

테이블은 처음에 계약 종료일(UTC) 열을 기준으로 정렬됩니다. 계약은 내림차순으로 표시되며, 향후 가장 멀리 종료되는 계약은 목록 상단에 표시됩니다. 계약 종료일(UTC) 열 제목을 선택하여 정렬 순서를 변경할 수 있습니다.

계약 세부 정보 보기

계약 세부 정보를 보려면 계약 ID 열에서 링크를 선택합니다. 또는 계약 옆의 옵션을 선택하고 세부 정보 보기를 선택할 수 있습니다.

> **Note**
>
> 계약을 수정하려면 섹션을 참조하세요에서 계약 수정 AWS Marketplace.

계약 세부 정보 다운로드

다음 절차에 따라 .pdf 파일로 계약 세부 정보를 다운로드합니다.

계약 세부 정보를 다운로드하려면

1. 에 로그인AWS Marketplace Management Portal하고 계약을 선택합니다.
2. 계약 테이블에서 계약 옆의 옵션을 선택하고 세부 정보 보기를 선택합니다. 또는 계약 ID 열에서 계약 링크를 선택할 수 있습니다.
3. 계약 세부 정보 페이지에서 PDF 다운로드를 선택합니다.

에서 계약 수정 AWS Marketplace

AWS Marketplace 판매자, 채널 파트너 및 ISVs 활성 계약을 대체하기 위한 업그레이드, 갱신 및 수정을 제공할 수 있습니다. 예를 들어, 표준화된 라이선스 조건을 사용하도록 새 권한을 부여하거나, 요금 할인을 제공하거나, 결제 일정을 조정하거나, 최종 사용자 라이선스 계약(EULA)을 변경할 수 있습니다. 또한 유닛 수 및 결제 일정을 변경하고 사용자 지정 종료 날짜를 추가할 수도 있습니다.

서비스형 소프트웨어(SaaS) 계약 및 소비 제품이 포함된 SaaS 계약은 공개 및 비공개 제안 수정을 지원합니다. 모든 AWS Marketplace 판매자는 독립 소프트웨어 공급업체(ISVs) 및 채널 파트너를 포함하여 이러한 제품 유형에 대한 비공개 제안을 업그레이드, 갱신 또는 수정할 수 있습니다. 다음 섹션에서는 프로세스에 대한 정보를 제공합니다.

제안과 계약의 차이점은 구매자가 조건을 수락했는지 여부입니다.

- 제안은 구매자의 제품 사용에 대한 일련의 조건입니다. 제안은 공개 또는 비공개일 수 있습니다.
- 계약은 구매자가 수락한 제안입니다. 계약에는 판매자가 공개 또는 비공개 제안을 사용하여 제공한 구매 제품 및 무료 제품이 포함됩니다.

원래 계약의 등록 판매자와 다른 등록 판매자를 지정하기 위해 계약을 수정할 수는 없습니다. 이 기능을 사용하려면 AWS Marketplace Management Portal에서 계약 탭을 사용할 수 있는 권한이 필요합니다. 자세한 내용은 AWS Marketplace 판매자에 대한 권한 단원을 참조하세요.

주제

- 공개 및 비공개 제안 수정에 지원되는 제품 유형
- 공개 및 비공개 제안 업그레이드, 갱신 및 수정 생성
- 업그레이드, 갱신 및 수정에 대한 보고

공개 및 비공개 제안 수정에 지원되는 제품 유형

다음 제품 유형만 제안 수정을 지원합니다.

- SaaS 계약
- 소비가 포함된 SaaS 계약

AWS Marketplace Management Portal의 계약 탭에서 다음과 같은 추가 제품 유형을 확인할 수 있습니다. 그러나 이러한 제품 유형은 수정을 지원하지 않습니다.

- SaaS 사용량 기반 제품
- AMI 기반 제품
- 컨테이너 기반 제품
- 서버 계약
- 전문 서비스 제품

공개 및 비공개 제안 업그레이드, 갱신 및 수정 생성

다음 절차를 AWS Marketplace Management Portal 사용하여에서 제안 업그레이드, 갱신 및 수정을 생성할 수 있습니다. 채널 파트너 비공개 제안의 경우 채널 파트너는 수정을 생성할 때 판매 권한에 정의된 통화를 사용해야 합니다.

> **Note**
>
> - 수정된 제안은 향후 계약 시작일을 가질 수 없습니다. 수정된 제안은 대상 활성 계약만 제공하며, 수락되면 수정된 계약은 즉시 활성 상태가 되어 향후 데이트가 불가능합니다.
> - 수락된 공개 제안을 수정하면 비공개 제안이 되고 더 이상 자동 갱신되지 않습니다. 자동 갱신을 유지하려면 구매자가 공개 제안을 구독해야 합니다. 자세한 정보는 비공개 제안 작동 방식 섹션을 참조하세요.

제안 업그레이드, 갱신 및 수정을 생성하려면

1. AWS Marketplace Management Portal에 로그인하고 계약을 선택합니다.
2. 계약 페이지에서 계약 옆의 확인란을 선택한 다음 세부 정보 보기를 선택합니다.
3. 계약 보기 페이지에서 계약 수정을 선택합니다.
4. 계약 세부 정보 수정 페이지에서 서비스 날짜, 제품 차원, 제안 통화(AWS MP 다이렉트 PO의 경우), 결제 일정, 사용량 차원, 갱신 상태, EULA 및 제안 만료 날짜를 변경할 수도 있습니다.
5. 제안을 검토하고 제안 생성을 선택합니다.

> **Tip**
>
> 설명이 포함된 사용자 지정 제안 이름을 입력하면 제안 페이지에서 활성 제안들을 구분하는 데 도움이 됩니다. 사용자 지정 제안 이름은 구매자도 볼 수 있습니다.

AWS 에서는 자체 IDs 및 구매 주문 번호와 같은 추가 식별 세부 정보가 포함된 사용자 지정 제안 이름을 사용할 것을 권장합니다. **upgrade** 또는 **renewal** 및 사용자 지정 회사 이름과 같은 상위 수준 설명을 사용하는 것도 좋습니다. 개인 식별 데이터(예: 이름, 성, 전화 번호 또는 주소)는 사용하지 마세요. 이 필드에는 최대 150자를 입력할 수 있습니다.

수정된 제안에 대한 추가 정보

수정된 제안은 약 45분 이내에 비공개 제안 페이지에 표시됩니다. 제안을 보려면에 로그인 AWS Marketplace Management Portal 하고 제안을 선택합니다. 비공개 제안 페이지에서 구매자는 제안을 수락하거나 원래 계약을 계속할 수 있습니다.

구매자가 공개 또는 비공개 제안 업그레이드 또는 갱신을 수락하면 새 계약이 즉시 적용되며 계약은의 계약 페이지에 나열됩니다 AWS Marketplace Management Portal. 이전 계약에서 남은 예정된 지불 항목은 취소됩니다. 구매자는 비공개 제안을 수락하는 것과 동일한 방식으로 수정을 수락합니다. 비공개 제안의 구매자 경험에 대한 자세한 내용은 AWS Marketplace 구매자 설명서의 비공개 제안을 참조하세요.

구매자가 만료되기 전에 공개 또는 비공개 제안 업그레이드 또는 갱신을 수락하지 않는 경우 원래 계약은 변경되지 않습니다.

> **Note**
>
> SaaS 제품에 대한 Amazon SNS 알림의 경우 구매자가 수정을 수락`offer-identifier`하면 새와 함께 `subscribe-success` 메시지가 전송됩니다.

업그레이드, 갱신 및 수정에 대한 보고

공개 및 비공개 제안 업그레이드 및 갱신은 기존 판매자 보고서 및 제안과 관련된 보고서에 표시됩니다. 일별 고객 구독자 보고서 및는 매일 일별 비즈니스 보고서 생성됩니다. 월별 청구 수익 보고서는 매월 생성됩니다.

일일 고객 구독자 보고서에서 구독 의도 필드는 보고서 항목이 새 비공개 제안인지 여부를 나타냅니다. 이전 제안 ID 필드에는 이전 제안의 ID가 표시됩니다. 모든 비공개 제안은 보고서에서 "프라이빗"으로 레이블이 지정됩니다.

> **Important**
>
> 수정은 구매자의 현재 구독을 대체합니다. 기존 인보이스는 변경되지 않습니다. 그러나 수정의 결제 일정은 이전 구독에서 보류 중인 인보이스를 대체합니다.

의 AMI 기반 제품 AWS Marketplace

AWS Marketplace 판매자는 Amazon Machine Image(AMIs)를 사용하여 구매자에게 제품을 제공할 수 있습니다. AMI는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 판매자가 제품에 대한 사용자 지정 AMI를 생성하면, 구매자가 이 AMI를 사용하여 제품이 이미 설치되어 사용 가능한 상태로 EC2 인스턴스를 생성할 수 있습니다. 이 주제에서는 AMIs 사용하여 AWS Marketplace 제품을 전달하는 방법에 대한 정보를 제공합니다.

구매자가 판매자 제공 AMI를 사용하는 경우 판매자가 제품에 대해 생성한 요금 및 측정 옵션에 따라 구매자가 생성하는 인스턴스에 대한 요금이 청구됩니다. 구매자는 AMI의 새 사용자 지정 버전을 AWS만드는 등에서 다른 AMIs 사용하는 것과 동일한 방식으로 제품 AMI를 사용할 수 있습니다. AMI에서 생성된 EC2 인스턴스는 AMI 제품 코드에 따라 여전히 판매자의 제품으로 요금이 청구됩니다.

AMI 기반 제품 제공 방법

다음 방법 중 하나로 AMI 기반 제품을 제공할 수 있습니다.

- 단일 AMI - 구매자가 AMI를 선택하여 EC2 인스턴스용 템플릿으로 사용합니다. 구매자는 Amazon 머신 이미지(AMI) 제공 방법 필터를 사용하여 이러한 제품을 찾을 수 있습니다. 자세한 내용은 AMI 기반 제품 생성 단원을 참조하십시오.
- AWS CloudFormation 템플릿 - 구매자가 서로 다른 역할을 가진 여러 인스턴스 시스템을 단일 단위로 설치할 수 있는 템플릿을 생성합니다. 구매자는 CloudFormation 제공 방법 필터를 사용하여 이러한 제품을 찾을 수 있습니다. 자세한 내용은 제품에 CloudFormation 템플릿 추가 단원을 참조하세요.

추가 리소스

AMI 제품에 대한 자세한 내용은 다음 주제를 참조하세요.

AWS Marketplace

- 에 대한 제품 요금 AWS Marketplace
- AWS Marketplace Metering Service를 사용하여 AMI 제품에 대한 사용자 지정 측정 구성

AMI 기반 제품

- 의 AMI 기반 제품 이해 AWS Marketplace
- AMI 기반 제품 생성

- 판매자로서 AWS Marketplace AMI 기반 제품 관리
- 제품에 CloudFormation 템플릿 추가
- 와 함께 사용할 AMIs 빌드 모범 사례 AWS Marketplace
- 에 대한 AMI 제품 요금 AWS Marketplace
- 에서 AMI 제품에 대한 Amazon SNS 알림 수신 AWS Marketplace
- 용 AMI 제품 체크리스트 AWS Marketplace
- 에 대한 AMI 기반 제품 요구 사항 AWS Marketplace

의 AMI 기반 제품 이해 AWS Marketplace

AWS Marketplace 판매자는 Amazon Machine Image(AMIs)를 사용하여 구매자에게 제품을 제공할 수 있습니다. AMI는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 다음 섹션에서는 AMI 기반 제품 작업을 위한 주요 개념을 설명합니다.

> **Note**
>
> AMI 제품에는 하나의 AMI만 사용할 수 있지만 해당 AMI의 버전을 제품에 추가할 수 있습니다.

주제

- 제품 수명 주기
- AMI 제품 코드
- 변경 요청
- 제품 로드 양식
- 연간 계약 수정

제품 수명 주기

AMI 기반 제품에는 하나 이상의 소프트웨어 버전과 제품 전체에 대한 메타데이터가 포함됩니다. 제품을 생성할 때 제품 이름, 설명 및 요금을 AWS Marketplace 포함하여에서 해당 속성을 구성합니다. 또한 제품이 관련 검색에서 나타나도록 제품의 적절한 범주를 결정하고 키워드를 추가합니다.

소프트웨어의 첫 번째 버전도 생성합니다. 이는 AMI일 뿐이거나 구매자가 자체 AMIs를 생성하는 데 사용할 수 있는 AWS CloudFormation 템플릿 또는 소프트웨어 패키지를 포함할 수 있습니다. 자세한 내용은 AMI 기반 제품 제공 방법 단원을 참조하십시오.

유료 제품의 경우 설치한 인스턴스 수에 따라 구매자에게 요금이 청구됩니다. 제품 사용자 수와 같이 소프트웨어가 추적하는 다른 차원을 측정하려면 제품을 AWS Marketplace 측정 서비스와 통합합니다. 자세한 내용은 AWS Marketplace Metering Service를 사용하여 AMI 제품에 대한 사용자 지정 측정 구성 단원을 참조하십시오.

제품과 첫 번째 버전의 소프트웨어를 생성할 때, 처음에는 판매자의 계정만 액세스할 수 있도록 제한된 범위에 게시됩니다. 준비가 되면 AWS Marketplace 카탈로그에 게시하여 구매자가 제품을 구독하고 구매할 수 있도록 할 수 있습니다.

서버 제품 페이지를 사용하여 제품 목록을 볼 수 있습니다. 제품에는 다음 상태 중 하나가 있습니다.

- 스테이징 - 아직 판매자가 정보를 추가하고 있는 불완전한 제품. 셀프 서비스 생성 경험에서 처음으로 저장 후 종료할 때, 변경 요청이 성공하면 제출한 전체 단계의 정보가 포함된, 게시되지 않은 제품이 생성됩니다. 이 상태에서 제품에 정보를 계속 추가할 수도 있고 변경 요청을 통해 이미 제출한 세부 정보를 변경할 수도 있습니다.
- 제한 - 제품이 시스템에 제출되고 시스템의 모든 검증을 통과하면 완성되었습니다. 그 후 제품이 제한 상태로 릴리스되었습니다. 이제 판매자 계정과 판매자가 허용한 사람만 제품 세부 정보 페이지에 액세스할 수 있습니다. 세부 정보 페이지를 통해 제품을 테스트할 수 있습니다. 자세한 정보 및 도움이 필요하면 AWS Marketplace 판매자 운영 팀에 문의하세요.
- 퍼블릭 - 구매자가 제품을 살펴보고 구독할 수 있도록 제품을 게시할 준비가 되면 표시 여부 업데이트 변경 요청을 사용합니다. 이렇게 하면 AWS Marketplace 판매자 운영 팀이 정책에 따라 제품을 검토하고 감사할 수 있는 워크플로가 시작됩니다. 제품이 승인되고 변경 요청이 처리되면 제품 상태가 제한에서 퍼블릭으로 바뀝니다.
- 제한 - 신규 사용자가 제품을 구독하지 못하게 하려면 표시 여부 업데이트 변경 요청을 사용하여 제품을 제한하면 됩니다. 제한 상태는 기존 사용자는 제품을 계속 사용할 수 있다는 것을 의미합니다. 하지만 제품이 더 이상 대중에게 표시되거나 신규 사용자에게 제공되지 않습니다.

용 AMI 기반 제품의 수명 주기는 첫 번째 버전을 게시한 후 종료되지 AWS Marketplace 않습니다. 소프트웨어의 새 버전과 기본 운영 체제에 대한 보안 패치를 사용하여 제품을 최신 상태로 유지해야 합니다.

완전한 AMI 기반 제품 수명 주기의 예시로, 한 판매자가 AWS Marketplace에서 AMI 기반 제품을 판매하려 한다고 가정하겠습니다. 시간이 지나면서 판매자가 제품을 생성하고 유지 관리하는 방법은 다음과 같습니다.

1. 제품 생성 - 판매자가 제품을 생성하고 버전 1.0.0을에 게시합니다 AWS Marketplace. 구매자는 버전 1.0.0 인스턴스를 생성하여 사용할 수 있습니다.

2. 새 버전 추가 - 나중에 판매자는 제품에 새 기능을 추가하고, 새 기능이 포함된 새 버전 1.1.0을 추가합니다. 구매자는 여전히 원래 버전인 1.0.0을 사용할 수도 있고 새 버전인 1.1.0을 선택할 수도 있습니다.

> **Note**
>
> 새 제품과 달리 새 버전은 모두에게 완전히 공개됩니다. 전체 제품이 제한된 릴리스에 있는 경우에만 고객이 보지 AWS Marketplace 않고에서 테스트할 수 있습니다.

3. 제품 정보 업데이트 - 버전 1.1.0이 공개되면 판매자는 기능에 대해 설명하는 새로운 하이라이트 텍스트로 제품 정보를 업데이트하여 구매자에게 새 기능에 대해 알립니다.
4. 마이너 버전 추가 - 판매자는 버전 1.1.0의 버그를 수정하면 새 버전 1.1.1을 추가하여 공개합니다. 이제 구매자는 1.0.0, 1.1.0 또는 1.1.1 버전 중에 선택할 수 있습니다.
5. 버전 제한 - 판매자는 구매자가 버전 1.1.0을 사용할 수 없게 해야 할 정도로 버그가 심각하다고 판단하면 해당 버전을 제한합니다. 기존 구매자는 1.1.0 버전을 계속 이용할 수 있지만 신규 고객은 구매할 수 없습니다(1.0.0 또는 1.1.1 버전만 선택 가능).
6. 버전 정보 업데이트 - 기존 구매자를 돕기 위해 판매자는 1.1.1 버전으로 업그레이드할 것을 권장하는 내용으로 1.1.0의 버전 정보를 업데이트합니다.
7. 사용량 모니터링 - 구매자가 제품을 구매하고 사용할 때 판매자는를 사용하여 판매, 사용량 및 기타 지표를 모니터링합니다 AWS Marketplace 의 판매자 보고서, 데이터 피드 및 대시보드 AWS Marketplace.
8. 제품 제거 - 제품이 더 이상 필요하지 않으면 판매자가 제품을 제거합니다 AWS Marketplace.

이 예제에서는 판매자가 제품에 세 가지 버전의 AMI를 생성했지만 두 가지 버전만 신규 구매자가 사용할 수 있었습니다(제품을 삭제하기 전).

버전 또는 제품 정보를 수정하려면 AWS Marketplace Management Portal에서 변경 요청을 만들어야 합니다.

AMI 기반 제품을 생성하고 관리하는 단계에 대한 자세한 지침은 AMI 기반 제품 생성 섹션을 참조하세요.

AMI 제품 코드

AWS Marketplace에서 제품을 생성하면 제품에 고유한 제품 코드가 할당됩니다. 해당 제품 코드는 제품의 AMI와 연결되며 제품 사용량을 추적하는 데 사용됩니다. 구매자가 소프트웨어로 작업할 때 제품

코드가 자동으로 전파됩니다. 예를 들어 AMI를 구독 및 시작하고 구성하며 새 AMI를 생성합니다. 새 AMI에는 원래 제품 코드가 여전히 포함되어 있으므로 올바른 사용 추적 및 권한이 그대로 유지됩니다.

> **Note**
>
> 제품 코드는 제품의 제품 ID와 다릅니다. 의 각 제품에는 고유한 제품 ID가 할당 AWS Marketplace 됩니다. 제품 ID는 AWS Marketplace 카탈로그, 고객 청구서 및 판매자 보고서에서 제품을 식별하는 데 사용됩니다. 제품 코드는 AMI에서 생성된 인스턴스에 인스턴스 메타데이터로 연결됩니다. 해당 제품 코드가 있는 AMI를 사용하여 인스턴스를 생성하면 고객은 연결된 제품 ID가 표시된 청구서를 받게 됩니다. 제품을 생성한 후 해당 제품의 AWS Marketplace Management Portal 페이지에서 제품 코드와 제품 ID를 찾습니다.

판매자의 소프트웨어는 런타임에 인스턴스 메타데이터에서 실행 중인 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스의 제품 코드를 가져올 수 있습니다. 판매자는 제품을 시작할 때 제품을 검증하는 등 제품 코드를 사용하여 보안을 강화할 수 있습니다. 테스트를 위해 제품이 제한 상태로 게시될 때까지는 AMI 제품 코드에 대한 API 직접 호출이 불가능합니다. 제품 코드 확인에 대한 자세한 내용은 AWS Marketplace AMI에서 소프트웨어가 실행되고 있는지 확인 섹션을 참조하세요.

변경 요청

에서 제품 또는 버전을 변경하려면를 통해 변경 요청을 AWS Marketplace제출합니다 AWS Marketplace Management Portal. 변경 요청은 대기열에 추가되며 요청 유형에 따라 변경 요청이 해결될 때까지 몇 분에서 며칠까지 걸릴 수 있습니다. AWS Marketplace Management Portal에서 요청 상태를 확인할 수 있습니다.

> **Note**
>
> 외에도 AWS Marketplace Catalog API를 사용하여 변경 요청을 생성할 수도 AWS Marketplace Management Portal있습니다.

AMI 기반 제품에 대해 요청할 수 있는 변경 유형은 다음과 같습니다.

- 구매자에게 표시되는 제품 정보를 업데이트합니다.
- 구매자에게 표시되는 버전 정보를 업데이트합니다.
- 새로운 제품 버전을 추가합니다.

- 신규 구매자가 더 이상 해당 버전에 액세스할 수 없도록 버전을 제한합니다.
- 제품을 사용할 수 AWS 리전 있는를 업데이트합니다.
- 제품의 요금 및 인스턴스 유형을 업데이트합니다.
- 에서 제품을 제거합니다 AWS Marketplace.

> **Note**
>
> 일부 변경 요청은 제품 로드 양식을 사용하여 요청을 생성해야 합니다. 자세한 내용은 제품 로드 양식 단원을 참조하세요.

변경 요청 업데이트

업데이트로 시작하는 변경 요청은 현재 세부 정보를 로드합니다. 그러면 기존 세부 정보를 덮어쓰는 업데이트를 수행할 수 있습니다.

변경 요청 추가 또는 제한

추가 및 제한 요청 페어는 각 요청이 성공한 후에 프로비저닝되는 단계와 업데이트를 위한 것입니다. 셀프 서비스 환경에서 저장 후 종료 및 제출 작업을 선택한 후 요청이 성공합니다.

예를 들어 제품이 추가된 후 AMI 자산이 인스턴스 및 리전에 프로비저닝되면 제품을 완전히 제거할 수는 없고 제한하는 것만 가능합니다. 즉, 기존 구독자와 사용자는 구독 또는 계약이 만료될 때까지 제품을 계속 사용할 수 있습니다. 하지만 제한 상태인 제품에 신규 구독자를 추가할 수 없습니다.

제품 로드 양식

일반적으로 AWS Marketplace Management Portal를 사용하여 제품을 생성하거나 편집합니다. 그러나 몇 가지 작업에서는 제품 로드 양식(PLF)을 사용해야 합니다.

PLF는 제품에 대한 모든 정보가 들어 있는 스프레드시트입니다. PLF를 가져오려면 다음을 수행할 수 있습니다.

- 의 제품 세부 정보 페이지에서 기존 제품의 PLF를 다운로드합니다 AWS Marketplace Management Portal.
- PLF가 필요한 작업의 메뉴 항목을 선택합니다. 예를 들어, 월별 청구 서버 제품을 새로 생성하는 경우 시스템에서 적절한 PLF를 다운로드하라는 메시지를 표시합니다.

기존 제품을 편집하는 작업인 경우 해당 제품에 대한 정보가 PLF에 미리 채워지므로 업데이트를 원하는 세부 정보만 변경하면 됩니다.

- 빈 새 PLF가 필요한 경우 AWS Marketplace Management Portal 파일 업로드 페이지로 이동합니다. 페이지에는 생성하려는 제품 유형의 PLFs 대한 링크가 포함되어 있습니다.

PLF를 완료한 후 AWS Marketplace Management Portal 파일 업로드 페이지에 업로드합니다. PLF의 지침 탭에 더 자세한 지침이 있습니다.

연간 계약 수정

시간당 연간(연간) 플랜 수정을 통해 판매자와 구매자는 기존 플랜을 다음과 같이 변경할 수 있습니다.

- Amazon EC2 인스턴스 유형 패밀리 간 전환
- Amazon EC2 인스턴스 유형 크기 간 전환
- 새 인스턴스 유형 추가
- 계약에서 기존 인스턴스 유형의 수량 증가

비례 배분된 변경 비용이 0보다 큰 경우 구매자는 변경할 수 있습니다(구독 가격을 낮출 수 없음). 새로 추가된 Amazon EC2 인스턴스의 비례 계산된 비용은 계약의 나머지 기간에 대해 조정된 인스턴스 유형의 연간 비용을 기준으로 합니다. 인스턴스 유형을 전환할 때 제거된 Amazon EC2 인스턴스 유형의 비례 계산된 비용은 새로 추가된 Amazon EC2 인스턴스 유형의 비례 계산된 비용에서 차감됩니다.

AMI 연간 제품에 대한 수정을 활성화하기 위해 추가 조치가 필요하지 않습니다. 수정은 할부 플랜을 사용하지 않는 비공개 제안의 공개 제안 및 계약에서 이루어진 모든 계약에 지원됩니다.

다음 대시보드에서 구매자의 수정 사항을 확인할 수 있습니다.

- 계약 및 갱신 대시보드 - 수정된 계약의 목록입니다.
- 청구된 수익 대시보드 - 고객에게 청구하는 요금입니다.
- 수금 및 지급 대시보드 - 지급액입니다.

AMI 기반 제품 생성

Amazon Machine Image(AMI) 셀프 서비스 경험은 제품 목록을 생성하고 변경 요청을 수행할 때 방법을 안내합니다. 셀프 서비스 환경을 사용하면 AWS Marketplace 판매자 운영 팀이 처리하는 데 필요

한 시간을 단축하여 제품 목록을 직접 업데이트할 수 있습니다. 셀프 서비스 경험의 많은 단계는의 카탈로그 시스템과 일치 AWS Marketplace하므로 AWS Marketplace 판매자 운영 팀의 처리 및 검증을 기다리는 대신 직접 검증이 용이합니다. 이 주제에서는 AMI 셀프 서비스 경험을 사용하여 단일 AMI에 대한 제품 목록을 생성하는 방법을 설명합니다. 고객은 AMI를 사용하여 제품이 이미 설치되고 구성된 Amazon EC2 인스턴스를 만듭니다.

주제

- 사전 조건
- 셀프 서비스 경험 이해
- 리스팅 만들기
- 추가 리소스

사전 조건

AMI 제품 목록을 생성하려면 먼저 다음과 같은 필수 조건을 충족해야 합니다.

1. AWS Marketplace Management Portal에 대한 액세스 권한이 있습니다. 판매자로 등록하고 판매하는 제품을 관리하는 데 사용하는 도구입니다 AWS Marketplace. 에 액세스하는 방법에 대한 자세한 내용은 섹션을 AWS Marketplace Management Portal참조하세요AWS Marketplace 판매자를 위한 정책 및 권한.
2. 판매자로 등록하고, 제품 요금을 청구하려면 세금 및 은행 정보를 제출합니다. 판매자가 되는 방법에 대해 자세히 알아보려면 AWS Marketplace 판매자로 시작하기 섹션을 참조하세요.
3. 판매하고 싶은 제품을 보유하고 있습니다. AMI 기반 제품의 경우 이는 일반적으로 서버 소프트웨어를 만들거나 수정했으며 고객이 사용할 AMI를 만들었다는 뜻입니다. 에서 사용할 AMI 준비에 대한 자세한 내용은 섹션을 AWS Marketplace참조하세요와 함께 사용할 AMIs 빌드 모범 사례 AWS Marketplace.

셀프 서비스 경험 이해

셀프 서비스 경험은에서 제품을 생성하는 과정을 안내합니다 AWS Marketplace. 단계를 진행하면서 AWS 리전인스턴스 유형 및 AMI 세부 정보와 같은 제품 정보 및 AMI 배포 설정을 지정합니다. 요금, 국가 가용성, EULA 및 환불 정책을 포함한 거래 세부 정보도 구성합니다. 옵션으로 제한된 상태인 제품에 액세스하고 테스트할 수 있는 AWS 계정 IDs 허용 목록을 지정할 수 있습니다.

시작하기 전에, 셀프 서비스 경험의 다음과 같은 주요 측면을 검토합니다.

- 현재 단계의 필수 필드를 모두 입력한 후에만 다음 단계로 이동할 수 있습니다. 각 단계가 끝날 때 페이지 수준 검증이 진행되기 때문입니다. 완료되지 않은 단계는 저장하거나 제출할 수 없습니다.
- 프로세스의 모든 단계를 완료하지 못했는데 세션을 종료해야 하는 경우 저장 후 종료를 선택하여 완료한 단계를 스테이징 영역에 제출할 수 있습니다.
- 완료되지 않아 검증을 통과하지 못한 단계는 시스템에 제출되지 않습니다. 일부만 완료된 단계는 유효하지 않으며 저장할 수 없습니다.
- 저장 후 종료를 선택하면 저장 후 종료 대화 상자에 검증 확인을 통과한 단계가 표시됩니다. 마지막으로 완료하고 검증한 단계까지 검토하고 저장하도록 선택할 수 있습니다. 검증 오류가 있거나 세부 정보가 누락된 경우 수정을 선택하여 해당 단계로 돌아갈 수 있습니다.
- 저장 후 종료하면 요청이 처리되는 동안 검토됩니다. 처리가 완료될 때까지 몇 분 또는 몇 시간이 걸릴 수 있습니다. 요청이 성공하기 전에는 단계를 계속하거나 변경할 수 없습니다. 첫 번째 저장 후 종료의 요청은 완료한 단계와 병렬로 제품을 생성하는 것입니다.
  - 요청이 성공하면 저장이 완료된 것입니다. 제품 개요 페이지에서 변경을 재개하려면 제품 생성 재개를 선택하거나 변경 요청을 사용하여 이전 세션에서 전에 제출한 세부 정보를 업데이트합니다. 재개하면 완료한 단계에 녹색 성공 레이블이 표시된 것을 볼 수 있습니다. 이전에 제출한 단계를 업데이트하려면 변경 요청을 사용합니다. 이 단계를 계속하려면 먼저 이전 저장 후 종료 요청을 완료해야 합니다.
- 모든 단계를 완료했으면 다음을 선택하여 리뷰를 볼 수 있습니다. 시스템에서 최종 검증을 수행할 것을 요청하는 제출을 선택합니다. 성공 응답을 받으면 제품이 제한 상태로 전환됩니다. 세부 정보 페이지를 보면, 이제 허용 목록에 있는 모든 사용자가 제품을 사용할 수 있다고 나옵니다. 요청이 실패해도 제품은 준비 중 상태로 유지되므로 수정한 후 다시 제출해야 합니다.

리스팅 만들기

이 섹션의 단계에서는 단일 AMI 제품의 리스팅을 만드는 방법을 설명합니다.

> **Note**
>
> 현재 단계의 필수 필드를 모두 완료한 후에만 다음 단계로 이동할 수 있습니다. 완료되지 않은 단계는 저장하거나 제출할 수 없습니다. 프로세스의 모든 단계를 완료하기 전에 세션을 종료해야 하는 경우 저장 후 종료를 선택하여 완료한 단계를 스테이징 영역에 제출할 수 있습니다. 자세한 내용은 셀프 서비스 경험 이해 단원을 참조하세요.

단일 AMI 제품을 만들려면

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 제품 메뉴에서 서버를 선택합니다. 또는 서버 제품 페이지로 직접 이동할 수도 있습니다.
3. 서버 제품 탭에서 서버 제품 생성을 선택하고 Amazon Machine Image(AMI)를 선택한 다음, 단일 AMI 제품의 라이선스 유형 중 하나를 선택합니다.
   - 자체 라이선스 사용(BYOL) - 사용자가 외부에서 사용자로부터 라이선스를 받는 제품 AWS Marketplace. 라이선스는 유료일 수도 있고 무료일 수도 있습니다.
   - 무료 - 구독자가 무료로 사용할 수 있는 제품입니다. (연결된 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 또는 기타 AWS 리소스에 대한 요금은 계속 지불합니다.)
   - 시간당 또는 시간당-연간 지불 - 구매자가 연간 계약을 통해 시간당 또는 시간당으로 지불하는 제품.는 AMI의 제품 코드를 기반으로 측정 작업을 AWS 수행합니다.
   - 월별 결제 - AWS에서 구매자에게 월 단위로 요금을 청구하는 제품입니다.
   - 유료 사용량 - 사용자, 데이터, 대역폭 또는 호스트의 네 가지 사용량 범주 중 하나와 함께 제공한 값에 대해 소프트웨어의 요금이 직접 부과됩니다. 제품에 대해 최대 24개의 차원을 정의할 수 있습니다. 모든 요금은 고객에 의해 발생합니다.
   - 계약 요금이 적용된 AMI - 구매자가 선결제 요금을 지불하는 AWS CloudFormation 스택이 포함된 단일 AMI 제품 또는 단일 AMI입니다.
4. 셀프 서비스 경험은 AWS Marketplace 목록 생성 단계를 안내합니다. 제품 정보(메타데이터), 제품 배포 세부 정보(AWS 리전, 인스턴스 및 AMI 세부 정보) 및 공개 제안 세부 정보(가격, EULA, 국가별 가용성, EULA, 환불)를 입력해야 합니다. 선택 사항으로, 허용 목록에 계정을 추가하여 제품을 테스트할 수 있습니다. 프로세스의 다음 단계로 이동하려면 각 단계를 완료해야 합니다.

   > **Note**
   >
   > 프로세스의 모든 단계를 완료하지 못했는데 세션을 종료해야 하는 경우 저장 후 종료 기능을 사용하여 완료한 단계를 스테이징 영역에 제출할 수 있습니다. 이렇게 하면 입력한 정보를 검증하라는 요청이 생성됩니다. 요청이 진행 중인 동안에는 제품을 편집할 수 없습니다. 요청이 성공하면 제품 생성 재개를 선택하여 제품을 계속 생성할 수 있습니다.
   > 요청이 실패하면 검증 오류로 인해 제품이 업데이트되지 않았다는 뜻입니다. 이 내용은 제품의 요청 로그에서 확인할 수 있습니다. 오류를 보는 요청을 선택하고, 작업에서 새로 복사를 선택하여 오류를 수정하고 요청을 다시 제출할 수 있습니다. 단계를 재개하면 마지막 세션에서 저장한 단계의 다음 단계부터 계속할 수 있습니다. 이전 단계를 업데이트하려면

제품 개요 페이지로 이동한 후 변경 요청을 제출하여 이전에 제출한 단계를 업데이트합니다.

5. 모든 변경 요청 단계에 필요한 정보를 입력한 후 제출을 선택합니다. 이 제출은 AWS Marketplace 카탈로그 시스템에 정보를 검증하고 검증에 통과할 경우 제품을 제한 상태로 릴리스하라는 요청을 생성합니다. 요청이 처리되는 동안에는 제품을 편집할 수 없습니다. 요청이 성공하면 제품이 제한 상태로 전환됩니다.

   - 제품이 처음 게시되면 사용자 AWS 계정 (제품을 생성하는 데 사용한 계정)와 AWS Marketplace 판매자 운영 팀의 테스트 계정만 액세스할 수 있습니다. 서버 제품 페이지에서 제품을 보는 경우에서 보기를 AWS Marketplace 선택하여 구매자를 AWS Marketplace 위해에 표시되는 제품 세부 정보를 볼 수 있습니다. 이 세부 정보 목록은 다른 AWS Marketplace 사용자에게 표시되지 않습니다.
   - 이 기능을 사용하면 제품을 테스트(및 여러 버전을 게시하여 테스트)한 후 공개적으로 릴리스할 수 있습니다.

6. 제품을 제한된 상태로 테스트하고 AWS Marketplace AMI 기반 제품 요구 사항 및 제품 체크리스트를 따르는지 확인합니다. 그런 다음, 제품을 퍼블릭에 게시하도록 요청하려면 표시 여부 업데이트를 선택합니다. AWS Marketplace 판매자 운영 팀은 공개로 승인하기 전에 제품을 검토해야 합니다.

   > **Note**
   >
   > 제품 확인 및 게시는 AWS Marketplace 판매자 운영 팀에서 처리하는 수동 프로세스입니다. 오류가 없으면 초기 제품 버전을 게시하는 데 7~10영업일이 소요될 수 있습니다. 시기에 대한 자세한 내용은 시간 및 기대를 참조하세요.

단일 AMI 제품 및 제품 정보를 준비하고 제출하는 자세한 내용은 추가 리소스 섹션을 참조하세요.

추가 리소스

게시할 제품 정보를 준비하고 제출하는 방법에 대한 자세한 내용은 다음 리소스를 참조하세요.

- 용 제품 준비 AWS Marketplace
- 에 게시할 제품 제출 AWS Marketplace

단일 AMI 제품을 AWS Marketplace에 제출할 수 있도록 준비하는 방법에 대한 자세한 내용은 다음 리소스를 참조하세요.

- 와 함께 사용할 AMIs 빌드 모범 사례 AWS Marketplace
- 용 AMI 제품 체크리스트 AWS Marketplace
- 에 대한 AMI 기반 제품 요구 사항 AWS Marketplace

제품에 CloudFormation 템플릿 추가

AWS Marketplace 판매자는 AWS CloudFormation 템플릿을 사용하여 AWS Marketplace 구매자에게 제공되는 AMI 기반 제품을 나열할 수 있습니다. AMI 기반 제품에 CloudFormation 템플릿을 추가하면 구매자가 리소스와 종속성을 수동으로 구성하지 않고도 솔루션을 배포할 수 있습니다. 템플릿을 사용하여 제품에 대한 클러스터 또는 분산형 아키텍처를 정의할 있고, 또는 다른 AMI 조합이나 제품 구성을 선택할 수 있습니다. 하나의 AMI 솔루션에는 최대 3개의 CloudFormation 템플릿이 포함될 수 있습니다.

CloudFormation 템플릿을 구성하여 단일 Amazon Machine Image(AMI)를 연결된 구성 파일 및 Lambda 함수와 함께 제공할 수 있습니다. 또한 각 템플릿에 대한 아키텍처 다이어그램을 포함해야 합니다.

주제

- CloudFormation 템플릿 준비
- 아키텍처 다이어그램
- 기존 제품의 CloudFormation 템플릿 변환
- 서버리스 애플리케이션 구성 요소 추가

CloudFormation 템플릿 준비

CloudFormation 템플릿을 빌드하려면 템플릿 사전 조건을 충족하고 필수 입력 및 보안 파라미터를 제공해야 합니다. CloudFormation 템플릿을 제출할 때 다음 섹션의 지침을 따릅니다.

템플릿 사전 조건

- 제품에 대해 AWS 리전 활성화된 모든의 AWS CloudFormation 콘솔을 통해 템플릿이 성공적으로 시작되었는지 확인합니다. TaskCat 도구를 사용하여 템플릿을 테스트할 수 있습니다.

- CloudFormation 템플릿에 포함된 AMI는 게시하는 제품의 AMI이거나 최신 Amazon Linux 2와 같은 AWS 관리형 AMI여야 합니다. 사용자 또는 다른 제3자가 소유하고 공유하는 AMI 또는 커뮤니티 AMI는 포함하지 마세요. AWS 관리형 AMI를 사용하려면 AMI IDs 하드코딩하는 대신 AWS Systems Manager Parameter Store에서 퍼블릭 파라미터를 사용합니다. 예를 들어, AMI ID를 지정하는 CloudFormation 템플릿 내에서 동적 참조 `ImageId: '{{resolve:ssm:/aws/service/ecs/optimized-ami/amazon-linux-2/recommended/image_id}}'`를 사용합니다.
- 특정 가용 영역(AZ) 사용 시 의존하지 않도록 템플릿을 빌드합니다. 모든 고객이 모든 AZ에 액세스할 수 있는 것이 아니고, AZ는 다른 계정에 대해 다르게 매핑됩니다.
- Auto Scaling 그룹을 사용하여 클러스터 솔루션을 빌드하는 경우 규모 조정 이벤트를 고려하는 것이 좋습니다. 새 노드가 자동으로 실행 중인 클러스터에 연결됩니다.
- 단일 노드 제품인 경우에도 Auto Scaling 그룹을 사용하는 것이 좋습니다.
- 솔루션에 여러 인스턴스의 클러스터가 포함된 경우 낮은 네트워크 지연 시간, 높은 네트워크 처리량 또는 인스턴스에서 두 가지 모두를 원할 때 배치 그룹 사용을 고려하세요.
- AWS Marketplace 팀이 쉽게 검토하고 고객에게 투명성을 제공하려면 UserData 섹션에 의견을 추가하는 것이 좋습니다.

AMI 세부 정보 요구 사항

> **Note**
>
> 판매자 포털의 서버 제품 페이지에서 CloudFormation을 사용하여 Amazon Machine Image(AMI) 또는 AMI를 생성하고 제품 로드 양식을 다운로드하라는 메시지가 표시되면 the section called “제품 로드 양식을 사용한 AMI 세부 정보 요구 사항” 대신 섹션을 참조하세요.

AWSAWS::EC2::Instance2::Instance, AWS::AutoScaling::LaunchConfiguration 및 AWS::EC2::LaunchTemplate 리소스와 같은 ECAWS::EC2::LaunchTemplate 인스턴스에 AMI를 배포하는 리소스의 `ImageId` 속성을 지정할 때는 템플릿 파라미터를 참조해야 합니다. AWS::AutoScaling::LaunchConfiguration 파라미터 유형은 `AWS::EC2::Image::Id`, `AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`또는 중 하나여야 합니다`String`.

이 템플릿 파라미터에 유효한 파라미터 이름을 지정할 수 있습니다.는 템플릿을 자체 Amazon S3 버킷에 AWS Marketplace 복사하고 지정된 파라미터를 AWS Systems Manager 파라미터 스토어 파라미터로 바꿉니다. AWS Marketplace 또한는 설명 및 제약 조건 텍스트를 업데이트하여 템플릿을 배포하는 구매자가 올바른 값을 명확하게 알 수 있도록 합니다. 구매자가 템플릿을 배포하면 해당 파라미터는 게시된 제품의 AWS 리전특정 AMI ID로 확인됩니다.

다음 템플릿 예제에서는 내장 함수 Ref를 사용하여 템플릿 파라미터를 참조하는 `ImageId` 속성을 보여줍니다.

YAML 예제:

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

JSON 예제:

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

루트 스택 대신 중첩 스택 내에 EC2 인스턴스를 배포하는 경우 AMI ID는 루트 스택에서 값을 동적으로 상속해야 합니다. 루트 스택에서 템플릿 파라미터 값을 설정하면이 중첩 스택에 사용된 AMI ID가 재정의되도록 루트 및 중첩 스택을 편집합니다.

제품 로드 양식을 사용한 AMI 세부 정보 요구 사항

> **Note**
>
> 판매자 포털의 서버 제품 페이지에서 CloudFormation을 사용하여 Amazon Machine Image(AMI) 또는 AMI를 생성하고 제품 로드 양식을 다운로드하라는 메시지가 즉시 표시되지 않는 경우 the section called "AMI 세부 정보 요구 사항" 대신 섹션을 참조하세요.

AMI는 각 리전의 매핑 테이블에 있어야 합니다. AWS Marketplace 팀은 AMI IDs 후 업데이트합니다. 소스 AMI는 `us-east-1` 리전에 있어야 합니다. 다른 리전에서는 자리 표시자를 사용할 수 있습니다.

YAML 예제:

```
Mappings:
  RegionMap:
      us-east-1:
          ImageId: ami-0123456789abcdef0
      us-west-1:
          ImageId: ami-xxxxxxxxxxxxxxxxx
      eu-west-1:
          ImageId: ami-xxxxxxxxxxxxxxxxx
      ap-southeast-1:
          ImageId: ami-xxxxxxxxxxxxxxxxx
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap
        - RegionMap
        - !Ref AWS::Region
        - ImageId
```

중첩 스택 템플릿에 대한 요구 사항

> **Note**
>
> 이 섹션은 제품 로드 양식을 사용하지 않는 요금 모델에만 적용됩니다. 제품 로드 양식을 사용하는 요금 모델의 경우 중첩 스택 `TemplateURL` 속성에는 고정 문자열만 허용됩니다.

템플릿에 중첩 스택이 포함된 경우 중첩 스택 리소스의 `TemplateURL` 속성은 Amazon S3 버킷 이름, 버킷 리전 및 Amazon S3 객체 키 접두사에 대한 템플릿 파라미터를 참조해야 합니다. 버킷 이름의 파라미터 이름은 , `MPS3BucketName`버킷 리전은 `MPS3BucketRegion`, 객체 키 접두사는 이어야 합니다`MPS3KeyPrefix`.

중첩된 템플릿이 저장된 Amazon S3 버킷에 맞게 이러한 파라미터의 기본값을 설정합니다. 모든 중첩 템플릿은 공개적으로 액세스할 수 있어야 합니다. 게시를 위해 템플릿을 제출하면는 템플릿을 자체 Amazon S3 버킷에 AWS Marketplace 복사하고 기본값과 허용된 값이 복사본이 저장된 위치에 맞게 설정되도록 세 파라미터의 속성을 수정합니다. AWS Marketplace 또한는 설명 및 제약 조건 텍스트를 업데이트하여 템플릿을 배포하는 구매자에게 올바른 값을 명확히 합니다.

여러 수준의 중첩 스택이 있는 경우 추가 중첩 스택을 생성하는 모든 중첩 스택은 `TemplateURL` 속성이 루트 스택에서 Amazon S3 버킷 이름, Amazon S3 버킷 리전 및 Amazon S3 객체 키의 값을 동적으로 상속하도록 구성해야 합니다. 루트 스택에서 템플릿 파라미터 , 및의 값을 설정하여 `MPS3KeyPrefix` 추가 중첩 스택을 생성하도록 루트 `MPS3BucketName` `MPS3BucketRegion`및 중첩 스택을 편집합니다.

다음 템플릿 예제에서는 내장 함수 Fn::Sub를 사용하여 템플릿 파라미터를 참조하는 `TemplateURL` 속성을 보여줍니다.

YAML 예제:

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: AWS Marketplace Parameters
        Parameters:
          - ImageId
          - MPS3BucketName
          - MPS3BucketRegion
          - MPS3KeyPrefix
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example1234567890
    Description: The AMI that will be used to launch EC2 resources.
  MPS3BucketName:
    Type: String
    Default: sellerbucket
```

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

JSON 예제:

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Metadata": {
        "AWS::CloudFormation::Interface": {
            "ParameterGroups": [
                {
                    "Label": {
                        "default": "AWS Marketplace Parameters"
                    },
                    "Parameters": [
                        "ImageId",
                        "MPS3BucketName",
                        "MPS3BucketRegion",
                        "MPS3KeyPrefix"
                    ]
                }
            ]
        }
    },
```

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

```
}
```

> **Note**
>
> AWS::CloudFormation::Interface는 구매자가 템플릿을 배포할 때 AWS CloudFormation 콘솔에서 파라미터를 그룹화하고 정렬하는 방법을 정의하는 데 사용됩니다.

템플릿 입력 파라미터

- 템플릿에 대한 입력 파라미터에는 AWS Marketplace 고객의 자격 증명(예: 암호, 퍼블릭 키, 프라이빗 키 또는 인증서)이 포함되어서는 안 됩니다 AWS .
- 암호와 같은 민감한 입력 파라미터의 경우 `NoEcho` 속성을 선택하고 더 강력한 정규 표현식을 활성화합니다. 기타 입력 파타미터의 경우 적절한 헬퍼 텍스트와 함께 가장 일반적인 입력을 설정합니다.
- 사용 가능한 경우 입력에 AWS CloudFormation 파라미터 유형을 사용합니다.
- `AWS::CloudFormation::Interface`를 사용하여 입력 파라미터를 그룹화 및 정렬하세요.
- 다음 입력 파라미터의 기본값을 설정하지 마세요.

> **Note**
>
> 고객이 입력 파라미터로 제공해야 합니다.

  - 퍼블릭 인터넷에서 원격 액세스 포트로 진입할 수 있는 기본 CIDR 범위
  - 퍼블릭 인터넷에서 데이터베이스 연결 포트로 진입할 수 있는 기본 CIDR 범위
  - 사용자 또는 데이터베이스의 기본 암호

네트워크 및 보안 파라미터

- 기본 SSH 포트(22) 또는 RDP 포트(3389)가 0.0.0.0에 개방되지 않았는지 확인합니다.
- 기본 가상 프라이빗 클라우드(VPC)를 사용하는 대신 적절한 액세스 통제 목록(ACL) 및 보안 그룹을 포함하여 VPC를 빌드하는 것이 좋습니다.
- 템플릿은 사용자에게 장기 액세스 키를 요청하거나 AWS 리소스에 액세스하기 위해 생성할 수 없습니다. AMI 애플리케이션이 구매자 계정의 AWS 서비스에 액세스해야 하는 경우 Amazon EC2에 대한 IAM 역할을 사용해야 합니다.

- IAM 역할 및 정책을 설정하여 최소 권한을 부여하고 절대적으로 필요한 경우에만 쓰기 액세스를 활성화합니다. 예를 들어 애플리케이션에 `S3:GET`, `PUT` 및 `DELETE` 작업만이 필요한 경우 이러한 작업만을 지정합니다. 이 경우 `S3:*` 사용을 권장하지 않습니다.

템플릿을 받은 후는 제품 구성 및 정보를 AWS Marketplace 검증하고 필요한 개정에 대한 피드백을 제공합니다.

아키텍처 다이어그램

각 템플릿의 아키텍처 다이어그램을 제공해야 합니다. 다이어그래밍에 대한 자세한 내용은 아키텍처 다이어그래밍이란 무엇인가요?를 참조하세요.

이 다이어그램은 다음 조건을 충족해야 합니다.

- 표준 배포를 보여줍니다 AWS.
- 리소스가 배포되는 논리적 위치를 보여줍니다. 예를 들어 Amazon EC2 인스턴스와 같은 리소스는 올바른 서브넷에 있습니다.
- AWS CloudFormation 템플릿을 통해 AWS 서비스 배포된 각에 대해 최신 AWS 제품 아이콘을 사용합니다. 현재 아키텍처 아이콘 세트를 다운로드하려면 AWS 아키텍처 아이콘을 참조하세요.
- AWS CloudFormation 템플릿에서 배포한 모든 서비스에 대한 메타데이터를 포함합니다.
- AWS CloudFormation 템플릿에 의해 배포된 모든 네트워크, VPCs 및 서브넷을 포함합니다.
- 타사 자산, API, 온프레미스, 하이브리드 자산을 비롯한 통합 지점을 표시합니다.
- 다이어그램 크기는 1100 x 700픽셀이어야 합니다. 다이어그램을 늘리거나 잘라내지 않고 원래 비율을 유지합니다.

기존 제품의 CloudFormation 템플릿 변환

> **Note**
>
> 이 섹션은 제품 로드 양식을 사용하여 템플릿을 게시하고 이제 제품 로드 양식을 사용하지 않고 해당 템플릿을 업데이트하려는 CloudFormation 제품이 포함된 기존 AMI를 보유한 판매자를 위한 것입니다. 새 제품을 게시하는 경우 CloudFormation 템플릿 준비를 참조하세요.
> 판매자 포털의 서버 제품 페이지에서 CloudFormation을 사용하여 Amazon Machine Image(AMI) 또는 AMI를 생성하고 제품 로드 양식을 다운로드하라는 메시지가 표시되면 섹션을 참조하세요the section called “제품 로드 양식을 사용한 AMI 세부 정보 요구 사항”.

셀프 서비스 환경을 사용하여 이전에 제품 로드 양식을 사용하여 게시한 기존 제품을 업데이트하려면 기존 CloudFormation 템플릿을 변경해야 합니다.

다음 표에서는 제품 로드 양식 사용과 셀프 서비스 경험 간의 차이점을 설명합니다.

	제품 로드 양식	셀프 서비스 경험
EC2 리소스의 **ImageId** 속성 값	AMI ID에 대한 매핑 테이블을 참조합니다. 자세한 내용은 the section called "제품 로드 양식을 사용한 AMI 세부 정보 요구 사항" 단원을 참조하십시오.	AMI ID에 대한 템플릿 파라미터를 참조합니다. 자세한 내용은 the section called "AMI 세부 정보 요구 사항" 단원을 참조하십시오.
중첩 스택의 **TemplateURL** 속성 값	고정 문자열이어야 하며 내장 함수를 사용할 수 없습니다.	내장 함수를 사용하여 동적일 수 있습니다. 템플릿 파라미터 세트를 참조해야 합니다. 자세한 내용은 the section called "중첩 스택 템플릿에 대한 요구 사항" 섹션을 참조하세요.

다음 예제 템플릿은 제품 로드 양식을 사용하여 템플릿을 게시한 기존 제품의 예를 보여줍니다. 이 예제에서 AMI ID는 `ami-example123456`이고 중첩된 템플릿은의 위치에 있는 판매자의 S3 버킷에 있습니다`https://sellerbucket.s3.us-east-1.amazonaws.com/sellerproductfolder/nested-template.yaml`.

제품 로드 양식과 함께 게시된 YAML 예제:

```
AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-example123456
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !FindInMap
        - RegionMap
        - !Ref AWS::Region
```

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

제품 로드 양식과 함께 게시된 JSON 예제:

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

다음 템플릿 예제에서는 셀프 서비스 경험을 사용하여 제품을 업데이트하는 데 필요한 변경 사항을 보여줍니다.

셀프 서비스 환경과 함께 게시된 YAML 예제:

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: AWS Marketplace Parameters
        Parameters:
          - ImageId
          - MPS3BucketName
          - MPS3BucketRegion
          - MPS3KeyPrefix
Parameters:
  ImageId:
    Type: AWS::EC2::Image::Id
    Default: ami-example123456
    Description: The AMI that will be used to launch EC2 resources.
  MPS3BucketName:
    Type: String
    Default: sellerbucket
    Description: Name of the S3 bucket for your copy of the nested templates.
  MPS3BucketRegion:
    Type: String
    Default: us-east-1
    Description: AWS Region where the S3 bucket for your copy of the nested templates
 is hosted.
  MPS3KeyPrefix:
    Type: String
    Default: sellerproductfolder/
    Description: S3 key prefix that is used to simulate a folder for your copy of the
 nested templates.
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
```

```
      TemplateURL: !Sub https://${MPS3BucketName}.s3.${MPS3BucketRegion}.
${AWS::URLSuffix}/${MPS3KeyPrefix}nested-template.yaml
```

셀프 서비스 환경과 함께 게시된 JSON 예제:

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

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

서버리스 애플리케이션 구성 요소 추가

> **Important**
>
> AWS Marketplace 는 더 이상 리소스를 배포하는 CloudFormation 템플릿으로 신제품 게시를 지원하지 않습니다 AWS Serverless Application Repository. 판매자는 향후 발표될 날짜 AWS Serverless Application Repository 까지 리소스를 배포하는 CloudFormation 템플릿으로 기존 제품을 계속 게시할 수 있습니다.

제품에 서버리스 구성 요소가 통합된 상태에서 AWS CloudFormation 템플릿을 사용하여 제공되는 Amazon Machine Image(AMI)가 포함된 제품을 생성할 수 있습니다. 예를 들어 AMI가 컨트롤러 서버로 구성된 제품을 생성하고 AWS CloudFormation 스택으로 제공할 수 있습니다. 스택을 생성하는 데 사용되는 AWS CloudFormation 템플릿에는 서버의 이벤트에 의해 트리거되는 AWS Lambda 함수를

설정하는 정의가 포함될 수 있습니다. 이 접근 방식을 사용하여 제품을 설계하면 아키텍처를 간소화할 수 있으므로 구매자가 손쉽게 시작할 수 있습니다. 또한 제품을 손쉽게 업데이트할 수 있습니다. 다음 섹션에서는 이러한 유형의 제품을 생성하고 제공하는 방법을 보여줍니다.

제품의 AMI 생성에 대한 자세한 내용은 섹션을 참조하세요의 AMI 기반 제품 AWS Marketplace. 제품의 AWS CloudFormation 템플릿 작성에 대한 자세한 내용은 섹션을 참조하세요제품에 CloudFormation 템플릿 추가.

서버리스 애플리케이션을 정의할 때에 저장하는 AWS Serverless Application Model (AWS SAM) 템플릿을 사용합니다 AWS Serverless Application Repository. AWS SAM 는 서버리스 애플리케이션을 빌드하기 위한 오픈 소스 프레임워크입니다. 배포 중에 AWS Serverless Application Model 는 구문을 AWS SAM 변환하고 AWS CloudFormation 구문으로 확장합니다. AWS Serverless Application Repository 는 서버리스 애플리케이션을 위한 관리형 리포지토리입니다. 이 리포지토리를 사용하면 재사용 가능한 애플리케이션을 저장 및 공유할 수 있으므로 구매자가 서버리스 아키텍처를 어셈블하고 배포할 수 있습니다.

> **Note**
>
> - AWS Marketplace 는 목록이 생성되기 전에 제품을 검토하고 검증합니다. 제안이 나열되기 전에 해결해야 하는 문제가 있는 경우 이메일 메시지를 보내 드립니다.
> - 구독 이행의 일환으로 AMIs, 서버리스 애플리케이션 및 AWS CloudFormation 템플릿을 각각 AWS Marketplace에 있는 소유 리포지토리에 복사합니다 AWS 리전. 구매자가 제품을 구독할 경우 구매자에게 액세스 권한을 부여하고 소프트웨어 업데이트 시 구매자에게 알립니다.

주제

- 1단계: 서버리스 애플리케이션 만들기
- 2단계: 리포지토리에 애플리케이션 게시
- 3단계: CloudFormation 템플릿 만들기
- 4단계: CloudFormation 템플릿과 구성 파일 제출
- 5단계: AWS Serverless Application Repository 애플리케이션 권한 업데이트
- 6단계: AMI 공유
- 7단계: AMI 및 서버리스 애플리케이션과 함께 CloudFormation 제품 제출

1단계: 서버리스 애플리케이션 만들기

첫 번째 단계는 서버리스 애플리케이션을 생성하는 데 사용되는 AWS Lambda 함수를 패키징하는 것입니다. 애플리케이션은 작업을 수행하는 데 함께 작동하는 Lambda 함수, 이벤트 소스 및 기타 리소스의 조합입니다. 서버리스 애플리케이션은 하나의 Lambda 함수만큼 간단하거나 APIs, 데이터베이스 및 이벤트 소스 매핑과 같은 다른 리소스와 함께 여러 함수를 포함할 수 있습니다.

AWS SAM 를 사용하여 서버리스 애플리케이션의 모델을 정의합니다. 속성 이름 및 유형에 대한 설명은 GitHub의 AWS::Serverless::Application in AWS Labs를 참조하세요. 다음은 단일 Lambda 함수 및 AWS Identity and Access Management (IAM) 역할이 있는 AWS SAM 템플릿의 예입니다.

```
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"
```

2단계: 리포지토리에 애플리케이션 게시

애플리케이션을 게시하려면 우선 애플리케이션 코드를 업로드합니다. 코드 아티팩트(예: Lambda 함수, 스크립트, 구성 파일)를 계정 소유의 Amazon S3 버킷에 저장합니다. 애플리케이션을 업로드하면 처음에는 프라이빗으로 설정됩니다. 즉, 애플리케이션을 AWS 계정 생성한 에서만 사용할 수 있습니다. 업로드한 아티팩트에 액세스할 수 있는 AWS Serverless Application Repository 권한을 부여하는 IAM 정책을 생성해야 합니다.

서버리스 애플리케이션을 서버리스 애플리케이션 리포지토리에 게시하려면

1. https://console.aws.amazon.com/s3/에서 S3 콘솔을 엽니다.
2. 애플리케이션을 패키징하는 데 사용한 Amazon S3 버킷을 선택합니다.
3. 권한 탭을 선택합니다.
4. 버킷 정책(Bucket Policy)을 선택합니다.
5. 다음 정책 설명 예시를 붙여넣습니다.

> **Note**
>
> 이 정책 설명 예시는 다음 단계에서 `aws:SourceAccount` 및 `Resource` 값이 업데이트될 때까지 오류를 일으킵니다.

JSON

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

```
            "Principal": {
                "Service":  "serverlessrepo.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition" : {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

a. `Resource` 속성 값의 amzn-s3-demo-bucket을 버킷의 버킷 이름으로 바꿉니다.

b. `Condition` 요소의 *123456789012*을 AWS 계정 ID로 바꿉니다. `Condition` 요소는 AWS Serverless Application Repository 만 지정된의 애플리케이션에 액세스할 수 있는 권한을 갖도록 합니다 AWS 계정.

6. 저장을 선택합니다.
7. 에서 AWS Serverless Application Repository 콘솔을 엽니다https://console.aws.amazon.com/serverlessrepo.
8. Applications(애플리케이션) 페이지에서 Publish application(애플리케이션 게시)을 선택합니다.
9. 필수 필드와 선택 필드(해당하는 경우)를 완료합니다. 필수 필드는 다음과 같습니다.
   - 애플리케이션 이름
   - 작성자
   - 설명
   - 소스 코드 URL
   - SAM 템플릿
10. Publish Application(애플리케이션 게시)을 선택합니다.

애플리케이션의 후속 버전을 게시하는 방법

1. 에서 AWS Serverless Application Repository 콘솔을 엽니다https://console.aws.amazon.com/serverlessrepo.
2. 탐색 창의 My Applications(내 애플리케이션)에서 애플리케이션을 선택합니다.

3. [새 버전 발행]을 선택합니다.

자세한 내용은 AWS SAM CLI를 사용하여 서버리스 애플리케이션 게시를 참조하세요.

3단계: CloudFormation 템플릿 만들기

CloudFormation 템플릿을 빌드하려면 템플릿 사전 조건을 충족하고 필수 입력 및 보안 파라미터를 제공해야 합니다. 자세한 내용은 AWS CloudFormation 사용 설명서의 템플릿 구조를 참조하세요.

CloudFormation 템플릿에서 서버리스 애플리케이션 및 AMI를 참조할 수 있습니다. 또한 중첩된 CloudFormation 템플릿을 사용하고 루트 템플릿과 중첩된 템플릿에서 서버리스 애플리케이션을 참조할 수 있습니다. 서버리스 애플리케이션을 참조하려면 AWS SAM 템플릿을 사용합니다. 에서 애플리케이션에 대한 AWS SAM 템플릿을 자동으로 생성할 수 있습니다 AWS Serverless Application Repository. 다음은 템플릿의 예입니다.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: An example root template for a SAR application

Resources:
  SampleSARApplication:
    Type: AWS::Serverless::Application
    Properties:
      Location:
         ApplicationId: arn:aws:serverlessrepo:us-east-1:1234567890:applications/
TestApplication
         SemanticVersion: 1.0.0
  SampleEC2Instance:
    Type: AWS::EC2::Instance
      Properties:
        ImageId: "ami-79fd7eee"
        KeyName: "testkey"
        BlockDeviceMappings:
          - DeviceName: "/dev/sdm"
            Ebs:
              VolumeType: "io1"
              Iops: "200"
              DeleteOnTermination: "false"
              VolumeSize: "20"
          - DeviceName: "/dev/sdk"
            NoDevice: {}
```

AWS SAM 템플릿에는 다음 요소가 포함되어 있습니다.

- `ApplicationID` - 애플리케이션의 Amazon 리소스 이름(ARN)입니다. 이 정보는 AWS Serverless Application Repository의 내 애플리케이션 섹션에 있습니다.
- `SemanticVersion` - 서버리스 애플리케이션의 버전입니다. AWS Serverless Application Repository의 내 애플리케이션 섹션에서 찾을 수 있습니다.
- `Parameter`(선택 사항) - 애플리케이션 파라미터입니다.

> **Note**
>
> `ApplicationID` 및 `SemanticVersion`의 경우 내장 함수가 지원되지 않습니다. 이러한 문자열은 하드코딩해야 합니다. `ApplicationID`는 복제될 때 업데이트됩니다 AWS Marketplace.

CloudFormation 템플릿의 구성 및 스크립트 파일을 참조할 계획이면 다음 형식을 사용합니다. 중첩 템플릿(`AWS::Cloudformation::Stack`)의 경우 내장 함수 없는 `TemplateURLs`만 지원됩니다. 템플릿의 `Parameters` 내용을 메모해 둡니다.

```
AWSTemplateFormatVersion: '2010-09-09'
Metadata:
  Name: Seller test product
Parameters:
  CFTRefFilesBucket:
    Type: String
    Default: "seller-bucket"
  CFTRefFilesBucketKeyPrefix:
    Type: String
    Default: "cftsolutionFolder/additionCFfiles"
Resources:
  TestEc2:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        addCloudAccount:
          files:
            /etc/cfn/set-aia-settings.sh:
              source:
                Fn::Sub:
```

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

4단계: CloudFormation 템플릿과 구성 파일 제출

CloudFormation 템플릿과 구성 및 스크립트 파일을 제출하려면 이러한 파일이 저장된 Amazon S3 버킷에 대한 읽기 권한을 AWS Marketplace 에 부여합니다. 이렇게 하려면 다음 권한을 포함하도록 버킷 정책을 업데이트합니다.

JSON

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

```
    ]
}
```

5단계: AWS Serverless Application Repository 애플리케이션 권한 업데이트

AWS Serverless Application Repository 애플리케이션을 제출하려면 애플리케이션을 읽을 수 있는 AWS Marketplace 권한을 부여 AWS Marketplace해야 합니다. 이렇게 하려면 서버리스 애플리케이션에 연결된 정책에 권한을 추가합니다. 두 가지 방법으로 애플리케이션 정책을 업데이트할 수 있습니다.

- AWS Serverless Application Repository로 이동합니다. 목록에서 서버리스 애플리케이션을 선택합니다. 공유 탭을 선택하고 문 생성을 선택합니다. 문 구성 페이지의 계정 ID 필드에 서비스 주체 **`assets.marketplace.amazonaws.com`**을 입력합니다. 그런 다음 저장을 선택합니다.
- 다음 AWS CLI 명령을 사용하여 애플리케이션 정책을 업데이트합니다.

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

6단계: AMI 공유

빌드되고에 제출된 모든 AMIs 모든 제품 정책을 준수해야 AWS Marketplace 합니다. 셀프 서비스 AMI 검사는 AWS Marketplace Management Portal에서 사용할 수 있습니다. 이 기능을 사용하면 AMI 검사를 시작할 수 있습니다. 단일 위치에서 분명한 피드백과 함께 검사 결과를 신속하게(대개 1시간 미만) 받을 수 있습니다. AMI를 성공적으로 스캔한 후 제품 로드 양식을 업로드하여 AWS Marketplace 판매자 운영 팀이 처리할 수 있도록 AMI를 제출합니다.

7단계: AMI 및 서버리스 애플리케이션과 함께 CloudFormation 제품 제출

제품을 제출하기 전에 다음에 유의하세요.

- 각 템플릿의 아키텍처 다이어그램을 제공해야 합니다. 다이어그램은 CloudFormation 템플릿을 통해 배포된 AWS 각 AWS 서비스의 제품 아이콘을 사용해야 합니다. 또한 다이어그램에는 서비스의 메타데이터가 포함되어야 합니다. 공식 AWS 아키텍처 아이콘을 다운로드하려면 AWS 아키텍처 아이콘을 참조하세요.

- 구매자에게 표시되는 각 템플릿의 인프라 추정 비용은 AWS 요금 계산기를 사용하여 제공하는 추정치를 바탕으로 계산됩니다. 추정에는 일반 배포에 대한 기본 값과 함께 템플릿의 일부로 배포되는 서비스의 목록이 포함되어야 합니다.
- 제품 로드 양식을 완료합니다. 제품 로드 양식은 AWS Marketplace Management Portal에서 찾을 수 있습니다. 단일 AMI 제품과 복수 AMI 제품에 다른 제품 로드 양식이 필요합니다. 제품 로드 양식에서 CloudFormation 템플릿의 퍼블릭 URL을 제공합니다. CloudFormation 템플릿은 퍼블릭 URL 형식으로 제출해야 합니다.
- AWS Marketplace Management Portal 를 사용하여 목록을 제출합니다. Assets(자산)에서 File upload(파일 업로드)를 선택하고 파일을 첨부한 다음 Upload(업로드)를 선택합니다. 템플릿과 메타데이터를 수신하면가 요청 처리를 AWS 시작합니다.

목록을 제출한 후는 제품 로드 양식을 AWS Marketplace 검토하고 검증합니다. 또한는 AMIs 및 서버리스 애플리케이션을 AWS Marketplace 리전화하고 사용자를 대신하여 AWS CloudFormation 템플릿에 대한 리전 매핑을 업데이트합니다. 문제가 발생하면 AWS Marketplace 판매자 운영 팀에서 이메일로 연락을 드립니다.

판매자로서 AWS Marketplace AMI 기반 제품 관리

AWS Marketplace 판매자는 단일 Amazon Machine Image(AMIs) 제품을 관리하고 업데이트할 수 있습니다. AMI 기반 제품에는 하나 이상의 소프트웨어 버전 세트와 제품 전체에 대한 메타데이터가 포함됩니다. 제품을 생성할 때 제품 이름, 설명 및 요금을 AWS Marketplace포함하여에서 해당 속성을 구성합니다. 또한 제품이 관련 검색에서 나타나도록 제품의 적절한 범주를 결정하고 키워드를 추가합니다. 단일 AMI 제품을 만든 후 변경 요청을 제출하여 제품 또는 버전을 변경할 수 있습니다.

AMI 기반 제품에 대해 요청할 수 있는 변경 유형은 다음과 같습니다.

- 구매자에게 표시되는 제품 정보를 업데이트합니다.
- 구매자에게 표시되는 버전 정보를 업데이트합니다.
- 새로운 제품 버전을 추가합니다.
- 신규 구매자가 더 이상 해당 버전에 액세스할 수 없도록 버전을 제한합니다.
- 제품을 사용할 수 AWS 리전 있는를 업데이트합니다.
- 제품의 요금 및 인스턴스 유형을 업데이트합니다.
- 에서 제품을 제거합니다 AWS Marketplace.

다음 주제는 단일 AMI 제품을 관리하고 업데이트하는 방법을 보여줍니다.

주제

- 에서 AMI 기반 제품에 대한 변경 요청 생성 AWS Marketplace
- AMI 기반 제품 가시성 업데이트
- 에 대한 AMI 인스턴스 추가 및 제한 AWS Marketplace
- 에서 AMI 기반 제품의 버전 관리 AWS Marketplace
- 에서 AMI 기반 제품 정보 업데이트 AWS Marketplace
- 및 국가별 AMI 기반 제품 가용성 AWS 리전 관리
- AMI 기반 AWS Marketplace 제품에 대한 최종 사용자 라이선스 계약(EULA) 업데이트
- 에서 AMI 기반 제품에 대한 환불 정책 업데이트 AWS Marketplace
- AMI에 대한 AWS Marketplace 액세스 권한 부여
- AWS Marketplace에서 제품 제거
- AWS Marketplace에서의 변경 요청과 관련하여 흔히 발생하는 오류 해결

에서 AMI 기반 제품에 대한 변경 요청 생성 AWS Marketplace

에서 제품 또는 버전을 변경하려면를 통해 변경 요청을 AWS Marketplace제출합니다 AWS Marketplace Management Portal. 변경 요청은 대기열에 추가되며 요청 유형에 따라 변경 요청이 해결될 때까지 몇 분에서 며칠까지 걸릴 수 있습니다. AWS Marketplace Management Portal에서 요청 상태를 확인할 수 있습니다. 이 주제에서는 셀프 서비스 경험을 사용하는 옵션을 AWS Marketplace포함하여에서 단일 AMI 제품에 대한 변경 요청을 생성하는 데 사용할 수 있는 절차를 제공합니다.

다음과 같은 상황에서 변경 요청을 만듭니다.

- 셀프 서비스 경험을 사용하여 단일 AMI 제품 목록을 생성하는 동안 진행 중인 단계를 저장했지만, 전체 프로세스를 완료하지는 못했습니다. 나머지 단계를 완료하려면 변경 요청을 생성합니다.
- 제한 또는 퍼블릭 상태인 제품의 제품 정보를 수정하려 합니다. 정보를 업데이트하려면 변경 요청을 생성합니다. AMI 기반 제품에 대해 요청할 수 있는 변경 유형에 대한 자세한 내용은 변경 요청 생성 섹션을 참조하세요.

> **Note**
>
> 외에도 AWS Marketplace Catalog API를 사용하여 변경 요청을 생성할 수도 AWS Marketplace Management Portal있습니다.

주제

- 셀프 서비스를 사용하여 변경 요청 생성
- 변경 요청 생성
- 변경 요청의 상태 가져오기
- 추가 리소스

셀프 서비스를 사용하여 변경 요청 생성

버전 또는 제품 정보를 수정하려면 AWS Marketplace Management Portal에서 변경 요청을 생성해야 합니다. 변경 요청은 제품을 변경할 때 사용하는 셀프 서비스 목록의 구성 요소입니다. 단계에서 저장 후 종료를 선택하거나 업데이트의 제출을 선택할 때마다 변경 요청을 하는 것입니다. 요청 탭에서 AWS Marketplace Management Portal 요청을 찾을 수 있습니다.

셀프 서비스를 사용하여 변경 요청을 생성하는 방법

1. AWS Marketplace Management Portal 에서를 열고 판매자 계정에 https://aws.amazon.com/marketplace/management/tour/로그인한 다음 서버 제품 페이지로 이동합니다.
2. 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 옵션을 선택합니다.
4. 변경 요청을 한 후에는 시스템에서 요청을 처리할 때까지 기다려야 하며, 이 기간에는 검토 중으로 표시됩니다. 요청이 완료되면 성공 또는 실패로 표시됩니다.
    - 요청이 제출되면 처리가 시작되고 검토 중, 변경 준비 중, 변경 사항 적용 중 상태를 거치게 됩니다.
    - 성공은 요청된 변경이 처리되었으며 변경 사항이 시스템에 반영되었다는 뜻입니다.
    - 실패는 요청에 문제가 발생하여 변경 사항이 처리되지 않았다는 뜻입니다. 상태가 실패이면 요청을 선택하여 오류 수정 방법에 대한 권장 사항을 제공하는 오류 코드를 찾을 수 있습니다. 이 시점에 오류를 해결하고 변경 요청을 새로 생성할 수 있습니다. 프로세스를 더 빠르게 진행하려면 새 요청에 복사를 선택하여 실패한 요청의 세부 정보를 복사하면 됩니다. 그런 다음, 요청을 조정하고 다시 제출할 수 있습니다.

변경 요청 생성

> **Important**
>
> 2023년 6월 15일에 AWS Marketplace 는 다음 절차를 중단합니다. 2023년 6월 15일 이후에는 the section called "셀프 서비스를 사용하여 변경 요청 생성" 절차를 사용하세요.

버전 또는 제품 정보를 수정하려면 AWS Marketplace Management Portal에서 변경 요청을 생성해야 합니다.

변경 요청을 생성하려면

1. AWS Marketplace Management Portal 에서를 열고 판매자 계정에 https://aws.amazon.com/marketplace/management/tour/로그인한 다음 서버 제품 페이지로 이동합니다.
2. 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 옵션을 선택합니다.

   대부분의 변경 요청은 사용자 인터페이스에서 양식을 작성하고 제출하면 됩니다. 하지만 특정 변경은 제품 로드 양식(PLF)을 다운로드하여 작성한 후 업로드해야 합니다. 이 스프레드시트에는 필수 정보를 입력하는 양식이 들어 있습니다. 이러한 변경 요청 중 하나를 선택하면 생성하려는 요청에 맞는 올바른 PLF를 다운로드하라는 메시지가 표시됩니다. PLF는 기존 제품 세부 정보의 정보가 미리 채워져 있습니다. 완료된 PLF를 AWS Marketplace Management Portal 파일 업로드 페이지에 업로드할 수 있습니다.

   > **Note**
   >
   > 가장 최신 PLF를 다운로드하여 사용하는 것이 좋습니다. 양식은 인스턴스 유형과 AWS 리전 을 비롯한 새로운 정보가 제공되는 대로 정기적으로 업데이트됩니다. 서버 제품 페이지에서 제품을 선택한 후 제품 로드 양식 다운로드를 선택하여 제품의 최신 PLF를 찾을 수 있습니다.

변경 요청의 상태에 대한 자세한 내용은 the section called "변경 요청의 상태 가져오기" 섹션을 참조하세요. 변경 요청과 관련된 잠재적 문제에 대한 자세한 내용은 AWS Marketplace에서의 변경 요청과 관련하여 흔히 발생하는 오류 해결 섹션을 참조하세요.

변경 요청의 상태 가져오기

> **Important**
>
> 2023년 6월 15일에 AWS Marketplace 는 다음 절차를 중단합니다. 이 절차는 셀프 서비스 경험에 더 이상 필요하지 않습니다.

변경 요청을 제출한 후에는 AWS Marketplace Management Portal의 서버 제품 페이지에 있는 요청 탭에서 요청 상태를 볼 수 있습니다. 요청의 상태는 다음 중 하나입니다.

- 검토 중은 요청을 검토 중이라는 뜻입니다. 일부 요청은 AWS Marketplace 팀의 수동 검토가 필요하지만 대부분은 시스템에서 자동으로 검토됩니다.
- 성공은 요청이 완료되었다는 뜻입니다. 요청하신 대로 제품 또는 버전을 업데이트했습니다.
- 조치 필요는 요청을 업데이트하여 문제를 해결하거나 요청에 대한 질문에 대답해야 한다는 뜻입니다. 요청을 선택하면 문제를 포함한 세부 정보를 볼 수 있습니다.
- 실패는 요청에 문제가 있다는 뜻이므로 동일한 데이터로 변경 요청을 새로 생성해야 합니다.

추가 리소스

특정 업데이트 유형의 변경 요청에 대한 자세한 내용은 다음 리소스를 참조하세요.

- 에서 AMI 기반 제품 정보 업데이트 AWS Marketplace
- 버전 정보 업데이트
- 새 버전 추가
- 버전 제한

AMI 기반 제품 가시성 업데이트

AWS Marketplace 판매자는 제품 가시성을 업데이트하여 단일 AMI 제품을 볼 수 있는 구매자를 변경할 수 있습니다 AWS Marketplace. 가시성 상태가 퍼블릭으로 설정된 경우 모든 AWS Marketplace 구매자에게 제품이 표시됩니다. 제품 가시성이 제한으로 설정된 경우 허용 목록에 있는 AWS 계정 IDs에만 제품이 표시됩니다. 또한 제품을 볼 수 AWS 계정 IDs 있는이 ID 허용 목록을 관리하고 업데이트할 수 있습니다. 다음 섹션에서는 제품 가시성 및 제한됨 가시성 허용 목록을 업데이트하는 방법을 보여줍니다.

주제

- 제품 표시 여부 업데이트
- 허용 목록 업데이트(계정 미리 보기)

제품 표시 여부 업데이트

제품을 볼 수 있는 구매자를 변경하려면 업데이트 가시성을 사용할 AWS Marketplace수 있습니다.

표시 여부를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 표시 여부 업데이트를 선택합니다.

> **Note**
>
> 이 변경 요청을 사용하여 제품을 제한 상태에서 퍼블릭 상태로 전환하라고 요청할 수 있습니다. 그러나 변경 요청은 AWS Marketplace 판매자 운영 팀 승인 프로세스를 거쳐 퍼블릭으로 이동해야 합니다.

4. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
5. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

허용 목록 업데이트(계정 미리 보기)

제한된 상태에서 제품을 볼 수 있는 AWS 계정 IDs 목록을 변경하려면 허용 목록 업데이트를 사용합니다.

허용 목록을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 허용 목록 업데이트를 선택합니다. 현재 목록에는 현재 허용 목록에 있는 계정 목록이 제공됩니다.

4. 표시에 선호되는 AWS 계정 IDs를 추가하고 IDs를 쉼표로 구분합니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

에 대한 AMI 인스턴스 추가 및 제한 AWS Marketplace

AWS Marketplace 판매자는 구매자가 단일 Amazon Machine Imagine(AMI) 제품에 사용할 수 있는 인스턴스를 관리할 수 있습니다. 구매자가 사용할 수 있는 단일 AMI 제품에 대한 새로운 인스턴스를 추가할 수 있습니다. 이와 비슷하게, 신규 구매자가 특정 인스턴스의 단일 AMI 제품을 사용하지 못하도록 하려면 인스턴스를 제한하면 됩니다.

Amazon EC2 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 사용 설명서의 사용 가능한 인스턴스 유형을 참조하세요.

다음 섹션에서는 인스턴스를 추가하고 제한하는 방법을 설명합니다.

주제

- 인스턴스 추가
- 인스턴스 제한

인스턴스 추가

구매자가 단일 AMI로 사용할 수 있는 새 인스턴스를 추가할 수 있습니다.

인스턴스를 추가하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 인스턴스 추가를 선택합니다.
4. 인스턴스 아키텍처를 선택합니다.
5. 사용 가능한 인스턴스 목록에서 추가할 인스턴스 유형을 선택합니다.
6. 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
7. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

> **Note**
>
> - 현재 요금 모델이 무료가 아니거나 기존 보유 라이선스 사용(BYOL) 모델을 사용하는 경우 가격도 추가해야 합니다.
> - 새 인스턴스의 가격이 포함된 인스턴스 추가 또는 요금을 인상하기 위한 요금 업데이트를 생성한 경우 변경한 날로부터 90일 이내에 셀프 서비스를 사용하여 인스턴스를 추가할 수 없습니다. 변경하려면 AWS Marketplace 판매자 운영 팀에 문의하세요.
> - 새 인스턴스 유형에 대한 지원을 추가하면 제품에 대한 비공개 제안을 이미 구독한 고객은 새로 추가된 인스턴스를 자동으로 시작할 수 없습니다. 고객이 액세스하려는 인스턴스로 다른 비공개 제안을 생성해야 합니다. 새 제안을 수락한 후 고객은 새로 추가된 인스턴스를 시작할 수 있습니다. 향후 제품을 구독하는 고객은 인스턴스가 비공개 제안에 포함되어 있는 한 인스턴스를 시작할 수도 있습니다. 새 비공개 제안을 생성하는 방법에 대한 자세한 내용은이 가이드의 에서 계약 수정 AWS Marketplace뒷부분을 참조하세요.

인스턴스 제한

신규 구매자가 AMI 제품의 인스턴스를 사용하지 못하도록 인스턴스를 제한할 수 있습니다. 나중에 필요할 때 인스턴스를 다시 추가할 수 있습니다. 제한된 인스턴스에 있는 단일 AMI의 기존 사용자는 구독 기간 동안 리전의 제품을 계속 사용할 수 있습니다.

인스턴스를 제한하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 인스턴스 제한을 선택합니다.
4. 제한하려는 인스턴스를 선택하고 제한을 선택합니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

> **Note**
>
> 확인란이 흐리게 표시되면 인스턴스가 권장 인스턴스 유형으로써 하나 또는 여러 버전과 연결되어 있다는 뜻입니다. 이러한 인스턴스를 제한하려면 버전 업데이트를 사용하여 다른 권장 인스턴스 유형을 선택합니다. 변경 요청이 완료되고 제한하려는 인스턴스가 더 이상 권장 인스턴스 유형이 아니면 인스턴스 제한으로 돌아가서 선택한 인스턴스를 제한할 수 있습니다.

에서 AMI 기반 제품의 버전 관리 AWS Marketplace

Amazon Machine Image(AMI) 기반 제품을 생성할 때 특정 버전의 소프트웨어를 AWS Marketplace포함합니다. 용 AMI 기반 제품의 수명 주기는 첫 번째 버전을 게시한 후에도 종료 AWS Marketplace 되지 않습니다. 새 버전의 소프트웨어로 제품을 최신 상태로 유지해야 합니다. 다음 섹션에서는 버전 정보 업데이트(예: 설명 및 날짜), 새 버전 추가, 이전 버전에 대한 액세스 제한 등 버전을 관리하는 방법을 보여 줍니다.

주제

- 버전 정보 업데이트
- 새 버전 추가
- 버전 제한

버전 정보 업데이트

버전을 생성한 후에는 버전과 관련된 정보를 수정하여 구매자에게 업데이트된 정보를 제공하는 것이 좋습니다. 예를 들어 1.1 버전이 출시된 후 1.0 버전을 제한하려면 1.0 버전의 설명을 업데이트하여 구매자를 1.1 버전으로 안내하고 버전이 제한되는 날짜를 알려주면 됩니다. AWS Marketplace Management Portal에서 버전 정보를 업데이트합니다.

버전 정보를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 현재 서버 제품 페이지로 이동한 후 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 메뉴에서 버전 정보 업데이트를 선택합니다.
4. 버전 업데이트 페이지에서 업데이트할 버전을 선택합니다.

5. 다음 중 수정해야 하는 정보를 업데이트합니다.
   - 릴리스 정보
   - 사용 지침
   - 64비트(x86) Amazon Machine Image(AMI) - 사용 및 보안 그룹에 대한 세부 정보
6. 제출을 선택합니다.
7. 요청이 요청 탭에 표시되고 상태가 검토 중인지 확인합니다.

> **Note**
> 이 절차를 사용하여 버전 제목 또는 버전과 연결된 AMI를 업데이트할 수 없습니다. 그 대신 새 버전을 생성하고 이전 버전을 제한하세요.

언제든지 서버 제품 페이지의 요청 탭에서 요청 상태를 확인할 수 있습니다. 자세한 내용은 변경 요청의 상태 가져오기 단원을 참조하세요.

새 버전 추가

제품 또는 기본 이미지를 변경하거나 제품의 AMI를 수정해야 할 때 새 버전의 제품을 추가할 수 있습니다. 에서 제품의 새 버전을 추가합니다 AWS Marketplace Management Portal.

> **Note**
> 에 대한 AMI 생성에 대한 자세한 내용은 섹션을 AWS Marketplace참조하세요와 함께 사용할 AMIs 빌드 모범 사례 AWS Marketplace.

새 버전을 추가하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 새 버전 추가를 선택합니다. 최신 버전의 정보로 채워진 새 버전 추가 양식이 나타납니다.

> **Note**
>
> 테스트 '버전 추가'를 선택하여 새 버전을 추가하지 않고 AMI 및 AWS CloudFormation 템플릿(들)을 스캔할 수도 있습니다. 자세한 내용은 게시 요구 사항에 대한 AMI 스캔을 참조하세요.

4. 버전 정보 섹션에서 다음 정보를 입력합니다.
   - 버전 제목 - 유효한 문자열(예: `1.1` 또는 `## 2.0`)을 입력합니다. 제품 전체에서 고유해야 합니다.
   - 릴리스 정보 - 이 버전에 대한 세부 정보를 제공하는 텍스트를 입력합니다.
5. 전송 옵션 섹션에서 구매자가 솔루션을 배포하는 방법, 즉 AMI(독립 실행형), CloudFormation을 사용하는 AMI 또는 둘 다를 선택합니다. CloudFormation 전송 옵션을 사용하여 최대 3개의 AMI를 선택할 수 있습니다.

> **Note**
>
> 새 전송 옵션은 기존 버전에 추가할 수 없습니다. 단일 버전에 대한 모든 전송 옵션은 동일한 요청으로 제출해야 합니다.

6. Amazon Machine Image(AMI) 섹션에서 다음 정보를 제공합니다.
   - Amazon Machine Image ID - 이 버전에 사용할 AMI의 AMI ID를 입력합니다. AMI ID는 콘솔의 AMI 목록에서 찾을 수 있습니다. AMI는 미국 동부(버지니아 북부) 리전과 AWS Marketplace 판매자 계정에 있어야 합니다. 이 AMI와 연결된 스냅샷은 암호화할 수 없습니다.
   - IAM 액세스 역할 ARN -가 AMI에 액세스할 수 있도록 허용하는 (IAM) 역할 AWS Marketplace의 Amazon 리소스 이름 AWS Identity and Access Management (ARN)을 입력합니다. IAM 역할을 생성하는 방법에 대한 지침은 AMI에 대한 AWS Marketplace 액세스 권한 부여 섹션을 참조하세요. IAM ARN의 표준 형식(예: `arn:aws:iam::123456789012:role/RoleName`)을 사용합니다. ARN은 AWS Marketplace 판매자 계정에 있어야 합니다.
   - 운영 체제(OS) - AMI의 기본 운영 체제를 입력합니다.
   - OS 버전 - AMI에서 사용 중인 운영 체제 버전을 입력합니다.
   - OS 사용자 이름 - Linux 기반 AMI의 경우 인스턴스에 로그인하는 데 사용할 수 있는 사용자 이름을 입력합니다. ec2-user를 사용하는 것이 좋습니다.
   - 스캔 포트 - 운영 체제에 로그인하는 데 사용할 수 있는 포트 번호(Linux AMI는 SSH 포트, Windows AMI는 RDP 포트)를 입력합니다.

7. 해당하는 경우 AMI 전송 옵션 섹션에 대해 다음 구성을 제공합니다.
   - 권장 인스턴스 유형 - 구매자에게 기본적으로 제공되는 인스턴스 유형을 선택합니다.
   - 사용 지침 - AMI 사용 지침 또는 AMI 사용에 대한 자세한 정보 링크를 입력합니다. 예: *### ### ## https://example.com/usage.htm## #####.*
   - 엔드포인트 URL - 구매자가 인스턴스를 생성한 후 소프트웨어에 액세스할 수 있는 방법에 대한 정보를 제공합니다. 구매자가 제품에 액세스하는 데 사용할 수 있는 프로토콜(https 또는 http), 상대 URL(예: */index.html*) 및 포트(예: *443*)를 입력합니다. (호스트 이름은 EC2 인스턴스에 따라 다르므로 상대 경로만 제공하면 됩니다.)
   - 보안 그룹 권장 사항 - 프로토콜(TCP 또는 UDP), 허용할 포트 범위, IPv4 CIDR IP 목록(xxx.xxx.xxx.xxx/nn 형식, 예: *192.0.2.0/24*)을 포함하여 하나 이상의 권장 사항에 대한 정보를 입력합니다.
8. 해당하는 경우 CloudFormation 제공 옵션을 사용하여 AMI에 대해 다음 구성 설정을 제공합니다.
   - 전송 옵션 제목 -이 이름은이 단일 AMI CloudFormation 템플릿 전송 옵션을 식별하기 위한 고유한 이름입니다.
   - 간략한 설명 - CloudFormation 템플릿에 대한 간략한 설명입니다.
   - 자세한 설명 - CloudFormation 템플릿에 대한 자세한 설명입니다.
   - 권장 인스턴스 유형 - 기본적으로 구매자에게 제공되는 인스턴스 유형입니다.
   - 사용 지침 - 솔루션 사용 지침 또는 솔루션 사용에 대한 자세한 정보 링크입니다.
   - CloudFormation 템플릿 링크 - Amazon S3에서 CloudFormation 템플릿의 위치에 대한 URL입니다. 중첩 스택 또는 템플릿의 경우 진입점 또는 상위 템플릿을 제공합니다. 파일이 상주하는 Amazon S3 버킷은 공개적으로 액세스할 수 있어야 합니다. 자세한 내용은 제품에 CloudFormation 템플릿 추가를 참조하세요.
   - AMI 파라미터 이름 - 템플릿의 EC2 인스턴스 리소스가 `ImageId` 속성 값을 참조하는 CloudFormation 템플릿에 파라미터 이름을 추가합니다. 지정된 파라미터를 AWS Systems Manager 파라미터 스토어 파라미터로 바 AWS Marketplace 꿉니다. 구매자가 템플릿을 배포하면 해당 파라미터는 게시된 제품의 AWS 리전별 AMI ID로 확인됩니다. 자세한 내용은 AMI 세부 정보 요구 사항을 참조하세요.
   - 아키텍처 다이어그램 링크 - Amazon S3의 아키텍처 다이어그램 위치에 대한 URL입니다. 최대 이미지 파일은 크기가 1560x878 또는 1560x3120픽셀이고 비율은 16:9 또는 1:2여야 합니다. 이미지 파일이 있는 Amazon S3 버킷은 공개적으로 액세스할 수 있어야 합니다. 다이어그램 요구 사항은 아키텍처 다이어그램을 참조하세요.
9. 제출을 선택하여 새 버전 추가 요청을 제출합니다.

> **Note**
> 새 버전을 추가하면 AMI가 스캔됩니다. 자세한 내용은 게시 요구 사항에 대한 AMI 스캔을 참조하세요.

10. 요청이 요청 탭에 표시되고 상태가 검토 중인지 확인합니다. 수정해야 하는 오류가 있으면 페이지 상단의 표에 오류가 표시되고, 업데이트해야 하는 특정 필드가 빨간색으로 표시됩니다.

언제든지 서버 제품 페이지의 요청 탭에서 요청 상태를 확인할 수 있습니다. 새 버전이 검토되고, 성공하면 제품의 새 공개 버전으로 게시됩니다. 문제가 있으면 상태가 조치 필요일 수 있습니다. 요청을 선택하면 문제를 포함한 세부 정보를 볼 수 있습니다.

요청이 성공하면 기존 사용자에게 다음과 같은 이메일 메시지가 전송됩니다. 이 메시지는 고객에게 새 버전을 사용할 수 있다고 알리고, 버전의 릴리스 정보로 연결되는 링크를 제공하며, 최신 버전으로 업그레이드할 것을 제안합니다. AWS 계정 루트 사용자는와 연결된 이메일 계정의 이메일 메시지 사본도 받게 됩니다 AWS 계정.

```
Greetings from AWS Marketplace,

Thank you for subscribing to <product-title>

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

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

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

<release-notes>

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

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

버전 제한

구매자가 특정 버전의 공개 제품에 액세스하지 못하게 하려면 해당 버전을 제한하면 됩니다.

> **Note**
>
> 모든 구독자는 제한 상태에 관계없이 최신 버전을 사용할 수 있습니다. AWS Marketplace 가이드라인에 따라 버전을 제한한 후 90일 동안 기존 구매자에게 지원을 계속 제공해야 합니다. 버전이 제한되면 AMI는 더 이상 사용되지 않는 것으로 표시됩니다. 자세한 내용은 Windows 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 AMI 사용 중지를 참조하세요.

버전을 제한하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 버전 제한을 선택합니다.
4. 버전 제한 페이지에서 제한하려는 버전(또는 여러 버전)을 선택합니다.
5. 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공입니다.

> **Note**
>
> 제품의 모든 버전을 제한할 수는 없습니다. 제품의 마지막 남은 공개 버전을 제한하려고 하면 오류가 발생합니다. 제품을 완전히 제거하려면 the section called “제품 제거” 섹션을 참조하세요.

언제든지 서버 제품 페이지의 요청 탭에서 요청 상태를 확인할 수 있습니다. 자세한 내용은 변경 요청의 상태 가져오기 단원을 참조하세요.

> **Note**
>
> 버전 제한이 완료될 때까지 최대 3일이 소요될 수 있습니다.

요청이 성공하면 기존 사용자에게 버전 제한 소식을 알리고 사용 가능한 최신 버전을 사용할 것을 권장하는 다음과 같은 이메일 메시지가 전송됩니다. AWS 계정 루트 사용자는와 연결된 이메일 계정의 이메일 메시지 사본도 받게 됩니다 AWS 계정.

```
Greetings from AWS Marketplace,

Thank you for subscribing to <product-title>.

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

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

Thank you,
The AWS Marketplace Team
https://aws.amazon.com/marketplace
AWS, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of
 Amazon.com, Inc. This message was produced and distributed by AWS Inc., 410 Terry Ave.
 North, Seattle, WA 98109-5210
```

에서 AMI 기반 제품 정보 업데이트 AWS Marketplace

단일 Amazon Machine Image(AMI) 제품을 만든 후 AWS Marketplace에서 연결된 일부 정보를 변경할 수 있습니다. 예를 들어 새 버전이 제품의 설명이나 주요 내용을 수정하는 경우 제품 정보를 새 데이터로 편집할 수 있습니다. 제품 제목, SKU 설명, 범주, 키워드 등 다른 제품 정보를 업데이트할 수도 있습니다. 자세한 내용은 다음 절차를 참조하세요.

제품 정보를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.

2. 서버 제품 페이지로 이동한 후 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 메뉴에서 제품 정보 업데이트를 선택합니다.
4. 변경해야 하는 다음 필드를 업데이트합니다.
   - 제품 제목
   - SKU
   - 간단한 설명
   - 자세한 설명
   - 제품 로고 이미지 URL
   - 하이라이트
   - 제품 범주
   - 키워드
   - 제품 비디오 URL
   - 리소스
   - 지원 정보

   > **Note**
   > 로고 형식에 대한 자세한 내용은 회사 및 제품 로고 요구 사항 섹션을 참조하세요.

5. 제출을 선택합니다.
6. 요청이 요청 탭에 표시되고 상태가 검토 중인지 확인합니다. 목록에 요청이 보이지 않으면 페이지를 새로 고쳐야 할 수도 있습니다.

언제든지 서버 제품 페이지의 요청 탭에서 요청 상태를 확인할 수 있습니다. 자세한 내용은 변경 요청의 상태 가져오기 단원을 참조하세요.

및 국가별 AMI 기반 제품 가용성 AWS 리전 관리

에서 제품을 생성할 때 제품을 사용할 수 있는 AWS 리전을 AWS Marketplace선택합니다. 구매자가 제품을 구매할 수 있는 국가도 선택할 수 있습니다. 이 두 속성은 비슷하지만 동일하지는 않습니다. 예를 들어 구매자가 미국에 살면서 미국에서 제품을 구매하지만 유럽(프랑크푸르트) 리전에 제품을 설치할 수 있습니다. 이 구매자가 제품을 구매할 수 있으려면 국가 목록에 미국을 포함시키고 리전 목록에 유

럽(프랑크푸르트) 리전을 포함시켜야 합니다. 다음 섹션을 사용하여 리전 및 국가별로 제품 가용성을 업데이트할 수 있습니다.

주제

- 추가 AWS 리전
- 제한 AWS 리전
- 향후 AWS 리전지원 업데이트
- 국가별 가용성 업데이트

추가 AWS 리전

구매자가 제품을 사용할 수 있는 리전을 추가할 수 있습니다.

리전을 추가하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 리전 추가를 선택합니다.
4. 사용 가능한 리전 목록에서 추가할 리전을 선택합니다.
5. 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

> **Note**
>
> 새에 대한 지원을 추가하면 제품에 대한 비공개 제안을 이미 구독한 AWS 리전고객은 새로 추가된 리전에 자동으로 액세스할 수 없습니다. 고객이 액세스하려는 리전을 사용하여 다른 비공개 제안을 만들어야 합니다. 고객은 새 제안을 수락한 후 새로 추가된 리전에 액세스할 수 있습니다. 향후 제품을 구독하는 고객은 해당 리전이 비공개 제안에 포함되는 한 해당 리전에 액세스할 수도 있습니다. 새 비공개 제안을 만드는 방법에 대한 자세한 내용은 Private offer upgrades, renewals, and amendments를 참조하세요.

제한 AWS 리전

신규 구매자가 특정에서 제품을 사용하지 못하도록 리전을 제한 AWS 리전할 수 있습니다. 나중에 리전을 다시 추가할 수 있습니다. 해당 리전의 기존 제품 구독자는 구독을 유지하는 한 해당 리전의 제품을 계속 사용할 수 있습니다.

리전을 제한하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 리전 제한을 선택합니다.
4. 드롭다운 메뉴를 선택하여 현재 제품을 사용할 수 있는 리전 목록을 확인합니다.
5. 제한하려는 리전을 선택합니다.
6. 선택한 리전이 토큰으로 표시됩니다. 제한 중인 리전 목록을 검토하고, 제한하지 않을 지역에 X를 입력합니다.
7. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
8. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

요청이 성공하면 기존 사용자에게 제한될 리전을 알리는 다음과 같은 이메일 메시지가 전송됩니다. 기존 사용자는 구독을 유지하는 한 제품을 계속 사용할 수 있지만, 구독을 취소하면 다시 구독할 수 없습니다.

```
Greetings from AWS Marketplace,

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

This impacts you in the following ways:

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

Regards,
```

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

향후 AWS 리전지원 업데이트

제품을 새로 출시된에 온보딩하려면 향후 리전 지원 업데이트를 사용할 AWS 리전수 있습니다.

향후 리전 지원을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 향후 리전 지원 업데이트를 선택합니다.
4. 향후 리전 지원을 활성화하여가 사용자를 대신하여 새로 출시된 AWS Marketplace 에 제품을 온보딩할 수 있도록 선택할 수 AWS 리전 있습니다.
5. 이 기능을 활성화한 후에는 모든 향후 리전을 선택할 수도 있고 미국 리전으로만 제한할 수도 있습니다.
6. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
7. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

국가별 가용성 업데이트

제품을 구독하고 제공할 수 있는 국가를 변경하려면 가용성 업데이트를 사용하면 됩니다.

국가별 가용성을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 메뉴에서 가용성 업데이트를 선택합니다.
4. 다음 옵션 중 하나를 선택하세요.

    1. 모든 국가 - 지원되는 모든 국가에서 사용할 수 있습니다.

   2. 일부 국가를 제외한 모든 국가 - 일부 국가를 제외하고 지원되는 모든 국가에서 사용할 수 있습니다.
   3. 사용자 지정 목록 - 특정 목록의 국가에서만 제품을 사용할 수 있습니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

AMI 기반 AWS Marketplace 제품에 대한 최종 사용자 라이선스 계약(EULA) 업데이트

AWS Marketplace 판매자는 단일 Amazon Machine Image(AMI) 제품의 사용에 적용되는 최종 사용자 라이선스 계약(EULA)을 업데이트할 수 있습니다. EULA는 AWS Marketplace의 공개 소프트웨어 리스팅에 대한 제품 리스팅 페이지에 있습니다. 판매자 고유의 EULA를 적용할 수도 있고 AWS Marketplace 표준 계약(SCMP)을 사용할 수도 있습니다. 다음 절차에서는 단일 AMI 제품의 EULA를 업데이트하는 방법을 보여줍니다.

EULA에 대한 자세한 내용은 에서 표준화된 계약 사용 AWS Marketplace 섹션을 참조하세요.

EULA를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 탭을 선택한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 최종 사용자 라이선스 계약 업데이트를 선택합니다.
4. 표준 계약 AWS Marketplace (SCMP)을 선택하거나 사용자 지정 EULA를 제출할 수 있습니다. 사용자 지정 EULA의 경우 Amazon S3 버킷의 사용자 지정 계약에 대한 URL을 제공해야 합니다.

   > **Note**
   >
   > Amazon S3 버킷에 대한 공개 액세스가 활성화되어 있어야 합니다.

5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

에서 AMI 기반 제품에 대한 환불 정책 업데이트 AWS Marketplace

AWS Marketplace 판매자는 단일 Amazon Machine Image(AMI) 제품에 대한 환불 정책을 설정할 수 있습니다. 제품의 환불 정책을 변경하려면 환불 정책 업데이트를 사용하면 됩니다. 다음 절차에서는 환불 정책을 업데이트하는 방법을 보여줍니다.

환불 정책을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 환불 정책 업데이트를 선택합니다.
4. 현재 환불 정책 세부 정보가 텍스트 상자에 제공되며 이 정보를 편집할 수 있습니다. 요청을 제출하면 현재 환불 정책을 덮어씁니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

AMI에 대한 AWS Marketplace 액세스 권한 부여

에 새 Amazon Machine Image(AMI)를 추가하는 요청이 포함된 요청을 생성할 때 AMI AWS Marketplace를 AWS Marketplace 시스템에 복사한 다음 보안 문제를 스캔해야 합니다. AMI에 대한 작업을 수행할 수 있는 권한이 있는 AWS Identity and Access Management (IAM) 역할과가 역할을 AWS Marketplace 수임하도록 허용하는 신뢰 정책을 생성하여 AMI에 대한 AWS Marketplace 액세스 권한을 부여해야 합니다. IAM 역할은 한 번만 만들면 됩니다. 다음 절차에서는 AMI에 대한 AWS Marketplace 액세스 권한을 부여하는 AWS Marketplace 자산 수집에 대한 역할을 생성하는 방법을 보여줍니다.

AWS Marketplace AMI 자산 수집을 위한 역할을 생성하려면

1. 에 로그인하고 IAM 콘솔을 AWS Management Console열고 역할 페이지로 이동합니다.
2. 규칙 생성을 선택합니다.
3. 역할 생성 페이지에서 다음을 선택합니다.
   - 신뢰할 수 있는 엔터티 유형 선택 - AWS 서비스를 선택합니다.
   - 사용 사례 선택 -를 선택합니다AWS Marketplace.
   - 사용 사례 선택 - Marketplace - AMI 자산 수집을 선택합니다.

- 다음: 권한을 선택하여 다음 페이지로 이동합니다.

4. AWSMarketplaceAmiIngestion 정책을 선택합니다. 필요한 경우 권한 경계를 추가하고 다음: 태그를 선택하여 계속합니다.

> **Note**
>
> 권한 경계를 사용하여 이 역할로 AWS Marketplace 에 부여하는 액세스 권한을 제한할 수 있습니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 엔터티의 권한 경계를 참조하세요.

5. 다음: 검토를 선택하여 계속합니다.
6. 역할의 이름을 지정하고 역할 생성을 선택합니다.
7. 페이지 상단에 *rolename* 역할이 생성되었다는 메시지가 표시되고, 역할 목록에 역할이 표시됩니다.

이 페이지에서 방금 생성한 역할을 선택하면 arn:aws:iam::123456789012:role/exampleRole 형식의 ARN을 확인할 수 있습니다. 변경 요청을 생성할 때(예: 제품에 새 버전 추가) IAM 액세스 역할 ARN에 ARN을 사용합니다.

AWS Marketplace에서 제품 제거

제품을 게시한 후에서 제거할 수 있습니다 AWS Marketplace. 이를 일몰이라고도 합니다. 제품을 제거하려면 제품을 식별하고 제거 이유 및 자신의 연락 이메일 주소와 함께 제거 요청을 제출합니다. 현재 제품을 새 제품으로 교체하려는 경우 교체 제품 ID를 제공할 수도 있습니다.

제품 제거를 요청하면 신규 고객은 제품을 구독할 수 없습니다. 최소 90일 동안 기존 고객을 지원해야 합니다.

> **Note**
>
> AWS Marketplace 관리 포털에서는 제한된 제품을 삭제할 수 없습니다. 포털은 계정의 게시 기록의 일부로 이를 보관합니다.

에서 제품을 제거하는 요청은 다음 조건으로 처리됩니다 AWS Marketplace .

- 검색 AWS Marketplace , 찾아보기 및 기타 검색 도구에서 제품이 제거됩니다. 구독 버튼이나 기능이 비활성화되고, 제품을 더 이상 사용할 수 없다고 분명하게 밝히는 메시지가 페이지에 표시됩니다. 계

속 URL을 사용하여 제품 세부 정보 페이지에 액세스할 수 있으며 공개 검색 엔진에 이 페이지가 인덱싱됩니다.

- 제거 이유(예: 지원 종료, 제품 업데이트 종료 또는 교체 제품)를 명시해야 합니다. 제거된 제품에 대한 지속적인 지원 요구 사항은 AWS Marketplace 판매자 이용 약관을 참조하세요.
- AWS Marketplace 는 제품 제거, 제거 이유 및 판매자 연락처 정보를 알리는 이메일 메시지를 통해 현재 구매자에게 연락합니다.
- 현재 구매자는 구독을 취소할 때까지 소프트웨어에 계속 액세스할 수 있습니다. 구독자는 어떤 방식으로도 제품 제거의 영향을 받지 않습니다.

를 사용하여 생성된 제품을 제거하려면 AWS Marketplace Management Portal

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 제품 탭을 선택하고 서버를 선택합니다.
3. 서버 제품의 제품 페이지에서 제거할 제품을 찾습니다. 변경 요청 드롭다운 목록에서 제품 가시성 업데이트를 선택합니다.
4. 제품 가시성 업데이트 페이지에서 제한됨을 선택합니다.
5. (선택 사항) 제거하려는 제품을 대신할 다른 제품이 있는 경우 교체 제품 ID를 입력합니다.
6. 정보가 정확한지 검토한 후 제출을 선택합니다.

제품 제거 요청을 제출하면 What’s next(다음 단계) 정보 페이지가 표시됩니다. AWS Marketplace 판매자 작업은 요청을 검토하고 처리합니다. 요청을 보고 제출의 상태를 확인합니다.

제품이 제거되고 나면 AWS Marketplace Management Portal의 현재 제품 목록에 제품이 표시됩니다. 현재 제품에서는 제품에 대한 스프레드시트 다운로드를 하는 작업만 수행할 수 있습니다. 다른 만료 요청을 편집하거나 제출할 수는 없습니다.

제품 제거에 대해 궁금한 점이 있는 경우 AWS Marketplace 판매자 작업 팀에 문의하세요.

AWS Marketplace에서의 변경 요청과 관련하여 흔히 발생하는 오류 해결

AWS Marketplace에서 제품 정보를 변경할 때 오류가 발생할 수 있습니다. 이 주제에서는 몇 가지 일반적인 오류를 설명하고 수정 방법을 제안합니다.

- AMI 스캔 - AMI를 스캔할 때 여러 가지 문제가 발생할 수 있습니다.

- AMI를 스캔할 AWS Marketplace 권한을 부여하지 않았습니다. AMI에 액세스할 수 있도록 AWS Marketplace 권한을 부여하세요. 또는 권한을 부여했지만 권한 경계가 너무 제한적입니다. 자세한 내용은 AMI에 대한 AWS Marketplace 액세스 권한 부여 단원을 참조하십시오.
- 스캔을 통해 AMI에서 보안 문제 또는 일반적인 취약점 및 노출(CVE)이 발견되면 이미지의 운영 체제에 최신 패치를 사용합니다. 자세한 내용은 에 대한 AMI 기반 제품 요구 사항 AWS Marketplace 단원을 참조하십시오.

AMI 빌드에 대한 일반 지침은 와 함께 사용할 AMIs 빌드 모범 사례 AWS Marketplace 섹션을 참조하세요.

- AWS Marketplace Management Portal 필드 - AWS Marketplace Management Portal의 일부 필드는 매우 구체적인 정보가 필요합니다.
  - 필드에서 무엇을 요청하는지 잘 모르겠으면 콘솔에서 세부 정보를 확인하세요. 대부분의 필드 위에는 텍스트 설명이 있고, 필드 아래에는 서식 요구 사항이 있습니다.
  - 유효하지 않은 필드가 하나 이상 있는 양식을 제출하려고 하면 문제 목록이 표시됩니다. 문제 해결에 도움이 되는 권장 조치가 제공됩니다.
  - ARN을 입력하라는 요청을 받으면 일반적으로 콘솔의 다른 곳에서 ARN을 찾을 수 있습니다. 예를 들어 AWS Marketplace에 AMI 액세스 권한을 부여하기 위해 생성한 IAM 역할의 ARN은 IAM 콘솔의 역할 페이지에서 찾을 수 있습니다. 모든 ARN은 형식이 비슷합니다. 예를 들어 IAM 역할 ARN은 arn:aws:iam::123456789012:role/exampleRole 형식입니다.
  - 로고와 비디오는 콘텐츠로 직접 연결되는 URL로 제공해야 합니다. 로고 형식에 대한 자세한 내용은 회사 및 제품 로고 요구 사항 섹션을 참조하세요.

  제품 및 버전 변경 요청 제출에 대한 자세한 내용은 에 게시할 제품 제출 AWS Marketplace 섹션을 참조하세요.

- 제품 로드 양식(PLF) 문제 - PLF에는 스프레드시트에 포함된 지침이 들어 있습니다. 전체 지침은 지침 표에 나와 있습니다. 각 필드에는 작성 방법에 대한 지침이 있습니다. 필드를 선택하면 지침이 표시됩니다.
- 요청 진행 중 - 일부 요청은 병렬로 발생할 수 없습니다. 제품의 특정 정보 업데이트 요청은 한 번에 하나씩만 진행할 수 있습니다. AWS Marketplace Management Portal의 서버 제품 페이지에 있는 요청 탭에서 아직 검토 중인 모든 요청을 볼 수 있습니다. 의도하지 않은 보류 중인 요청이 있는 경우 요청을 취소하고 원하는 변경 내용이 포함된 새 요청을 제출할 수 있습니다.
  - (추가 또는 제한하는) 버전 업데이트가 진행 중이면 버전 정보를 업데이트할 수 없습니다.
  - AWS Marketplace 판매자 운영 팀에서 보류 중인 요청이 있는 경우 새로운 변경 사항을 제출할 수 없습니다.

- 설명 없는 오류 - 제출이 설명 없이 실패하면 다시 시도하세요. 서버 부하 때문에 제출이 실패하는 경우가 가끔 있습니다.

여전히 변경 요청에 문제가 있으면 AWS Marketplace 판매자 운영 팀에 문의하세요.

와 함께 사용할 AMIs 빌드 모범 사례 AWS Marketplace

이 주제에서는와 함께 사용할 Amazon Machine Image(AMIs)를 빌드하는 데 도움이 되는 모범 사례와 참조를 제공합니다 AWS Marketplace. 에 빌드 및 제출된 AMIs 모든 AWS Marketplace 제품 정책을 준수해야 AWS Marketplace 합니다. 자세한 내용은 다음 섹션을 참조하세요.

주제

- 재판매 권한 확보
- AMI 작성
- 용 AMI 준비 및 보안 AWS Marketplace
- AMI를 스캔하여 게시 요구 사항 확인
- AWS Marketplace AMI에서 소프트웨어가 실행되고 있는지 확인

재판매 권한 확보

무료가 아닌 Linux 배포판의 경우 AWS사용자는 제공된 Amazon Linux, RHEL 및 SUSE를 제외하고 해당 배포판에 대한 재판매 권한을 보호할 책임이 있습니다. Windows AMIs에 대한 재판매 권한을 보호할 필요가 없습니다.

AMI 작성

AMI 빌드에 대한 다음 지침을 따릅니다.

- AMI가 모든 AWS Marketplace 정책을 충족하는지 확인합니다.
- 미국 동부(버지니아 북부) 리전에서 AMI를 생성합니다.
- AWS Marketplace와 같이 신뢰할 수 있고 평판이 좋은 출처에서 제공하며 수명 주기가 분명히 정의되고 잘 유지되는 기존의 Amazon Elastic Block Store(Amazon EBS) 기반 AMI에서 제품을 생성합니다.
- 최신 운영 체제, 패키지, 소프트웨어를 사용하여 AMI를 개발합니다.

- AMI가 하드웨어 가상 머신(HVM) 가상화 및 64비트 아키텍처를 사용하는 퍼블릭 Amazon EC2 AMI를 기반으로 하는지 확인합니다.
- AMI 개발, 업데이트, 리퍼블리싱을 위한 반복적인 프로세스를 개발합니다.
- 모든 버전과 제품에서 일관성 있는 운영 체제(OS) 사용자 이름을 사용합니다. 권장되는 기본 사용자 이름은 Linux 및 기타 Unix 유사 시스템의 `ec2-user` 경우 , Windows의 `Administrator` 경우 입니다.
- 게시할 최종 AMI를 제출하기 전에 AMI에서 인스턴스를 AWS Marketplace 시작하고 테스트하여 의도한 최종 사용자 경험을 확인합니다. 이 인스턴스에서 모든 설치 방법, 기능 및 성능을 테스트합니다.
- 다음과 같이 포트 설정을 확인합니다.
  - 개방형 방화벽, 역방향 프록시 및 SSRF 취약성에 대한 모범 사례 보안 구성으로 IMDS 지원 옵션을 IMDSv2로만 설정해야 합니다. 최종 빌드 단계에서 새 AMI를 등록할 때 다음 CLI를 사용할 수 있습니다.
    - `aws ec2 register-image --name my-image --root-device-name /dev/xvda --block-device-mappings DeviceName=/dev/xvda,Ebs={SnapshotId=snap-0123456789example} --architecture x86_64 --imds-support v2.0`

AMI 생성에 대한 자세한 내용은 다음 리소스를 참조하세요.

- Amazon EC2 사용 설명서의 Amazon EBS 지원 AMI 생성
- Amazon EC2 사용 설명서의 Windows Sysprep을 사용하여 Amazon EC2 AMI 생성
- EBS 기반 인스턴스에서 Amazon Machine Image(AMI)를 생성하려면 어떻게 해야 합니까?
- Amazon Linux AMI
- Amazon EC2 인스턴스 유형 및 인스턴스 유형
- 기본적으로 IMDS V2용 AMI 사용 구성

용 AMI 준비 및 보안 AWS Marketplace

보안 AMI 생성 시 다음 지침을 따르는 것이 좋습니다.

- Amazon EC2 사용 설명서의 공유 Linux AMI 지침 사용
- 최소 설치로 배포하도록 AMI를 설계하여 공격 대상 영역을 축소합니다. 필요 없는 서비스와 프로그램을 비활성화하거나 제거합니다.

- 가능할 때마다 네트워크 트래픽에 종단 간 암호화를 사용합니다. 예를 들어 Secure Socket Layer(SSL)를 사용하여 귀사와 구매자 간 HTTP 세션을 보호합니다. 서비스에 유효한 최신 인증서만 사용하도록 합니다.
- AMI 제품을 문서화할 때 구매자가 인스턴스에 대한 인바운드 트래픽 액세스를 제어할 수 있도록 보안 그룹 권장 사항을 제공합니다. 권장 사항은 다음을 지정해야 합니다.
  - 서비스가 작동하는 데 필요한 최소 포트 세트입니다.
  - 관리 액세스를 위한 권장 포트 및 소스 IP 주소 범위입니다.

  이러한 보안 그룹 권장 사항은 구매자가 적절한 액세스 제어를 구현하는 데 도움이 됩니다. AMI 제품에 새 버전을 추가하는 방법에 대한 자세한 내용은 새 버전 추가 섹션을 참조하세요.
- AWS 컴퓨팅 환경에 대해 정기적으로 침투 테스트를 수행하거나 사용자를 대신하여 이러한 테스트를 수행하기 위해 타사를 고용하는 것이 좋습니다. 침투 테스트 요청 양식을 포함한 자세한 내용은 AWS 침투 테스트를 참조하세요.
- 웹 애플리케이션의 10대 취약성에 주의하며 애플리케이션을 빌드합니다. 자세한 내용은 Open Web Application Security Project (OWASP) - Top 10 Web Application Security Risks를 참조하세요. 새로운 인터넷 취약성이 발견되면 AMI에 제공되는 모든 웹 애플리케이션을 즉시 업데이트합니다. 이 정보를 포함하는 리소스로는 SecurityFocus 및 NIST National Vulnerability Database가 있습니다.

보안과 관련된 자세한 내용은 다음 리소스를 참조하세요.

- AWS 클라우드 보안
- Center for Internet Security(CIS): 보안 벤치마크
- Open Web Application Security Project(OWASP): Secure Coding Practices - 빠른 참조 가이드
- OWASP Top 10 Web Application Security Risks
- CWE/SANS TOP 25 Most Dangerous Software Errors
- SecurityFocus
- NIST National Vulnerability Database

AMI를 스캔하여 게시 요구 사항 확인

AWS Marketplace 카탈로그에 AMI를 게시하려면 AMI 스캔을 완료해야 합니다. AMI 스캔은 패치되지 않은 일반적인 취약성 및 노출(CVEs)이 있는지 확인하고 AMI가 보안 모범 사례를 따르는지 확인합니다. 자세한 내용은 용 AMI 준비 및 보안 섹션을 참조하세요. AWS Marketplace

AMI 스캔을 수행하려면 다음 옵션 중 하나를 선택합니다.

옵션 1: 자산 메뉴

이 방법을 사용하면 제품 생성 흐름 외부에서 AMIs를 스캔할 수 있습니다. 또한 AMI 제품을 생성하지 않고 자산을 스캔해야 하는 SAAS Quick Launch를 사용하는 SaaS 판매자에게도 유용합니다.

1. 에서 자산 메뉴로 AWS Marketplace Management Portal이동하여 Amazon Machine Image를 선택합니다.
2. 스캔 프로세스를 시작하려면 AMI 추가를 선택합니다.
3. 이 페이지로 돌아가서 AMIs 상태를 볼 수 있습니다.

옵션 2: 변경 요청 메뉴

이 옵션은 이미 AMI 제품을 생성한 판매자가 사용할 수 있습니다. 자세히 알아보려면 AMI 기반 제품 생성 단원을 참조하십시오.

1. 에서 제품 메뉴로 AWS Marketplace Management Portal이동하여 서버를 선택합니다.
2. 서버 제품에서 제품을 선택합니다. AMI 기반 제품이어야 합니다. 제품은 모든 상태일 수 있으며 다음 단계를 위해 공개 게시 상태일 필요는 없습니다.
3. 변경 요청 메뉴로 이동하여 버전 업데이트를 선택합니다.
4. 테스트 '버전 추가'를 선택합니다. 프롬프트에 따라 AMI 세부 정보가 포함된 요청을 제출합니다. 요청이 성공하면 AMI가 스캔을 성공적으로 통과했음을 나타냅니다. 새 버전 추가 옵션과 달리 테스트 '버전 추가'는 스캔이 성공하면 AMI 기반 제품에 새 버전을 추가하지 않습니다.

> **Note**
>
> AMI에 대한 AWS Marketplace 액세스 권한을 부여하는 방법에 대한 자세한 내용은 섹션을 참조하세요AMI에 대한 AWS Marketplace 액세스 권한 부여.

AWS Marketplace AMI에서 소프트웨어가 실행되고 있는지 확인

소프트웨어가 AMI 제품에서 생성된 Amazon EC2 인스턴스에서 실행되고 있는지 런타임에 확인하려는 경우가 있습니다.

Amazon EC2 인스턴스가 AMI 제품에서 생성되었는지 확인하려면 Amazon EC2에 내장된 인스턴스 메타데이터 서비스를 사용합니다. 다음 단계를 따라 검증을 수행할 수 있습니다. 메타데이터 서비스 사

용에 대한 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 메타데이터 및 사용자 데이터를 참조하세요.

1. 인스턴스 자격 증명 문서 획득

   실행 중인 각 인스턴스에는 인스턴스에서 액세스할 수 있는 자격 증명 문서가 있으며, 이 문서는 인스턴스 자체에 대한 데이터를 제공합니다. 다음 예제는 인스턴스에서 curl을 사용하여 인스턴스 자격 증명 문서를 검색하는 방법을 보여줍니다.

   IMDSv2: (권장)

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/dynamic/
instance-identity/document
 {
   "accountId" : "0123456789",
   "architecture" : "x86_64",
   "availabilityZone" : "us-east-1e",
   "billingProducts" : null,
   "devpayProductCodes" : null,
   "marketplaceProductCodes" : [ "0vg0000000000000000000000" ],
   "imageId" : "ami-0123456789abcdef1",
   "instanceId" : "i-0123456789abcdef0",
   "instanceType" : "t2.medium",
   "kernelId" : null,
   "pendingTime" : "2020-02-25T20:23:14Z",
   "privateIp" : "10.0.0.2",
   "ramdiskId" : null,
   "region" : "us-east-1",
   "version" : "2017-09-30"
}
```

   IMDSv1:

```
curl http://169.254.169.254/latest/dynamic/instance-identity/document{
   "accountId" : "0123456789",
   "architecture" : "x86_64",
   "availabilityZone" : "us-east-1e",
   "billingProducts" : null,
   "devpayProductCodes" : null,
   "marketplaceProductCodes" : [ "0vg0000000000000000000000" ],
```

```
    "imageId" : "ami-0123456789abcdef1",
    "instanceId" : "i-0123456789abcdef0",
    "instanceType" : "t2.medium",
    "kernelId" : null,
    "pendingTime" : "2020-02-25T20:23:14Z",
    "privateIp" : "10.0.0.2",
    "ramdiskId" : null,
    "region" : "us-east-1",
    "version" : "2017-09-30"
}
```

2. 인스턴스 자격 증명 문서 확인

   서명을 사용하여 인스턴스 자격 증명이 올바른지 확인할 수 있습니다. 이 프로세스에 대한 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 자격 증명 문서를 참조하세요.

3. 제품 코드 확인

   게시할 AMI 제품을 처음으로 제출하면 AWS Marketplace에서 제품에 제품 코드를 할당합니다. 인스턴스 자격 증명 문서의 `marketplaceProductCodes` 필드를 확인하여 제품 코드를 확인할 수도 있고, 메타데이터 서비스에서 직접 제품 코드를 가져올 수도 있습니다.

   IMDSv2:

   ```
   TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/product-codes
   ```

   제품 코드가 AMI 제품 코드와 일치하면 인스턴스가 제품에서 생성된 것입니다.

인스턴스 자격 증명 문서의 다른 정보(예: `instanceId` 및 인스턴스 `privateIp`)를 확인하려는 경우도 있습니다.

에 대한 AMI 제품 요금 AWS Marketplace

AWS Marketplace 에는 Amazon Machine Image(AMI) 제품에 대한 여러 요금 모델이 있습니다. 판매자 비공개 제안의 경우 다년 및 사용자 지정 기간 계약에 사용할 수 있는 몇 가지 옵션이 있습니다. 다년 및 사용자 지정 기간 계약에 대한 자세한 내용은 AWS Marketplace 제품에 대한 비공개 제안 준비 및 비공개 제안 할부 플랜 단원을 참조하세요. 다음 섹션에서는 AMI 기반 제품의 요금 모델에 대한 정보를 제공합니다.

> **Note**
>
> AWS Marketplace 판매자로 등록에 설명된 대로 W-9 세금 신고서(미국 내 사업체인 경우) 또는 W-8 양식(EU 내 사업체인 경우)을 제공할 수 있어야 합니다.

주제

- AMI 요금 모델
- AWS 요금 및 소프트웨어 요금
- AWS Marketplace Metering Service를 통한 AMI 제품에 대한 사용자 지정 측정 요금
- 의 AMI 제품에 대한 계약 요금 AWS Marketplace
- 를 사용하여 AMI 기반 제품과 라이선스 연결 AWS License Manager

AMI 요금 모델

다음 표에서는 AMI 기반 제품의 요금 모델에 대한 일반 정보를 제공합니다.

요금 모델	설명
무료	고객은 추가 소프트웨어 요금 없이 Amazon Elastic Compute Cloud(Amazon EC2)가 지원하는 수만큼 인스턴스를 실행할 수 있습니다. **Note** 무료 평가판 및 연간 요금은 월별 요금과 결합할 수 없습니다.
기존 보유 라이선스 사용(BYOL)	AWS Marketplace 는 소프트웨어 사용에 대해 고객에게 요금을 부과하지 않지만 고객은 제품을 활성화하려면 라이선스 키를 제공해야 합니다. 이 키는 외부에서 구매됩니다 AWS Marketplace. 권한 부여 및 라이선스 집행과 모든 요금 및 청구 업무는 판매자가 처리합니다.

요금 모델	설명
시간당 또는 시간당-연간 요금	시간당 - 시간을 기준으로 소프트웨어 요금이 부과됩니다. 각 인스턴스 유형에 다른 요금을 책정할 수 있으며(필수 사항은 아님) 사용량은 가장 근접한 전체 시간으로 올림됩니다. 무료 평가판 포함 시간당 - 고객은 요금 부과 없이 정확하게 1개의 소프트웨어 인스턴스를 실행하도록 제한됩니다. 판매자가 5일에서 30일 사이로 기간을 정의합니다. 무료 평가판은 실행 중인 가장 비싼 인스턴스 유형에 적용되며, 인스턴스 1개 이외의 동시 사용은 시간당 요금으로 청구됩니다. 참고: 이 모델은 고객에게 매월 750시간의 무료 사용이 제공되는 Amazon EC2 사용량 프리 AWS 티어와 다릅니다. 시간당 월별 - 시간당 요금과 월별 요금이 독립적으로 적용됩니다. 월 사용료는 사용량에 관계없이 매달 청구되며, 시간당 요금은 시간당 사용을 기준으로만 적용됩니다. 연간 시간당 - 고객은 인스턴스 유형 1개의 Amazon EC2 인스턴스 하나에 대해 연간 사용량을 선결제로 구매할 수 있습니다. 판매자는 각 인스턴스 유형에 대한 요금을 설정하며, 시간당 가격에 대한 순 절감을 제공할 수 있습니다. 구매한 연간 구독 수를 초과하는 고객 사용은 판매자가 해당 인스턴스 유형에 대해 설정한 시간당 요금으로 청구됩니다. 다년 및 사용자 지정 기간 포함 시간당 - 이 유형의 제안은 판매자 비공개 제안을 통해서만 사용할 수 있습니다. 판매자 비공개 제안을 사용하면 사용자 지정 계약 기간을 최대 3년까지 지정할 수 있습니다. 선결제를 지정하거나 유연한 결제 일정을 포함할 수 있습니다. 판매자가 각 인스턴스 유형의 요금을 설정합니다. 제안이 유연한 결제 일정을 포함하는 경우 인보이스 날짜, 결제 금액, 제안에 포함된 각 인스턴스 유형의 인스턴스 수도 설정합니다. 유연한 결제 일정을 포함하는 활성 판매자 비공개 제안의 경우 고객이 지정된 수의 인스턴스를 시작한 후 추가로 시작하는 추가 인스턴스는 판매자 비공개 제안에 지정된 시간당 요금이 부과됩니다. 다년 및 사용자 지정 기간 계약에 대한 자세한 내용은 AWS Marketplace 제품에

요금 모델	설명
	대한 비공개 제안 준비 및 비공개 제안 할부 플랜 단원을 참조하세요. 연간 시간당(무료 평가판 포함) - 연간 옵션이 있는 시간당 모델과 동일합니다. 단, 이 모델에는 고객이 사용자가 결정한 일수에 대해 모든 인스턴스 유형 중 1개의 인스턴스를 무료로 실행할 수 있는 무료 평가판이 포함되어 있습니다. 연간 구독은 언제든지 구매할 수 있으며, 무료 평가판 구독과 결합됩니다. 시간당 연간 - 연간 시간당 요금 모델과 동일합니다. 고객은 인스턴스 유형 1개의 Amazon EC2 인스턴스 하나에 대해 연간 사용량을 선결제로 구매할 수 있습니다. 판매자는 각 인스턴스 유형에 대한 요금을 설정하며, 시간당 가격에 대한 순 절감을 제공할 수 있습니다. 그러나 절감 제공은 필수 사항이 아닙니다. 구매한 연간 구독 수를 초과하는 고객 사용은 판매자가 해당 인스턴스 유형에 대해 설정한 시간당 요금으로 청구됩니다. 시간당 포함 다년 및 사용자 지정 기간 - AWS Marketplace 제품에 대한 비공개 제안 준비을 통해서만 사용할 수 있습니다. 판매자 비공개 제안을 사용하면 최대 3년까지 사용자 지정 기간 계약을 지정할 수 있습니다. 고객에게 선결제를 요구하거나 유연한 결제 일정을 제공할 수 있습니다. 계약 기간 동안 각 인스턴스 유형의 요금과 추가로 시작되는 인스턴스에 대한 시간당 요금을 설정합니다. 유연한 결제 일정을 제공하는 경우 인보이스 날짜, 결제 금액, 제안에 포함된 각 인스턴스 유형의 인스턴스 수도 설정합니다. 유연한 결제 일정을 포함하는 활성 비공개 제안의 경우 고객이 지정된 수의 인스턴스를 시작한 후 추가로 시작하는 추가 인스턴스는 비공개 제안에 지정된 시간당 요금이 부과됩니다. 다년 및 사용자 지정 기간 계약에 대한 자세한 내용은 AWS Marketplace 제품에 대한 비공개 제안 준비 및 비공개 제안 할부 플랜 단원을 참조하세요.

요금 모델	설명
	Note 무료 평가판 및 연간 요금은 월별 요금과 결합할 수 없습니다.
월별 결제	월별 - 고객이 실행하는 인스턴스의 수에 관계없이 고정된 월별 기준으로 소프트웨어의 요금을 지불합니다. 월별 요금은 가입 및 취소 시 비례 할당됩니다. 예: 한 달 중 하루를 구독한 고객에게는 한 달의 1/30에 대해 요금이 부과됩니다. 시간당 월별 - 시간당 요금과 월별 요금이 독립적으로 적용됩니다. 월 사용료는 사용량에 관계없이 매달 청구되며, 시간당 요금은 시간당 사용을 기준으로만 적용됩니다. Note 무료 평가판 및 연간 요금은 월별 요금과 결합할 수 없습니다.
유료 사용 요금	사용자, 데이터, 대역폭 또는 호스트의 네 가지 사용량 범주 중 하나와 함께 판매자가 제공하는 가치에 해당하는 요금이 소프트웨어에 직접 부과됩니다. 제품에 대해 최대 24개의 차원을 정의할 수 있습니다. 모든 요금은 고객에 의해 시간당 발생합니다. 모든 사용량은 매월 계산되며 기존 AWS Marketplace 소프트웨어와 동일한 메커니즘을 사용하여 매월 청구됩니다. 사용 요금을 AWS Marketplace 측정 서비스라고도 합니다.
계약 요금 모델	계약 요금이 적용되는 AMI - 구매자가 선결제 요금을 지불하는 단일 AMI 제품 또는 AWS CloudFormation 스택이 있는 단일 AMI입니다.

AWS 요금 및 소프트웨어 요금

Amazon Machine Image(AMI) 기반 상품의 요금은 다음과 같은 두 가지 범주로 나뉩니다.

- 인프라 요금 세부 정보 - 모든 AMI 기반 제품에는 사용된 서비스 및 AWS 인프라에 따라 관련 인프라 요금이 부과됩니다. 이러한 요금 및 수수료는에서 정의 및 제어하며 AWS마다 다를 수 있습니다 AWS 리전. 자세한 내용은 Amazon EC2 요금을 참조하세요.
- 소프트웨어 요금 세부 정보 - 유료 제품의 경우 판매자가 소프트웨어 사용 요금을 정의합니다.

이 두 제품 요금 범주는 구매자가 제품 사용의 잠재적 비용을 이해하는 데 도움이 되도록 AWS Marketplace 세부 정보 페이지에 별도로 표시됩니다.

AMI 시간당 제품의 무료 평가판

AMI 시간당 제품은 선택적 무료 평가판 프로그램을 이용할 수 있습니다. 무료 평가판에서 고객은 제품에 대한 소프트웨어 요금 없이 최대 31일 동안 제품을 구독하고 인스턴스 하나를 사용할 수 있습니다. 적용 가능한 AWS 인프라 요금이 여전히 적용됩니다. 무료 평가판 기간이 만료되면 자동으로 유료 구독으로 전환됩니다. 제공된 무료 사용량을 초과하여 사용할 경우 추가 사용량에 대한 요금이 고객에게 부과됩니다. 시간당 제품 무료 평가판을 제공하려면 평가 기간을 정의하고 AWS Marketplace 판매자 운영 팀에 알려야 합니다. 가능한 평가 기간은 5~31입니다.

고객이 무료 평가판 제품을 구독하면 무료 평가판의 조건, 계산된 만료 날짜 및 구독 해지에 대한 세부 정보가 포함된 환영 이메일 메시지가 고객에게 전송됩니다. 미리 알림 이메일 메시지는 만료 날짜 3일 전에 전송됩니다.

에서 무료 평가판 제품을 제공하는 경우 환불 정책에 설명된 특정 환불 정책에 AWS Marketplace동의하는 것입니다.

추가 리소스

AMI 제품 요금에 대한 자세한 내용은 다음 주제를 참조하세요.

- AWS Marketplace Metering Service를 통한 AMI 제품에 대한 사용자 지정 측정 요금
- 의 AMI 제품에 대한 계약 요금 AWS Marketplace

AMI 기반 제품의 청구, 측정 및 라이선싱 통합에 대한 자세한 내용은 다음 항목을 참조하세요.

AMI 제품 요금에 대해서는 다음 주제를 참조하세요.

- AWS Marketplace Metering Service를 사용하여 AMI 제품에 대한 사용자 지정 측정 구성
- 를 사용하여 AMI 기반 제품과 라이선스 연결 AWS License Manager

AWS Marketplace Metering Service를 통한 AMI 제품에 대한 사용자 지정 측정 요금

AWS Marketplace Metering Service를 통해 측정 레코드를 엔드포인트에 전송하여 사용량을 캡처하도록 소프트웨어를 수정할 수 있습니다. 사용량 범주를 선택하고 한 범주에서 최대 24개의 차원을 정의할 수 있습니다. 이러한 차원은 시간당 한 번 측정되고 집계되며 사용자가 정의한 요금제에 대해 부과됩니다. 판매자로서 해야 할 일은 사용할 차원을 결정하는 것입니다. AMI가 게시된 이후에는 이를 변경할 수 없습니다. 다음 섹션에서는 AWS Marketplace Metering Service에 대한 정보를 제공합니다.

주제

- 측정 서비스 개념
- 소프트웨어 요금 책정
- 에 제품 추가 AWS Marketplace
- 측정 서비스를 사용하도록 소프트웨어 수정
- 공급업체 측정 태깅(선택 사항)
- 사용자 지정 측정 구성
- AWS Marketplace Metering Service를 사용하여 AMI 제품에 대한 사용자 지정 측정 구성

측정 서비스 개념

AWS Marketplace Metering Service를 사용하기 전에 다음과 같은 중요한 서비스 개념에 유의하세요.

- 사용 범주 - 측정 서비스를 사용하여 요금이 책정되는 소프트웨어 제품은 고객에게 요금을 청구하는 적절한 방법을 결정하는 한 가지 사용 범주에 따라 분류됩니다. 사용 범주에는 다음이 포함되지만 이에 국한되지는 않습니다.
  - 사용자 - 단일 식별자로 연결되어 있는 정의된 권한 세트입니다. 이 범주는 고객의 사용자가 소프트웨어에 직접 연결하는 소프트웨어에 적합합니다(예: 고객 관계 관리 또는 비즈니스 인텔리전스 보고).
  - 호스트 – 모든 서버, 노드, 인스턴스, 엔드포인트 또는 기타 컴퓨팅 시스템의 일부입니다. 이 범주는 여러 고객 소유 인스턴스를 모니터링 또는 스캔하는 소프트웨어에 적합합니다(예: 성능 또는 보안 모니터링).

  - 데이터 – MB, GB 또는 TB로 측정된 스토리지 또는 정보입니다. 이 범주는 저장된 데이터를 관리하거나 데이터를 배치로 처리하는 소프트웨어에 적합합니다.
  - 대역폭 - Mbps 또는 Gbps 단위로 측정됩니다. 이 범주는 고객이 프로비저닝할 대역폭의 양을 지정할 수 있는 소프트웨어에 적합합니다.
  - 단위 - 측정 단위입니다. 다음에 설명된 예제를 참조하세요.
- 사용량 단위 – 소프트웨어 제품의 특정 사용량 단위는 선택한 사용 범주에 해당합니다. 이 사용량 단위는 소프트웨어가 요금을 부과하는 단위를 설명합니다. 그러한 예는 다음과 같습니다.
  - NodesHrs(호스트 범주에 해당)
  - UserHrs(사용자 범주에 해당)
  - GBStored(데이터 범주에 해당)
- 소비 - 측정 서비스를 사용하여 가격이 책정되는 소프트웨어 제품은 다음 세 가지 방법 중 하나로 사용량에 따른 요금을 부과합니다.
  - 프로비저닝됨 - 고객은 소프트웨어를 통해 사용할 특정 리소스 수(예: 사용자 수 또는 고정 대역폭 수)를 구성할 수 있습니다. 매시간 고객은 프로비저닝한 서비스에 대해 비용을 지불합니다.
  - 동시 - 소프트웨어를 통해 구분된 호스트 또는 사용자는 수에 상관없이 소프트웨어에 연결할 수 있습니다. 매시간 고객은 소프트웨어에 액세스한 호스트 또는 사용자 수를 기준으로 비용을 지불합니다.
  - 누적 - 고객은 소프트웨어를 통해 처리되거나 저장된 모든 양의 데이터를 사용할 수 있습니다. 매시간 고객은 집계된 양에 대해 비용을 지불합니다.
- 요금 - 측정 서비스를 사용하여 요금이 책정되는 소프트웨어 제품은 단일 요금을 지정하거나 최대 24개의 차원을 정의하고 각 요금을 지정해야 합니다. 다음은 요금 옵션에 대한 자세한 내용입니다.
  - 단일 차원 - 가장 단순한 요금 옵션입니다. 고객은 크기 또는 볼륨에 관계없이 시간당 리소스 단위마다 단일 가격을 지불합니다(예: 시간당 사용자마다 0.014 USD 또는 시간당 호스트마다 0.070 USD).
  - 여러 차원 - 이 요금 옵션은 선택한 사용 범주가 여러 축에 따라 다를 때 적합합니다. 예를 들어, 호스트 모니터링의 경우 호스트 크기에 따라 다른 가격이 설정될 수 있습니다. 또는 사용자 기반 요금의 경우 사용자 유형(예: 관리자, 파워 유저 및 읽기 전용 사용자)에 따라 다른 가격이 설정될 수 있습니다.
- 측정 - 모든 사용량은 시간당 1회 측정 이벤트로 기록됩니다. 사용자 소프트웨어는 적합한 크기 및 사용량을 AWS Marketplace 전송하도록 구성되어야 합니다.
  - 할당 - 선택 사항으로, 추적하는 속성별로 사용량을 할당량에 분배할 수 있습니다. 이러한 할당량은 구매자에게 태그로 표시됩니다. 태그를 통해 구매자는 태그에 따라 사용량으로 분할된 비용을 볼 수 있습니다. 예를 들어 사용자별로 요금을 부과하고 사용자에게 "Department" 속성이 있는 경

우 키가 "Department"인 태그와 값마다 하나의 할당을 사용하여 사용량 할당을 생성할 수 있습니다. 이 접근 방식은 판매자가 보고하는 가격, 차원 또는 총 사용량을 변경하지 않습니다. 하지만 고객은 제품에 적합한 범주별로 비용을 볼 수 있습니다.

소프트웨어 요금 책정

AWS Marketplace 측정 서비스로 소프트웨어의 요금을 책정하는 경우 먼저 사용 범주 및 소비 방식을 결정해야 합니다. 이 서비스는 6가지 요금 시나리오를 지원합니다. 제품에 대해 이러한 시나리오 중 하나만 선택해야 합니다.

- 프로비저닝된 사용자(시간당)
- 동시 사용자(시간당)
- 프로비저닝된 호스트(시간당)
- 동시 호스트(시간당)
- 프로비저닝된 대역폭(시간당)
- 누적 데이터(시간당)

그런 다음 선택한 사용 범주에 대해 가격을 책정하는 방식을 결정해야 합니다.

- 단일 가격
- 다중 차원(최대 24개)

에 제품 추가 AWS Marketplace에는 차원 및 요금에 대한 고객 친화적 설명을 제공하는 방법이 설명되어 있습니다.

예: 비선형 요금이 있는 프로비저닝된 대역폭

네트워크 어플라이언스 소프트웨어를 제공하는 경우를 생각해 봅니다. 프로비저닝된 대역폭을 기준으로 요금을 청구하도록 선택합니다. 사용 범주의 경우 대역폭을 선택합니다. 대역폭에 따라 요금을 청구하는 것 외에도 구매자가 확장할 때 다른 가격으로 청구하려고 합니다. 대역폭 범주 내에 여러 차원을 정의할 수 있습니다. 25Mbps, 100Mbps 및 1Gbps에 대해 구별된 가격을 정의할 수 있습니다.

예: 여러 차원이 있는 동시 호스트

다른 Amazon EC2 인스턴스를 모니터링하는 소프트웨어를 제공하는 경우를 생각해 봅니다. 모니터링되는 호스트 수를 기준으로 요금을 청구하도록 선택합니다. 사용 범주의 경우 호스트를 선택합니다. 호

스트에 따라 요금을 청구하는 것 외에도 대형 호스트를 모니터링하기 위한 추가 값을 청구하려고 합니다. 호스트 범주 내에 여러 차원을 사용할 수 있습니다. 마이크로, 스몰, 중간, 라지, x-라지, 2XL, 4XL, 8XL 인스턴스에 대해 구별된 가격을 정의할 수 있습니다. 소프트웨어는 각 특정 호스트를 정의된 차원 하나에 매핑할 책임이 있습니다. 소프트웨어는 사용 범주의 각 차원에 대해 별도의 측정 레코드(있는 경우)를 전송할 책임이 있습니다.

에 제품 추가 AWS Marketplace

측정 서비스를 활용하려면가 나열 AWS Marketplace 할 새 제품을 생성해야 합니다. 제품이 이미에 있는 경우 AWS Marketplace현재 제품 외에 새 AWS Marketplace 측정 서비스 제품을 사용할 수 있는지 또는 새 사용자가 사용할 수 있는 유일한 버전으로 현재 제품을 교체할지 결정해야 합니다. 교체를 선택하면 새 구매자가 더 이상 사용할 수 AWS Marketplace 없도록 기존 제품이에서 제거됩니다. 기존 고객은 이전 제품과 인스턴스에 계속 액세스할 수 있지만 편의를 위해 새 제품으로 마이그레이션할 수 있습니다. 새 제품은 측정 서비스를 사용하도록 소프트웨어 수정에 설명된 대로 AWS Marketplace 측정 서비스의 사용을 측정해야 합니다.

AMI를 보유한 경우 표준 프로세스에 따라 셀프 서비스 도구를 사용하여 AMI를 공유하고 스캔합니다. 또한 관리 포털에서 사용 가능한 템플릿을 사용하여 제품 로드 양식을 작성하고 업로드하여 수집 프로세스를 시작합니다.

다음 정의를 사용하여 AWS Marketplace 측정 서비스에 대한 제품 로드 양식의 필드를 작성합니다. 제품 로드 양식에서 이러한 필드는 시간당 및 월별 요금 제품과 구분하기 위해 유연한 소비 요금(FCP)으로 레이블이 지정됩니다.

- 제목 -에 이미 제품이 AWS Marketplace 있고와 동일한 제품을 추가하는 경우 AWS Marketplace Metering Service괄호 안에 FCP 범주와 차원을 포함하여 구분합니다(예: “PRODUCT TITLE (Data)”).
- 요금 모델 - 드롭다운 목록에서 사용량을 선택합니다.
- FCP 범주 - 사용량 요금 구성 요소가 있는 유료 제품에 대해 고객에게 요금이 청구되는 범주입니다. 드롭다운 목록에서 사용자, 호스트, 데이터 또는 대역폭을 선택합니다.
- FCP 단위 - 사용 요금 구성 요소로 유료 제품에 대해 고객에게 요금이 청구되는 측정 단위입니다. 옵션은 선택한 FCP 범주에 따라 드롭다운 목록에 나타납니다. 다음 표에는 각 범주에 대한 유효 단위가 나열되어 있습니다.

카테고리	유효 단위
Users	UserHrs

카테고리	유효 단위
호스트	HostHrs
Data	MB, GB, TB
대역폭	Mbps, Gbps

- FCP 차원 이름 - `MeterUsage` 작업을 호출하여 측정 레코드를 전송할 때 사용되는 이름입니다. 이 이름은 결제 보고서에 공개됩니다. 하지만 외부에 공개되지는 않으므로 사용자에게 친숙한 이름일 필요는 없습니다. 이름은 15자를 초과할 수 없으며 영숫자 및 밑줄만 포함할 수 있습니다. 이름을 설정하고 제품을 공개한 후에는 변경할 수 없습니다. 이름을 변경하려면 새 AMI가 필요합니다.
- FCP 차원 설명 - 제품에 대한 차원을 설명하는 고객용 설명문입니다. 설명은 70자를 초과할 수 없으며 사용자에게 친숙해야 합니다. 설명의 예로는 시간당 관리자 수, 프로비저닝된 Mbps당 대역폭 등이 있습니다. 제품을 게시한 후 이 설명을 변경할 수 있습니다.
- FCP 요금 - 이 제품의 단위당 소프트웨어 요금입니다. 이 필드는 소수 자릿수 8자리를 지원합니다.

참고:

- 시간당 및 연간 요금 필드를 작성할 필요가 없습니다.
- 무료 평가판 및 연간 요금은 호환되지 않습니다.
- AMI, 클러스터 및 AWS 리소스 기능을 사용하는 제품은 AWS Marketplace 측정 서비스를 사용할 수 없습니다.
- 가격, 인스턴스 유형 또는 AWS 리전 변경 사항은 다른 AWS Marketplace 제품과 동일한 프로세스를 따릅니다.
- AWS Marketplace 측정 서비스가 있는 제품은 시간당, 월별 또는 기존 보유 라이선스 사용(BYOL)과 같은 다른 요금 모델로 변환할 수 없습니다.
- AWS Marketplace 에서는 사용 지침 또는 문서에 IAM 정책 정보를 추가할 것을 권장합니다.
- 총 24개의 FCP 차원을 포함할 수 있습니다. 차원을 생성하고 게시한 후에는 기존 차원을 수정할 수 없지만 새 차원을 추가할 수 있습니다(최대 24개).

궁금한 점은 AWS Marketplace 판매자 작업 팀에 문의하세요.

측정 서비스를 사용하도록 소프트웨어 수정

고객 사용을 기록하고, 시간당 사용 보고서를 측정 서비스에 전송하고, 새 실패 모드를 처리하려면 소프트웨어를 수정해야 합니다. 소프트웨어는 요금과 관계없이 작동하지만, 소프트웨어에서 사용 범주, 소비 방식 및 차원에 대해 알고 있어야 합니다.

소비 측정

소프트웨어에서는 선택한 사용 범주의 양과 고객이 소비한 차원을 결정해야 합니다. 이 값은 매 시간 한 번 측정 서비스로 전송됩니다. 어떤 경우에도 소프트웨어에 시간당 기준으로 측정 서비스에 리소스 소비를 전송하기 위해 리소스 소비를 측정, 기록 및 읽을 수 있는 기능이 있다고 가정합니다.

프로비저닝된 소비의 경우 일반적으로 소프트웨어 구성에서 샘플링된 값으로 판독되지만, 매 시간 기록된 최대 구성 값일 수도 있습니다. 동시 소비의 경우 주기적 샘플 또는 매 시간 기록되는 최대 값일 수 있습니다. 누적 소비의 경우 매 시간 누적되는 값입니다.

여러 차원에 대한 요금의 경우 여러 값을 측정하여 차원마다 하나를 측정 서비스로 전송해야 합니다. 이렇게 하려면 AMI를 제공할 때 알려진 차원 집합으로 소프트웨어를 프로그래밍하거나 구성해야 합니다. 제품을 생성한 후에는 차원 집합을 변경할 수 없습니다.

각 요금 시나리오와 관련하여 다음 표에는 시간당 사용량을 측정하는 권장 방식이 설명되어 있습니다.

시나리오	측정 방법
프로비저닝된 사용자	현재 프로비저닝된 사용자 수입니다(샘플링됨). -또는- 최대 프로비저닝된 사용자 수입니다(해당 시간 표시).
동시 사용자	현재 동시 사용자 수입니다(샘플링됨). -또는- 최대 동시 사용자 수입니다(해당 시간 표시). -또는- 구분된 사용자의 총 수입니다(해당 시간 표시).

시나리오	측정 방법
프로비저닝된 호스트	현재 프로비저닝된 호스트 수입니다(샘플링됨). -또는- 최대 프로비저닝된 호스트 수입니다(해당 시간 표시).
동시 호스트	현재 동시 호스트 수입니다(샘플링됨). -또는- 최대 동시 호스트 수입니다(해당 시간 표시). -또는- 구분된 호스트의 총 수입니다(해당 시간 표시).
프로비저닝된 대역폭	현재 프로비저닝된 대역폭 설정입니다(샘플링됨). -또는- 최대 프로비저닝된 대역폭입니다(해당 시간 표시).
누적 데이터	현재 저장된 데이터의 GB입니다(샘플링됨). -또는- 저장된 데이터의 최대 GB입니다(해당 시간 표시). -또는- 해당 시간에 추가 또는 처리된 데이터의 총 GB입니다. -또는- 해당 시간에 처리된 데이터의 총 GB입니다.

공급업체 측정 태깅(선택 사항)

공급업체 측정 태깅을 사용하면 독립 소프트웨어 개발 판매 회사(ISV)는 구매자에게 소프트웨어 사용에 대한 보다 세밀한 통찰력을 제공하고 구매자의 비용 할당을 도와줄 수 있습니다.

구매자의 소프트웨어 사용에 태그를 지정하려면 비용 할당 방법을 결정해야 합니다. 먼저 구매자에게 비용 할당에서 보고 싶은 것이 무엇인지 물어봅니다. 그 후 구매자 계정에 대해 추적하는 모든 속성에 사용량을 분할할 수 있습니다. 속성의 예로는 `Account ID`, `Business Unit`, `Cost Centers` 및 기타 제품 관련 메타데이터가 있습니다. 이러한 속성은 구매자에게 태그로 노출됩니다. 구매자는 태그를 사용하여 AWS 결제 콘솔(https://console.aws.amazon.com/costmanagement/)에서 태그 값을 기준으로 사용량으로 분할된 비용을 볼 수 있습니다. 공급업체 측정 태깅은 판매자가 신고한 가격, 크기 또는 총 사용량을 변경하지 않습니다. 이를 통해 고객은 제품에 적합한 범주별로 비용을 볼 수 있습니다.

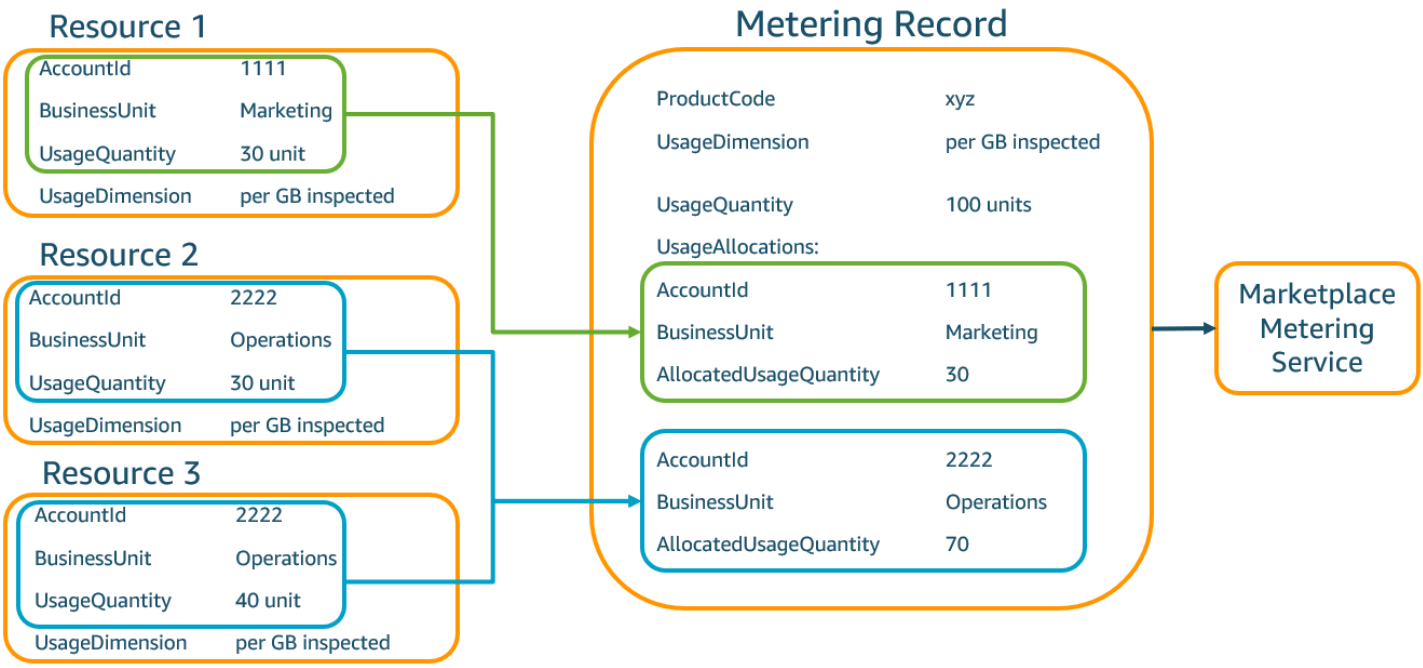
일반적으로 구매자는 AWS 계정하나로 제품을 구독합니다. 또한 구매자의 수많은 사용자가 동일한 제품 구독에 연결됩니다. `Account ID` 키가 있는 태그를 사용하여 사용량 할당을 생성한 다음, 각 사용자에게 사용량을 할당할 수 있습니다. 이 경우 구매자는 과금 정보 및 비용 관리 콘솔에서 `Account ID` 태그를 활성화하고 개별 사용자 사용량을 분석할 수 있습니다.

판매자 환경

판매자는 모든 리소스 사용량을 집계하는 대신 동일한 태그 세트를 사용하여 리소스 측정 기록을 집계할 수 있습니다. 예를 들어 판매자는 `UsageAllocations`의 여러 버킷을 포함하는 측정 기록을 구성할 수 있습니다. 각 버킷은 태그 세트의 `UsageQuantity`(예: `AccountId`, `BusinessUnit`)를 나타냅니다.

다음 다이어그램의 리소스 1은 고유한 `AccountId` 및 `BusinessUnit` 태그 세트가 있으며 Metering Record(측정 기록)에 단일 항목으로 표시됩니다.

리소스 2와 리소스 3은 동일한 `AccountId`, 2222, `BusinessUnit` 및 `Operations` 태그를 갖고 있습니다. 따라서 두 리소스는 측정 기록의 단일 `UsageAllocations` 항목으로 결합됩니다.



판매자는 태그가 없는 리소스를 사용량 수량이 할당된 단일 `UsageAllocation`으로 결합하고 `UsageAllocations`의 항목 중 하나로 전송할 수도 있습니다.

다음과 같은 제한이 있습니다.

- 태그 수 - 5
- `UsageAllocations` 크기(카디널리티) - 2,500

다음 사항을 확인합니다.

- 태그 키 및 값에 허용되는 문자 – a-zA-Z0-9+ -=._:\/@
- `UsageAllocation` 목록의 최대 태그 수 - 5
- 두 `UsageAllocations`의 태그가 같으면(즉, 동일한 태그 키와 값의 조합) 안 됩니다. 같을 경우 둘 다 동일한 `UsageAllocation`을 사용해야 합니다.
- `UsageAllocation`의 `AllocatedUsageQuantity` 합계는 총 사용량인 `UsageQuantity`와 같아야 합니다.

구매자 경험

다음 표는 구매자가 `AccountId` 및 `BusinessUnit` 공급업체 태그를 활성화한 이후의 구매자 경험 예시를 보여줍니다.

이 예에서 구매자는 비용 사용 보고서에서 할당된 사용량을 볼 수 있습니다. 공급업체 측정 태그는 “`aws:marketplace:isv`” 접두사를 사용합니다. 구매자는 과금 정보 및 비용 관리의 비용 할당 태그 아래에 있는 AWS생성 비용 할당 태그에서 태그를 활성화할 수 있습니다.

비용 사용 보고서의 첫 번째 행과 마지막 행은 판매자 환경 예제처럼 판매자가 측정 서비스에 보내는 내용과 관련이 있습니다.

비용 사용 보고서(단순)

ProductCode	구매자	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	네트워크: 검사한 크기(GB)당	70	2222	운영
xyz	111122223333	네트워크: 검사한 크기(GB)당	30	3333	Finance
xyz	111122223333	네트워크: 검사한 크기(GB)당	20	4444	IT
xyz	111122223333	네트워크: 검사한 크기(GB)당	20	5555	마케팅
xyz	111122223333	네트워크: 검사한 크기(GB)당	30	1111	마케팅

코드 예제는 사용량 할당 태그 지정을 사용하는 `MeterUsage`(선택 사항) 섹션을 참조하세요.

사용자 지정 측정 구성

AWS Marketplace Metering Service 작업에 대한 자세한 내용은 AWS Marketplace Metering Service를 사용하여 AMI 제품에 대한 사용자 지정 측정 구성 섹션을 참조하세요.

AWS Marketplace Metering Service를 사용하여 AMI 제품에 대한 사용자 지정 측정 구성

> **Note**
>
> 사용자 지정 측정 요금이 적용되는 AMI 기반 제품의 경우 소프트웨어는 Amazon EC2 인스턴스에 연결된 Amazon Elastic Compute Cloud에 대한 IAM 역할의 임시 AWS 자격 증명을 사용하여 MeterUsage API를 호출해야 합니다. 장기 액세스 키 사용은 지원되지 않습니다.

AWS Marketplace Metering Service는 사용 범주별로 소프트웨어에 직접 요금을 부과하는 데 사용할 수 있는 요금 및 측정 기능입니다. 사용자, 데이터, 대역폭, 호스트, 단위의 다섯 가지 사용 범주가 있습니다. Amazon Machine Image(AMI) 기반, 컨테이너 기반 및 서비스형 소프트웨어(SaaS) 기반 제품에서 측정 서비스를 사용할 수 있습니다. 다음 섹션에서는 AWS Marketplace Metering Service를 사용하여 사용자 지정 측정을 구성하는 방법에 대한 자세한 정보를 제공합니다.

AWS Marketplace 측정 서비스는 여러 가지 새로운 요금 시나리오를 활성화합니다. 예를 들어, 소프트웨어에서 호스트를 모니터링하는 경우, 모니터링된 각 호스트에 대해 요금을 부과합니다. 호스트 크기에 따라 다른 요금을 가질 수 있으며 매시간 모니터링되는 동시 호스트 수에 대해 요금을 부과할 수 있습니다. 마찬가지로 소프트웨어가 조직 전체에서 많은 사용자를 허용하는 경우 사용자 수에 따라 요금을 부과할 수 있습니다. 시간마다 고객에게 프로비저닝된 사용자의 총 수에 대해 요금이 부과됩니다.

자세한 내용은 AWS Marketplace 측정 서비스 API 참조를 참조하세요.

AMI 기반 제품의 AWS Marketplace Metering Service API를 사용자 지정 측정 요금과 통합하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 워크숍의 사용자 지정 단위 랩에서 가격이 책정된 AMI 제품 나열을 참조하세요.

주제

- 요구 사항
- AWS Marketplace 측정 서비스 호출
- 장애 처리
- 제한 사항

- 코드 예제

요구 사항

측정 서비스를 사용하는 모든 AMI 기반 소프트웨어는 다음 요구 사항을 충족해야 합니다.

- 소프트웨어는 Amazon Machine Image(AMI)를 AWS Marketplace 통해에서 시작해야 합니다.
- 에 기존 제품이 있는 경우 새 AMI를 제출하고 새 제품을 생성하여이 기능을 활성화해야 AWS Marketplace합니다.
- 모든 소프트웨어는 AWS Identity and Access Management (IAM) 역할로 프로비저닝해야 합니다. 최종 고객은 사용자가 소프트웨어를 통해 프로비저닝하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 IAM 역할을 추가해야 합니다. 소프트웨어를 배포할 때 IAM 역할 사용은 선택 사항입니다 AWS Marketplace. AWS Marketplace 측정 서비스 소프트웨어를 배포할 때에는 필수입니다.
- 사용자 소프트웨어는 몇 가지 방식으로 소비를 결정할 수 있어야 합니다.

AWS Marketplace 측정 서비스 호출

소프트웨어에서는 시간별로 측정 서비스를 호출하고 해당 시간에 대한 소비 값을 기록해야 합니다.

소프트웨어가 시작할 때 시작된 시간-분을 기록해야 합니다. 이를 시작-분이라고도 합니다. 시작-분에서 매 시간마다 소프트웨어는 해당 시간의 소비 값을 확인하고 측정 서비스를 호출해야 합니다. 이 값을 구하는 방법에 대한 자세한 내용은 측정 서비스를 사용하도록 소프트웨어 수정을 참조하세요.

시작-분에서 매시간 설정 상태로 유지하려면 소프트웨어에서 다음 접근 방식 중 하나를 사용해야 합니다.

- 소프트웨어 내의 스레드
- 인스턴스 또는 소프트웨어로 시작하는 데몬 프로세스
- 애플리케이션을 시작하는 동안 구성되는 cron 작업

> **Note**
>
> 소프트웨어는 고객 인스턴스에서 구성된 IAM 역할을 사용하여 AWS Marketplace 측정 서비스를 호출하고 소비 차원 및 양을 지정해야 합니다.

소프트웨어는 SDK를 사용하여 다음 예제 구현과 마찬가지로 AWS Marketplace 측정 서비스를 AWS 호출할 수 있습니다.

1. 인스턴스 프로파일을 사용하여 서비스 클라이언트를 생성합니다. 이렇게 하려면 EC2 인스턴스에 대해 구성된 역할이 필요합니다. 역할 자격 증명은 SDK에 의해 자동으로 새고 고쳐집니다.
2. 매 시간 소프트웨어 구성 및 상태를 읽고 해당 시간에 대한 소비 값을 결정합니다. 여기에는 차원당 값 수집이 포함될 수 있습니다.
3. 다음과 같은 파라미터를 사용하여 SDK 클라이언트에서 `meterUsage` 메서드를 호출합니다(사용이 있는 각 차원에 대해 추가적으로 호출).
   - `timestamp` - 기록되는 시간의 타임스탬프(UTC)입니다.
   - `productCode` - 소프트웨어에 할당된 제품 코드입니다.
   - `dimension` - 소프트웨어에 할당된 차원입니다.
   - `quantity` - 시간의 소비 값입니다.
   - `allocations` - (선택 사항) 추적하는 속성 전체에 걸쳐 사용량을 할당할 수 있습니다. 이러한 할당량을 합산하여 레코드의 총 소비량을 구해야 합니다. 구매자에게 이러한 태그는 결제 도구(예: AWS 결제 및 비용 관리 콘솔)에 잠재적 비용 할당 태그로 표시됩니다. 구매자는 이러한 태그를 사용하여 비용을 추적하려면 계정에서 태그를 활성화해야 합니다.

또한 소프트웨어는 리전 내 AWS Marketplace 측정 서비스 엔드포인트를 호출해야 합니다. `us-east-1`이 레코드를 `us-east-1` 엔드포인트로 전송하고 `us-west-2`가 레코드를 `us-west-2` 엔드포인트로 전송하도록 제품의 리전 엔드포인트를 올바르게 설정해야 합니다. 리전 내 호출을 수행하면 구매자에게 더 안정적인 경험을 제공하며, 관련되지 않은 리전의 가용성이 다른 리전에서 실행되는 소프트웨어에 영향을 주는 상황을 방지합니다.

서비스에 측정 레코드를 전송할 때 사용자의 리전에 있는 AWS Marketplace 측정 서비스에 연결해야 합니다. `getCurrentRegion()` 헬퍼 메서드를 사용하여 EC2 인스턴스가 실행 중인 리전을 결정한 다음 이 리전 정보를 `MeteringServiceClient` 생성자로 전달합니다. SDK 생성자 AWS 리전 에서를 지정하지 않으면 기본 `us-east-1` 리전이 사용됩니다. 애플리케이션에서 서비스에 대해 교차 리전 호출을 수행할 경우 해당 호출이 거부됩니다. 자세한 내용은 애플리케이션의 현재 리전 확인 및 getCurrentRegion()을 참조하세요.

장애 처리

제품은 사용량을 캡처하고 요금을 청구할 수 있도록 퍼블릭 인터넷 엔드포인트인 서비스에 측정 레코드를 전송해야 합니다. 고객이 측정 레코드를 전달되지 않도록 하는 방식으로 네트워크 설정을 수정할 수 있으므로, 제품은 장애 모드를 선택하여 이를 설명해야 합니다.

> **Note**
>
> 일부 측정 실패는에 연결할 때 일시적인 문제일 수 있습니다 AWS Marketplace Metering Service. AWS Marketplace 에서는 단기 중단 또는 네트워크 문제를 방지하기 위해 지수 백오프를 통해 최대 30분 동안 재시도를 구현할 것을 적극 권장합니다.

일반적으로 소프트웨어는 장애 시 열림(경고 메시지를 제공하지만 전체 기능은 유지) 또는 장애 시 닫힘(연결이 다시 설정될 때까지 애플리케이션의 모든 기능을 비활성화)을 수행할 수 있습니다. 장애 시 열림, 장애 시 닫힘 또는 애플리케이션에 고유한 사항을 선택할 수 있습니다. 2시간 이내의 측정 실패 후에는 장애 시 닫힘을 사용하지 않는 것이 좋습니다.

장애 시 부분적으로 열림의 예로, 소프트웨어에 대한 액세스를 계속 허용할 수 있지만 구매자가 소프트웨어 설정을 수정하는 것을 허용하지 않을 수 있습니다. 또는 구매자가 계속 소프트웨어에 액세스할 수 있지만 추가 사용자를 생성할 수 없습니다. 소프트웨어는 이 장애 모드를 정의하고 적용할 책임이 있습니다. 소프트웨어의 장애 모드에는 AMI를 제출할 시기가 포함되어야 하며, 나중에 변경할 수 없습니다.

제한 사항

측정 서비스 활성화 소프트웨어를 설계 및 제출할 때 이 제한 사항을 고려하세요.

- 고객에 대한 IAM 역할 및 인터넷 게이트웨이 요구 사항 - 고객은 인터넷 게이트웨이를 보유해야 하고 특정 권한을 가진 IAM 역할로 소프트웨어를 시작해야 합니다. 자세한 내용은 AWS Marketplace 측정 및 권한 부여 API 권한 단원을 참조하세요. 두 가지 조건을 충족하지 못하면 소프트웨어에서 측정 서비스에 연결할 수 없습니다.
- 기존 측정 서비스 제품에 새로운 사용 범주를 추가하거나 사용 범주를 변경할 수 없음 - 고객이 소프트웨어 제품을 구독하는 경우 이용 약관에 동의하는 것으로 간주됩니다. 측정 서비스를 통해 제품의 차원을 변경하려면 새 제품과 새 구독이 필요합니다.
- 기존 측정 서비스 제품의 차원을 수정할 수 없음 - 고객이 소프트웨어 제품을 구독하는 경우 이용 약관에 동의하는 것으로 간주됩니다. 측정 서비스를 통해 제품의 차원을 변경하려면 새 제품과 새 구독이 필요합니다. 기존 제품에 새 차원을 최대 24개까지 추가할 수 있습니다.
- 무료 평가판 및 연간 구독 없음 - 측정 서비스 제품은 시작 시 무료 평가판 및 연간 구독을 지원하지 않습니다.
- 다중 인스턴스 또는 클러스터 기반 배포 고려 사항 - 일부 소프트웨어는 다중 인스턴스 배포의 일환으로 배포됩니다. 소프트웨어 설계 시 사용량 측정 방식 및 위치, 측정 기록을 내보내는 위치를 고려하세요.

코드 예제

다음 코드 예제는 AMI 제품을 제품 게시 및 유지 관리에 필요한 AWS Marketplace API와 통합하는 데 도움이 되도록 제공됩니다.

사용량 할당 태그 지정을 사용하는 **MeterUsage**(선택 사항)

다음 코드 예제는 소비 요금 모델이 적용되는 AMI 제품과 관련이 있습니다. Python 예제에서는 적절한 사용량 할당 태그가 달린 측정 레코드를 AWS Marketplace 에 전송하여 고객에게 사용한 만큼만 지불 요금을 청구합니다.

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited

# Import AWS Python SDK
import boto3
import time

usageRecord = [
    {
        "AllocatedUsageQuantity": 2,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "IT" },
                { "Key": "AccountId", "Value": "123456789" },
            ]

    },
    {
        "AllocatedUsageQuantity": 1,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "Finance" },
                { "Key": "AccountId", "Value": "987654321" },
            ]

    }
]
```

```
marketplaceClient = boto3.client("meteringmarketplace")

response = marketplaceClient.meter_usage(
    ProductCode="testProduct",
    Timestamp=int(time.time()),
    UsageDimension="Dimension1",
    UsageQuantity=3,
    DryRun=False,
    UsageAllocations=usageRecord
)
```

`MeterUsage`에 대한 자세한 내용은 AWS Marketplace 측정 서비스 API 참조의 MeterUsage를 참조하세요.

응답의 예

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

의 AMI 제품에 대한 계약 요금 AWS Marketplace

Amazon Machine Image(AMI) 기반 제품의 계약 요금은 구매자가 AWS CloudFormation 스택이 있는 단일 AMI 제품 또는 단일 AMI에 대해 선결제 요금을 지불한다는 것을 의미합니다. 계약 요금이 적용되는 AMI 기반 제품의 경우는 사용자와 고객 간의 계약에 따라 사용자가 정의한 결제 일정에 따라 고객에게 선결제 요금을 AWS Marketplace 청구합니다. 그 이후 고객에게 해당 리소스를 사용할 권한이 부여됩니다. 이 주제에서는 계약 요금에 대한 자세한 정보를 제공합니다.

요금을 설정하려면 고객에게 제공하는 하나 이상의 계약 기간을 선택합니다. 각 계약 기간에 다른 요금을 입력할 수 있습니다. 1개월, 12개월, 24개월, 36개월 중에 선택할 수 있습니다. 비공개 제안의 경우 사용자 지정 기간을 월 단위로(최대 60개월) 지정할 수 있습니다.

> **Note**
>
> 계약 요금이 적용되는 AMI 제품에는 무료 평가판을 사용할 수 없습니다.

제품 요금을 가장 잘 설명하는 범주를 선택합니다. 요금 범주는 AWS Marketplace 웹 사이트에 고객에게 표시됩니다. 대역폭(GB/s, MB/s), 데이터(GB, MB, TB), 호스트, 요청, 티어 또는 사용자 중에서 선택할 수 있습니다. 미리 정의된 범주 중 요구 사항에 적합한 것이 없는 경우 더 일반적인 단위 범주를 선택할 수 있습니다.

이 제안에는 최대 24개 차원을 추가할 수 있습니다. 각 차원에는 다음 데이터가 필요합니다.

- 계약 범주 - 사용량 기반 요금이 없는 계약 제품의 경우 계약의 차원 범주와 가장 유사한 범주를 선택하거나, 계약의 차원 단위와 유사한 값이 없으면 단위를 선택하면 됩니다.
- 계약 단위 - 선택한 범주에 따라 차원과 가장 유사한 단위의 사용 가능한 값 중에서 하나를 선택합니다.
- 계약 차원 복수 구매 허용 - 이 필드는 제안이 계층별 요금 제안인지 아니면 비계층 제안인지 나타내는 데 사용됩니다.

  계층형 제안 - 구매자가 제안에 제공된 차원 중에서 하나만 구독할 수 있습니다. 계층형 제안의 차원은 수량이라는 개념이 없습니다. 특정 차원이 포함된 계약을 체결한다는 것은 본질적으로 구매자가 해당 차원에 지정된 특정 기능을 선택했다는 뜻입니다.

  비계층형 제안 - 고객이 차원을 2개 이상 조달하고 각 차원의 여러 단위를 조달할 수 있도록 계약서에 허용합니다.

  이 필드에 값을 true로 설정하면 해당 제안은 비계층형 제안이라는 뜻입니다. 이 필드에 값을 false로 설정하면 해당 제안은 계층형 제안이라는 뜻입니다.

제품 로드 양식(PLF)을 사용하여 AMI 제품에 대한 계약을 생성할 때, 요금 차원에 대한 다음 필드를 정의해야 합니다.

- 계약 DimensionX API 이름 - 구매자의 AWS License Manager 계정에서 생성된 라이선스에 표시되어야 하는 이름입니다. 이 이름은 `Checkoutlicense` API 직접 호출에서 `Entitlement`의 `Name` 값으로도 사용됩니다.
- 계약 DimensionX 표시 이름 - AWS Marketplace 웹 사이트의 제품 세부 정보 페이지와 조달 페이지에 표시되는 차원의 고객용 이름입니다. 사용자에게 친숙한 이름을 지정합니다. 이름의 최대 길이는 24자입니다. 목록이 공개된 후에는 `Name` 값을 변경할 수 없습니다.
- 계약 DimensionX 설명 - 특정 차원이 제공하는 기능과 같이 제품의 차원에 대한 추가 정보를 제공하는 차원의 고객용 설명입니다. 설명의 최대 길이는 70자입니다.
- 계약 차원X 수량 - 제품 계약을 수정하는 경우 비례 배분을 계산하는 데 사용됩니다. 모든 계약 제안에서 이 필드의 값을 1로 설정해야 합니다. 편집하면 안 됩니다.
- 계약 차원X 1개월 요율 — 이 차원을 기준으로 1개월간의 자격에 대해 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 세 자리를 지원합니다.
- 계약 DimensionX 12개월 요율 - 이 차원에 대한 권한을 12개월 동안 부여할 때 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 세 자리를 지원합니다.

- 계약 DimensionX 24개월 요율 - 이 차원에 대한 권한을 24개월 동안 부여할 때 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 세 자리를 지원합니다.
- 계약 DimensionX 36개월 요율 - 이 차원에 대한 권한을 36개월 동안 부여할 때 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 세 자리를 지원합니다.

예: 데이터 스토리지 애플리케이션

	1개월 요금	12개월 요금	24개월 요금	36개월 요금
암호화되지 않은 데이터(GB)	1.50 USD/GB	16.00 USD/GB	30.00 USD/GB	GB당 60.00 USD
암호화된 데이터(GB)	1.55 USD/GB	16.60 USD/GB	31.20 USD/GB	GB당 61.20 USD

예제: 로그 모니터링 제품

	1개월 요금	12개월 요금	24개월 요금	36개월 요금
기본(10개의 호스트 모니터링, 5개의 컨테이너 모니터링)	100 USD	1000 USD	2000 USD	4000 USD
스탠다드(20개의 호스트 모니터링, 10개의 컨테이너 모니터링)	$200	2000 USD	4000 USD	8,000 USD
프로(40개의 호스트 모니터링, 20개의 컨테이너 모니터링)	400 USD	4000 USD	8,000 USD	16,000 USD
시간당 추가 호스트 모니터링	10 USD	100 USD	$200	400 USD

	1개월 요금	12개월 요금	24개월 요금	36개월 요금
시간당 추가 컨테이너 모니터링	10 USD	100 USD	$200	400 USD

> **Note**
>
> 요금은 1개월, 12개월, 24개월, 36개월 기간일 수 있습니다. 제품에 대해 이러한 옵션을 하나 이상 제공하도록 선택할 수 있습니다. 기간은 각 차원에 대해 동일해야 합니다.
>
> Example
>
> 예를 들어 `ReadOnlyUsers` 및 `AdminUsers` 차원이 있는 경우 ReadOnlyUsers에 대한 연간 가격을 제공하면 `AdminUsers`에 대해서도 연간 가격을 제공해야 합니다.

자동 갱신

고객이 AMI 계약을 AWS Marketplace 사용하여 제품을 구매할 때 계약 조건을 자동으로 갱신하는 데 동의할 수 있습니다. 고객은 매달, 1년마다, 2년마다 또는 3년마다 권한 부여에 대한 요금을 계속 지불합니다.

고객은 언제든지 갱신 설정을 변경할 수 있습니다. 자세한 내용은 AWS Marketplace 구매자 안내서의 기존 계약 수정을 참조하세요.

를 사용하여 AMI 기반 제품과 라이선스 연결 AWS License Manager

계약 요금이 적용되는 Amazon Machine Image(AMI) 기반 제품의 경우 AWS License Manager를 사용하여 라이선스를 제품과 연결할 수 있습니다. AWS License Manager 는 애플리케이션이 고객이 구매한 라이선스(권한이라고도 함)를 추적하고 업데이트할 수 있는 라이선스 관리 도구입니다. 통합이 완료되면 제품 목록을 AWS Marketplace에 게시할 수 있습니다. 다음 섹션에서는 AWS License Manager를 사용하여 라이선스를 AMI 기반 제품과 연결하는 방법에 대한 자세한 정보를 제공합니다.

에 대한 자세한 내용은 AWS License Manager 사용 설명서 및 AWS CLI 명령 AWS License Manager 참조의 AWS License Manager 섹션을 참조하세요.

AMI 기반 제품과 계약 요금 AWS License Manager 의 통합에 대한 자세한 내용은 AWS Marketplace 판매자 워크숍의 선결제 랩에서 가격이 책정된 AMI 제품 나열을 참조하세요.

> **Note**
>
> - 계약이 만료된 후에는 고객이 새 AMI 인스턴스를 시작할 수 없습니다. 하지만 계약 기간에는 인스턴스를 원하는 만큼 시작할 수 있습니다. 이러한 라이선스는 노드 잠금되거나 특정 인스턴스에 묶여 있지 않습니다.
> - 비공개 제안 생성 - 판매자는 AWS Marketplace Management Portal의 비공개 제안 생성 도구를 사용하여 제품에 대한 비공개 제안을 생성할 수 있습니다.
> - 보고 - AWS Marketplace Management Portal의 보고서 섹션에서 Amazon S3 버킷을 설정하여 데이터 피드를 설정할 수 있습니다. 자세한 정보는 의 판매자 보고서, 데이터 피드 및 대시보드 AWS Marketplace 섹션을 참조하세요.

주제

- 라이선스 모델
- 통합 워크플로
- License Manager 통합 사전 조건
- AMI 기반 제품을와 통합 AWS License Manager
- 라이선스 갱신 및 업그레이드

라이선스 모델

AWS Marketplace 와의 통합은 두 가지 라이선스 모델을 AWS License Manager 지원합니다.

- 구성 가능한 라이선스 모델
- 계층형 라이선스 모델

구성 가능한 라이선스 모델

구성 가능한 라이선스 모델(정량화 가능한 라이선스 모델이라고도 함)은 구매자가 라이선스를 조달한 후 구매자에게 특정 수량의 리소스를 부여합니다.

판매자는 요금 차원과 단가를 설정합니다. 그러면 구매자는 구매하려는 리소스 수량을 선택할 수 있습니다.

Example 요금 차원 및 단가 예시

판매자는 요금 차원(예: 데이터 백업) 및 단가(예: 30 USD/단위)를 설정할 수 있습니다.

구매자는 5, 10 또는 20단위를 구매하기로 선택할 수 있습니다.

판매자의 제품은 사용량을 추적하고 측정하여 소비된 리소스 양을 측정합니다.

구성 모델의 경우 다음 두 가지 방법 중 하나로 권한이 계산됩니다.

- 드로다운 라이선스
- 플로팅 라이선스

드로다운 라이선스

라이선스를 사용한 만큼 허용된 라이선스 수량 풀에서 차감됩니다. 해당 권한은 영구적으로 체크아웃되며 라이선스 풀로 반환할 수 없습니다.

Example 제한된 양의 데이터 처리 예시

한 사용자에게 500GB의 데이터를 처리할 권한이 있습니다. 사용자가 데이터를 계속 처리하면 500GB 라이센스가 모두 소진될 때까지 500GB 풀에서 수량이 차감됩니다.

드로다운 라이선스의 경우 `CheckoutLicense` API 작업을 사용하여 소비된 라이선스 단위를 체크아웃할 수 있습니다.

Example 단위 수/년 동안 Amazon S3로 백업

1년 동안 데이터에 대해 최대 1,024단위까지 Amazon Simple Storage Service에 백업할 수 있는 스토리지 제품이 있습니다. Amazon EC2 인스턴스를 여러 개 사용하여 애플리케이션을 시작할 수 있습니다. 애플리케이션에는 데이터를 추적하고 집계하는 메커니즘이 있습니다. 소프트웨어는 백업이 발생할 때마다 또는 고정된 간격으로 제품 ID로 `CheckoutLicense` API 작업을 호출하여 소비량을 업데이트합니다.

이 예시에서는 소프트웨어가 `CheckoutLicense`를 호출하여 데이터 10단위를 체크아웃합니다. 총 용량이 고객이 구매한 백업 한도에 도달하면 API 직접 호출이 실패합니다.

요청

```
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"
```

응답

```
{
  "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"
}
```

플로팅 라이선스

라이선스가 사용 후 허용된 라이선스 수량 풀로 반환됩니다.

Example 고정된 상한의 사용자 수 예시

사용자는 애플리케이션에서 동시 사용자 500명을 이용할 수 있습니다. 사용자가 로그인하고 로그아웃하면 사용자가 차감된 후 사용자 500명 풀로 반환됩니다. 하지만 고정된 상한은 동시 사용자 500명이기 때문에 애플리케이션이 풀에서 500명을 초과하는 사용자를 차감할 수 없습니다.

플로팅 라이선스의 경우 `CheckInLicense` API 작업을 사용하여 라이선스 단위를 권한 풀에 반환할 수 있습니다.

Example 1년간 동시 사용자 수 예시

제품 요금은 동시 사용자 수를 기준으로 책정됩니다. 고객이 1년간 사용자 10명 라이선스를 구매합니다. 고객이 AWS Identity and Access Management (IAM) 권한을 제공하여 소프트웨어를 시작합니다. 사용자가 로그인하면 애플리케이션이 `CheckoutLicense` API 작업을 호출하여 수량을 1만큼 줄입니다. 사용자가 로그아웃하면 애플리케이션이 `CheckInLicense` API 작업을 호출하여 해당 라이선스를 풀에 반환합니다. `CheckInLicense`를 호출하지 않으면 1시간 후에 라이선스 단위가 자동으로 체크인됩니다.

> **Note**
>
> 다음 요청에서 `key-fingerprint`는 자리 표시자 값이 아니라 모든 라이선스를 게시하는 데 사용되는 fingerprint의 실제 값입니다.

요청

```
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"
```

응답

```
{
  "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"
}
```

계층형 라이선스 모델

계층형 라이선스 모델은 구매자가 라이선스를 조달한 후 특정 수준 또는 계층의 애플리케이션 기능을 이용할 수 있는 권한을 구매자에게 부여합니다.

판매자가 제품 계층(예: 기본, 중급, 프리미엄)을 생성합니다. 그 후 구매자가 사전 정의된 계층 중 하나를 선택합니다.

애플리케이션은 애플리케이션 사용량을 추적하거나 측정할 필요가 없습니다.

계층형 라이선스 모델의 경우 권한은 계산되지 않고, 대신 고객이 조달한 서비스 계층을 나타냅니다.

기능을 번들로 제공하려는 경우 계층형 라이선스 모델을 사용하는 것이 좋습니다.

Example 기본, 중급 및 프리미엄 계층의 예시

고객은 세 가지 소프트웨어 티어인 기본, 중급 또는 프리미엄 중 하나를 계약할 수 있습니다. 티어마다 자체 요금이 책정되어 있습니다. 소프트웨어는 `CheckoutLicense` API 작업을 간접적으로 호출하고 가능한 모든 등급을 요청에서 지정하여 고객이 가입한 등급을 식별할 수 있습니다.

요청 응답에는 고객이 구매한 계층에 해당하는 권한이 포함됩니다. 소프트웨어는 이 정보를 기반으로 적절한 고객 경험을 프로비저닝할 수 있습니다.

요청

```
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"
```

응답

```
{
  "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"
}
```

통합 워크플로

다음 단계는 AMI 제품을 AWS License Manager와 통합하는 워크플로를 보여줍니다.

1. 판매자는 AWS License Manager 통합이 포함된 제품을 생성합니다.

2. 판매자가 제품을 나열합니다 AWS Marketplace.
3. 구매자가에서 제품을 찾아 AWS Marketplace 구매합니다.
4. 라이선스가 구매자의 AWS 계정에 전송됩니다.
5. 구매자가 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, Amazon Elastic Container Service(Amazon ECS) 작업 또는 Amazon Elastic Kubernetes Service(Amazon EKS) 포드 소프트웨어를 시작하여 소프트웨어를 사용하고, 고객은 IAM 역할을 사용하여 배포합니다.
6. 소프트웨어는 구매자 AWS License Manager 계정의 라이선스를 읽고, 구매한 권한을 검색하고, 그에 따라 기능을 프로비저닝합니다.

> **Note**
>
> License Manager는 추적 또는 업데이트를 수행하지 않습니다. 추적 또는 업데이트는 판매자의 애플리케이션에서 수행됩니다.

License Manager 통합 사전 조건

제품을 게시하기 전에 다음을 수행해야 합니다.

1. 에서 새 AMI 제품을 AWS Marketplace Management Portal생성하고 해당 제품 코드를 기록해 둡니다.
2. 필요한 가격 정보와 함께 제품 로드 양식(PLF)을 작성하고, AWS에 반환하여 처리합니다.
3. `CheckoutLicense`, `ExtendLicenseConsumption` 및 `CheckInLicense`를 호출하는 데 필요한 IAM 권한이 있는 애플리케이션을 실행하는 작업 또는 포드에 IAM 역할을 사용합니다.

   필요한 IAM 권한은 다음 IAM 정책에 자세히 설명되어 있습니다.

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"VisualEditor0",
         "Effect":"Allow",
         "Action":[
            "license-manager:CheckoutLicense",
            "license-manager:GetLicense",
```

```
                "license-manager:CheckInLicense",
                "license-manager:ExtendLicenseConsumption",
                "license-manager:ListReceivedLicenses"
            ],
            "Resource":"*"
        }
    ]
}
```

4. 정의하는 모든 요금 차원에 대한 레코드를 사용하여 `RegisterUsage` API 작업 호출을 테스트합니다.

AMI 기반 제품을와 통합 AWS License Manager

AWS License Manager API를 사용하여 AMI 기반 제품을 License Manager와 통합할 수 있습니다. AWS Marketplace AMI 기반 제품을 사용하여 Amazon EC2 인스턴스를 시작합니다.

> **Note**
>
> 다음 절차를 수행하려면 먼저 the section called “License Manager 통합 사전 조건”을 완료해야 합니다.

AMI 기반 제품을 License Manager와 통합하는 방법

1. the section called “License Manager에서 테스트 라이선스 생성”의 절차를 완료합니다. License Manager에서 통합을 테스트하는 데 사용할 테스트 라이선스를 생성해야 합니다.
2. 1단계에서 얻은 라이선스 Amazon 리소스 이름(ARN)을 사용하여 GetLicense API 작업을 실행합니다. 나중에 사용할 수 있도록 `GetLicense` 응답의 `KeyFingerprint` 속성 값을 기록해 둡니다.
3. 최신 퍼블릭 AWS SDK를 다운로드하여 애플리케이션에 포함합니다.
4. 구매자가 애플리케이션용 라이선스를 사용할 자격이 있는지 확인하려면 CheckoutLicense API 작업을 실행합니다. 1단계에서 얻은 테스트 라이선스의 자격 세부 정보와 key fingerprint를 사용합니다.

   라이선스에 대한 권한이 없거나 최대 권한 수를 초과한 경우 `CheckoutLicense` API 작업이 `NoEntitlementsAllowedException`을 반환합니다. 권한이 유효하거나 사용 가능한 경우 `CheckoutLicense` 작업은 요청된 권한 및 해당 값이 포함된 성공 응답을 반환합니다.

5. (플로팅 권한에만 필요) `CheckoutLicense` 응답에서 받은 `LicenseConsumptionToken`을 사용하여 CheckInLicense API 작업을 실행합니다. 이 작업은 이전에 체크아웃한 권한을 사용 가능한 권한 풀로 해제합니다.
6. 1단계에서 생성한 테스트 라이선스로 License Manager 통합을 성공적으로 확인한 후에는 코드의 key fingerprint를 `aws:294406891311:AWS/Marketplace:issuer-fingerprint`로 업데이트합니다. 이제 AWS Marketplace에서 발급한 라이선스로 작업할 수 있습니다.

AMI 제품에 애플리케이션을 빌드하는 릴리스 프로세스를 수행한 다음 제품 게시 프로세스에 AWS Marketplace 따라에 제품을 제출합니다.

License Manager에서 테스트 라이선스 생성

AWS Command Line Interface (AWS CLI) 버전 2를 사용하여에서 테스트 라이선스를 생성합니다 AWS License Manager. 이 테스트 라이선스는 AWS License Manager 통합 확인 및 테스트에만 사용됩니다. 테스트가 완료된 후에는 테스트 라이선스를 삭제해도 됩니다. 실제 라이선스는 AWS Marketplace 에서 다른 key fingerprint를 사용하여 생성됩니다.

AWS Marketplace 는에서 두 가지 유형의 권한을 지원합니다 AWS License Manager. 하지만 한 제품에 대해 한 가지 유형만 활성화할 수 있습니다. 테스트 라이선스를 포함하여 라이선스를 생성할 때 다음 자격 유형 중 하나를 지정해야 합니다.

계층형 권한 - 계층형 라이선스 모델은 고객에게 특정 애플리케이션 기능에 대한 권한을 부여합니다. 고객은 구매하려는 단위의 수량을 정의할 수 없습니다. 하지만 사전 정의된 패키지 또는 계층을 하나 선택할 수 있습니다. 고객은 나중에 계약을 수정하여 다른 계층을 구독할 수 있습니다.

구성 가능한 권한 - 구성 가능한 라이선스 모델은 고객이 라이선스를 구매할 때 특정 수량의 리소스에 대한 권한을 부여합니다. 고객은 구독 프로세스에서 구매하려는 수량 단위를 선택할 수 있으며 단가를 기준으로 요금이 청구됩니다. 고객은 여러 차원을 구독할 수도 있습니다.

`CheckoutLicense` API 작업에 꼭 사용해야 하는 필수 파라미터는 다음과 같습니다.

- `CheckoutType` - 유효한 값은 `Perpetual` 또는 `Provisional`입니다.
  - `Perpetual` - 체크아웃된 권한 수량이 풀에서 모두 소진될 때 사용됩니다. 예: 구매자에게 500GB의 데이터를 처리할 권한이 있습니다. 구매자가 데이터를 계속 처리하다 보면 수량이 감소하고 500GB 풀이 모두 소진됩니다. 구매한 라이선스의 만료 또는 만료 임박 여부에 대한 상태를 가져와서 고객에게 알림을 보냅니다.
  - `Provisional` - 풀에서 권한을 체크아웃하고 사용 후 다시 반환하는 플로팅 라이선스 권한에 사용됩니다. 예: 사용자는 애플리케이션에서 동시 사용자 500명을 이용할 수 있습니다. 사용자가 로

그인하고 로그아웃하면 사용자가 차감된 후 사용자 500명 풀로 반환됩니다. 유동 라이선스 권한에 대한 자세한 내용은 AWS License Manager에서 판매자가 발급한 라이선스를 참조하세요.

- `ClientToken` - 대소문자를 구분하는 고유의 식별자로, 몇 번을 시도해도 항상 동일한 결과가 나옵니다. 각 요청에 임의의 범용 고유 식별자(UUID)를 사용하는 것이 좋습니다.
- `Entitlements` - 체크아웃할 권한 목록입니다.
  - 계층형 권한의 경우 다음과 같이 `Name` 및 `Unit` 속성을 제공합니다.

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

  - 계층형 권한의 경우 다음과 같이 `Name`, `Unit` 및 `Value` 속성을 제공합니다.

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

- `KeyFingerprint` - 이 key fingerprint를 사용하여 라이선스를 AWS Marketplace에서 발급했는지 확인합니다. 에서 발급한 라이선스의 키 지문 AWS Marketplace 은 다음과 같습니다.

  `aws:294406891311:AWS/Marketplace:issuer-fingerprint`

- `Product SKU` - 제품과 연결된 GUID(Globally Unique Identifier) 형식의 AWS Marketplace 제품 ID 입니다.

Example 구성 가능한 권한의 예시

다음은 `CheckoutLicense` API 작업을 사용하여 `PowerUsers`라는 구성 가능한 권한을 체크아웃하는 요청의 예입니다.

```
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 계층형 자격의 예시

다음은 `CheckoutLicense` API 작업을 사용하여 `EnterpriseEdition`라는 기능 권한을 체크아웃하는 요청의 예입니다.

```
aws license-manager checkout-license \
   --product-sku "2205b290-19e6-4c76-9eea-377d6bf71a47" \
   --checkout-type "PROVISIONAL" \
   --client-token "79464194dca9429698cc774587a603a1" \
   --entitlements "Name=EnterpriseEdition,Unit=None" \
   --key-fingerprint "aws:294406891311:AWS/Marketplace:issuer-fingerprint"
```

AMI 기반 제품에 대한 테스트 라이선스를 생성하는 방법

1. AWS CLI v2가 설치된 로컬 환경에서 다음 스크립트를 실행합니다. 이 스크립트는 테스트 라이선스를 생성하고 적절한 제품 세부 정보를 구성합니다.

   > **Note**
   >
   > 소프트웨어를 배포하고 테스트 AWS 계정 하는 테스트 AWS 계정 와 다른를 사용합니다. 동일한 AWS 계정에서 라이선스를 생성하고, 부여하고, 체크아웃할 수 없습니다.

   ```
   #!/bin/bash

   # Replace with intended product ID on AWS Marketplace
   PRODUCT_ID=<REPLACE-WITH-PRODUCT-ID>

   # Replace with license recipient's AWS Account ID
   BENEFICIARY_ACCOUNT_ID=<REPLACE-WITH-BENEFICIARY-ACCOUNT-ID>

   # Replace with your product's name
   PRODUCT_NAME="Test Product"

   # Replace with your seller name on AWS Marketplace
   ```

```
SELLER_OF_RECORD="Test Seller"

# Replace with intended license name
LICENSE_NAME="AWSMP Test License"

# Replace the following with desired contract dimensions
# More info here: https://docs.aws.amazon.com/license-manager/latest/APIReference/
API_Entitlement.html
# Example "configurable entitlement"
ENTITLEMENTS='[
  {
    "Name": "ReadOnly",
    "MaxCount": 5,
    "Overage": false,
    "Unit": "Count",
    "AllowCheckIn": true
  }
]'
# Example "tiered entitlement"
# ENTITLEMENTS='[
#   {
#     "Name": "EnterpriseUsage",
#     "Value": "Enabled",
#     "Unit": "None"
#   }
# ]'

# Format "yyyy-mm-ddTHH:mm:ss.SSSZ"
# This creates a validity period of 10 days starting the current day
# Can be updated to desired dates
VALIDITY_START=$(date +%Y-%m-%dT%H:%M:%S.%SZ)
VALIDITY_END=$(date --date="+10 days" +%Y-%m-%dT%H:%M:%S.%SZ)

# Configuration for consumption of the license as set on Marketplace products
CONSUMPTION_CONFIG='{
 "RenewType": "None",
 "ProvisionalConfiguration": {
   "MaxTimeToLiveInMinutes": 60
 }
}'

# License's home Region
HOME_REGION=us-east-1
```

```
# License issuer's name
ISSUER=Self

# Run AWS CLI command to create a license
aws license-manager create-license \
  --license-name "${LICENSE_NAME}" \
  --product-name "${PRODUCT_NAME}" \
  --product-sku "${PRODUCT_ID}" \
  --issuer Name="${ISSUER}" \
  --beneficiary "${BENEFICIARY_ACCOUNT_ID}" \
  --validity 'Begin="'"${VALIDITY_START}"'",End="'"${VALIDITY_END}"'"' \
  --entitlements "${ENTITLEMENTS}" \
  --home-region "${HOME_REGION}" \
  --region "${HOME_REGION}" \
  --consumption-configuration "${CONSUMPTION_CONFIG}" \
  --client-token $(uuidgen)
```

2. AWS License Manager 콘솔을 사용하여 라이선스를 부여합니다. 자세한 내용은 License Manager 사용 설명서의 권한 배포를 참조하세요.
3. 소프트웨어를 배포하고 테스트할 구매자 계정 역할을 AWS 계정 하는에 로그인합니다. 이는 라이선스를 생성하고 부여 AWS 계정 한와 AWS 계정 달라야 합니다.
4. AWS License Manager 콘솔로 이동하여 부여된 라이선스를 수락하고 활성화합니다. 자세한 내용은 License Manager 사용 설명서의 부여된 라이선스 관리를 참조하세요.
5. 현재 환경에서 다음 명령을 실행합니다.

```
# 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)
```

이전 명령은 `PROVISIONAL`을 `CheckoutType` 파라미터의 값으로 사용합니다. 권한에 드로다운 라이선스가 사용되는 경우 `PERPETUAL`을 값으로 사용합니다.

License Manager API 직접 호출

고객의 License Manager 계정에 저장된 라이선스를 관리하려면 소프트웨어에서 다음 API 직접 호출을 사용하면 됩니다.

- `GetLicense` - 구매한 라이선스의 만료 또는 만료 임박 여부에 대한 상태를 가져와서 고객에게 알림을 보냅니다.
- `CheckoutLicense` - 사용자가 구매한 라이선스를 검색합니다. 사용자가 라이선스를 어느 정도 사용한 경우 이 호출을 사용하여 라이선스 수량을 업데이트할 수도 있습니다. `CheckoutLicense`를 사용하면 고객이 사용한 라이선스의 수량을 계속 체크아웃할 수 있습니다. 고객이 라이선스를 모두 소진하면 이 호출에서 오류가 반환됩니다. `CheckoutLicense` 실행에 권장되는 케이던스에 대한 자세한 내용은 the section called “라이선스 갱신 및 업그레이드” 섹션을 참조하세요.
- `ExtendLicenseConsumption` - 플로팅 차원의 경우 소프트웨어는 라이선스를 체크아웃한 후 60분 후에 자동으로 라이선스를 풀에 반환합니다. 라이선스가 체크아웃된 상태로 유지되는 시간을 연장하려면 소프트웨어에서 `ExtendLicenseConsumption`을 호출하여 라이선스를 60분 더 연장할 수 있습니다.
- `CheckInLicense` - 플로팅 차원의 경우 라이선스를 권한 풀에 반환하려면 `CheckInLicense`를 사용합니다.
- `ListReceivedLicenses` - 구매자가 구매한 라이선스를 나열합니다.

라이선스 갱신 및 업그레이드

고객은 AWS Marketplace Management Portal에서 라이선스를 갱신하거나 업그레이드할 수 있습니다. 추가 구매를 한 후는 새 권한을 반영하는 라이선스의 새 버전을 AWS Marketplace 생성합니다. 소프트웨어는 동일한 API 직접 호출을 사용하여 새 권한을 읽습니다. 갱신 및 업그레이드를 처리하기 위해 License Manager 통합과 관련하여 아무 것도 할 필요가 없습니다.

라이선스 갱신, 업그레이드, 취소 등으로 인해 제품이 사용되는 동안 제품에서 정기적으로 `CheckoutLicense` API 직접 호출을 수행하는 것이 좋습니다. `CheckoutLicense` API 작업을 정기적으로 사용하면 제품에서 업그레이드 및 만료와 같은 권한 변경을 감지할 수 있습니다.

15분마다 `CheckoutLicense` API 직접 호출을 수행하는 것이 좋습니다.

에서 AMI 제품에 대한 Amazon SNS 알림 수신 AWS Marketplace

제품의 고객 구독 변경 사항에 대한 알림을 받으려면 제품 생성 중에 AWS Marketplace 제공되는에 대한 Amazon Simple Notification Service(Amazon SNS) 주제를 구독하면 됩니다. 예를 들어 고객이 비

공개 제안을 수락하는 시점을 알 수 있습니다. Amazon SNS 주제: aws-mp-subscription-notification은 AMI 제품에 사용할 수 있는 Amazon SNS 주제입니다. 이 주제는 구매자가 제품을 구독하거나 구독 해지할 때 알려줍니다. 이 알림은 시간당 및 연간 시간당을 포함한 시간당 요금 모델에 사용할 수 있습니다. 자세한 내용은 다음 섹션을 참조하세요.

> **Note**
>
> 제품 생성 프로세스 중에 제품에 대한 Amazon SNS 주제가 만들어집니다. 알림을 구독하려면 Amazon SNS 주제(예: `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`)의 Amazon 리소스 이름(ARN)이 필요합니다. 서버 제품의 판매자 포털에서는 ARN을 사용할 수 없습니다. AWS Marketplace 운영 팀에 문의하여 ARN을 요청합니다.

주제

- Amazon SNS 주제: aws-mp-subscription-notification
- Amazon SQS 대기열을 Amazon SNS 주제에 구독

Amazon SNS 주제: aws-mp-subscription-notification

`subscribe-success` 및 `subscribe-fail` 작업에 대한 `aws-mp-subscription-notification` 주제의 각 메시지는 다음과 같은 형식입니다.

```
{
    "action": "action-name",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123"
}
```

*<action-name>*은 알림에 따라 달라집니다. 가능한 작업은 다음과 같습니다.

- `subscribe-success`
- `subscribe-fail`
- `unsubscribe-pending`
- `unsubscribe-success`

`offer-identifier`는 작업이 `subscribe-success` 또는 인 경우에만 알림에 포함됩니다`subscribe-fail`. 작업이 `unsubscribe-pending` 또는 일 때는 알림에 포함되지 않습니다`unsubscribe-success`. 2024년 1월 이전에 생성된 제안의 경우이 식별자는 비공개 제안에 대한 알림에만 포함됩니다. 2024년 1월 이후에 생성된 제안의 경우이 식별자는 비공개 제안 및 공개 제안을 포함한 모든 제안에 대한 알림에 포함됩니다.

제안 유형에 대한 자세한 내용은 계약 갱신 대시보드에서 DescribeEntity API의 응답 또는 계약의 제안 가시성을 참조하세요. https://docs.aws.amazon.com/marketplace/latest/userguide/agreements-renewals-dashboard.html

> **Note**
>
> DescribeEntity API의 경우 해당 제안에 대한 대상 지정 규칙의 패싯을 대상으로 하는를 계정 AWS 계정 에서 찾으면 비공개 제안입니다. 해당 제안 AWS 계정 에 대한 대상 지정 규칙의 패싯을 대상으로 하는가 계정에 없는 경우 공개 제안입니다.

Amazon SQS 대기열을 Amazon SNS 주제에 구독

Amazon SQS 대기열을 제공된 SNS 주제에 구독하는 것이 좋습니다. SQS 대기열을 생성하고 대기열을 주제에 구독하는 방법에 대한 자세한 지침은 Amazon Simple Notification Service 개발자 안내서의 Amazon SQS 대기열을 Amazon SNS 주제에 구독을 참조하세요.

> **Note**
>
> 제품을 판매하는 데 AWS 계정 사용되는 에서만 AWS Marketplace SNS 주제를 구독할 수 있습니다. 하지만 메시지를 다른 계정에 전달할 수 있습니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 다른 계정의 Amazon SQS 대기열로 Amazon SNS 메시지 전송을 참조하세요.

알림을 위한 SQS 대기열 폴링

SQS 대기열을 SNS 주제에 구독하면 메시지가 SQS에 저장됩니다. 대기열을 지속적으로 폴링하는 서비스를 정의하고, 메시지를 찾고, 그에 따라 메시지를 처리해야 합니다.

용 AMI 제품 체크리스트 AWS Marketplace

Amazon Machine Image(AMI) 제품을 제출하기 전에이 체크리스트를 AWS사용하여 제출을 검증합니다. 이러한 요구 사항을 따르면 효율적인 게시 프로세스를 보장하는 데 도움이 됩니다.

제품 사용

- AMI는 프로덕션에 사용할 수 있습니다.
- 제품 사용량은 시간 또는 기타 측정값으로 제한되지 않습니다.
- AMI는 원클릭 이행 환경과 호환됩니다.
- 제품을 활용하는 데 필요한 모든 것이 고객 애플리케이션을 비롯하여 소프트웨어에 있습니다. 소프트웨어 패키지 또는 클라이언트 애플리케이션과 같은 외부 종속성이 필요한 제품은 적절한 공개제품 사용 정책가 포함된를 따릅니다.
- AMI는 필수를 충족합니다에 대한 AMI 기반 제품 요구 사항 AWS Marketplace.
- 제품을 사용하기 위해 추가 라이센스가 필요하지 않음.
- 구매자는 제품을 사용하기 위해 개인 식별 정보(예: 이메일 주소)를 제공할 필요가 없습니다.

AMI 준비

- 제품 이름과 설명은 제공하려는 AMI 제품의 설명 필드와 일치해야 합니다.
- 하드웨어 가상 머신(HVM) 가상화 및 64비트 아키텍처를 사용합니다.
- 제품에 알려진 취약성, 맬웨어 또는 바이러스가 포함되어 있지 않습니다.
- 고객은 네트워크를 통해 인스턴스에 액세스하고 관리 액세스를 사용할 수 있습니다.
- AMI에는 하드 코딩된 보안 암호가 없습니다. 하드 코딩된 보안 암호의 예로는 시스템 사용자 및 서비스의 기본 암호, 프라이빗 키 및 자격 증명이 있습니다.
- AMI에는 하드 코딩된 SSH 인증 퍼블릭 키가 없습니다.
- 테스트 '버전 추가' 검증 테스트가 문제 없이 성공적으로 완료되었습니다.

Windows AMI

- Windows Server 2012 이상 운영 체제의 경우 AMI는 최신 버전의 EC2Launch v2를 사용합니다.
- EC2Launch v2를 사용하는 경우 다음을 완료하세요.

  - Amazon EC2Launch 설정의 관리자 계정 설정에서 임의를 선택하여 런타임 시 관리자 암호를 생성합니다.
  - Amazon EC2Launch 설정에서 다시 활성화를 선택하고 Sysprep 후 SSM 서비스를 시작합니다.
  - EC2 v2 작업 구성에 UserData를 추가합니다.
- Windows Server 2012 이상 버전의 경우 EC2Config를 사용하지 마십시오. EC2Config가 필요한 경우 최신 버전을 사용해야 합니다.
- EC2Config를 사용하는 경우 AMI의 설정 파일에서 다음 파라미터를 활성화합니다.
  - `Ec2SetPassword`
  - `Ec2WindowsActivate`
  - `Ec2HandleUserData`
- 게스트 계정이나 원격 데스크톱 사용자가 없어야 합니다.

Linux AMI

- 수퍼유저로 원격 로그인은 금지됩니다.
- 암호 기반 원격 액세스는 금지됩니다.

제품 로드 양식 또는 제품 탭

- 모든 필수 필드가 완료되었습니다.
- 모든 값이 지정된 문자 제한 내에 있습니다.
- 모든 URL이 오류 없이 로드됩니다.
- 제품 이미지는 최소 110픽셀 너비/1:1 ~ 2:1 비율입니다.
- 활성화된 모든 인스턴스 유형의 요금(시간별, 시간 기반 월별 요금 및 시간 기반 연별 요금 모델)이 지정되었습니다.
- 월별 요금(시간 기반 월별 및 월별 요금 모델)이 지정되었습니다.

에 대한 AMI 기반 제품 요구 사항 AWS Marketplace

AWS Marketplace 는 모든 Amazon Machine Image(AMI) 제품 및 제품에 대해 다음 정책을 유지합니다. 이 섹션의 정책은 고객에게 안전하고 신뢰할 수 있는 컴퓨팅 플랫폼을 제공하기 위한 것입니다.

제출 시 모든 제품 및 관련 메타데이터를 검토하여 현재 AWS Marketplace 정책을 충족하거나 초과하는지 확인합니다. 이러한 정책은 진화하는 보안 지침에 따라 정기적으로 업데이트됩니다.는 제품을 AWS Marketplace 지속적으로 스캔하여 기존 목록이 이러한 요구 사항에 대한 변경 사항을 계속 충족하는지 확인합니다. 제품이 규정을 준수하지 않는 경우 AWS Marketplace 는 판매자에게 문의하여 새 표준을 충족하도록 제품을 업데이트합니다. 경우에 따라 문제가 해결될 때까지 새 구독자가 제품을 일시적으로 사용할 수 없게 될 수 있습니다. 이 프로세스는 모든 사용자에 대해 AWS Marketplace 플랫폼의 보안 및 신뢰성을 유지하는 데 도움이 됩니다.

제품을 제출하기 전에의 '버전 추가' 테스트 기능을 사용하여 현재 정책을 준수하는 AWS Marketplace Management Portal 것이 좋습니다.

주제

- AMI 제품 판매자 정책
- 보안 정책
- 아키텍처 정책
- AMI 제품 사용 지침
- AMI 제품 버전 정책
- 고객 정보 정책
- 제품 사용 정책

AMI 제품 판매자 정책

모든 AMIs 다음 판매자 정책을 준수해야 합니다.

- 기본적으로 AWS Marketplace 판매자는 최대 75개의 퍼블릭 AMI 제품 목록으로 제한됩니다. 한도를 초과하는 모든 판매자는 정기적인 성과 검토를 받아야 하며 성능이 저조한 리스팅을 제한해야 할 수 있습니다.는 단독 재량에 따라이 한도에 대한 증가를 부여하고 취소할 AWS Marketplace 수 있습니다.

보안 정책

일반 정책

모든 AMIs 다음 정책을 준수해야 합니다.

- AMIs 알려진 취약성이나 맬웨어가 없는 AWS Marketplace AMI 스캔 도구에서 수행하는 모든 보안 검사를 통과해야 합니다.
- AMIs 현재 지원되는 운영 체제 및 소프트웨어를 사용해야 합니다. 수명이 다한 운영 체제 및 소프트웨어는 허용되지 않습니다.
- 인스턴스 서비스에 대한 암호 기반 인증은 금지됩니다. 이는 시작 시 사용자가 암호를 생성, 재설정 또는 정의한 경우에도 적용됩니다. Null 암호와 빈 암호는 허용되지 않습니다.

  예외:

  - Windows 인스턴스`EC2Config`/`EC2Launch`에서가 생성한 관리자 암호입니다.
  - 다른 인증 방법이 없는 경우 호스트 서비스(예: 웹 애플리케이션)에 대한 비관리자 액세스. 강력한 암호를 사용하는 경우 각 인스턴스에 대해 무작위로 생성하고, 서비스 관리자가 초기 인증을 위해 한 번 사용하고, 처음 로그인한 후 즉시 변경해야 합니다.
- AMI에는 시스템 사용자 및 서비스 암호(해시된 암호 포함), 프라이빗 키 또는 자격 증명과 같은 하드코딩된 보안 암호가 포함되어서는 안 됩니다.
- AMIs AWS 서비스에 액세스하기 위해 AWS 자격 증명을 요청해서는 안 됩니다. 제품에 AWS 서비스에 대한 액세스가 필요한 경우 인스턴스에 최소 권한 AWS Identity and Access Management (IAM) 역할이 할당되어야 합니다. 사용자는 수동으로 또는 AWS CloudFormation 템플릿을 사용하여 역할을 생성할 수 있습니다. CloudFormation 제공 방법을 사용하는 제품에 대해 단일 AMI 시작이 활성화된 경우 사용 지침에 최소 권한의 IAM 역할을 생성하기 위한 명확한 지침이 포함되어야 합니다. 자세한 내용은AWS CloudFormation을 사용하여 AMI 기반 제품 제공을 참조하세요.
- 판매자는 고객이 실행하는 인스턴스에 액세스할 수 없어야 합니다. 지원 또는 기타 목적으로 이러한 액세스가 필요한 경우 고객에게 명시적으로 활성화하도록 지시할 수 있습니다.

SSH(Secure Shell) 액세스 정책

SSH(Secure Shell) 액세스를 제공하는 AMIs 일반 정책 외에도 다음 보안 정책을 준수해야 합니다.

- AMIs SSH를 사용한 암호 기반 인증을 허용해서는 안 됩니다. 이렇게 하려면 `sshd_config` 파일에서를 `PasswordAuthentication`로 설정합니다no.
- AMIs 수퍼유저 계정의 암호 기반 원격 로그인을 비활성화해야 합니다. 자세한 내용은 루트 사용자의 암호 기반 원격 로그인 비활성화를 참조하세요.
- AMIs SSH 액세스를 위한 승인된 퍼블릭 키가 포함되어서는 안 됩니다.
- AMIs SSH는 AWS Marketplace 내부 심사 절차에서 액세스할 수 있어야 합니다.
  - SSH 서비스는 AMI 스캔을 위해 지정된 TCP 포트에서 수신 대기해야 합니다. 자세한 내용은 새 버전 추가를 참조하세요.

- SSH는 인스턴스 시작 시 서브넷`10.0.0.0/16`과 Amazon Elastic Compute Cloud(Amazon EC2)에서 할당한 IP 주소`10.2.0.0/16`에서 액세스할 수 있어야 합니다.

Linux 및 기타 Unix 유사 운영 체제 기반 AMIs 정책

일반 정책 외에도 Linux 및 기타 Unix와 유사한 운영 체제를 기반으로 하는 AMIs는 다음 보안 정책을 준수해야 합니다.

- AMIs 사용자가 완전한 액세스 권한을 갖도록 허용해야 합니다(예: `sudo` 액세스 허용).

Windows 기반 AMIs에 대한 정책

Windows 기반 AMIs 일반 정책 외에도 다음 보안 정책을 준수해야 합니다.

- AMIs 게스트 계정이 포함되어서는 안 됩니다.
- 관리자 계정만 인스턴스에 대한 원격 데스크톱 액세스 권한을 부여받을 수 있습니다.
- Windows AMIs EC2Launch(또는 Windows 2016 및 이전 버전의 경우 EC2Config)에서 이러한 옵션을 활성화하여 관리자 암호를 생성해야 합니다.
  - `Ec2SetPassword`
  - `Ec2WindowsActivate`
  - `Ec2HandleUserData`
- 자동 심사에 AMIs 사용할 수 있어야 합니다. 다음 요구 사항 중 하나 이상을 구현해야 합니다.
  - (권장 옵션) SSM 에이전트가 설치되어 있고 관리 권한과 아웃바운드 네트워크 액세스 권한이 있습니다.
  - Windows 원격 관리(WinRM) 서비스가 활성화되고 TCP 포트에서 수신 5985대기하며, 인스턴스 시작 시 Amazon Elastic Compute Cloud(Amazon EC2)에서 할당한 서브넷 `10.0.0.0/16` 및 IP 주소`10.2.0.0/16`에서 액세스할 수 있습니다.
  - Microsoft Server Message Block(SMB) Protocol 및 Common Internet File System(CIFS) Protocol 서비스가 활성화되어 있고, TCP 포트를 수신 139 대기445하며, 인스턴스 시작 시 Amazon Elastic Compute Cloud(Amazon EC2)에서 할당한 서브넷 `10.0.0.0/16` 및 IP 주소`10.2.0.0/16`에서 액세스할 수 있습니다.

아키텍처 정책

모든 AMI는 다음 아키텍처 정책을 준수해야 합니다.

- 에 대한 소스 AMIs는 미국 동부(버지니아 북부) 리전에서 제공해야 AWS Marketplace 합니다.
- AMI는 HVM 가상화를 사용해야 합니다.
- AMIs x86-64 또는 64비트 ARM 아키텍처를 사용해야 합니다.
- AMI는 Amazon Elastic Block Store(Amazon EBS)에서 지원하는 AMI여야 합니다. Amazon Simple Storage Service에서 지원하는 AMIs 지원되지 않습니다.
- AMI는 암호화된 EBS 스냅샷을 사용하면 안 됩니다.
- AMI는 암호화된 파일 시스템을 사용하면 안 됩니다.
- AMIs 모든에서 실행 AWS 리전 되고 리전에 구애받지 않도록 빌드되어야 합니다. 리전에 따라 다르게 빌드된 AMI는 허용되지 않습니다.

AMI 제품 사용 지침

AMI 제품에 대한 사용 지침을 생성할 때에는 the section called “AMI 및 컨테이너 제품 사용 지침”의 단계와 지침을 따릅니다.

AMI 제품 버전 정책

AWS Marketplace 는 S-AMI, CloudFormation 템플릿이 포함된 AMI 및 컨테이너 제품을 사용하여 AWS 고객과 판매자의 버전 관리 환경을 자동화합니다. 자동 버전 아카이브를 사용하면 판매자가 2년 이상 제한한 모든 제품 버전이 자동으로 아카이브됩니다. 보관된 버전은 더 이상 신규 고객을 AWS Marketplace 위해에서 시작할 수 없지만 기존 사용자는 AMI ID를 지정하여 시작 템플릿 및 Amazon EC2 Auto Scaling 그룹을 통해 아카이브된 버전을 계속 사용할 수 있습니다. 지난 13개월 동안 새 인스턴스를 시작하는 데 사용되지 않은 아카이브된 버전은 모두 삭제됩니다. 아카이브된 버전이 삭제되면 신규 또는 기존 사용자에 대해 더 이상 시작할 수 없습니다.

고객 정보 정책

모든 AMI는 다음과 같은 고객 정보 정책을 준수해야 합니다.

- 소프트웨어는 기존 보유 라이선스 사용(BYOL)에서 요구하는 경우를 제외하고 고객의 인지와 분명한 동의 없이 고객 데이터를 수집하거나 내보내면 안 됩니다. 고객 데이터를 수집하거나 내보내는 애플리케이션은 다음 지침을 준수해야 합니다.
  - 고객 데이터 수집은 셀프 서비스이고, 자동화되고, 안전해야 합니다. 구매자는 판매자가 소프트웨어 배포를 승인할 때까지 기다릴 필요가 없어야 합니다.
  - 고객 데이터 수집은 Marketplace 이용 약관 AWS, AWS 서비스 약관, AWS 개인 정보 보호 고지 및 AWS 고객 계약을 포함하되 이에 국한되지 않는 와의 계약과 일치해야 합니다. AWS

- 결제 정보를 수집하면 안 됩니다.

제품 사용 정책

모든 AMI는 다음과 같은 제품 사용 정책을 준수해야 합니다.

- 제품은 시간, 사용자 수 또는 그 밖의 제한 사항에 따라 제품이나 제품 기능에 대한 액세스를 제한해서는 안 됩니다. 베타 및 출시 전 제품이나, 시험판 또는 평가 기능을 제공하는 것이 유일한 목적인 제품은 지원되지 않습니다. AWS Marketplace에서 동일한 유료 버전을 사용할 수 있는 경우 상용 소프트웨어의 개발자, 커뮤니티 및 BYOL 에디션이 지원됩니다.
- 모든 AMI는 웹 사이트에서 실행 환경 또는 AWS CloudFormation을 통한 AMI 기반 제공과 호환되어야 합니다. 웹 사이트에서 실행의 경우, AMI는 인스턴스를 생성할 때 올바르게 기능하기 위해 고객 또는 사용자 데이터를 요구할 수 없습니다.
- AMI와 해당 소프트웨어는 셀프 서비스 방식으로 배포할 수 있어야 하며 추가 결제 방법이나 비용이 필요 없어야 합니다. 배포 시 외부 종속성이 필요한 애플리케이션은 다음 지침을 준수해야 합니다.
  - 요구 사항은 목록의 설명 또는 사용 지침에 명시되어야 합니다. 예: 이 제품을 올바르게 배포하려면 인터넷 연결이 필요합니다. 배포 시 <패키지 목록> 패키지가 다운로드됩니다.
  - 판매자는 모든 외부 종속성의 사용과 가용성 및 보안에 대한 책임이 있습니다.
  - 외부 종속성을 더 이상 사용할 수 없는 경우 제품 AWS Marketplace 도에서 제거해야 합니다.
  - 외부 종속성은 추가 결제 방법이나 비용이 필요 없어야 합니다.
- 외부 APIs 또는 판매자나 제3자가 AWS 서비스 관리하는 등 구매자가 직접 제어하지 않는 외부 리소스에 지속적으로 연결해야 하는 AMIs는 다음 지침을 따라야 합니다.
  - 요구 사항은 목록의 설명 또는 사용 지침에 명시되어야 합니다. 예: 이 제품은 지속적인 인터넷 연결이 필요합니다. 정상적으로 작동하려면 <리소스 목록>과 같은 지속적인 외부 서비스가 필요합니다.
  - 판매자는 모든 외부 리소스의 사용과 가용성 및 보안에 대한 책임이 있습니다.
  - 외부 리소스를 더 이상 사용할 수 없는 경우 제품 AWS Marketplace 도에서 제거해야 합니다.
  - 외부 리소스는 추가 결제 방법이나 비용이 필요 없어야 하며 연결 설정이 자동화되어야 합니다.
- AWS Marketplace에서 사용할 수 없는 업셀 서비스, 추가 제품 또는 다른 클라우드 플랫폼으로 사용자를 유도하는 언어가 제품 소프트웨어 및 메타데이터에 포함되면 안 됩니다.
- 제품이 다른 제품이나 다른 ISV 제품의 추가 기능인 경우 제품 설명에 해당 제품은 다른 제품의 기능을 확장한다는 내용과 다른 제품이 없으면 해당 제품의 유용성이 매우 제한된다는 내용을 명시해야 합니다. 예: 이 제품은 <제품 이름>의 기능을 확장하며, <제품 이름> 제품이 없으면 이 제품의 유

용성이 매우 제한됩니다. 이 목록의 모든 기능을 사용하려면 <제품 이름>의 자체 라이선스가 필요할 수 있습니다.

의 EC2 Image Builder 구성 요소 기반 제품 AWS Marketplace

AWS Marketplace 판매자는 Image Builder 구성 요소를 사용하여 구매자에게 제품을 제공할 수 있습니다. Image Builder 구성 요소는 Image Builder 서비스에서 사용자 지정 EC2 Image Builder 구성 요소를 생성하고 유지 관리하는 데 사용되는 기본 구성 요소입니다. 구성 요소는 이미지 생성 프로세스 중에 수행할 특정 작업 세트를 정의하는 재사용 가능한 모듈식 구성 요소입니다.

구성 요소는 이미지를 만들기 전에 인스턴스를 사용자 지정(빌드 구성 요소)하거나 생성된 이미지에서 시작된 인스턴스를 테스트(테스트 구성 요소)하는 데 필요한 단계 순서를 정의합니다. Image Builder 구성 요소는 다음 세 가지 항목으로 구성될 수 있습니다.

- 구성 요소 문서, 구성 요소를 사용하여 이미지를 빌드할 때 실행할 단계와 작업을 지정하는 선언적 YAML 또는 JSON 파일입니다.
- 구성 요소에 대한 구성 옵션을 제공하는 구성 요소 속성입니다. 구성 요소의 특성, 동작 및 호환성을 정의하는 데 도움이 됩니다.
- 소프트웨어 패키지, 구성 요소가 올바르게 작동하기 위해 설치해야 하는 종속성입니다.

에 Image Builder 구성 요소 게시 AWS Marketplace 는 AWS Marketplace 카탈로그 API를 사용하여 지원됩니다. 카탈로그 API를 사용하여 구성 요소를 게시하는 지침은 API 참조의 EC2 Image Builder 구성 요소 제품 작업을 AWS Marketplace 참조하세요.

에서 EC2 Image Builder 제품에 대한 변경 요청 생성 AWS Marketplace

에서 제품 또는 버전을 변경하려면를 통해 변경 요청을 AWS Marketplace제출합니다 AWS Marketplace Management Portal. 변경 요청은 대기열에 추가되며 요청 유형에 따라 변경 요청이 해결될 때까지 몇 분에서 며칠까지 걸릴 수 있습니다. AWS Marketplace Management Portal에서 요청 상태를 확인할 수 있습니다. 이 주제에서는 셀프 서비스 경험을 사용하는 옵션을 AWS Marketplace포함하여에서 단일 제품에 대한 변경 요청을 생성하는 데 사용할 수 있는 절차를 제공합니다.

다음과 같은 상황에서 변경 요청을 만듭니다.

- 진행 중인 단계를 저장했지만 셀프 서비스 경험을 사용하여 단일 제품 목록을 생성하는 동안 전체 프로세스를 완료하지 못했습니다. 나머지 단계를 완료하려면 변경 요청을 생성합니다.

- 제한 또는 퍼블릭 상태인 제품의 제품 정보를 수정하려 합니다. 정보를 업데이트하려면 변경 요청을 생성합니다. EC2 Image Builder 제품에 요청할 수 있는 변경 유형에 대한 자세한 내용은 섹션을 참조하세요변경 요청 생성.

> **Note**
>
> 외에도 AWS Marketplace 카탈로그 API를 사용하여 변경 요청을 생성할 수도 AWS Marketplace Management Portal있습니다.

주제

- 셀프 서비스를 사용하여 변경 요청 생성
- 변경 요청 생성
- 변경 요청의 상태 가져오기
- 추가 리소스

셀프 서비스를 사용하여 변경 요청 생성

버전 또는 제품 정보를 수정하려면 AWS Marketplace Management Portal에서 변경 요청을 생성해야 합니다. 변경 요청은 제품을 변경할 때 사용하는 셀프 서비스 목록의 구성 요소입니다. 단계에서 저장 후 종료를 선택하거나 업데이트의 제출을 선택할 때마다 변경 요청을 하는 것입니다. 요청 탭에서 AWS Marketplace Management Portal 요청을 찾을 수 있습니다.

셀프 서비스를 사용하여 변경 요청을 생성하는 방법

1. AWS Marketplace Management Portal 에서를 열고 판매자 계정에 https://aws.amazon.com/marketplace/management/tour/로그인한 다음 서버 제품 페이지로 이동합니다.
2. 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 옵션을 선택합니다.
4. 변경 요청을 한 후에는 시스템에서 요청을 처리할 때까지 기다려야 하며, 이 기간에는 검토 중으로 표시됩니다. 요청이 완료되면 성공 또는 실패로 표시됩니다.
    - 요청이 제출되면 처리가 시작되고 검토 중, 변경 준비 중, 변경 사항 적용 중 상태를 거치게 됩니다.
    - 성공은 요청된 변경이 처리되었으며 변경 사항이 시스템에 반영되었다는 뜻입니다.

- 실패는 요청에 문제가 발생하여 변경 사항이 처리되지 않았다는 뜻입니다. 상태가 실패이면 요청을 선택하여 오류 수정 방법에 대한 권장 사항을 제공하는 오류 코드를 찾을 수 있습니다. 이 시점에 오류를 해결하고 변경 요청을 새로 생성할 수 있습니다. 프로세스를 더 빠르게 진행하려면 새 요청에 복사를 선택하여 실패한 요청의 세부 정보를 복사하면 됩니다. 그런 다음, 요청을 조정하고 다시 제출할 수 있습니다.

변경 요청 생성

> **Important**
>
> 2023년 6월 15일에 AWS Marketplace 는 다음 절차를 중단합니다. 2023년 6월 15일 이후에는 the section called “셀프 서비스를 사용하여 변경 요청 생성” 절차를 사용하세요.

버전 또는 제품 정보를 수정하려면 AWS Marketplace Management Portal에서 변경 요청을 생성해야 합니다.

변경 요청을 생성하려면

1. AWS Marketplace Management Portal 에서를 열고 판매자 계정에 https://aws.amazon.com/marketplace/management/tour/로그인한 다음 서버 제품 페이지로 이동합니다.
2. 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 옵션을 선택합니다.

   대부분의 변경 요청의 경우 사용자 인터페이스에서 양식을 작성하고 제출합니다.

변경 요청의 상태에 대한 자세한 내용은 the section called “변경 요청의 상태 가져오기” 섹션을 참조하세요. 변경 요청과 관련된 잠재적 문제에 대한 자세한 내용은 AWS Marketplace에서의 변경 요청과 관련하여 흔히 발생하는 오류 해결 섹션을 참조하세요.

변경 요청의 상태 가져오기

> **Important**
>
> 2023년 6월 15일에 AWS Marketplace 는 다음 절차를 중단합니다. 이 절차는 셀프 서비스 경험에 더 이상 필요하지 않습니다.

변경 요청을 제출한 후에는 AWS Marketplace Management Portal의 서버 제품 페이지에 있는 요청 탭에서 요청 상태를 볼 수 있습니다. 요청의 상태는 다음 중 하나입니다.

- 검토 중은 요청을 검토 중이라는 뜻입니다. 일부 요청은 AWS Marketplace 팀의 수동 검토가 필요하지만 대부분은 시스템에서 자동으로 검토됩니다.
- 성공은 요청이 완료되었다는 뜻입니다. 요청하신 대로 제품 또는 버전을 업데이트했습니다.
- 조치 필요는 요청을 업데이트하여 문제를 해결하거나 요청에 대한 질문에 대답해야 한다는 뜻입니다. 요청을 선택하면 문제를 포함한 세부 정보를 볼 수 있습니다.
- 실패는 요청에 문제가 있다는 뜻이므로 동일한 데이터로 변경 요청을 새로 생성해야 합니다.

추가 리소스

특정 업데이트 유형의 변경 요청에 대한 자세한 내용은 다음 리소스를 참조하세요.

- 에서 EC2 Image Builder 제품 정보 업데이트 AWS Marketplace

EC2 Image Builder 제품 가시성 업데이트

AWS Marketplace 판매자는 제품 가시성을 업데이트하여 제품을 볼 수 있는 구매자를 변경할 수 있습니다 AWS Marketplace. 가시성 상태가 공개로 설정된 경우 모든 AWS Marketplace 구매자가 제품을 볼 수 있습니다. 제품 가시성이 제한으로 설정된 경우 허용 목록에 있는 AWS 계정 IDs에만 제품이 표시됩니다. 제품을 볼 수 AWS 계정 IDs 있는이 ID 허용 목록을 관리하고 업데이트할 수도 있습니다. 다음 섹션에서는 제품 가시성 및 제한됨 가시성 허용 목록을 업데이트하는 방법을 보여줍니다.

주제

- 제품 표시 여부 업데이트
- 허용 목록 업데이트(계정 미리 보기)

제품 표시 여부 업데이트

제품을 볼 수 있는 구매자를 변경하려면 업데이트 가시성을 사용할 AWS Marketplace수 있습니다.

표시 여부를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.

2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 표시 여부 업데이트를 선택합니다.

> **Note**
>
> 이 변경 요청을 사용하여 제품을 제한 상태에서 퍼블릭 상태로 전환하라고 요청할 수 있습니다. 그러나 변경 요청은 AWS Marketplace 판매자 운영 팀 승인 프로세스를 거쳐야 퍼블릭으로 이동할 수 있습니다.

4. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
5. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

허용 목록 업데이트(계정 미리 보기)

제한된 상태에서 제품을 볼 수 있는 AWS 계정 IDs 목록을 변경하려면 허용 목록 업데이트를 사용합니다.

허용 목록을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 허용 목록 업데이트를 선택합니다. 현재 목록에는 현재 허용 목록에 있는 계정 목록이 제공됩니다.
4. 표시에 선호되는 AWS 계정 IDs를 추가하고 IDs를 쉼표로 구분합니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

에 대한 Amazon EC2 인스턴스 추가 및 제한 AWS Marketplace

AWS Marketplace 판매자는 구매자가 사용할 수 있는 인스턴스를 관리할 수 있습니다. 구매자가 사용할 수 있는 제품의 새 인스턴스를 추가할 수 있습니다. 마찬가지로 신규 구매자가 특정 인스턴스에서 제품을 사용하지 못하도록 하려면 인스턴스를 제한할 수 있습니다.

Amazon EC2 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 사용 설명서의 사용 가능한 인스턴스 유형을 참조하세요.

다음 섹션에서는 인스턴스를 추가하고 제한하는 방법을 설명합니다.

주제

- 인스턴스 추가
- 인스턴스 제한

인스턴스 추가

구매자가 EC2 Image Builder 구성 요소로 사용할 수 있는 새 인스턴스를 추가할 수 있습니다.

인스턴스를 추가하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 인스턴스 추가를 선택합니다.
4. 인스턴스 아키텍처를 선택합니다.
5. 사용 가능한 인스턴스 목록에서 추가할 인스턴스 유형을 선택합니다.
6. 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
7. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

> **Note**
>
> - 현재 요금 모델이 무료가 아니거나 기존 보유 라이선스 사용(BYOL) 모델을 사용하는 경우 요금도 추가해야 합니다.
> - 새 인스턴스에 대한 요금이 포함된 인스턴스 추가 또는 요금을 인상하기 위한 요금 업데이트를 생성한 경우 변경한 날로부터 90일 이내에 셀프 서비스를 사용하여 인스턴스를 추가할 수 없습니다. 변경하려면 AWS Marketplace 판매자 운영 팀에 문의하세요.
> - 새 인스턴스 유형에 대한 지원을 추가하면 제품에 대한 비공개 제안을 이미 구독한 고객은 새로 추가된 인스턴스를 자동으로 시작할 수 없습니다. 고객이 액세스하려는 인스턴스로 다른 비공개 제안을 생성해야 합니다. 새 제안을 수락한 후 고객은 새로 추가된 인스턴스를 시작할 수 있습니다. 향후 제품을 구독하는 고객은 인스턴스가 비공개 제안에

포함되어 있는 한 인스턴스를 시작할 수도 있습니다. 새 비공개 제안을 생성하는 방법에 대한 자세한 내용은이 가이드의 에서 계약 수정 AWS Marketplace뒷부분을 참조하세요.

인스턴스 제한

새 구매자가 인스턴스를 사용하지 못하도록 인스턴스를 제한할 수 있습니다. 나중에 필요할 때 인스턴스를 다시 추가할 수 있습니다. 제한된 인스턴스에 있는 EC2 Image Builder 구성 요소의 기존 사용자는 구독 기간 동안 리전의 제품을 계속 사용할 수 있습니다.

인스턴스를 제한하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 인스턴스 제한을 선택합니다.
4. 제한하려는 인스턴스를 선택하고 제한을 선택합니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

> **Note**
>
> 확인란이 흐리게 표시되면 인스턴스가 권장 인스턴스 유형으로써 하나 또는 여러 버전과 연결되어 있다는 뜻입니다. 이러한 인스턴스를 제한하려면 버전 업데이트를 사용하여 다른 권장 인스턴스 유형을 선택합니다. 변경 요청이 완료되고 제한하려는 인스턴스가 더 이상 권장 인스턴스 유형이 아니면 인스턴스 제한으로 돌아가서 선택한 인스턴스를 제한할 수 있습니다.

에서 EC2 Image Builder 제품의 버전 관리 AWS Marketplace

EC2 Image Builder 구성 요소에 대한 버전 관리는 UI에서 수행할 수 없습니다. 자세한 내용은 다음 카탈로그 API 섹션을 참조하세요.

- 새 버전 추가
- 기존 버전에 대한 정보 업데이트

- 버전 제한

에서 EC2 Image Builder 제품 정보 업데이트 AWS Marketplace

단일 EC2 Image Builder 구성 요소를 생성한 후 해당 구성 요소와 연결된 일부 정보를 변경할 수 있습니다 AWS Marketplace. 예를 들어 새 버전이 제품의 설명이나 주요 내용을 수정하는 경우 제품 정보를 새 데이터로 편집할 수 있습니다. 제품 제목, SKU 설명, 범주, 키워드 등 다른 제품 정보를 업데이트할 수도 있습니다. 자세한 내용은 다음 절차를 참조하세요.

제품 정보를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 메뉴에서 제품 정보 업데이트를 선택합니다.
4. 변경해야 하는 다음 필드를 업데이트합니다.
   - 제품 제목
   - SKU
   - 간단한 설명
   - 자세한 설명
   - 제품 로고 이미지 URL
   - 하이라이트
   - 제품 범주
   - 키워드
   - 제품 비디오 URL
   - 리소스
   - 지원 정보

   > **Note**
   > 로고 형식에 대한 자세한 내용은 회사 및 제품 로고 요구 사항 섹션을 참조하세요.

5. 제출을 선택합니다.

6. 요청이 요청 탭에 표시되고 상태가 검토 중인지 확인합니다. 목록에 요청이 보이지 않으면 페이지를 새로 고쳐야 할 수도 있습니다.

및 국가별 EC2 Image Builder 제품 가용성 AWS 리전 관리

에서 제품을 생성할 때 제품을 사용할 수 있는 AWS 리전을 AWS Marketplace선택합니다. 구매자가 제품을 구매할 수 있는 국가도 선택할 수 있습니다. 이 두 속성은 비슷하지만 동일하지는 않습니다. 예를 들어 구매자가 미국에 살면서 미국에서 제품을 구매하지만 유럽(프랑크푸르트) 리전에 제품을 설치할 수 있습니다. 이 구매자가 제품을 구매할 수 있으려면 국가 목록에 미국을 포함시키고 리전 목록에 유럽(프랑크푸르트) 리전을 포함시켜야 합니다. 다음 섹션을 사용하여 리전 및 국가별로 제품 가용성을 업데이트할 수 있습니다.

주제

- 추가 AWS 리전
- 제한 AWS 리전
- 향후 AWS 리전지원 업데이트
- 국가별 가용성 업데이트

추가 AWS 리전

구매자가 제품을 사용할 수 있는 리전을 추가할 수 있습니다.

리전을 추가하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 리전 추가를 선택합니다.
4. 사용 가능한 리전 목록에서 추가할 리전을 선택합니다.
5. 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

> **Note**
>
> 새에 대한 지원을 추가하면 제품에 대한 비공개 제안을 이미 구독한 AWS 리전고객은 새로 추가된 리전에 자동으로 액세스할 수 없습니다. 고객이 액세스하려는 리전을 사용하여 다른 비공개 제안을 만들어야 합니다. 고객은 새 제안을 수락한 후 새로 추가된 리전에 액세스할 수 있습니다. 향후 제품을 구독하는 고객은 해당 리전이 비공개 제안에 포함되는 한 해당 리전에 액세스할 수도 있습니다. 새 비공개 제안을 만드는 방법에 대한 자세한 내용은 Private offer upgrades, renewals, and amendments를 참조하세요.

제한 AWS 리전

신규 구매자가 특정에서 제품을 사용하지 못하도록 리전을 제한 AWS 리전할 수 있습니다. 나중에 리전을 다시 추가할 수 있습니다. 해당 리전의 기존 제품 구독자는 구독을 유지하는 한 해당 리전의 제품을 계속 사용할 수 있습니다.

리전을 제한하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 리전 제한을 선택합니다.
4. 드롭다운 메뉴를 선택하여 현재 제품을 사용할 수 있는 리전 목록을 확인합니다.
5. 제한하려는 리전을 선택합니다.
6. 선택한 리전이 토큰으로 표시됩니다. 제한 중인 리전 목록을 검토하고, 제한하지 않을 지역에 X를 입력합니다.
7. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
8. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

요청이 성공하면 기존 사용자에게 제한될 리전을 알리는 다음과 같은 이메일 메시지가 전송됩니다. 기존 사용자는 구독을 유지하는 한 제품을 계속 사용할 수 있지만, 구독을 취소하면 다시 구독할 수 없습니다.

```
Greetings from AWS Marketplace,
```

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

This impacts you in the following ways:

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

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

향후 AWS 리전지원 업데이트

제품을 새로 출시된에 온보딩하려면 향후 리전 지원 업데이트를 사용할 AWS 리전수 있습니다.

향후 리전 지원을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 향후 리전 지원 업데이트를 선택합니다.
4. 향후 리전 지원을 활성화하여가 사용자를 대신하여 새로 출시된 AWS Marketplace 에 제품을 온보딩할 수 있도록 선택할 수 AWS 리전 있습니다.
5. 이 기능을 활성화한 후에는 모든 향후 리전을 선택할 수도 있고 미국 리전으로만 제한할 수도 있습니다.
6. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
7. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

국가별 가용성 업데이트

제품을 구독하고 제공할 수 있는 국가를 변경하려면 가용성 업데이트를 사용하면 됩니다.

국가별 가용성을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 메뉴에서 가용성 업데이트를 선택합니다.
4. 다음 옵션 중 하나를 선택하세요.
   1. 모든 국가 - 지원되는 모든 국가에서 사용할 수 있습니다.
   2. 일부 국가를 제외한 모든 국가 - 일부 국가를 제외하고 지원되는 모든 국가에서 사용할 수 있습니다.
   3. 사용자 지정 목록 - 특정 목록의 국가에서만 제품을 사용할 수 있습니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

EC2 Image Builder AWS Marketplace 제품의 최종 사용자 라이선스 계약(EULA) 업데이트

AWS Marketplace 판매자는 EC2 Image Builder 구성 요소의 사용에 적용되는 최종 사용자 라이선스 계약(EULA)을 업데이트할 수 있습니다. EULA는 AWS Marketplace의 공개 소프트웨어 리스팅에 대한 제품 리스팅 페이지에 있습니다. 판매자 고유의 EULA를 적용할 수도 있고 AWS Marketplace 표준 계약(SCMP)을 사용할 수도 있습니다. 다음 절차에서는 EC2 Image Builder 구성 요소의 EULA를 업데이트하는 방법을 보여줍니다.

EULA에 대한 자세한 내용은 에서 표준화된 계약 사용 AWS Marketplace 섹션을 참조하세요.

EULA를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 탭을 선택한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 최종 사용자 라이선스 계약 업데이트를 선택합니다.
4. 표준 계약 AWS Marketplace (SCMP)을 선택하거나 사용자 지정 EULA를 제출할 수 있습니다. 사용자 지정 EULA의 경우 Amazon S3 버킷의 사용자 지정 계약에 대한 URL을 제공해야 합니다.

> **Note**
>
> Amazon S3 버킷에 대한 공개 액세스가 활성화되어 있어야 합니다.

5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

에서 EC2 Image Builder 제품에 대한 환불 정책 업데이트 AWS Marketplace

AWS Marketplace 판매자는 EC2 Image Builder 구성 요소에 대한 환불 정책을 설정할 수 있습니다. 제품의 환불 정책을 변경하려면 환불 정책 업데이트를 사용하면 됩니다. 다음 절차에서는 환불 정책을 업데이트하는 방법을 보여줍니다.

환불 정책을 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 환불 정책 업데이트를 선택합니다.
4. 현재 환불 정책 세부 정보가 텍스트 상자에 제공되며 이 정보를 편집할 수 있습니다. 요청을 제출하면 현재 환불 정책을 덮어씁니다.
5. 변경 요청 제출을 선택하여 요청을 제출하고 검토를 받습니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

AWS Marketplace에서 제품 제거

제품을 게시한 후에서 제품을 제거할 수 있습니다 AWS Marketplace. 이를 일몰이라고도 합니다. 제품을 제거하려면 제품을 식별하고 제거 이유 및 자신의 연락 이메일 주소와 함께 제거 요청을 제출합니다. 현재 제품을 새 제품으로 교체하려는 경우 교체 제품 ID를 제공할 수도 있습니다.

제품 제거를 요청한 후에는 신규 고객이 제품을 구독할 수 없습니다. 최소 90일 동안 기존 고객을 지원해야 합니다.

> **Note**
>
> AWS Marketplace 관리 포털에서는 제한된 제품을 삭제할 수 없습니다. 포털은 계정의 게시 기록의 일부로 이를 보관합니다.

에서 제품을 제거하는 요청은 다음 조건으로 처리됩니다 AWS Marketplace .

- 검색 AWS Marketplace , 찾아보기 및 기타 검색 도구에서 제품이 제거됩니다. 구독 버튼이나 기능이 비활성화되고, 제품을 더 이상 사용할 수 없다고 분명하게 밝히는 메시지가 페이지에 표시됩니다. 계속 URL을 사용하여 제품 세부 정보 페이지에 액세스할 수 있으며 공개 검색 엔진에 이 페이지가 인덱싱됩니다.
- 제거 이유(예: 지원 종료, 제품 업데이트 종료 또는 교체 제품)를 명시해야 합니다. 제거된 제품에 대한 지속적인 지원 요구 사항은 AWS Marketplace 판매자 이용 약관을 참조하세요.
- AWS Marketplace 는 제품 제거, 제거 이유 및 판매자 연락처 정보를 알리는 이메일 메시지를 통해 현재 구매자에게 연락합니다.
- 현재 구매자는 구독을 취소할 때까지 소프트웨어에 계속 액세스할 수 있습니다. 구독자는 어떤 방식으로도 제품 제거의 영향을 받지 않습니다.

를 사용하여 생성된 제품을 제거하려면 AWS Marketplace Management Portal

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 제품 탭을 선택하고 서버를 선택합니다.
3. 서버 제품의 제품 페이지에서 제거할 제품을 찾습니다. 변경 요청 드롭다운 목록에서 제품 가시성 업데이트를 선택합니다.
4. 제품 가시성 업데이트 페이지에서 제한됨을 선택합니다.
5. (선택 사항) 제거하려는 제품을 대신할 다른 제품이 있는 경우 교체 제품 ID를 입력합니다.
6. 정보가 정확한지 검토한 후 제출을 선택합니다.

제품 제거 요청을 제출하면 What’s next(다음 단계) 정보 페이지가 표시됩니다. AWS Marketplace 판매자 작업은 요청을 검토하고 처리합니다. 요청을 보고 제출의 상태를 확인합니다.

제품이 제거되고 나면 AWS Marketplace Management Portal의 현재 제품 목록에 제품이 표시됩니다. 현재 제품에서는 제품에 대한 스프레드시트 다운로드를 하는 작업만 수행할 수 있습니다. 다른 만료 요청을 편집하거나 제출할 수는 없습니다.

제품 제거에 대해 궁금한 점이 있는 경우 AWS Marketplace 판매자 작업 팀에 문의하세요.

의 컨테이너 기반 제품 AWS Marketplace

AWS Marketplace 는 Docker 컨테이너를 사용하는 소프트웨어 제품을 지원합니다. 컨테이너 제품은 함께 사용되는 컨테이너 이미지와 배포 템플릿 세트의 제공 옵션으로 구성됩니다. 제품의 제공 옵션을 1개 이상, 최대 4개 제출해야 합니다. 각 제공 옵션에 대해 고객이 해당 제공 옵션을 시작할 수 있도록 컨테이너 이미지, 사용 지침 및 배포 템플릿 링크 세트를 제공합니다. 이 주제에서는 AWS Marketplace 의 컨테이너 기반 제품에 대한 정보를 제공합니다.

AWS Marketplace 구매자는 게시된 제품 세부 정보 페이지에서 사용 가능한 제공 옵션을 볼 수 있습니다. 구매자가 제품을 구독하고 선호하는 제공 옵션을 선택하면 제품을 시작하고 사용하는 방법에 대한 정보와 지침이 제공됩니다. 컨테이너 이미지 제공 옵션의 경우 사용 가능한 배포 템플릿과 컨테이너 이미지 URL의 링크가 구매자에게 제공됩니다. 또한 개별 컨테이너 이미지를 가져오는 방법에 대한 지침이 구매자에게 제공됩니다. 차트 Helm 제공 옵션의 경우 구매자는 Helm을 시작하는 방법에 대한 단계별 지침을 볼 수 있습니다.

구매 경험에 대한 연습은 Amazon ECS 클러스터에 AWS Marketplace 컨테이너 배포(3:34) 동영상을 참조하세요.

모든 환경의 모든 Kubernetes 클러스터 AWS Marketplace 에서의 타사 Kubernetes 애플리케이션을 찾고, 구독하고, 배포할 수 있습니다. Amazon EKS Anywhere(EKS Anywhere)를 사용하여 Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS) AWS Fargate및 온프레미스에 타사 Kubernetes 애플리케이션을 배포할 수 있습니다. 온프레미스의 자체 관리형 Kubernetes 클러스터 또는 Amazon Elastic Compute Cloud(Amazon EC2)에 배포할 수도 있습니다.

Docker 호환 런타임에 무료 및 기존 보유 라이선스 사용(BYOL) 컨테이너 제품을 실행할 수 있습니다.

주제

- 도움말 가져오기
- 컨테이너 제품 시작하기
- 에 대한 컨테이너 기반 제품 요구 사항 AWS Marketplace
- 에 대한 컨테이너 제품 요금 AWS Marketplace
- 컨테이너 제품 결제, 측정 및 라이선스 통합
- 컨테이너 제품에 대한 Amazon SNS 알림

도움말 가져오기

컨테이너 제품과 관련하여 지원을 받으려면 AWS Marketplace 의 비즈니스 개발 파트너 또는 AWS Marketplace 판매자 운영 팀에 문의하세요.

컨테이너 제품 시작하기

AWS Marketplace 판매자는 컨테이너 기반 소프트웨어 제품을 만들 수 있습니다. 컨테이너 제품은 컨테이너 이미지 및 배포 템플릿 세트인 전송 옵션으로 구성됩니다. 다음 주제에서는 컨테이너 제품을 시작하는 방법을 보여줍니다.

- 제품 수명 주기
- 사전 조건
- 1단계: 컨테이너 제품의 제품 ID 및 제품 코드 만들기
- 2단계: 초기 리스팅 만들기
- 3단계: 제품의 초기 버전 추가
- 4단계: (유료 제품만 해당) 측정 또는 계약 요금 통합
- 다음 단계
- 컨테이너 제품을 스캔하여 보안 문제 확인

제품 수명 주기

에서 제품을 생성하면 AWS Marketplace처음에 제품이 제한된 가시성으로 게시되므로 제품을 생성한 계정을 포함하여 허용 목록의 계정이 제품을 볼 수 있습니다. 준비가 되면 AWS Marketplace 카탈로그에 게시하여 구매자가 제품을 구독하고 구매할 수 있도록 할 수 있습니다.

서버 제품 페이지에서 제품 목록을 볼 수 있습니다. 제품의 현재 스테이지에 따라 제품의 상태는 다음 중 하나입니다.

- 준비 중 - 아직 판매자가 정보를 추가하고 있는 불완전한 제품. 셀프 서비스 경험에서 처음으로 저장 후 종료할 때, 변경 요청이 성공하면 판매자가 제출한 완료된 단계의 정보가 포함된, 게시되지 않은 제품이 생성됩니다. 이 상태에서 제품에 정보를 계속 추가할 수도 있고 변경 요청을 통해 이미 제출한 세부 정보를 변경할 수도 있습니다.
- 제한 - 제품이 시스템에 제출되고 시스템의 모든 검증을 통과하면 완성되었습니다. 그 후 제품이 제한 상태로 릴리스되었습니다. 이제 판매자 계정과 판매자가 허용한 사람만 제품 세부 정보 페이지에

액세스할 수 있습니다. 세부 정보 페이지를 통해 제품을 테스트할 수 있습니다. 자세한 내용이나 도움이 필요하면 AWS Marketplace 판매자 운영 팀에 문의하세요.

- 퍼블릭 - 구매자가 제품을 살펴보고 구독할 수 있도록 제품을 게시할 준비가 되면 표시 여부 업데이트 변경 요청을 사용합니다. 이 요청은 AWS Marketplace 판매자 운영 팀이 AWS 정책을 기준으로 제품을 검토하고 감사하는 워크플로를 시작합니다. 제품이 승인되고 변경 요청이 처리되면 제품 상태가 제한에서 퍼블릭으로 바뀝니다. AWS 지침에 대한 자세한 내용은 단원을 참조하십시오에 대한 컨테이너 기반 제품 요구 사항 AWS Marketplace.
- 제한 - 신규 사용자가 제품을 구독하지 못하게 하려면 표시 여부 업데이트 변경 요청을 사용하여 제품을 제한하면 됩니다. 제한 상태는 허용 목록에 있는 기존 사용자는 제품을 계속 사용할 수 있다는 것을 의미합니다. 하지만 제품이 더 이상 대중에게 표시되거나 신규 사용자에게 제공되지 않습니다.

사전 조건

시작하기 전에 다음 사전 조건을 완료해야 합니다.

1. AWS Marketplace Management Portal에 액세스하고 사용합니다. 판매자로 등록하고 판매하는 제품을 관리하는 데 사용하는 도구입니다 AWS Marketplace. 자세한 내용은 AWS Marketplace Management Portal 단원을 참조하십시오.
2. 판매자로 등록하고 세금 및 은행 정보를 제출합니다. 자세한 내용은 AWS Marketplace 판매자로 등록 단원을 참조하세요.
3. Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS) 또는 AWS Fargate에서 컨테이너를 하나 이상 생성합니다. 관련 이미지에 대한 링크가 있는지 확인합니다.
4. AWS Marketplace에서 컨테이너 제품을 생성하고 통합하는 방법을 계획합니다.

   제품을 공개적으로 게시하기 전에 요금, 권한 부여 및 측정 전략을 계획하는 것이 좋습니다.

   - 컨테이너 기반 제품의 요구 사항에 대한 자세한 내용은 에 대한 컨테이너 기반 제품 요구 사항 AWS Marketplace 섹션을 참조하세요.
   - 제품의 요금 설정에 대한 자세한 내용은 에 대한 컨테이너 제품 요금 AWS Marketplace 섹션을 참조하세요.
   - 유료 컨테이너 기반 제품의 사용자 지정 측정에 대한 자세한 내용은 AWS Marketplace Metering Service로 시간당 및 사용자 지정 측정 섹션을 참조하세요.

개요: 컨테이너 제품 만들기

컨테이너 제품을 생성하는 단계는 다음과 같습니다.

1. 1단계: 컨테이너 제품의 제품 ID 및 제품 코드 만들기
2. 2단계: 초기 리스팅 만들기
3. 3단계: 제품의 초기 버전 추가
4. 4단계: (유료 제품만 해당) 측정 또는 계약 요금 통합
5. 제품 표시 여부 업데이트

제품 수명 주기에 대한 자세한 내용은 제품 수명 주기 섹션을 참조하세요.

1단계: 컨테이너 제품의 제품 ID 및 제품 코드 만들기

컨테이너 제품을 시작하려면에서 제품 ID와 제품 코드 레코드를 생성해야 합니다 AWS Marketplace. 제품 ID는 제품 수명 주기 전반에 걸쳐 제품을 추적하는 데 사용됩니다.

다음 절차에 따라에서 새 컨테이너 제품을 AWS Marketplace Management Portal생성하고 제품 ID를 생성합니다.

> **Note**
> 제품과 페어링되는 컨테이너의 퍼블릭 키도 이 프로세스에서 생성합니다.

컨테이너 제품 ID를 생성하는 방법

1. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
2. 메뉴 표시줄에서 제품을 선택하고 서버를 선택합니다.
3. 서버 제품 생성을 선택하고 컨테이너를 선택합니다.
4. 컨테이너 제품 ID 및 코드를 생성합니다.

   > **Note**
   > (선택 사항) 태그 기반 권한 부여를 사용하려면 제품에 태그를 지정합니다. 자세한 내용은 AWS 리소스에 태그 지정을 참조하십시오.

5. 계속을 선택하여 제품 생성을 계속합니다.

2단계: 초기 리스팅 만들기

제품 ID, 제품 코드 및 퍼블릭 키를 생성한 후에는 마법사를 사용하여 초기 목록을 생성합니다.

1. 제품 목록의 제품 정보를 제공합니다.
2. 제품의 요금 모델을 결정합니다.

> **Note**
>
> 자세한 내용은 컨테이너 제품 요금을 참조하세요.

> **Note**
>
> 유료 제품의 경우 판매자와 AWS Marketplace 판매자 운영 팀이 높은 비용을 들이지 않고도 제품을 테스트할 수 있도록 제품이 0.01 USD로 시작합니다. 제품을 대중에 공개할 때 실제 요금을 입력합니다.

3. 환불 정책, EULA 및 제안 가용성을 비롯한 추가 제안 정보를 제공합니다.
4. 컨테이너 제품의 초기 리포지토리를 추가합니다.
5. 마지막 단계에서 제출을 선택하여 제품의 표시 여부를 제한으로 전환합니다.

> **Note**
>
> 컨테이너 제품은 처음에 자리 표시자 버전으로 생성됩니다. 판매자는 제품의 표시 여부가 제한일 때 최종 버전을 추가하게 됩니다.

3단계: 제품의 초기 버전 추가

제품의 수명 주기 동안 여러 제품 버전이 출시될 수 있습니다. 각 버전에는 해당 버전과 관련된 컨테이너 이미지 세트가 있습니다. 제품의 초기 버전을 추가하려면 에서 컨테이너 제품의 새 버전 추가 AWS Marketplace 섹션을 참조하세요.

4단계: (유료 제품만 해당) 측정 또는 계약 요금 통합

사용량 요금이 적용되는 컨테이너 기반 제품의 경우 AWS Marketplace Metering Service를 사용하여 제품 사용 권한을 확인하고 요금 청구를 위해 사용량을 측정합니다. 요금 정보를 설정할 때 생성한 요

금 모델로 측정해야 합니다. 자세한 내용은 AWS Marketplace Metering Service로 시간당 및 사용자 지정 측정 단원을 참조하세요.

계약 요금

계약 가격이 적용되는 컨테이너 기반 제품의 경우 AWS License Manager 를 사용하여 라이선스를 제품과 연결합니다.

와의 통합에 대한 자세한 내용은 단원을 AWS License Manager참조하십시오를 사용한 컨테이너 제품의 계약 요금 AWS License Manager.

5단계: 제품 가시성 업데이트

에서 제품을 생성하면 AWS Marketplace처음에 제품이 제한된 가시성으로 게시되므로 제품을 생성한 계정을 포함하여 허용 목록의 계정이 제품을 볼 수 있습니다. 제품 가시성을 업데이트하여 구매자가 제품을 구독하고 구매하도록 할 수 있습니다. 또는 추가할 제품 허용 목록을 업데이트할 수 있습니다 AWS 계정. 이 주제에서는 AWS Marketplace에서 제품을 볼 수 있는 구매자를 관리하는 방법을 보여줍니다.

제품 가시성 및 수명 주기에 대한 자세한 내용은 제품 수명 주기 섹션을 참조하세요.

주제

- 제품 표시 여부 업데이트
- AWS 계정 ID 허용 목록 업데이트

제품 표시 여부 업데이트

표시 여부를 업데이트하는 방법

1. AWS Marketplace Management Portal 에서를 연 https://aws.amazon.com/marketplace/management/tour/다음 판매자 계정에 로그인합니다.
2. 서버 제품 페이지로 이동한 후 현재 서버 제품 탭에서 수정하려는 컨테이너 기반 제품을 선택합니다.
3. 변경 요청 드롭다운에서 표시 여부 업데이트를 선택합니다.

> **Note**
>
> 이 변경 요청을 사용하여 제품을 제한 상태에서 퍼블릭 상태로 전환하라고 요청할 수 있습니다. 그러나 변경 요청은 AWS Marketplace 판매자 운영 팀 승인 프로세스를 거쳐야 퍼블릭으로 이동할 수 있습니다.

4. 제출을 선택하여 검토 요청을 제출합니다.
5. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

AWS 계정 ID 허용 목록 업데이트

제한된 상태에서 제품을 볼 수 있는 AWS 계정 IDs 목록을 변경할 수 있습니다. 허용 목록에 있는 계정은 제품 세부 정보 페이지에 제품 버전과 함께 제한 배지가 표시됩니다.

1. https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace Management Portal를 열고 판매자 계정에 로그인합니다.
2. 서버 제품 페이지에서 수정하려는 컨테이너 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 허용 목록 업데이트를 선택합니다. 현재 허용 목록에 있는 계정 목록이 표시됩니다.
4. 허용 목록에 있는 AWS 계정 필드에 AWS 계정 ID를 입력하고, 쉼표로 ID를 구분합니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

다음 단계

컨테이너 제품을 만든 후 다음 주제의 정보를 사용하여 구성하고 관리할 수 있습니다.

- AWS Marketplace에서 컨테이너 제품의 제품 정보 업데이트
- 에서 컨테이너 제품의 새 버전 추가 AWS Marketplace
- AWS Marketplace에서 컨테이너 제품 요금 관리
- AWS Marketplace에서 국가별 컨테이너 제품 가용성 업데이트
- 에서 컨테이너 제품에 대한 최종 사용자 라이선스 계약(EULA) 업데이트 AWS Marketplace

- AWS Marketplace에서 컨테이너 제품 테스트 및 릴리스

컨테이너 제품을 스캔하여 보안 문제 확인

컨테이너 제품에 새 버전을 추가하기 위한 변경 요청을 생성하면 AWS는 새 버전에 포함된 컨테이너 이미지를 스캔하여 보안 취약성을 확인합니다. 이를 위해 AWS는 이미지를 한 계층씩 정적 스캔합니다. AWS는 원격으로 악용될 수 있는 위험 벡터가 포함된 치명적인 취약성을 발견하면 판매자에게 발견된 문제 목록을 제공합니다. 수집 및 게시 프로세스가 지연되지 않도록 Clair, Twistlock, Aqua Security 또는 Trend Micro와 같은 컨테이너 이미지 스캐너를 사용하여 자체적으로 보안 분석을 수행할 것을 강력하게 권장합니다.

컨테이너 이미지를 빌드할 때 선택하는 기본 이미지는 최종 이미지의 보안 프로파일에 커다란 영향을 미칠 수 있습니다. 이미 치명적인 것으로 알려진 취약성이 있는 기본 이미지를 선택하면 애플리케이션 소프트웨어 계층에 아무런 문제가 없더라도 기본 계층으로 인해 해당 기본 이미지에 플래그가 지정됩니다. 이미지를 빌드하고 제출하기 전에 취약성이 없는 기본 컨테이너로 시작하는 것이 좋습니다 AWS Marketplace.

AWS Marketplace에서 컨테이너 제품의 제품 정보 업데이트

제품 ID를 만들고 요금을 설정한 후에는 AWS Marketplace에서 컨테이너 제품에 대해 고객에게 표시되는 내용을 포함하여 제품 정보를 편집할 수 있습니다. 예를 들어 새 버전이 제품의 설명이나 주요 내용을 수정하는 경우 제품 정보를 새 데이터로 편집할 수 있습니다. 제품 제목, SKU 설명, 범주, 키워드 등 다른 제품 정보를 업데이트할 수도 있습니다. 다음은 제품 세부 정보를 생성하는 방법을 간략하게 설명하는 절차입니다.

컨테이너 제품의 제품 세부 정보를 생성하거나 업데이트하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 서버를 선택합니다.
3. 서버 제품 탭에서 수정하려는 제품을 선택합니다.
4. 변경 요청 드롭다운 메뉴에서 제품 정보 업데이트를 선택합니다.
5. 다음 중 변경하려는 필드를 업데이트합니다.
    - 제품 제목
    - SKU
    - 간단한 설명
    - 자세한 설명

- 제품 로고 이미지 URL
- 하이라이트
- 제품 범주
- 키워드
- 제품 비디오 URL
- 리소스
- 지원 정보

> **Note**
>
> 이미지 URL은 공개적으로 액세스할 수 있는 Amazon S3 버킷에 있어야 합니다. 로고 형식에 대한 자세한 내용은 회사 및 제품 로고 요구 사항 섹션을 참조하세요.

6. 제출을 선택합니다.
7. 요청이 요청 탭에 표시되고 상태가 검토 중인지 확인합니다. 목록에 요청이 보이지 않으면 페이지를 새로 고쳐야 할 수도 있습니다.

언제든지 서버 제품 페이지의 요청 탭에서 요청 상태를 확인할 수 있습니다.

에서 컨테이너 제품의 새 버전 추가 AWS Marketplace

AWS Marketplace 판매자는 컨테이너 제품의 새 버전을 추가하고, 버전을 관리하고, 버전 정보를 업데이트할 수 있습니다. 제품의 수명 주기 동안 여러 제품 버전이 출시될 수 있습니다. 각 버전에는 해당 버전과 관련된 컨테이너 이미지 세트가 있습니다. 다음 주제에서는 컨테이너 제품의 제품 버전을 관리하는 방법을 보여줍니다.

> **Note**
>
> 제품 ID와 제품 요금을 생성하기 전에는 제품에 버전을 추가할 수 없습니다. 이 단계에 대한 자세한 내용은 1단계: 컨테이너 제품의 제품 ID 및 제품 코드 만들기 섹션을 참조하세요.

주제

- 1단계: 리포지토리 추가
- 2단계: 컨테이너 이미지 및 아티팩트를 리포지토리에 업로드

- 3단계: 컨테이너 제품에 새 버전 추가
- 4단계: 버전 정보 업데이트
- Amazon EKS 추가 기능 버전 제한

1단계: 리포지토리 추가

제품의 컨테이너 이미지와 기타 아티팩트는 AWS Marketplace의 리포지토리에 저장됩니다. 일반적으로 필요한 아티팩트마다 리포지토리를 하나씩 생성하지만, 서로 다른 태그가 지정된 여러 아티팩트 버전을 리포지토리에 저장할 수 있습니다.

> **Note**
> 제품 배포의 모든 이미지는 AWS Marketplace 리포지토리의 이미지를 사용해야 합니다.

다음 절차에서는에 필요한 리포지토리를 추가하는 방법을 설명합니다 AWS Marketplace.

리포지토리를 추가하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 서버를 선택합니다.
3. 서버 제품 탭에서 수정하려는 제품을 선택한 다음, 변경 요청 드롭다운에서 리포지토리 추가를 선택합니다.
4. 생성할 리포지토리의 이름을 입력합니다. 새 리포지토리를 두 개 이상 생성하려는 경우:
   - 각 추가 리포지토리에 대해 새 리포지토리 추가를 선택합니다.
   - 고유한 이름을 지정합니다. 선택한 고유 이름은 판매자 계정의 모든 제품에 적용되어야 합니다.

> **Note**
> 리포지토리는 `<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>` 구조입니다. (다음 절차에서) 리포지토리에 항목을 추가하면 태그가 지정되고 `<repositoryID>.dkr.ecr.us-east-1.amazonaws.com/<sellerName>/<repositoryName>:<tag>` 구조를 갖게 됩니다. 는의 내부 IDrepositoryID입니다 AWS Marketplace. sellerName은 판매자

계정에 대해 생성한 이름을 기반으로 합니다. 이 단계에서 `respositoryName`을 정의합니다. `tag`는 리포지토리에 아티팩트를 업로드할 때 설정됩니다.

5. 제출을 선택합니다.

> **Note**
>
> 제품 하나에 가능한 리포지토리는 최대 50개입니다.

새 요청이 생성되고 요청 탭에 표시됩니다. 작업이 완료된 후 몇 분 내에, 앞에서 생성한 리포지토리에 컨테이너 이미지와 기타 아티팩트를 추가할 수 있습니다.

2단계: 컨테이너 이미지 및 아티팩트를 리포지토리에 업로드

컨테이너 이미지 및 아티팩트를 리포지토리에 업로드하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 서버를 선택합니다.
3. 서버 제품 탭에서 수정하려는 제품을 선택합니다.
4. 변경 요청 드롭다운에서 리포지토리 추가를 선택합니다.
5. 기존 리포지토리 보기를 선택합니다.
6. 컨테이너 이미지 및 아티팩트를 업로드할 리포지토리를 선택합니다.
7. Docker 컨테이너 이미지와 Helm 차트를 해당 리포지토리에 푸시하는 데 사용할 수 있는 명령을 포함한 지침 목록을 열려면 푸시 명령 보기를 선택합니다.

   컨테이너 이미지 및 기타 아티팩트를 리포지토리로 푸시하는 방법에 대한 일반적인 정보는 Amazon Elastic Container Registry 사용 설명서의 이미지 푸시를 참조하세요.

   > **Note**
   >
   > `docker pull` 또는 `docker push`를 호출할 때 다음 Amazon Elastic Container Registry(Amazon ECR) API 작업을 사용할 수 있습니다.
   >
   > - `DescribeImages` - 리포지토리의 이미지에 대한 메타데이터를 검토할 때 사용합니다.
   > - `GetAuthorizationToken`- 아티팩트를 리포지토리에 업로드하기 전에 인증에 사용하며, 그 후에는 `docker pull` 또는 `docker push` 명령을 사용합니다.

- `ListImages` - 푸시한 이미지 목록을 보는 데 사용합니다.

8. 나열된 명령을 사용하여 로컬 리포지토리에서 제품의 리포지 AWS Marketplace 토리로 필요한 아티팩트를 푸시합니다.

> **Note**
>
> `push` 명령에서 제공하는 **`tag`**는 리포지토리에 업로드하는 아티팩트 버전을 구분하는 데 사용됩니다. 아티팩트가 속한 버전에 적합한 태그를 사용합니다.

9. 버전에 필요한 컨테이너 이미지 또는 아티팩트마다 이 과정을 반복합니다.

> **Note**
>
> 버전의 각 제공 옵션에 최대 50개의 컨테이너 이미지 또는 아티팩트를 포함할 수 있습니다. 제공 옵션에 대한 자세한 내용은 다음 절차를 참조하세요.

아티팩트를 업로드했으면 제품 버전을 생성할 준비가 된 것입니다.

> **Note**
>
> 에 대한 컨테이너 기반 제품 요구 사항 AWS Marketplace을 충족하는지 확인하기 위해 컨테이너 이미지가 자동으로 스캔됩니다. 자세한 정보는 컨테이너 제품을 스캔하여 보안 문제 확인 섹션을 참조하세요.

새 제공 옵션 추가

컨테이너 제품의 각 버전에는 전송 옵션이 필요합니다. 전송 옵션은 구매자가 사용할 수 있는 배포 옵션을 지정합니다. 아래 전송 옵션 중 하나에 따라 적절한 아티팩트를 리포지토리에 업로드해야 합니다.

- 컨테이너 이미지 전송 옵션의 경우 제품 설치에 필요한 모든 컨테이너 이미지를 AWS Marketplace 콘솔에서 생성된 Amazon Elastic Container Registry(Amazon ECR) 리포지토리에 업로드합니다.
- Helm 차트 전송 옵션의 경우 AWS Marketplace 콘솔에서 생성된 Amazon ECR 리포지토리에 Helm 차트 및 컨테이너 이미지를 업로드합니다.
- Amazon EKS 콘솔 추가 기능 전송 옵션의 경우 AWS Marketplace 콘솔에서 생성된 Amazon ECR 리포지토리에 Helm 차트 및 컨테이너 이미지를 업로드합니다.

3단계: 컨테이너 제품에 새 버전 추가

> **Note**
> 컨테이너에 새 버전을 추가할 때 오류가 발생하면 AWS Marketplace Catalog API 참조의 새 버전의 비동기 오류 테이블 추가를 참조하세요.

컨테이너 제품에 새 버전을 추가하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 서버를 선택합니다.
3. 서버 제품 탭에서 버전을 추가하려는 제품을 선택합니다. 변경 요청 드롭다운에서 새 버전 추가를 선택합니다.
4. 새 버전 추가 페이지에서 사용 중인 버전의 버전 제목 및 릴리스 정보를 입력합니다.
5. 버전 세부 정보를 입력한 후에는 제공 옵션을 추가합니다. 제공 옵션은 구매자가 제품 버전에서 소프트웨어를 시작하는 데 사용할 수 있는 지침 및 정보 세트입니다. 제공 옵션은 구매자의 입장에서는 이행 옵션입니다.

   > **Note**
   > 판매자의 제품은 컨테이너 이미지가 서로 다른 여러 플랫폼(예: Kubernetes 및 Ubuntu 배포)을 지원할 수 있습니다. 고객이 제품을 설정할 수 있는 방법마다 한 가지 제공 옵션을 만들 수 있으며, 제품 버전당 최대 4개의 제공 옵션을 만들 수 있습니다.

   a. 제품에 이미 다른 버전의 제공 옵션이 있는 경우 기존 옵션을 템플릿으로 사용하여 새 버전에 제공 옵션을 추가할 수 있습니다. 제공 옵션의 목록에서 추가하려는 제공 옵션을 선택합니다. 다음 단계의 지침에 따라 옵션을 편집할 수 있습니다.

   b. 새 제공 옵션을 추가하려면 새 제공 옵션을 선택합니다. 옵션을 추가한 후에는 다음 단계의 지침에 따라 옵션을 구성합니다.

6. 제공 옵션에 사용할 제공 방법을 선택합니다. 제공 방법에 따라 구매자가 소프트웨어를 시작하는 방법이 결정됩니다.

   - 컨테이너 이미지 제공 옵션의 경우 AWS Marketplace 콘솔에서 생성한 Amazon Elastic Container Registry(Amazon ECR) 리포지토리에 있는 컨테이너 이미지의 경로를 입력합니다.

구매자는 이 컨테이너 이미지 경로를 사용하여 이미지를 환경으로 직접 가져와서 소프트웨어를 시작합니다.

- Helm 차트 전송 옵션의 경우 AWS Marketplace 콘솔에서 만들어진 Amazon ECR 리포지토리에 있는 Helm 차트에 대한 경로를 제공합니다. 구매자는 배포 환경에 Helm 차트를 설치하여 소프트웨어를 시작합니다.
- Amazon EKS 콘솔 추가 기능 제공 옵션의 경우 AWS Marketplace 콘솔에서 만든 Amazon ECR 리포지토리의 Helm 차트 경로를 입력합니다. 구매자는 Amazon EKS 콘솔 또는 기본 Amazon EKS 추가 기능 API를 사용하여 컨테이너를 설치하여 소프트웨어를 시작합니다. 자세한 내용은 Amazon EKS에서 사용 가능한 Amazon EKS 추가 기능을 참조하세요.

a. 컨테이너 이미지 제공 옵션을 추가하려면 다음 단계를 수행합니다.

   i. 컨테이너 이미지에서, 제품 버전 소프트웨어가 포함된 컨테이너 이미지에 Amazon ECR URL을 추가합니다.

   ii. 제공 옵션 제목 및 배포 옵션 설명에 이 제공 옵션의 제목과 설명을 입력합니다.

   iii. 사용 지침에서 구매자가 소프트웨어를 시작한 후 사용하는 데 도움이 되는 세부 정보를 입력합니다.

   iv. 지원되는 서비스에서 구매자가 소프트웨어를 시작할 수 있는 환경을 선택합니다.

   v. 배포 템플릿에서 구매자가 소프트웨어를 시작하는 데 사용할 수 있는 리소스를 추가합니다. 각 템플릿의 리소스에 대한 제목과 URL을 입력합니다.

b. Helm 차트 제공 옵션을 추가하려면 다음 단계를 수행합니다.

   i. Helm 차트에서 구매자가 소프트웨어를 시작하기 위해 배포 환경에 설치할 Helm 차트에 Amazon ECR URL을 추가합니다.

   ii. 컨테이너 이미지에서, 제품 버전 소프트웨어가 포함된 컨테이너 이미지에 Amazon ECR URL을 추가합니다.

   iii. 제공 옵션 제목 및 배포 옵션 설명에 이 제공 옵션의 제목과 설명을 입력합니다.

   iv. 사용 지침에서 구매자가 소프트웨어를 시작한 후 사용하는 데 도움이 되는 세부 정보를 입력합니다.

   v. 지원되는 서비스에서 구매자가 소프트웨어를 시작할 수 있는 환경을 선택합니다.

   vi. 선택 사항 - Helm 릴리스 이름에 Helm 차트가 설치될 Kubernetes 네임스페이스의 이름을 입력합니다.

   vii. 선택 사항 - Helm 설치 네임스페이스에 `helm install` 명령에서 사용할 Helm 릴리스의 이름을 입력합니다.

viii. 선택 사항 - Kubernetes 서비스 계정 이름에 AWS Identity and Access Management (IAM)에 연결하는 데 사용할 Kubernetes 서비스 계정의 이름을 입력합니다. Kubernetes 서비스 계정은 라이선싱 또는 측정과 같은 AWS 서비스를 호출합니다.

ix. 이 제품 버전에서 QuickLaunch를 활성화하도록 선택합니다. 빠른 시작은의 기능입니다 AWS Marketplace. 구매자는 빠른 시작을 사용하여 Amazon EKS 클러스터를 빠르게 생성하고를 사용하여 클러스터에서 소프트웨어를 시작할 수 있습니다 AWS CloudFormation. 자세한 내용은 에서 빠른 시작 AWS Marketplace을 참조하세요.

x. 재정의 파라미터에서 소프트웨어를 시작하는 Helm CLI 명령에 사용할 파라미터를 입력합니다. 이러한 파라미터를 사용하면 제공된 기본값을 구매자가 재정의할 수 있습니다. 빠른 시작을 활성화한 경우 CloudFormation 양식의 파라미터 이름과 설명도 입력합니다. AWS Marketplace Management Console을 사용할 때는 파라미터가 15개로 제한되지만 를 사용할 때는 제한이 없습니다 AWS Marketplace Catalog API. 자세한 내용은 컨테이너 기반 제품에 새 버전 추가를 참조하세요.

> **Note**
>
> 일부 재정의 파라미터는 필수입니다. Amazon EKS Anywhere 제품은 라이선스 보안 암호에 대한 재정의 파라미터가 필요하며 `DefaultValue`는 `"${AWSMP_LICENSE_SECRET}"`입니다. 유료 제품의 경우 `DefaultValue`가 `"${AWSMP_SERVICE_ACCOUNT}"`인 서비스 계정 구성에 재정의 파라미터 하나를 제공해야 합니다.

xi. 콘솔, 명령줄 도구 및 API에서 민감한 정보를 숨기려면 암호 및 보안 암호 숨기기를 선택합니다. 자세한 내용은 AWS CloudFormation 사용 설명서의 파라미터에서 `NoEcho` 파라미터 설명서를 참조하세요.

c. Amazon EKS 콘솔 추가 기능 전송 옵션을 추가하려면 아티팩트가 Amazon EKS 추가 기능 제품에 대한 요구 사항를 준수하는지 확인한 후 다음 단계를 수행합니다.

> **Note**
>
> 버전당 하나의 Amazon EKS 추가 기능 전송 옵션만 지원됩니다. 작업 중인 현재 버전이 Amazon EKS 콘솔에 게시될 때까지 새 버전을 추가할 수 없습니다.

i. Helm 차트에서 구매자가 소프트웨어를 시작하기 위해 배포 환경에 설치할 Helm 차트에 Amazon ECR URL을 추가합니다.

ii. 컨테이너 이미지에서, 제품 버전 소프트웨어가 포함된 컨테이너 이미지에 Amazon ECR URL을 추가합니다. Helm 차트 내의 모든 이미지가 나열되었는지 확인합니다.

iii. 제공 옵션 제목 및 배포 옵션 설명에 이 제공 옵션의 제목과 설명을 입력합니다.

iv. 가시성에서 선택한 제한의 기본값을 유지합니다.

v. 추가 기능 이름에 이 추가 기능의 고유한 이름을 입력합니다. 입력하는 추가 기능 이름은 Amazon EKS 콘솔에 표시될 때 판매자 이름과 함께 추가됩니다.

vi. 추가 기능 버전에서 이 추가 기능을 설치하거나 업그레이드할 때 표시되는 추가 기능의 버전을 입력합니다. `major.minor.patch` 형식을 따릅니다.

vii. 추가 기능 유형의 드롭다운 목록에서 추가 기능의 범주를 선택합니다.

viii. Kubernetes 버전에서 추가 기능이 지원할 모든 Kubernetes 버전을 선택합니다.

ix. 아키텍처에서 추가 기능이 지원하는 플랫폼 아키텍처를 선택합니다. 옵션은 AMD64 및 ARM64입니다. 호환성을 극대화하려면 두 아키텍처를 모두 지원하는 것이 좋습니다. 추가 기능이 ARM64 디바이스를 지원하지 않는 경우, 모든 상용 AWS 리전에 제품을 게시하려면 먼저 지원 추가를 위한 계획된 날짜를 지정해야 합니다.

x. 네임스페이스에 추가 기능을 설치할 고유한 Kubernetes 네임스페이스를 입력합니다. `default`, `kube-system` 및 `kube-public` 네임스페이스는 타사 추가 기능 설치에 지원되지 않습니다.

xi. 환경 재정의 파라미터에서 Amazon EKS 추가 기능 프레임워크의 파라미터를 최대 2개 선택할 수 있습니다. values.yaml의 파라미터 이름을 `${AWS_REGION}` 및 `${AWS_EKS_CLUSTER_NAME}`인 환경 변수에 매핑할 수 있습니다.

7. 필요에 따라 새 제공 옵션을 선택하여 추가 제공 옵션을 추가하고 이전 단계의 지침에 따라 옵션을 구성합니다.

8. 제출을 선택합니다.

4단계: 버전 정보 업데이트

버전을 생성한 후에는 버전과 관련된 정보를 수정하여 구매자에게 업데이트된 정보를 제공하는 것이 좋습니다. 예를 들어 1.1 버전이 릴리스된 후 1.0 버전을 제한하려면 1.0 버전의 설명을 업데이트하여 구매자를 1.1 버전으로 안내하면 됩니다. 1.0 버전이 제한되는 날짜를 입력합니다. AWS Marketplace Management Portal에서 버전 정보를 업데이트합니다.

버전 정보를 업데이트하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.

2. 제품 메뉴에서 서버를 선택합니다.
3. 서버 제품 탭에서 수정하려는 제품을 선택합니다.
4. 변경 요청 드롭다운 메뉴에서 버전 정보 업데이트를 선택합니다.
5. 버전 업데이트 페이지에서 업데이트할 버전을 선택합니다.
6. 선택한 버전을 업데이트합니다. 제품 버전 또는 제공 옵션의 상태에 따라 업데이트할 수 있는 필드가 달라집니다.
   a. 모든 버전의 릴리스 정보를 업데이트할 수 있습니다.
   b. 아직 공개되지 않은 버전은 버전 제목을 업데이트할 수 있습니다.
   c. 제한되지 않은 제공 옵션의 경우 다음 필드를 업데이트할 수 있습니다.
      - 설명
      - 사용 지침
      - 지원되는 서비스
   d. 아직 공개되지 않은 버전의 제공 옵션은 다음 필드를 업데이트할 수 있습니다.
      - 제공 옵션 제목
      - Helm 차트(Helm 차트 제공 옵션만 해당)
      - 컨테이너 이미지
      - 배포 리소스
      - 애드온 이름
      - 애드온 버전
      - 애드온 유형
      - Helm 차트 URI
      - CompatibleKubernetesVersions
      - SupportedArchitectures
      - 네임스페이스
      - EnvironmentOverrideParameters
   e. 공개적으로 사용할 수 있는 버전의 전송 옵션의 경우 SupportedArchitectures를 업데이트할 수 있습니다.
7. 제출을 선택합니다.
8. 요청이 요청 탭에 표시되고 상태가 검토 중인지 확인합니다.

언제든지 서버 제품 페이지의 요청 탭에서 요청 상태를 확인할 수 있습니다.

Amazon EKS 추가 기능 버전 제한

추가 기능으로 게시된 컨테이너 제품 버전을 제한하려면 AWS Marketplace Management Portal 하단의 문의 양식을 통해 AWS Marketplace 운영 팀에 문의하세요.

AWS Marketplace에서 컨테이너 제품 요금 관리

AWS Marketplace 판매자는 컨테이너 제품 요금을 관리하고 업데이트할 수 있습니다. 무료 제품, Bring Your Own License(BYOL) 모델 제품 그리고 Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS) 및 AWS Fargate에 대한 유료 제품을 나열할 수 있습니다. 제품마다 설정할 수 있는 요금은 한 가지로 제한됩니다. 요금 차원을 추가하거나 업데이트할 수 있습니다. 요금 모델에 따라 계약 및 사용량 차원을 추가할 수 있습니다. 요금 조건 또는 환불 정책을 업데이트할 수도 있습니다. 자세한 내용은 다음 섹션을 참조하십시오.

AWS Marketplace의 컨테이너 제품 요금 모델에 대한 자세한 내용은 에 대한 컨테이너 제품 요금 AWS Marketplace 섹션을 참조하세요.

주제

- 요금 차원 추가
- 차원 정보 업데이트
- 요금 조건 업데이트
- 제품의 환불 정책 업데이트

요금 차원 추가

제품 요금 청구에 사용하는 요금 모델에 요금 차원을 추가할 수 있습니다. 요금 모델에 대한 자세한 내용은 컨테이너 요금 모델 섹션을 참조하세요.

> **Note**
>
> AWS Marketplace Management Portal에서는 소비량 요금이 적용되는 계약(예: 추가 사용량에는 사용한 만큼만 지불)에 요금 차원을 추가할 수 없습니다.
> 계약, 사용량, 소비량 요금이 적용되는 계약 사이에서 요금 모델을 변경할 수 없습니다. AWS Marketplace 판매자 운영 팀에 지원을 요청하세요.

1. https://aws.amazon.com/marketplace/management/tour/에서 AWS Marketplace Management Portal을 열고 판매자 계정에 로그인합니다.
2. 서버 제품 페이지에서 수정하려는 컨테이너 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 요금 차원 업데이트를 선택하고 요금 차원 추가를 선택합니다.
4. 요금 모델에 따라 API 식별자, 표시 이름 및 설명을 제공하여 계약 차원 또는 사용량 차원을 추가할 수 있습니다.
5. 다음을 선택하고 계약 차원 요금을 입력합니다.
6. 제출을 선택하여 검토 요청을 제출합니다.
7. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

차원 정보 업데이트

제품의 차원 정보를 변경할 수 있습니다. 요금 모델에 대한 자세한 내용은 컨테이너 요금 모델 섹션을 참조하세요.

1. https://aws.amazon.com/marketplace/management/tour/에서 AWS Marketplace Management Portal을 열고 판매자 계정에 로그인합니다.
2. 서버 제품 페이지에서 수정하려는 컨테이너 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 요금 차원 업데이트를 선택하고 차원 정보 업데이트를 선택합니다.
4. 업데이트하려는 차원의 표시 이름 및 설명을 제공하여 차원 정보를 추가할 수 있습니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

요금 조건 업데이트

제품의 요금 조건을 변경할 수 있습니다. 제품이 제공되는 국가를 변경할 때 요금 조건을 업데이트해야 할 수도 있습니다.

1. https://aws.amazon.com/marketplace/management/tour/에서 AWS Marketplace Management Portal을 열고 판매자 계정에 로그인합니다.
2. 서버 제품 페이지에서 수정하려는 컨테이너 제품을 선택합니다.
3. 변경 요청 목록에서 공개 제안 업데이트를 선택하고 요금 조건 업데이트를 선택합니다.

4. 현재 요금이 편집할 수 있도록 미리 입력되어 있습니다. 현재 요금을 삭제하고 새 요금을 추가하면 됩니다. 요청하려는 요금을 검토한 후 요청을 제출하는 것이 좋습니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

> **Note**
>
> 요금 조건 업데이트를 사용하여 차원의 요금을 인상하면 90일 동안 요금을 업데이트할 수 없습니다. 요금을 인상하면 요금을 변경한 날로부터 90일 동안 가격이 동결됩니다. 이 가격 동결은 가격 인상에만 유효하며 가격 인하에는 적용되지 않습니다.

제품의 환불 정책 업데이트

제품의 환불 정책을 업데이트할 수 있습니다. 업데이트된 환불 정책은 모든 사용자에게 적용됩니다. 자세한 내용은 에서 제품 환불 AWS Marketplace 단원을 참조하십시오.

1. https://aws.amazon.com/marketplace/management/tour/에서 AWS Marketplace Management Portal을 열고 판매자 계정에 로그인합니다.
2. 서버 제품 페이지에서 수정하려는 컨테이너 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 공개 제안 업데이트를 선택하고 환불 정책 업데이트를 선택합니다.
4. 현재 환불 정책이 편집할 수 있도록 미리 입력되어 있습니다. 현재 환불 정책을 삭제하고 새 환불 정책을 추가할 수 있습니다. 요청하려는 환불 정책을 검토한 후 요청을 제출하는 것이 좋습니다. 요청을 제출하면 현재 환불 정책을 덮어씁니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

AWS Marketplace에서 국가별 컨테이너 제품 가용성 업데이트

AWS Marketplace 판매자는 컨테이너 제품을 제공하고 구독할 수 있는 국가를 변경할 수 있습니다. 자세한 내용은 국가 단원을 참조하십시오. 다음 절차에서는 국가별 컨테이너 제품 가용성을 업데이트하는 방법을 보여줍니다.

1. https://aws.amazon.com/marketplace/management/tour/에서 AWS Marketplace Management Portal을 열고 판매자 계정에 로그인합니다.
2. 서버 제품 페이지에서 수정하려는 컨테이너 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 공개 제안 업데이트를 선택하고 국가별 가용성 업데이트를 선택합니다.
4. 다음 옵션 중 하나를 선택합니다.
   - 모든 국가 - 지원되는 모든 국가에서 사용할 수 있습니다.
   - 일부 국가를 제외한 모든 국가 - 일부 국가를 제외하고 지원되는 모든 국가에서 사용할 수 있습니다.
   - 허용 목록에 있는 국가만 - 판매자가 지정한 국가의 구매자만 제품을 구매할 수 있습니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

에서 컨테이너 제품에 대한 최종 사용자 라이선스 계약(EULA) 업데이트 AWS Marketplace

AWS Marketplace 판매자는 최종 사용자 라이선스 계약(EULA)을 업데이트하여 에 대한 표준 계약 AWS Marketplace 또는 사용자 지정 EULA를 사용할 수 있습니다. 업데이트된 EULA는 제품을 구독하는 신규 사용자와 제품 갱신에 적용됩니다. 데이터 세트를 업데이트하려면 다음 절차를 수행합니다.

1. https://aws.amazon.com/marketplace/management/tour/ AWS Marketplace Management Portal를 열고 판매자 계정에 로그인합니다.
2. 서버 제품 페이지에서 수정하려는 컨테이너 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 공개 제안 업데이트를 선택하고 EULA 업데이트를 선택합니다.
4. 에 대한 표준 계약을 AWS Marketplace 선택하거나 사용자 지정 EULA를 제출합니다. 사용자 지정 EULA의 경우 Amazon S3 버킷의 계약을 제공해야 합니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

AWS Marketplace에서 컨테이너 제품 테스트 및 릴리스

AWS Marketplace에 새 버전의 컨테이너 제품을 추가한 후 제품을 테스트한 다음 일반에 릴리스할 수 있습니다. 이 주제에서는 제품을 테스트하고 일반에 릴리스하는 데 필요한 특정 단계와 프로세스를 간략하게 설명합니다.

주제

- 컨테이너 이미지 및 Helm 차트 전송 옵션
- Amazon EKS 추가 기능 전송 옵션

컨테이너 이미지 및 Helm 차트 전송 옵션

이 섹션에서는 컨테이너 이미지 및 Helm 차트 릴리스에 대한 지침을 제공합니다.

새 버전 요청이 생성되고 몇 분 내에 완료됩니다. 서버 제품 페이지의 요청 탭에서 요청을 추적할 수 있습니다. 추가 기능을 테스트하거나 릴리스할 때 오류가 발생하는 경우 AWS Marketplace Catalog API 참조의 새 버전 추가에서 비동기 오류 테이블을 참조하세요.

> **Note**
>
> 현재 제품이 제한된 가용성으로 설정된 경우 제품을 사용할 수 있는 구매자만 제품 버전에 액세스할 수 있습니다. 현재 제품이 공개로 설정된 경우 모든 AWS Marketplace 구매자가 제품 버전에 액세스할 수 있습니다.

이 버전이 첫 번째 버전이면 이제 제품을 게시할 준비가 된 것입니다.

Amazon EKS 추가 기능 전송 옵션

이 섹션에서는 Amazon EKS 추가 기능 테스트 및 릴리스에 대한 지침을 제공합니다.

추가 기능 테스트

- 추가 기능을 제출하면 AWS Marketplace는 요청을 처리하고 Amazon EKS 추가 기능 카탈로그에서 검증할 수 있도록 제한된 상태로 추가 기능을 게시합니다. AWS Marketplace Management Portal의 서버 제품 페이지의 요청 탭에서 요청을 추적할 수 있습니다. 수집 시간은 처리 중인 요청의 양에 따라 영업일 기준 5~10일 정도입니다.

요청이 검토 중 상태인 경우 AWS Marketplace의 AWS 팀이 Amazon EKS 추가 기능 카탈로그에 추가 기능을 계속 게시하고 있습니다. 추가 기능이 제한 상태로 게시되면 상태 변경을 성공으로 요청합니다. 이후 추가 기능 테스트를 시작할 수 있습니다.

- 추가 기능을 사용할 수 있게 되면 테스트 목적으로 아시아 태평양(서울) 리전에서 찾을 수 있습니다. AWS Marketplace는 소프트웨어 기능을 확인하기 위해 전문 지식을 신뢰합니다. 추가 기능을 테스트하려면 허용 목록에 추가 기능이 있는 판매자 계정의 아시아 태평양(서울) 리전에 Amazon EKS 클러스터를 생성해야 합니다. 추가 기능을 테스트하려면 다음 세부 지침을 따르세요. 소프트웨어가 지원하는 각 Kubernetes 버전에서 테스트해야 합니다.
- 유료 제품을 제공하는 경우 다음 내부 AWS 계정에 대한 비공개 제안을 생성합니다. 이러한 계정은 모든 상용 AWS 리전의 Amazon EKS 콘솔에 소프트웨어를 통합하는 데 도움이 됩니다.

```
288092140294, 288092140294, 408202761791
```

- AWS Marketplace가 추가 기능 버전을 승인하고 퍼블릭으로 이동할 때까지 추가 기능이 있는 테스트 클러스터를 활성 상태로 유지합니다.

> **Note**
>
> AWS Marketplace는 Amazon EKS 클러스터에서 컨테이너 제품을 테스트하는 동안 발생한 AWS 인프라 비용을 부담하지 않습니다. 올바른 크기 조정 메커니즘을 따라 테스트 결과를 확인하는 동안 노드를 최소 운영 비용으로 톤다운할 수 있습니다.

추가 기능을 퍼블릭 릴리스

Amazon EKS 클러스터를 통해 소프트웨어를 추가 기능으로 검증한 후 AWS Marketplace Management Portal 또는 AWS Marketplace Catalog API를 사용하여 Amazon EKS 추가 기능 버전을 퍼블릭으로 릴리스하라는 요청을 제출할 수 있습니다.

자세한 내용은 AWS Marketplace Catalog API 참조의 Update the visibility for an Amazon EKS add-on을 참조하세요.

AWS Marketplace Management Portal의 서버 제품 페이지의 요청 탭에서 요청을 추적할 수 있습니다. 수집 시간은 다를 수 있습니다.

에 대한 컨테이너 기반 제품 요구 사항 AWS Marketplace

AWS Marketplace 는의 모든 컨테이너 기반 제품 및 제품에 대해 다음 요구 사항을 유지합니다 AWS Marketplace. 이러한 요구 사항은 고객을 위해 안전하고 신뢰할 수 있는 디지털 카탈로그를 홍보하는 데 도움이 됩니다. 또한 판매자는 특정 제품의 요구 사항을 충족할 수 있도록 추가 컨트롤 및 프로토콜 구현을 검토하는 것이 좋습니다.

제출 시 모든 제품 및 관련 메타데이터를 검토하여 현재 AWS Marketplace 정책을 충족하거나 초과하는지 확인합니다. 이러한 정책은 진화하는 보안 지침에 따라 정기적으로 업데이트됩니다.는 제품을 AWS Marketplace 지속적으로 스캔하여 기존 목록이 이러한 요구 사항에 대한 변경 사항을 계속 충족하는지 확인합니다. 제품이 규정을 준수하지 않는 경우 AWS Marketplace 는 판매자에게 문의하여 새 표준을 충족하도록 제품을 업데이트합니다. 경우에 따라 문제가 해결될 때까지 신규 구독자가 제품을 일시적으로 사용할 수 없게 될 수 있습니다. 이 프로세스는 모든 사용자에 대해 AWS Marketplace 플랫폼의 보안 및 신뢰성을 유지하는 데 도움이 됩니다.

주제

- 보안 정책
- 고객 정보 요구 사항
- 제품 사용 요구 사항
- 아키텍처 요구 사항
- 컨테이너 제품 사용 지침
- Amazon EKS 추가 기능 제품에 대한 요구 사항

보안 정책

모든 컨테이너 기반 제품은 다음과 같은 보안 요구 사항을 준수해야 합니다.

- 컨테이너 이미지에는 알려진 취약성, 맬웨어 또는 End-of-Life(EoL) 소프트웨어 패키지 및 운영 체제가 포함되어서는 안 됩니다.
- 컨테이너는 AWS 서비스에 액세스하기 위해 AWS 자격 증명을 요청해서는 안 됩니다. 제품이 AWS 서비스에 액세스해야 하는 경우 다음 중 하나를 사용해야 합니다.
  - Amazon Elastic Kubernetes Service(Amazon EKS) 워크로드에 대한 서비스 계정의 IAM 역할입니다.
  - 작업에 대한 IAM 역할, Amazon Elastic Container Service(Amazon ECS) 워크로드에 대한 IAM 역할.

- 컨테이너 기반 제품을 실행하려면 최소 권한만 있으면 됩니다. 자세한 내용은 Amazon Elastic Container Service의 보안 및 Amazon EKS의 보안을 참조하세요.
- 컨테이너 이미지는 기본적으로 루트가 아닌 권한으로 실행되도록 구성해야 합니다.
- 컨테이너에는 시스템 사용자 및 서비스의 암호(해시된 암호 포함), 프라이빗 키, 자격 증명 등과 같은 하드 코딩된 보안 암호가 포함되어서는 안 됩니다.
- 컨테이너 내에서 실행되는 서비스의 인증은 시작 시 사용자가 암호를 생성, 재설정 또는 정의하더라도 암호 기반 인증을 사용해서는 안 됩니다. Null 암호와 빈 암호도 허용되지 않습니다.
- 컨테이너 이미지에는 지원되지 않는 아키텍처(예: in-to Attestation Framework 메타데이터)가 있는 계층이 포함되어서는 안 됩니다.

고객 정보 요구 사항

모든 컨테이너 기반 제품은 다음과 같은 고객 정보 요구 사항을 준수해야 합니다.

- 소프트웨어는 기존 보유 라이선스 사용(BYOL)에서 요구하는 경우를 제외하고 고객의 인지와 분명한 동의 없이 고객 데이터를 수집하거나 내보내면 안 됩니다. 고객 데이터를 수집하거나 내보내는 애플리케이션은 다음 지침을 준수해야 합니다.
  - 고객 데이터 수집은 셀프 서비스이고, 자동화되고, 안전해야 합니다. 구매자는 판매자가 소프트웨어 배포를 승인할 때까지 기다릴 필요가 없어야 합니다.
  - 고객 데이터 수집은 Marketplace 이용 약관 AWS, AWS 서비스 약관, AWS 개인 정보 보호 고지 및 AWS 고객 계약을 포함하되 이에 국한되지 않는 와의 계약과 일치해야 합니다. AWS
  - 결제 정보를 수집하면 안 됩니다.

제품 사용 요구 사항

모든 컨테이너 기반 제품은 다음과 같은 제품 사용 요구 사항을 준수해야 합니다.

- 판매자는 완전히 작동하는 제품만 등록할 수 있습니다. 시험 또는 평가 목적의 베타 또는 프리릴리스 제품은 허용되지 않습니다. 판매자가 무료 버전을 제공한 후 90일 AWS Marketplace 이내에에 동등한 유료 버전을 제공하는 경우 상용 소프트웨어의 개발자, 커뮤니티 및 BYOL 에디션이 지원됩니다.
- 모든 컨테이너 기반 제품의 사용 지침에는 컨테이너 기반 제품을 배포하는 모든 단계가 포함되어야 합니다. AWS Marketplace의 해당 컨테이너 이미지를 가리키는 명령 및 배포 리소스를 사용 지침에서 제공해야 합니다.

- 컨테이너 기반 제품에는 구독자가 소프트웨어를 사용하는 데 필요한 모든 컨테이너 이미지가 포함되어야 합니다. 또한 컨테이너 기반 제품은 사용자가 외부의 이미지 AWS Marketplace (예: 타사 리포지토리의 컨테이너 이미지)를 사용하여 제품을 시작할 필요가 없어야 합니다.
- 컨테이너와 해당 소프트웨어는 셀프 서비스 방식으로 배포할 수 있어야 하며 추가 결제 방법이나 비용이 필요 없어야 합니다. 배포 시 외부 종속성이 필요한 애플리케이션은 다음 지침을 준수해야 합니다.
  - 요구 사항은 목록의 설명 또는 사용 지침에 명시되어야 합니다. 예: 이 제품을 올바르게 배포하려면 인터넷 연결이 필요합니다. 배포 시 <패키지 목록> 패키지가 다운로드됩니다.
  - 판매자는 모든 외부 종속성의 사용과 가용성 및 보안에 대한 책임이 있습니다.
  - 외부 종속성을 더 이상 사용할 수 없는 경우 제품 AWS Marketplace 도에서 제거해야 합니다.
  - 외부 종속성은 추가 결제 방법이나 비용이 필요 없어야 합니다.
- 구매자가 직접 관리하지 않는 외부 리소스(예: 외부 API 또는 판매자나 타사가 관리하는 AWS 서비스)에 지속적으로 연결해야 하는 컨테이너는 다음 지침을 준수해야 합니다.
  - 요구 사항은 목록의 설명 또는 사용 지침에 명시되어야 합니다. 예: 이 제품은 지속적인 인터넷 연결이 필요합니다. 정상적으로 작동하려면 <리소스 목록>과 같은 지속적인 외부 서비스가 필요합니다.
  - 판매자는 모든 외부 리소스의 사용과 가용성 및 보안에 대한 책임이 있습니다.
  - 외부 리소스를 더 이상 사용할 수 없는 경우 제품 AWS Marketplace 도에서 제거해야 합니다.
  - 외부 리소스는 추가 결제 방법이나 비용이 필요 없어야 하며 연결 설정이 자동화되어야 합니다.
- AWS Marketplace에서 사용할 수 없는 업셀 서비스, 추가 제품 또는 다른 클라우드 플랫폼으로 사용자를 유도하는 언어가 제품 소프트웨어 및 메타데이터에 포함되면 안 됩니다.
- 제품이 다른 제품이나 다른 ISV 제품의 추가 기능인 경우 제품 설명에 해당 제품은 다른 제품의 기능을 확장한다는 내용과 다른 제품이 없으면 해당 제품의 유용성이 매우 제한된다는 내용을 명시해야 합니다. 예: 이 제품은 <제품 이름>의 기능을 확장하며, <제품 이름> 제품이 없으면 이 제품의 유용성이 매우 제한됩니다. 이 목록의 모든 기능을 사용하려면 <제품 이름>의 자체 라이선스가 필요할 수 있습니다.

아키텍처 요구 사항

모든 컨테이너 기반 제품은 다음과 같은 아키텍처 요구 사항을 준수해야 합니다.

- 에 대한 소스 컨테이너 이미지는가 소유한 Amazon Elastic Container Registry(Amazon ECR) 리포지토리로 푸시해야 AWS Marketplace 합니다 AWS Marketplace. 컨테이너 제품 목록마다 AWS Marketplace Management Portal 의 서버 제품 아래에서 이러한 리포지토리를 생성할 수 있습니다.

- 컨테이너 이미지는 Linux 기반이어야 합니다.
- 유료 컨테이너 기반 제품은 Amazon ECS, Amazon EKS 또는 AWS Fargate에 배포할 수 있어야 합니다.
- 계약 요금이 적용되고와 통합되는 유료 컨테이너 기반 제품은 Amazon EKS, Amazon ECS, AWS Fargate Amazon EKS Anywhere, Amazon ECS Anywhere, Red Hat OpenShift Service on AWS (ROSA), 자체 관리형 Kubernetes 클러스터 온프레미스 또는 Amazon Elastic Compute Cloud에 배포 AWS License Manager 해야 합니다.

컨테이너 제품 사용 지침

컨테이너 제품에 대한 사용 지침을 생성할 때에는 the section called “AMI 및 컨테이너 제품 사용 지침”의 단계와 지침을 따릅니다.

Amazon EKS 추가 기능 제품에 대한 요구 사항

Amazon EKS 추가 기능은 Kubernetes 애플리케이션에 운영 기능을 제공하는 소프트웨어이지만 해당 애플리케이션에 국한되지 않습니다. 예를 들어 Amazon EKS 추가 기능에는 클러스터가 네트워킹, 컴퓨팅 및 스토리지를 위한 기본 AWS 리소스와 상호 작용할 수 있도록 하는 관찰성 에이전트 또는 Kubernetes 드라이버가 포함됩니다.

컨테이너 제품의 판매자는 Amazon EKS를 비롯한 여러 배포 옵션 중에서 선택할 수 있습니다. Amazon EKS AWS Marketplace 추가 기능 카탈로그에 추가 기능으로 제품 버전을 게시할 수 있습니다. 추가 기능은 AWS 및 다른 공급업체가 유지 관리하는 추가 기능 옆에 있는 Amazon EKS 콘솔에 나타납니다. 구매자는 다른 추가 기능과 마찬가지로 소프트웨어를 추가 기능으로 쉽게 배포할 수 있습니다.

자세한 내용을 알아보려면 Amazon EKS 사용 설명서의 Amazon EKS 추가 기능을 참조하세요.

컨테이너 제품을 AWS Marketplace 추가 기능으로 준비

컨테이너 제품을 AWS Marketplace 추가 기능으로 게시하려면 다음 요구 사항을 충족해야 합니다.

- 컨테이너 제품을에 게시해야 합니다 AWS Marketplace.
- 컨테이너 제품은 AMD64 및 ARM64 아키텍처와 호환되도록 빌드되어야 합니다.
- 컨테이너 제품은 기존 보유 라이선스 사용(BYOL) 요금 모델을 사용하면 안 됩니다.

> **Note**
>
> BYOL은 Amazon EKS 추가 기능 전송에 지원되지 않습니다.

- 모든 컨테이너 이미지와 차트를 관리형 Amazon ECR 리포지토리로 푸시하는 등 모든 컨테이너 기반 제품 요구 사항을 준수해야 합니다. Helm AWS Marketplace 이 요구 사항에는 오픈 소스 이미지(예: `nginx`)도 포함됩니다. 이미지와 차트는 Amazon ECR 퍼블릭 갤러리, Docker Hub 및 Quay를 포함한(이에 국한되지 않음) 다른 외부 리포지토리에 호스팅할 수 없습니다.
- Helm 차트 - 소프트웨어를 Helm 차트로 준비하고 패키징합니다. Amazon EKS 추가 기능 프레임워크는 Helm 차트를 Kubernetes 매니페스트로 변환합니다. Amazon EKS 시스템에서는 일부 Helm 기능이 지원되지 않습니다. 다음 목록은 소프트웨어를 Amazon EKS 추가 기능으로 온보딩하기 전에 충족해야 하는 요구 사항을 설명합니다. 이 목록에서 모든 Helm 명령은 Helm 버전 3.8.1을 사용합니다.
  - `.APIVersions`를 제외한 모든 `Capabilities` 객체가 지원됩니다. `.APIVersions`는 기본이 아닌 사용자 지정 Kubernetes API에서는 지원되지 않습니다.
  - `Release.Name` 및 `Release.Namespace` 객체만 지원됩니다.
  - Helm 후크와 `lookup` 함수는 지원되지 않습니다.
  - 모든 종속 차트는 기본 Helm 차트(리포지토리 경로 file://...로 지정됨) 내에 있어야 합니다.
  - Helm 차트는 오류 없이 Helm Lint 및 Helm Template을 성공적으로 전달해야 합니다. 명령은 다음과 같습니다.
    - Helm Lint - `helm lint` *`helm-chart`*

      일반적인 문제에는 상위 차트의 메타데이터에 선언되지 않은 차트가 포함됩니다. 예: `chart metadata is missing these dependencies: chart-base Error: 1 chart(s) linted, 1 chart(s) failed`
    - Helm Template – `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`*

      `–f` 플래그로 재정의된 모든 구성을 전달합니다.
- 모든 컨테이너 바이너리를 AWS Marketplace Amazon ECR 리포지토리에 저장합니다. 매니페스트를 생성하려면 앞에 표시된 Helm 템플릿 명령을 사용합니다. 매니페스트에서 busybox 또는 `gcr` 이미지와 같은 외부 이미지 참조를 검색합니다. 요청 드롭다운에서 리포지토리 추가 옵션을

사용하여 생성된 AWS Marketplace Amazon ECR 리포지토리에 종속성과 함께 모든 컨테이너 이미지를 업로드합니다.

- 사용자 지정 구성 - 배포 중에 사용자 지정 변수를 추가할 수 있습니다. 최종 사용자 경험을 식별하는 방법, 소프트웨어 이름을 `aws_mp_configuration_schema.json`으로 지정하는 방법, Helm 차트가 있는 래퍼로 패키징하는 방법에 대한 자세한 내용은 Amazon EKS 추가 기능: 고급 구성을 참조하세요.

  '$schema' 키워드에 따라 `$schema`는 유효한 `application/schema+json` 리소스를 가리키는 URI여야 합니다.

  이 파일은 암호, 라이선스 키 및 인증서와 같은 민감한 정보를 수락해서는 안 됩니다.

  보안 암호 및 인증서 설치를 처리하려면 최종 사용자에게 추가 기능 설치 후 또는 설치 전 단계를 제공할 수 있습니다. 제품은 외부 라이선스에 의존해서는 안 됩니다. 제품은 AWS Marketplace 권한에 따라 작동해야 합니다.

  `aws_mp_configuration_schema.json`의 제한에 대한 자세한 내용은 추가 기능 공급자에 대한 추가 기능 구성 요구 사항 및 모범 사례 섹션을 참조하세요.
- 소프트웨어가 배포될 네임스페이스 식별 및 만들기 - 제품의 첫 번째 릴리스에서는 템플릿화된 네임스페이스를 추가하여 소프트웨어가 배포될 네임스페이스를 식별해야 합니다.
- 사용자 지정 리소스 정의(CRDs) - Amazon EKS 추가 기능 프레임워크는 동일한 추가 기능으로 적용된 CRDs를 기반으로 하는 사용자 지정 리소스 선언 및 CRDs 설치를 지원하지 않습니다. 추가 기능에 사용자 지정 리소스가 있고 CRDs를 사용하는 경우 다음 중 하나를 수행할 수 있습니다.
  - 두 개의 추가 기능 게시: CRD 정의를 별도의 추가 기능(별도의 헬름 차트)으로 분할하고 실제 사용자 지정 리소스 설치를 별도의 추가 기능에 게시합니다.
  - 추가 수동 지침이 포함된 단일 추가 기능 게시: 클러스터에 CRDs를 설치하는 단일 추가 기능을 게시합니다. 최종 사용자가 해당 CRDs에 의존하는 사용자 지정 리소스를 설정할 수 있도록 kubernetes 매니페스트 파일과 함께 사용 지침을 제공합니다.
- 해당하는 **`serviceAccount`** 경우 생성 - 소프트웨어가의 유료 소프트웨어 AWS Marketplace 이거나 다른에 연결해야 하는 경우 Helm 차트가 `serviceAccount` 기본적으로를 생성하는지 AWS 서비스확인합니다. `serviceAccount` 만들기가 `values.yaml` 파일의 파라미터로 처리되는 경우 파라미터 값을 `true`로 설정합니다. 예를 들어 `serviceAccount.create = true`입니다. 이것이 필요한 이유는 고객이 이미 필요한 권한이 있는 기본 노드 인스턴스에서 권한을 상속하여 추가 기능을 설치하도록 선택할 수 있기 때문입니다. 헬름 차트가 `serviceAccount`를 만들지 않으면 권한을 `serviceAccount`에 연결할 수 없습니다.

- 추적 가능한 배포 또는 DaemonSet - 헬름 차트에 DaemonSet 또는 배포가 있는지 확인합니다. Amazon EKS 애드온 프레임워크는 이를 사용하여 Amazon EKS 리소스의 배포를 추적합니다. 추적 가능한 배포 또는 DaemonSet이 없으면 애드온에 배포 오류가 발생합니다. 추가 기능에 배포 또는 DaemonSet이 없는 경우, 예를 들어 애드온에 추적 불가능한 사용자 지정 리소스 또는 Kubernetes 작업이 많이 배포되는 경우 더미 배포 또는 DaemonSet 객체를 추가합니다.
- AMD 및 ARM 아키텍처 지원 - 오늘날 많은 Amazon EKS 고객이 ARM64를 사용하여 AWS Graviton 인스턴스를 사용합니다. 타사 소프트웨어는 두 아키텍처를 모두 지원해야 합니다.
- 의 라이선스 또는 측정 APIs와 통합 AWS Marketplace -는 여러 결제 모델을 AWS Marketplace 지원합니다. 자세한 내용은 컨테이너 제품 결제, 측정 및 라이선스 통합 단원을 참조하십시오. PAYG 메커니즘을 통해 제품을 판매하려면 AWS Marketplace Metering Service를 사용하여 컨테이너 제품의 사용자 지정 측정 구성 섹션을 참조하세요. 선결제 또는 계약 모델을 통해 제품을 판매하려면 를 사용한 컨테이너 제품의 계약 요금 AWS License Manager 섹션을 참조하세요.
- 소프트웨어와 모든 아티팩트 및 종속성 업로드 - 헬름 차트는 자체적으로 포함되어야 하며 GitHub와 같은 외부 소스의 종속성이 필요하지 않아야 합니다. 소프트웨어에 외부 종속성이 필요한 경우 종속성을 동일한 AWS Marketplace 목록의 AWS Marketplace 프라이빗 Amazon ECR 리포지토리로 푸시해야 합니다.
- 웹 사이트에 배포 지침 제공 - 고객이 create-addon 명령을 통해 소프트웨어를 배포하는 방법을 식별할 수 있도록 배포 안내서를 호스팅할 것을 요청합니다.
- 추가 기능 권한/IAM 역할 -에서 게시된 추가 기능에 서비스에 대한 액세스가 AWS Marketplace AWS 필요한 경우 소프트웨어에는 서비스에 액세스하기 위한 IAM 정책으로 주석이 달린 Kubernetes AWS 서비스 계정이 있어야 합니다. 서비스 계정의 두 가지 옵션 중에서 선택하여 서비스에 API를 요청할 수 있습니다 AWS .
  - IRSA를 통한 자격 증명:이 옵션을 사용하면 소프트웨어가 Identity and Access Management(IAM) Role Service(IRSA)에서 수임 자격 증명을 얻을 수 있습니다. 자세한 내용은 서비스 계정의 IAM 역할을 참조하세요.
  - Amazon EKS 포드 자격 증명:이 옵션을 사용하면 소프트웨어가 Amazon EKS 포드의 포드 자격 증명을 사용하여 AWS 서비스에 API를 요청할 수 있습니다. 자세한 내용은 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

  추가 기능에는 Helm 차트의 `aws_mp_addon_parameters.json` 최상위 수준, 현재 사용자 지정 구성 스키마()와 동일한 디렉터리에 라는 추가 구성 파일이 있어야 합니다`aws_mp_configuration_schema.json`. 현재이 파일은 포드 자격 증명 호환 권한만 처리합니다. 파일 형식은 다음과 같습니다.

```
{
  "permissions": {
```

```
      "isPodIdentityCompatible" : true,
      "permissionsList": [
       {
        "serviceAccount" : "String",
        "managedPolicies" : ["Policy Arn"],
       }
     ]
    }
  }
```

파일 이름: **`aws_mp_addon_parameters.json`**

> **Note**
>
> `aws_mp_addon_parameters.json` 파일은 Amazon EKS 콘솔의 추가 기능 구성 설정 페이지에서 추가 기능 액세스 섹션을 활성화합니다.

필드 이름	유형	Notes	예시 값
isPodIdentityCompatible	불	지금은 `true`만 지원됩니다. 필드는 다음 permissionsList 목록에 설명된 권한이 pod-identity와 일치하는지 여부를 보여줍니다.	TRUE
serviceAccount	String	추가 기능이 권한에 액세스하는 데 사용할 서비스 계정의 이름입니다.	`kpow`
managedPolicies	목록<문자열>	EKS 추가 기능에서 수임할 수 있는이 서비스 계정에 사용할 정책 ARN 목록	`["arn:aws:iam::aws:policy/ReadOnlyAccess"]`

> **Note**
>
> 의 Pay-as-you-go(PAYG) 추가 기능 제품은 Amazon EKS Pod Identity를 사용할 AWS Marketplace 수 없으며 액세스 제어를 위해 서비스 계정에 대한 IAM 역할(IRSA)을 사용해야 합니다.

- 버전 업데이트 - Amazon EKS는 업스트림 릴리스 몇 주 후에 새 Kubernetes 버전을 릴리스합니다. 새로운 Amazon EKS 클러스터 버전이 일반적으로 제공됨에 따라 공급업체는 45일 이내에 새 Amazon EKS 클러스터 버전 릴리스와 호환되도록 소프트웨어를 인증하거나 업데이트해야 합니다. 추가 기능의 현재 버전이 새 Kubernetes 버전을 지원하는 경우 버전 호환성 매트릭스를 업데이트할 수 있도록 동일한 버전을 검증하고 인증합니다. 새 Kubernetes 버전 릴리스를 지원하기 위해 새 추가 기능 버전이 필요한 경우 온보딩을 위해 새 버전을 제출하세요.
- 파트너의 소프트웨어는 다음 유형 중 하나에 해당하거나 Kubernetes 또는 Amazon EKS를 향상시키는 운영 소프트웨어여야 합니다. Gitops | monitoring | logging | cert-management | policy-management | cost-management | autoscaling | storage | kubernetes-management | service-mesh | etcd-backup | ingress-service-type | load-balancer | local-registry| networking | Security | backup | ingress-controller | observability
- 소프트웨어는 컨테이너 네트워크 인터페이스(CNI)일 수 없습니다.
- 소프트웨어는 유료 제품의 라이선싱 AWS Marketplace 및 측정 APIs를 통해 판매되고 통합되어야 합니다. BYOL 제품은 허용되지 않습니다.

추가 기능 공급자에 대한 추가 기능 구성 요구 사항 및 모범 사례

Amazon EKS는 추가 기능 공급자의 Helm JSON 스키마 문자열로 구성해야 합니다. 필수 구성이 필요하거나 선택적 구성을 허용하는 추가 기능에는 Helm 차트가 제출된 `aws_mp_configuration_schema.json` 파일이 포함되어야 합니다 AWS Marketplace. Amazon EKS는 이 스키마를 사용하여 고객의 구성 입력을 검증하고 스키마를 준수하지 않는 입력 값이 있는 API 직접 호출을 거부합니다. 추가 기능 구성은 일반적으로 두 가지 범주로 분류됩니다.

- 레이블, 허용 오차, nodeSelector 등과 같은 일반 Kubernetes 속성에 대한 구성
- 라이선스 키, 기능 활성화, URL 등과 같은 추가 기능별 구성입니다.

이 섹션에서는 일반 Kubernetes 속성과 관련된 첫 번째 범주에 중점을 둡니다.

Amazon EKS는 Amazon EKS 추가 기능 구성에 대한 모범 사례를 따를 것을 권장합니다.

- 스키마 요구 사항
- 구성에 허용되는 공통 파라미터
- 구성에 허용되지 않는 공통 파라미터

스키마 요구 사항

JSON 스키마를 정의할 때 Amazon EKS 추가 기능에서 지원하는 JSON 스키마 버전을 사용해야 합니다.

지원되는 스키마 목록:

- 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

다른 JSON 스키마 버전을 사용하는 것은 Amazon EKS 추가 기능과 호환되지 않으며 이 문제가 해결될 때까지 추가 기능을 릴리스할 수 없습니다.

예제 헬름 스키마 파일

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

카멜케이스

구성 파라미터는 카멜케이스여야 하며 이 형식을 준수하지 않으면 거부됩니다.

설명 필수

스키마 속성에 대한 의미 있는 설명은 항상 포함하세요. 이 설명은 각 구성 파라미터에 대해 Amazon EKS 콘솔에서 레이블 이름을 렌더링하는 데 사용됩니다.

RBAC 정의

추가 기능 공급자는 최소 권한 원칙을 사용하여 추가 기능을 성공적으로 설치하는 데 필요한 RBAC 권한을 정의하고 제공해야 합니다. 최신 버전의 추가 기능 또는 CVE를 해결하기 위한 수정 사항에 대해 RBAC 권한을 변경해야 하는 경우 추가 기능 공급자는 Amazon EKS 팀에 이 변경 사항을 알려야 합니다. 각 Kubernetes 리소스에 필요한 권한은 객체의 리소스 이름으로 제한되어야 합니다.

```
apiGroups: ["apps"]
resources: ["daemonsets"]
resourceNames: ["ebs-csi-node"]
verbs: ["create", "delete", "get", "list", "patch", "update", "watch"]
```

보안 암호 관리

이 섹션은 고객이 애플리케이션 키, API 키, 암호 등의 비밀 정보를 구성해야 하는 추가 기능에만 적용됩니다. 현재 Amazon EKS API 보안 영향으로 인해 일반 텍스트로 비밀 정보를 전달하는 것은 지원하지 않습니다. 그러나 고객은 구성을 사용하여 추가 기능에 필요한 키를 포함하는 Kubernetes 보안 암호의 이름으로 전달할 수 있습니다. 고객은 사전 필수 단계와 동일한 네임스페이스를 가진 키를 포함하는 Kubernetes 보안 암호 객체를 만든 다음, 추가 기능을 만들 때 구성 블롭을 사용하여 보안 암호의 이름을 전달해야 합니다. 고객이 실수로 스키마 속성을 실제 키로 오인하지 않도록 하려면 추가 기능 공급자의 이름을 스키마 속성으로 지정하는 것이 좋습니다. 예: appSecretName, connectionSecretName 등

요약하면 추가 기능 공급자는 스키마를 활용하여 고객이 보안 암호의 이름을 전달할 수 있지만 보안 암호 자체를 실제로 보유하게 될 키는 전달할 수 없습니다.

예제 구성 값

스키마에 구성 예제를 포함하여 고객이 추가 기능을 구성하도록 도울 수 있습니다. 다음 예제는 AWS Distro for OpenTelemetry 추가 기능의 스키마에서 가져온 것입니다.

```
"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": []
      }
    ]
```

구성에 허용되는 공통 파라미터

다음은 고객 응대 헬름 스키마 파일의 권장 파라미터입니다.

파라미터	설명	기본값 필요 여부
additionalLabels	추가 기능으로 관리되는 모든 Kubernetes 객체에 Kubernetes 레이블을 추가합니다.	아니요
additionalAnnotations	추가 기능으로 관리되는 모든 Kubernetes 객체에 Kubernetes 주석을 추가합니다.	아니요
podLabels	추가 기능에서 관리하는 포드에 Kubernetes 레이블을 추가합니다.	아니요
podAnnotations	추가 기능으로 관리되는 포드에 Kubernetes 주석을 추가합니다.	아니요
logLevel	추가 기능에서 관리하는 구성 요소의 로그 수준입니다.	예
nodeSelector	노드 선택 제약의 가장 간단한 권장 형태입니다. 포드 사양에 nodeSelector 필드를 추가하고 대상 노드에 포함할 노드 레이블을 지정할 수 있습니다.	잠재적으로, 예를 들어 Linux 노드만
허용 오차	포드에는 허용 오차가 적용됩니다. 허용 오차를 사용하면 스케줄러가 일치하는 테인트가 있는 포드를 예약할 수 있습니다. 허용 오차는 예약을 허용하지만 예약을 보장하지는 않습니다.	아마도 DaemonSets에서 더 흔할 수 있습니다.
affinity	선호도 기능은 두 가지 유형의 선호도로 구성됩니다. 노드 선호도는 nodeSelector 필드와 유사하지만 더 표현력이 풍부하고 소프트 규칙을 지정할 수 있습니다. 포드 간 선호도/반선호도를 사용하면 포드를 다른 포드의 레이블에 대해 제한할 수 있습니다.	가능

파라미터	설명	기본값 필요 여부
topologySpreadConstraints	토폴로지 분산 제약 조건을 사용하면 리전, 영역, 노드 및 기타 사용자 정의 토폴로지 도메인과 같은 장애 도메인 간에 포드가 클러스터에 어떻게 분산되는지 제어할 수 있습니다. 이를 통해 고가용성과 효율적인 리소스 활용도를 달성할 수 있습니다.	가능
리소스 요청/제한	각 컨테이너에 필요한 CPU/메모리의 양을 지정합니다. 요청을 설정하는 것이 좋습니다. 한도는 선택 사항입니다.	예
복제본	추가 기능으로 관리되는 포드의 복제본 수입니다. DaemonSets에는 적용되지 않습니다.	예

> **Note**
>
> 워크로드 예약 구성 파라미터의 경우 필요한 경우 스키마에서 최상위 구성 요소를 분리해야 할 수 있습니다. 예를 들어 Amazon EBS CSI 드라이버에는 컨트롤러와 노드 에이전트라는 두 가지 주요 구성 요소가 포함되어 있습니다. 고객은 각 구성 요소에 대해 서로 다른 노드 선택기/허용 오차가 필요합니다.

> **Note**
>
> JSON 스키마에 정의된 기본값은 전적으로 사용자 설명서용이며, `values.yaml` 파일에 올바른 기본값이 있어야 할 필요성을 대체하지 않습니다. 기본 속성을 사용하는 경우 `values.yaml`의 기본값이 스키마의 기본값과 일치하는지, 헬름 차트를 변경할 때마다 두 아티팩트(`values.schema.json` 및 `values.yaml`)가 동기화되어 있는지 확인하세요.

```
"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"
            ]
        }
```

구성에 허용되지 않는 공통 파라미터

`clusterName`, `region`, `vpcId`, `accountId` 및 기타 클러스터 메타데이터 파라미터는 다양한 추가 기능(예: Elastic Load Balancing Controller)에 필요할 수 있습니다. Amazon EKS 서비스에서 알려진 파라미터와 유사한 파라미터는 Amazon EKS 추가 기능에서 자동으로 주입되며, 구성 옵션으로 지정할 책임은 사용자에게 없습니다. 파라미터는 다음과 같습니다.

- AWS 리전
- Amazon EKS 클러스터 이름
- 클러스터의 VPC ID
- 특히 네트워킹 추가 기능에 사용되는 빌드-프로덕션 계정용 컨테이너 레지스트리
- 특히 coredns 추가 기능에 사용되는 DNS 클러스터 IP
- Amazon EKS 클러스터 API 엔드포인트
- 클러스터에서 IPv4 활성화
- 클러스터에서 IPv6 활성화
- 클러스터에서 활성화된 IPv6에 대한 접두사 위임

추가 기능 공급자는 이러한 해당 파라미터에 템플릿이 정의되어 있는지 확인해야 합니다. 위의 각 파라미터에는 Amazon EKS에서 정의한 사전 정의된 `parameterType` 속성이 있습니다. 릴리스 메타데이터는 템플릿에서 `parameterType`과 파라미터의 이름/경로 간의 매핑을 지정합니다. 이렇게 하면 고객이 구성을 통해 값을 지정하지 않고도 Amazon EKS에서 값을 동적으로 전달할 수 있으며, 추가 공급자가 자체 템플릿 이름/경로를 유연하게 정의할 수 있습니다. 위와 같이 Amazon EKS가 동적으로 주입해야 하는 파라미터는 스키마 파일에서 제외해야 합니다.

릴리스 메타데이터의 매핑 예제

```
"defaultConfiguration": [
       {
            "key": "image.containerRegistry",
            "parameterType": "CONTAINER_REGISTRY"
       }
]
```

다음은 고객 응대 헬름 스키마 파일에서 구성하지 않는 파라미터입니다. 파라미터에는 수정할 수 없는 기본값이 있거나 추가 기능 템플릿에 전혀 포함되지 않아야 합니다.

파라미터	설명	기본값 필요 여부
image	Kubernetes 클러스터에 배포될 컨테이너 이미지입니다.	아니요, 추가 기능 정의를 통해 관리됨
imagePullSecrets	보안 암호를 사용하여 프라이빗 레지스트리에서 가져오도록 포드를 구성합니다.	N/A
livenessProbe	Kubelet 프로세스는 실시간 프로브를 사용하여 컨테이너를 다시 시작해야 하는 시기를 파악합니다. 예를 들어, 생동성 프로브는 애플리케이션이 실행 중이지만 진행할 수 없는 교착 상태를 포착할 수 있습니다. 이러한 상태에서 컨테이너를 다시 시작하면 버그에도 불구하고 애플리케이션을 더 쉽게 사용할 수 있습니다.	예
readinessProbe	컨테이너에 대한 준비 프로브가 있어야 합니다. 이렇게 하면 데이터 플레인에서 실행되는 Kubelet 프로세스가 컨테이너가 트래픽을 제공할 준비가 된 시점을 알 수 있습니다. 모든 컨테이너가 준비되면 포드가 준비된 것으로 간주됩니다. 이 신호의 한 가지 사용은 서비스의 백엔드로 사용되는 포드를 제어하는 것입니다. 포드가 준비되지 않으면 서비스 로드 밸런서에서 제거됩니다.	예
startupProbe	kubelet은 시작 프로브를 사용하여 컨테이너 애플리케이션이 시작된 시기를 파악합니다. 이러한 프로브가 구성	선택 사항

파라미터	설명	기본값 필요 여부
	된 경우, 성공할 때까지 실시간 및 준비 확인을 비활성화하여 해당 프로브가 애플리케이션 시작을 방해하지 않도록 합니다. 이를 사용하여 느린 시작 컨테이너에 대한 실시간 확인을 채택하여 kubelet이 가동 및 실행되기 전에 컨테이너가 멈추지 않도록 할 수 있습니다.	
podDisruptionBudget	자발적 중단 중에 최소 수의 포드가 계속 실행되도록 포드 중단 예산(PDB)을 정의합니다. PDB는 자발적 중단으로 인해 동시에 다운되는 복제된 애플리케이션의 포드 수를 제한합니다. 예를 들어, 쿼럼 기반 애플리케이션은 실행 중인 복제본 수가 쿼럼에 필요한 수보다 부족해지지 않도록 하려고 합니다. 웹 프런트엔드는 로드에 대한 복제본 수가 총계의 특정 백분율 미만으로 떨어지지 않도록 유지해야 할 수 있습니다.	예, 기본값이 두 개 이상의 복제본인 경우
serviceAccount(이름)	실행할 서비스 계정 포드의 이름입니다.	예

파라미터	설명	기본값 필요 여부
serviceAccount(주석)	서비스 계정에 적용되는 주석입니다. 일반적으로 서비스 계정 기능에 대한 IAM 역할에 사용됩니다.	아니요, IAM 서비스 계정 역할 ARN은 최상위 Amazon EKS 추가 기능 API에 설정되어 있습니다. 이 규칙의 예외는 추가 기능에 여러 배포/컨트롤러(예: Flux)가 있고 별도의 IRSA 역할 ARN이 필요한 경우입니다.
priorityClassName	우선 순위는 다른 포드와 비교한 포드의 중요성을 나타냅니다. 포드를 예약할 수 없는 경우, 스케줄러는 보류 중인 포드를 예약할 수 있도록 우선 순위가 낮은 포드를 선점(제거)하려고 시도합니다.	예. 대부분의 추가 기능은 클러스터 기능에 중요하며, 기본적으로 우선 순위 클래스가 설정되어 있어야 합니다.
podSecurityContext	포드 보안 컨텍스트는 포드 또는 컨테이너에 대한 권한과 액세스 제어 설정을 정의합니다. 일반적으로 v1.19 및 하위 클러스터의 IRSA에 필요한 fsGroup을 설정하는 데 사용됩니다.	가능성 낮음, Amazon EKS 가 더 이상 Kubernetes v1.19를 지원하지 않음
securityContext	포드 보안 컨텍스트는 포드 또는 컨테이너에 대한 권한과 액세스 제어 설정을 정의합니다.	예
updateStrategy	이전 포드를 새 포드로 대체하는 데 사용되는 전략을 지정합니다.	예

파라미터	설명	기본값 필요 여부
nameOverride	포드 이름을 재정의합니다.	아니요
podSecurityPolicy	파라미터에 대한 제한을 적용합니다.	아니요 - PSP가 더 이상 사용되지 않음
extraVolumeMounts/extraVolumes	Amazon EKS 클러스터가 아닌 IRSA에 사용됩니다.	아니요

에 대한 컨테이너 제품 요금 AWS Marketplace

에서 Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS) 및에 대한 무료 제품, BYOL(Bring Your Own License Model) 제품 및 유료 제품을 나열 AWS Marketplace할 수 있습니다 AWS Fargate. 제품마다 설정할 수 있는 요금은 한 가지로 제한됩니다. 이 주제에서는 컨테이너 제품에 사용할 수 있는 요금 모델을 간략하게 설명합니다.

> **Note**
>
> AWS Marketplace 측정 서비스를 사용하여 유료 제품에 대한 권한을 부여하고 사용량을 측정합니다. 작업별 또는 포드별 요금의 경우 AWS에서 자동으로 사용량을 측정합니다.

컨테이너 제품에 대해 설정한 가격이 모든에 적용됩니다 AWS 리전. 컨테이너 제품의 가격을 인하할 때마다 새로운 가격이 구매자에게 즉시 적용됩니다. 가격을 인상하는 경우 청구에 영향을 미치기 90일 전에 기존 구매자에게 변경 사항이 통지되며, 신규 구매자에게는 새 요금이 청구됩니다.

> **Note**
>
> 신규 구독자의 경우 변경된 요금이 즉시 적용됩니다. 기존 구독자의 경우 요금 변경 알림이 전송된 날짜부터 시작되는 90일 기간이 끝난 후 다음 달 1일부터 변경된 요금이 적용됩니다. 예를 들어 3월 16일에 요금 변경 알림을 보낸다고 가정하겠습니다. 3월 16일부터 약 90일 후는 6월 16일입니다. 변경된 요금은 90일 기간이 끝난 후 다음 달 1일부터 적용되므로 7월 1일부터 적용됩니다.

주제

- 컨테이너 요금 모델
- 컨테이너 제품의 계약 요금

컨테이너 요금 모델

AWS Marketplace 에는 컨테이너 제품에 대한 여러 요금 모델이 있습니다.

다음 표에서는 컨테이너 기반 제품의 요금 모델에 대한 일반 정보를 제공합니다.

컨테이너 제품에 적용되는 요금 모델

요금 모델	설명
기존 보유 라이선스 사용(BYOL)	BYOL은 구매자와 유지하는 외부 결제 관계를 AWS Marketplace 통해 외부에서 관리됩니다.
매월	고정 월별 요금 익월 동안 제품을 무제한으로 사용할 수 있는 고정 월별 요금 예: 제품 요금을 매월 99 USD로 설정합니다. 제품에는 Amazon ECS 작업 정의를 사용하여 배포되는 컨테이너 이미지가 3개 포함됩니다. 구매자가 제품을 구독하면 99 USD가 즉시 청구되며, 구독을 취소할 때까지 매월 반복됩니다. 구매자는 제품을 무제한으로 사용할 수 있습니다. 작업이 실행되는 인프라에 대해서는 요금이 별도로 청구됩니다. 구독자는 구독하는 동안 컨테이너 이미지에 액세스할 수 있습니다. 컨테이너 이미지는 어떤 구성에서든 Amazon ECS 또는 Amazon EKS 에 있는 해당 이미지에서 원하는 만큼 컨테이너를 시작하고 실행할 수 있습니다. 구매자가 한 달 중순에 구독을 취소하면가 컨테이너 이미지를 AWS Marketplace 저장하는 Amazon ECR 리포지토리에 액세스할 수 없습니다. 구매자가 원본 이미지를 가져와서 저장했을 수 있습니다. 그러나 더 이상 사용할 수 있는 새 컨테이너 이미지 버전을 가져올 수 없습니다 AWS Marketplace. 마지막 달의 미사용 부분에 대한 요금이 구매자에게 환불됩니다. 구매자의 사용량에서 합의된 AWS Marketplace 요금을 뺀 값을 기준으로 지불됩니다.
사용자 지정 지표 요금 차원	정의한 차원(예: 사용자, 노드, 리포지토리, GB)을 기준으로 하는 사용자 지정 측정 요금. 제품당 최대 24개의 차원을 정의할 수 있습니다.

요금 모델	설명
	예: 제품은 사용자별로 요금이 청구됩니다. 관리자와 일반 사용자가 있으며, 관리자의 경우 2 USD, 일반 사용자의 경우 1 USD로 가격을 정의합니다. 제품을 리스팅할 때 별도의 차원을 설정할 수 있습니다. 일간 로그인한 사용자별로 요금을 청구하고 일일 사용량을 측정합니다.
작업별 또는 포드별 시간당 가격	Amazon ECS 작업 또는 Amazon EKS 포드 설정된 시간당 요금으로 초 단위까지 측정하여 청구되는 Amazon ECS 작업별 또는 Amazon EKS 포드별 요금. 예: 제품에는 컨트롤러 노드, 워커 노드, 분석 노드 등 컨테이너 이미지가 3개 포함됩니다. 컨트롤러 노드가 없으면 제품이 실용적이거나 유용하지 않기 때문에 사용량에 대한 요금을 청구할 이미지로 컨트롤러 노드를 결정합니다. 요금은 시간당 6 USD로 설정합니다. 컨트롤러 노드가 AWS Marketplace Metering Service `RegisterUsage` API 운영과 통합되도록 컨테이너 이미지에서 소프트웨어를 수정합니다. 이렇게 하면 구독이 활성화된 구매자만 컨테이너 이미지를 시작 및 실행할 수 있으며, 사용량은 실행 기간에 따라 측정됩니다. 실행되는 Amazon EKS 컨트롤러 포드마다 시간당 6 USD가 구매자에게 청구됩니다. 구매자가 컨트롤러 노드 컨테이너가 포함된 Amazon EKS 컨트롤러 포드를 5개 시작하면 시간당 30 USD(포드마다 6 USD)가 청구됩니다. 포드가 실행되는 인프라에 대해서는 요금이 별도로 청구됩니다. 시간당 요금일 때는 초 단위로 청구되며, 최소 시간은 1분입니다. 고객이 20분 30초 동안 컨트롤러 컨테이너를 실행하였다면 `20 x ($6/60) + 30 x ($6/60/60) = $2 + $0.05 = $2.05` 가 청구됩니다. 구매자의 사용량에서 합의된 AWS Marketplace 요금을 뺀 값을 기준으로 지불됩니다.

요금 모델	설명
장기 계약 시간당/사용량 요금	선불 또는 정기 할부로 결제하는 장기 계약 요금(할인 요금). 사용자 지정 측정 요금 또는 작업별 및 포드별 요금이 적용되는 기존 제품에 장기 계약을 추가할 수 있습니다. 구매자는 장기 계약에서 구매한 것보다 더 많이 사용할 경우 측정 가격을 지불합니다. 예: 측정 요금 모델의 경우 구매자에게 장기 계약 요금을 추가하여 선결제 계약 할인을 제공할 수 있습니다. 일반적으로 1단위를 사용할 때마다 1 USD를 청구한다고 가정하겠습니다. 시간당 1단위를 사용하는 구매자는 연간 8760 USD(`365 days x 24 hours x $1 per hour`)를 지불하게 됩니다. 구매자가 365일 동안 시간당 1단위를 절반 요금(4380 USD)으로 사용할 수 있는 계약을 활성화할 수 있습니다. 이 경우 구매자가 1년 계약 요금을 선결제하기로 계약하면 요금이 단위당 1 USD에서 단위당 0.5 USD로 할인됩니다. 구매자가 이러한 계약을 여러 개 구매하게 할 수도 있습니다. 수량을 측정했더니, 구매자가 한 시간에 10단위를 사용했고 계약을 2개 체결한 경우 2단위가 2개 계약에 포함됩니다. 나머지 8개 단위는 시간당 1 USD가 청구되며, 따라서 이 시간의 요금으로 총 8 USD가 청구됩니다. 작업별 또는 포드별 예제의 경우 구매자가 선결제 계약 할인을 받을 수 있도록 장기 계약 요금을 추가할 수도 있습니다. 일반적으로 포드당 6 USD를 청구하는 경우 365일 장기 계약 기간의 요금을 13,140 USD(`365 days x 24 hours x $3 per pod per hour`)로 설정할 수 있습니다. 계약 하나는 고객에게 365일 동안 시간당 포드 1개를 사용할 수 있는 권한을 부여합니다. 고객은 계약을 여러 개 구매할 수 있습니다. 예를 들어 고객은 시간당 포드 2개를 사용할 수 있는 계약을 2개 구매할 수 있습니다. 고객이 허가된 계약보다 많은 수의 시간당 포드를 실행하면 초과된 포트에 대해서는 일반 시간당 요금이 청구됩니다. 두 경우 모두 장기 계약을 구매하는 구매자는 일회성 결제 또는 정기 일정에 따른 향후 결제로 사전에 요금이 청구됩니다. 또한 구매자는 계약 사항을 초과하는 추가 사용량에 대해 측정 요금이 청구됩니다.
컨테이너 계약 요금	계약 요금이 적용되는 컨테이너 - 구매자가 요금을 선결제하는 컨테이너 기반 제품입니다.

컨테이너 제품의 계약 요금

계약 요금이 적용되는 컨테이너 기반 제품의 경우는 고객과 고객 간의 계약에 따라 고객이 사전에 또는 정의한 결제 일정에 따라 요금을 AWS Marketplace 청구합니다. 그 이후 고객에게 해당 리소스를 사용할 권한이 부여됩니다.

요금을 설정하려면 고객에게 제공하는 하나 이상의 계약 기간을 선택합니다. 각 계약 기간에 다른 요금을 입력할 수 있습니다. 1개월, 12개월, 24개월, 36개월 중에 선택할 수 있습니다. 비공개 제안의 경우 사용자 지정 기간을 월 단위로(최대 60개월) 지정할 수 있습니다.

제품 요금을 가장 잘 설명하는 범주를 선택합니다. 요금 범주는 AWS Marketplace 웹 사이트에 고객에게 표시됩니다. 대역폭(GB/s, MB/s), 데이터(GB, MB, TB), 호스트, 요청, 티어 또는 사용자 중에서 선택할 수 있습니다. 미리 정의된 범주 중 요구 사항에 적합한 것이 없는 경우 더 일반적인 단위 범주를 선택할 수 있습니다.

이 제안에는 최대 24개 차원을 추가할 수 있습니다. 각 차원에는 다음 데이터가 필요합니다.

- 계약 범주 - 제품이 계약 요금에 더해서 사용량 기반 측정을 지원하는 경우 계약 범주는 제품을 측정하는 데 사용됩니다. 사용량 기반 요금이 적용되지 않는 계약 제품의 경우 계약의 차원 범주와 가장 유사한 범주를 선택하면 됩니다. 계약의 차원 단위와 비슷한 값이 없으면 `Units`를 선택합니다.
- 계약 단위 - 제품이 사용량 기반 측정을 지원하는 경우 계약 단위는 측정 범주와 함께 사용됩니다. 선택한 범주에 따라 차원과 가장 유사한 단위의 사용 가능한 값 중에서 하나를 선택합니다.
- 계약 차원 복수 구매 허용 - 이 필드는 제안이 계층별 요금 제안인지 아니면 여러 차원 구매를 허용하는 비계층 제안인지 나타내는 데 사용됩니다.

  계층형 제안 - 구매자가 제안에 제공된 차원 중에서 하나만 구독할 수 있습니다. 계층형 제안의 차원은 수량이라는 개념이 없습니다. 특정 차원이 포함된 계약을 체결한다는 것은 본질적으로 구매자가 해당 차원에 지정된 특정 기능을 선택했다는 뜻입니다.

  비계층형 제안 - 고객이 차원을 2개 이상 조달하고 각 차원의 여러 단위를 조달할 수 있도록 계약서에 허용합니다.

  이 필드에 값을 true로 설정하면 해당 제안은 비계층형 제안이라는 뜻입니다. 이 필드에 값을 false로 설정하면 해당 제안은 계층형 제안이라는 뜻입니다.

제품 로드 양식(PLF)을 사용하여 컨테이너 제품에 대한 계약을 생성할 때, 요금 차원에 대한 다음 필드를 정의해야 합니다.

- 계약 DimensionX API 이름 - 구매자 계정에서 생성된 라이선스에 표시되어야 하는 이름입니다 AWS License Manager . 이 이름은 `Checkoutlicense` API 직접 호출에서 `Entitlement`의 `Name` 값으로도 사용됩니다.
- 계약 DimensionX 표시 이름 - AWS Marketplace 웹 사이트의 제품 세부 정보 페이지와 조달 페이지에 표시되는 차원의 고객용 이름입니다. 사용자에게 친숙한 이름을 지정하세요. 이름의 최대 길이는 24자입니다. 목록이 공개된 후에는 `Name` 값을 변경할 수 없습니다.
- 계약 DimensionX 설명 - 특정 차원이 제공하는 기능과 같이 제품의 차원에 대한 추가 정보를 제공하는 차원의 고객용 설명입니다. 설명의 최대 길이는 70자입니다.
- 계약 차원X 수량 - 제품 계약을 수정하는 경우 비례 배분을 계산하는 데 사용됩니다. 모든 계약 제안에서 이 필드의 값을 1로 설정해야 합니다. 편집하면 안 됩니다.
- 계약 DimensionX 1개월 요율 - 이 차원에 대한 권한을 1개월 동안 부여할 때 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 이하 8자리를 지원합니다.
- 계약 DimensionX 12개월 요율 - 이 차원에 대한 권한을 12개월 동안 부여할 때 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 이하 8자리를 지원합니다.
- 계약 DimensionX 24개월 요율 - 이 차원에 대한 권한을 24개월 동안 부여할 때 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 이하 8자리를 지원합니다.
- 계약 DimensionX 36개월 요율 - 이 차원에 대한 권한을 36개월 동안 부여할 때 부과되는 계약 요율입니다. 비계층형 제안의 경우 조달되는 차원 1단위마다 이 요율로 부과됩니다. 이 필드는 소수점 이하 8자리를 지원합니다.

예: 데이터 스토리지 애플리케이션

	1개월 요금	12개월 요금	24개월 요금	36개월 요금
암호화되지 않은 데이터(GB)	1.50 USD/GB	16.00 USD/GB	30.00 USD/GB	GB당 60.00 USD
암호화된 데이터(GB)	1.55 USD/GB	16.60 USD/GB	31.20 USD/GB	GB당 61.20 USD

예제: 로그 모니터링 제품

	1개월 요금	12개월 요금	24개월 요금	36개월 요금
기본(10개의 호스트 모니터링, 5개의 컨테이너 모니터링)	100 USD	1000 USD	2000 USD	4000 USD
스탠다드(20개의 호스트 모니터링, 10개의 컨테이너 모니터링)	$200	2000 USD	4000 USD	8,000 USD
프로(40개의 호스트 모니터링, 20개의 컨테이너 모니터링)	400 USD	4000 USD	8,000 USD	16,000 USD
시간당 추가 호스트 모니터링	10 USD	100 USD	$200	400 USD
시간당 추가 컨테이너 모니터링	10 USD	100 USD	$200	400 USD

Note

요금은 1개월, 12개월, 24개월, 36개월 기간일 수 있습니다. 제품에 대해 이러한 옵션을 하나 이상 제공하도록 선택할 수 있습니다. 기간은 각 차원에 대해 동일해야 합니다.

Example

예를 들어 `ReadOnlyUsers` 및 `AdminUsers` 차원이 있는 경우 ReadOnlyUsers에 대한 연간 가격을 제공하면 `AdminUsers`에 대해서도 연간 가격을 제공해야 합니다.

자동 갱신

고객이 컨테이너 계약을 AWS Marketplace 사용하여 제품을 구매할 때 계약 조건을 자동으로 갱신하는 데 동의할 수 있습니다. 고객은 매달, 1년마다, 2년마다 또는 3년마다 권한 부여에 대한 요금을 계속 지불합니다.

고객은 언제든지 갱신 설정을 변경할 수 있습니다. 자세한 내용은 AWS Marketplace 구매자 안내서의 기존 계약 수정을 참조하세요.

컨테이너 계약이 종료되는 경우

컨테이너 계약 제품은 계약 만료일이 있습니다. 계약이 종료되면 다음과 같은 이벤트가 발생합니다.

1. 컨테이너 제품은 구매자의 권한이 변경되었다는 `entitlement-updated` 알림을 수신하고는 빈 응답을 AWS Marketplace Entitlement Service 반환합니다.
2. 1시간 동안 구매자의 나머지 사용량을 측정할 수 있습니다. 이후 이 고객에 대한 측정 레코드를 더 이상 전송할 수 없습니다.

컨테이너 제품 결제, 측정 및 라이선스 통합

AWS Marketplace 는 다른와 통합되어 컨테이너 제품에 대한 측정 및 계약 기반 요금을 모두 AWS 서비스 제공합니다. 사용량 요금이 적용되는 컨테이너 기반 제품의 경우 AWS Marketplace Metering Service를 사용하여 제품 사용 권한을 확인하고 요금 청구를 위해 사용량을 측정할 수 있습니다. 계약 요금이 적용되는 컨테이너 기반 제품의 경우 AWS License Manager 를 사용하여 라이선스를 제품과 연결할 수 있습니다. 다음 섹션에서는 AWS Marketplace Metering Service 를 사용한 시간당 및 사용자 지정 측정과 AWS License Manager를 사용한 계약 요금에 대한 자세한 정보를 제공합니다.

주제

- AWS Marketplace Metering Service로 시간당 및 사용자 지정 측정
- 를 사용한 계약 요금 AWS License Manager
- AWS Marketplace Metering Service를 사용한 시간당 측정 구성
- AWS Marketplace Metering Service를 사용하여 컨테이너 제품의 사용자 지정 측정 구성
- 를 사용한 컨테이너 제품의 계약 요금 AWS License Manager

AWS Marketplace Metering Service로 시간당 및 사용자 지정 측정

제품 사용 권한을 확인하고 요금을 청구할 사용량을 측정하려면 AWS Marketplace 측정 서비스를 사용합니다. 판매자 고유의 요금 단위를 정의하고 사용량을 측정한 후 AWS가 요금을 청구하게 하려면 MeterUsage API 작업을 사용하여 통합해야 합니다. 사용된 작업 또는 포드 수를 기준으로 제품의 가격을 책정하고 해당 사용량을 자동으로 AWS 측정하려면 RegisterUsage API 작업을 사용하여를 통합합니다. 두 가지 요금 유형 모두 AWS Marketplace Metering Service와 통합하는 방법을 변경하지 않고 장기 계약 요금을 추가할 수 있습니다.

에서 새 컨테이너 제품을 생성하면 제품을와 통합하는 데 사용되는 제품 식별자 세트(제품 코드 및 퍼블릭 키)가 AWS Marketplace Management Portal제공됩니다 AWS Marketplace Metering Service.

권한 부여

와 통합하면 유료 소프트웨어를 실행하는 고객이 제품을 구독하는지 확인할 AWS Marketplace Metering Service 수 있으므로 컨테이너 시작 시 무단 사용이 방지 AWS Marketplace됩니다. 권한을 확인하려면 요금 모델에 따라 MeterUsage 또는 RegisterUsage API 작업을 사용합니다. 시간당 요금 및 고정 월별 요금 모델의 경우 `RegisterUsage` API 작업을 사용합니다. 사용자 지정 측정 요금 모델의 경우 `MeterUsage` API 작업을 사용합니다.

구매자에게 제품에 대한 권한이 없는 경우 이러한 API 작업에서 `CustomerNotEntitledException` 예외가 반환됩니다.

> **Note**
>
> 구매자가 제품을 실행하는 동안 제품 가입을 취소하면 계속해서 실행할 수 있습니다. 그러나 제품에 대한 추가 컨테이너를 시작할 수는 없습니다.

통합 지침

컨테이너 제품을 생성 및 게시하고 권한 부여 및 측정을 위해 `MeterUsage` 또는 `RegisterUsage` API 작업을 사용하는 경우 다음 지침에 유의하세요.

- 소프트웨어 또는 Docker 컨테이너 이미지 내에서 AWS 자격 증명을 구성하지 마세요. 컨테이너 이미지가 Amazon ECS 작업 또는 Amazon EKS 포드 내에서 실행 중일 때 런타임 시 구매자의 AWS 자격 증명이 자동으로 획득됩니다.
- Amazon EKS에서 `MeterUsage` 또는 `RegisterUsage` API 작업을 호출하려면 지원되는 AWS SDK를 사용해야 합니다. Amazon EKS의 `MeterUsage` 또는 `RegisterUsage` 통합을 테스트하려

면 Kubernetes 1.13.x 이상을 실행하는 Amazon EKS 클러스터를 실행해야 합니다. 포드 지원을 위한 AWS Identity and Access Management (IAM) 역할에는 Kubernetes 1.13이 필요합니다. 실행 중인 포드가 Amazon EKS에서 이러한 작업을 호출하는 데 필요한 AWS 자격 증명을 얻으려면 IAM 역할이 필요합니다.

- 로컬 개발도 가능하지만 `PlatformNotSupportedException` 예외가 발생합니다. AWS 컨테이너 서비스(Amazon ECS, Amazon EKS 및 Fargate)에서 컨테이너를 시작할 때는이 예외가 발생하지 않습니다.

지원됨 AWS 리전

AWS Marketplace 지원되는 모든 목록의 경우 글로벌 인프라 웹 사이트의 리전 테이블을 AWS 리전참조하세요.

측정 AWS 리전 용 얻기

측정용 컨테이너를 `MeterUsage` 또는 `RegisterUsage` API 작업과 통합할 때 특정를 사용하도록 AWS SDK를 구성하지 마십시오 AWS 리전. 리전은 런타임에서 동적으로 가져와야 합니다.

Example

예를 들어 고객이 Amazon ECS 작업 또는 Amazon EKS 포드를 시작합니다. Amazon ECS 작업 또는 Amazon EKS 포드가 시작된 리전과 다른 리전에서 `RegisterUsage` API 작업이 호출됩니다. 따라서 `RegisterUsage` API 작업에서 `InvalidRegionException` 오류를 throw합니다.

AWS SDK 언어는 일관된 방식으로 `AWS_REGION`를 결정하지 않습니다. SDK가 `AWS_REGION`을 자동으로 선택하지 않으면 소프트웨어를 수동으로 작성하여 `AWS_Region`을 결정해야 합니다. 예를 들어 AWS SDK for Java 는 환경 변수 또는 기타 구성이 존재하지 않을 경우 자동으로 Amazon EC2 인스턴스 메타데이터(구체적으로 `ec2InstanceMetadata`)를 사용하여 리전을 결정합니다. 이 경우 `AWS_REGION` 환경 변수가 없을 때에만 `ec2InstanceMetadata`를 호출합니다.

런타임 AWS 리전 에를 동적으로 가져오는 방법에 대한 자세한 내용은 프로그래밍 언어에 대한 AWS SDK 개발자 안내서를 참조하세요.

측정 수정 방지

구매자가 `RegisterUsage` 또는 `MeterUsage` 호출을 수정하거나 재정의할 수 있도록 하면 바람직하지 않은 청구 및 결제 문제가 발생할 수 있습니다. 측정 및 권한 부여 로직을 통합하는 것이 좋습니다.

측정 수정을 방지하기 위해 제품을 엔지니어링할 때 다음 사항에 유의해야 합니다.

- 구매자가 `CMD` 또는 `ENTRYPOINT` 명령을 포함하고 있는 새 이미지 계층을 삽입할 수 있는 경우 구매자가 판매자의 컨테이너 이미지를 통해 실행하는 소프트웨어에 `RegisterUsage` 또는 `MeterUsage`를 직접 통합하세요. 그러지 않으면 기본 이미지에서 `CMD` 또는 `ENTRYPOINT`를 통해 실행되는 `RegisterUsage` 또는 `MeterUsage` 호출을 구매자가 재정의할 가능성이 있습니다.
- 소프트웨어에서 또는 구매자가 수정할 수 없는 `RegisterUsage` `MeterUsage` 방식으로 입력으로 사용하는 AWS Marketplace 제품 코드를 관리하는 것이 좋습니다. 그러나 제품이 , AWS CloudFormation차트 Helm 또는 Kubernetes 매니페스트와 같이 고객이 재정의할 수 있는 방식으로 제품 코드를 관리하는 경우 신뢰할 AWS Marketplace 수 있는 제품 코드 목록을 유지해야 합니다. 이는 소프트웨어가 `RegisterUsage` 또는 `MeterUsage`에 입력으로 전달하는 제품 코드의 유효성을 확보하기 위한 조치입니다.
- 신뢰할 수 있는 제품 코드 중에서 무료 제품에 사용되는 코드가 있다면 유료 제품 코드를 대신해서 사용하지 못하도록 해야 합니다.

를 사용한 계약 요금 AWS License Manager

계약 요금이 적용되는 컨테이너 기반 제품의 경우 AWS License Manager 를 사용하여 라이선스를 제품과 연결합니다.

AWS License Manager 는 애플리케이션이 고객이 구매한 라이선스(권한이라고도 함)를 추적하고 업데이트할 수 있는 라이선스 관리 도구입니다. 이 섹션에서는 제품을 AWS License Manager와 통합하는 방법에 대한 정보를 제공합니다. 통합이 완료되면 제품 목록을 AWS Marketplace에 게시할 수 있습니다.

에 대한 자세한 내용은 AWS License Manager 사용 설명서 및 AWS CLI 명령 AWS License Manager 참조의 AWS License Manager 섹션을 참조하세요.

> **Note**
>
> - 계약이 만료된 후에는 고객이 새 컨테이너 인스턴스를 시작할 수 없습니다. 하지만 계약 기간에는 인스턴스를 원하는 만큼 시작할 수 있습니다. 이러한 라이선스는 특정 노드나 인스턴스에 바인딩되지 않습니다. 노드의 컨테이너에서 실행되는 소프트웨어는 할당된 AWS 보안 인증 정보가 있는 한 라이선스를 체크아웃할 수 있습니다.
> - 비공개 제안 생성 - 판매자는 AWS Marketplace Management Portal의 비공개 제안 생성 도구를 사용하여 제품에 대한 비공개 제안을 생성할 수 있습니다.

- 보고 - AWS Marketplace Management Portal의 보고서 섹션에서 Amazon S3 버킷을 설정하여 데이터 피드를 설정할 수 있습니다. 자세한 내용은 의 판매자 보고서, 데이터 피드 및 대시보드 AWS Marketplace 단원을 참조하십시오.

통합 워크플로

다음 단계는 컨테이너 제품을 AWS License Manager와 통합하는 워크플로를 보여줍니다.

1. 판매자는 AWS License Manager 통합으로 제품을 생성합니다.
2. 판매자가 제품을 나열합니다 AWS Marketplace.
3. 구매자가에서 제품을 찾아 AWS Marketplace 구매합니다.
4. 라이선스가 구매자의 AWS 계정에 전송됩니다.
5. 구매자가 Amazon EC2 인스턴스, Amazon ECS 작업 또는 Amazon EKS 포드 소프트웨어를 시작하여 소프트웨어를 시작합니다. 고객이 IAM 역할을 사용하여 배포합니다.
6. 소프트웨어는 구매자 AWS License Manager 계정의 라이선스를 읽고, 구매한 권한을 검색하고, 그에 따라 기능을 프로비저닝합니다.

> **Note**
>
> License Manager는 추적 또는 업데이트를 수행하지 않습니다. 추적 또는 업데이트는 판매자의 애플리케이션에서 수행됩니다.

AWS Marketplace Metering Service를 사용한 시간당 측정 구성

> **Note**
>
> Amazon EKS 배포의 경우 소프트웨어는 서비스 계정에 대한 IAM 역할(IRSA)을 사용하여 API 작업에 대한 `RegisterUsage` API 호출에 서명해야 합니다. EKS Pod Identity, 노드 역할 또는 장기 액세스 키를 사용하는 것은 지원되지 않습니다.
> Amazon ECS 배포의 경우 소프트웨어는 Amazon ECS 태스크 IAM 역할을 사용하여 API 작업에 대한 `RegisterUsage` API 호출에 서명해야 합니다. 노드 역할 또는 장기 액세스 키 사용은 지원되지 않습니다.

컨테이너 제품이 사용자 지정 측정 요금 차원 대신 시간당 작업별 또는 포드별 요금을 사용하는 경우 사용자 지정 측정 차원을 정의할 필요가 없습니다. AWS Marketplace Metering Service를 사용하여 AWS Marketplace의 컨테이너 제품을 시간당 측정할 수 있습니다. 다음 섹션에서는 AWS Marketplace Metering Service를 사용하여 시간당 측정을 구성하는 방법을 보여줍니다.

`RegisterUsage` API 작업은 Amazon Elastic Container Service(Amazon ECS) 작업별, Amazon Elastic Kubernetes Service(Amazon EKS) 포드별 또는 시간당 소프트웨어 사용량을 측정하며 사용량은 초 단위로 비례 배분합니다. 실행 시간이 짧은 작업 또는 포드에 대해서는 최소 1분의 사용량이 적용됩니다. 소프트웨어 사용을 위한 연속 측정은에서 자동으로 처리됩니다 AWS Marketplace Metering Control Plane. 사용자 소프트웨어는 소프트웨어 사용 측정을 시작할 때 한 번 `RegisterUsage`를 호출하는 것 외에 다른 측정 작업이 필요하지 않습니다.

컨테이너를 시작할 때 `RegisterUsage`를 즉시 호출해야 합니다. 컨테이너 시작 후 처음 6시간 이내에 컨테이너를 등록하지 않으면 AWS Marketplace 측정 서비스는 이전 달에 대한 어떠한 측정도 보장하지 않습니다. 하지만 이번 달 이후에도 컨테이너가 종료될 때까지 측정이 계속됩니다.

는 AWS Marketplace Metering Control Plane 고객의 구독 상태에 관계없이 Amazon ECS 작업 및 Amazon EKS 포드 실행에 대한 요금을 고객에게 계속 청구합니다. 따라서 작업 또는 포드가 성공적으로 시작된 후에는 소프트웨어에서 권한 검사를 수행할 필요가 없습니다.

AWS Marketplace Metering Service API를 시간당 요금으로 컨테이너 제품과 통합하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 워크숍의 시간당 측정과 통합 랩을 참조하세요.

주제

- 시간당 측정을 위한 사전 조건
- 에 대한 통합 테스트 RegisterUsage
- RegisterUsage의 오류 처리
- 를 사용하여 컨테이너 제품을 AWS Marketplace 측정 서비스와 통합 AWS SDK for Java

시간당 측정을 위한 사전 조건

제품을 게시하기 전에 다음을 수행해야 합니다.

1. 에서 새 컨테이너 제품을 생성하고 해당 제품 코드를 기록해 AWS Marketplace Management Portal 둡니다.

   자세한 내용은 개요: 컨테이너 제품 만들기 단원을 참조하십시오.

2. 를 호출하는 데 필요한 IAM 권한을 사용하여 애플리케이션을 실행하는 작업 또는 포드에 AWS Identity and Access Management (IAM) 역할을 사용합니다`RegisterUsage`. IAM 관리형 정책 `AWSMarketplaceMeteringRegisterUsage`에 이러한 권한이 있습니다. 정책에 대한 자세한 내용은 관리형 정책 참조의 AWSMarketplaceMeteringFullAccess를 참조하세요. AWS
3. (선택 사항) 로깅을 보려면 작업 또는 포드 정의에서 AWS CloudTrail 로깅을 활성화하는 것이 좋습니다.
4. 정의하는 모든 요금 차원에 대한 레코드를 사용하여 `RegisterUsage` API 작업 호출을 테스트합니다.

에 대한 통합 테스트 `RegisterUsage`

`RegisterUsage` API 작업을 사용하여 게시를 AWS Marketplace 위해 이미지를에 제출하기 전에 통합을 테스트합니다.

Amazon ECS 또는 Amazon EKS에서 제품을 실행하여 컨테이너 이미지`RegisterUsage`에서를 호출합니다. 사용 중인 AWS 계정을 사용하여 제품을 나열합니다 AWS Marketplace. 측정 통합은 하드 코딩 AWS 리전이 아닌 동적으로를 설정해야 합니다. 하지만 테스트할 때는 미국 동부(버지니아 북부) 리전에서 유료 컨테이너가 포함된 Amazon ECS 작업 또는 Amazon EKS 포드를 하나 이상 시작합니다. 이렇게 하면 AWS Marketplace 운영 팀이 해당 리전의 로그 작업을 확인할 수 있습니다.

> **Note**
>
> 제품이 Amazon ECS와 Amazon EKS를 모두 지원하는 경우 판매자는 Amazon EKS에서 시작하기만 하면 됩니다. 그러면 AWS에서 통합을 검증할 수 있습니다.

제품과 함께 필요한 모든 메타데이터 및 요금 정보가 게시될 때까지는 통합에 대한 완전한 테스트가 어렵습니다. 요청된 경우 AWS Marketplace 카탈로그 운영 팀은 측정 레코드의 수신을 확인할 수 있습니다.

`RegisterUsage`의 오류 처리

컨테이너 이미지가와 통합 AWS Marketplace Metering Service 되고 컨테이너 시작 `ThrottlingException` 시 이외의 예외가 발생하는 경우 무단 사용을 방지하기 위해 컨테이너를 종료해야 합니다.

`ThrottlingException` 이외의 예외는 초기 `RegisterUsage` API 작업 호출에서만 발생합니다. 하지만 이후에 동일한 Amazon ECS 작업 또는 Amazon EKS 포드에서 호출할 때에는 작업 또는 포드가

여전히 실행 중인 동안 고객이 구독을 해지해도 `CustomerNotSubscribedException`이 throw되지 않습니다. 이러한 고객은 구독을 취소한 후에도 사용량을 추적하여 컨테이너 실행에 대한 요금이 계속 청구됩니다.

다음 표에는 `RegisterUsage` API 작업에서 발생할 수 있는 오류가 설명되어 있습니다. 각 AWS SDK 프로그래밍 언어에는 추가 정보를 위해 참조할 수 있는 오류 처리 지침 세트가 있습니다.

오류	설명
`InternalServiceErrorException`	`RegisterUsage` 를 사용할 수 없습니다.
`CustomerNotEntitledException`	고객에게 유효한 제품 구독이 없습니다.
`InvalidProductCodeException`	요청과 함께 전달된 `ProductCode` 값이 존재하지 않습니다.
`InvalidPublicKeyException`	요청과 함께 전달된 `PublicKeyVersion` 값이 존재하지 않습니다.
`PlatformNotSupportedException`	AWS Marketplace 는 기본 플랫폼의 측정 사용량을 지원하지 않습니다. Amazon ECS, Amazon EKS 및 만 지원 AWS Fargate 됩니다.
`ThrottlingException`	`RegisterUsage` 호출에 병목 현상이 발생했습니다.
`InvalidRegionException`	`RegisterUsage` 는 Amazon ECS 작업 또는 Amazon EKS 포드가 시작된 AWS 리전 것과 동일한에서 호출되어야 합니다. 그래야만 `RegisterUsage` 호출 시 컨테이너가 리전(예: `withRegion(“us-east-1”)`)을 선택하지 않습니다.

를 사용하여 컨테이너 제품을 AWS Marketplace 측정 서비스와 통합 AWS SDK for Java

AWS SDK for Java 를 사용하여 AWS Marketplace 측정 서비스와 통합할 수 있습니다. 소프트웨어 사용을 위한 연속 측정은에서 자동으로 처리됩니다 AWS Marketplace Metering Control Plane. 사

용자 소프트웨어는 소프트웨어 사용 측정을 시작할 때 한 번 `RegisterUsage`를 호출하는 것 외에 다른 측정 작업이 필요하지 않습니다. 이 주제에서는를 사용하여 AWS Marketplace 측정 서비스의 `RegisterUsage` 작업과 통합 AWS SDK for Java 하는 예제 구현을 제공합니다.

컨테이너를 시작할 때 `RegisterUsage`를 즉시 호출해야 합니다. 컨테이너 시작 후 처음 6시간 이내에 컨테이너를 등록하지 않으면 AWS Marketplace 측정 서비스는 이전 달에 대한 어떠한 측정도 보장하지 않습니다. 하지만 이번 달 이후에도 컨테이너가 종료될 때까지 측정이 계속됩니다.

전체 소스 코드는 RegisterUsage Java 예제 단원을 참조하세요. 이러한 단계 중 대부분은 AWS SDK 언어에 관계없이 적용됩니다.

AWS Marketplace 측정 서비스를 통합하는 단계 예제

1. AWS Marketplace Management Portal에 로그인합니다.
2. [Assets(자산)]에서 [Containers(컨테이너)]를 선택하여 새 컨테이너 제품을 생성합니다. 제품을 만들면 제품에 대한 제품 코드가 생성되어 컨테이너 이미지와 통합할 수 있습니다. IAM 권한 설정에 대한 자세한 내용은 the section called “AWS Marketplace 측정 및 권한 부여 API 권한” 섹션을 참조하세요.
3. 퍼블릭 AWS Java SDK를 다운로드합니다.

   > **Important**
   >
   > Amazon EKS에서 측정 APIs를 호출하려면 지원되는 AWS SDK를 사용하고 Kubernetes 1.13 이상을 실행하는 Amazon EKS 클러스터에서를 실행해야 합니다.

4. (선택 사항) `RegisterUsage` 작업과 통합하며 디지털 서명 확인을 원한다면 애플리케이션 클래스 경로에서 BouncyCastle 서명 확인 라이브러리를 구성해야 합니다.

   또한 JSON 웹 토큰(JWT)을 사용하고 싶다면 JWT Java 라이브러리를 애플리케이션 클래스 경로에 추가해야 합니다. JWT를 사용하면 서명을 더욱 간단하게 확인할 수 있는 방법이 있지만 반드시 필요하지는 않습니다. 대신에 독립적으로 실행되는 BouncyCastle을 사용할 수도 있습니다. JWT를 사용하든, BouncyCastle을 사용하든 상관없이 Maven 같은 빌드 시스템을 사용하여 BouncyCastle 또는 JWT의 전이 종속성을 애플리케이션 클래스 경로에 추가해야 합니다.

```
// 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. 제품의 각 유료 컨테이너 이미지에서 `RegisterUsage`를 호출합니다. `ProductCode`와 `PublicKeyVersion`는 필수 파라미터이지만 그 밖에 모든 입력 값은 선택 사항입니다. 다음은 `RegisterUsage`에 대한 페이로드 예제입니다.

```
{
    "ProductCode" : "string", // (required)
    "PublicKeyVersion": 1,    // (required)
    "Nonce": "string",        // (optional) to scope down the registration
                              //            to a specific running software
                              //            instance and guard against
                              //            replay attacks
}
```

> **Note**
>
> AWS Marketplace 측정 서비스에 연결할 때 일시적인 문제가 발생할 수 있습니다. AWS Marketplace 에서는 단기 중단 또는 네트워크 문제를 방지할 수 있도록 지수 백오프를 사용하여 최대 30분 동안 재시도하도록 구현할 것을 강력히 권장합니다.

6. `RegisterUsage`가 요청 진위를 확인할 수 있는 SHA-256을 사용해 RSA-PSS 디지털 서명을 생성합니다. 서명에는 `ProductCode`, `PublicKeyVersion` 및 `Nonce` 필드가 포함됩니다. 디지털 서명을 확인하려면 요청에서 이 세 가지 필드를 유지해야 합니다. 다음은 `RegisterUsage` 호출에 대한 응답 예제입니다.

```
{
"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. `RegisterUsage` 호출을 포함하는 새 버전의 컨테이너 이미지를 다시 빌드하고, 컨테이너에 태그를 지정하고, 컨테이너를 Amazon ECR 또는 Amazon ECR Public처럼 Amazon ECS 또는 Amazon EKS와 호환되는 컨테이너 레지스트리로 푸시합니다. Amazon ECR을 사용하는 경우에는 Amazon ECS 작업 또는 Amazon EKS 포드를 시작하는 계정에 Amazon ECR 리포지토리에 대한 권한이 있는지 확인합니다. 그렇지 않으면 시작되지 않습니다.
8. 다음 코드에 정의된 대로 컨테이너가 `RegisterUsage`를 호출할 수 있는 권한을 부여하는 IAM 역할을 생성합니다. Amazon ECS 작업 또는 Amazon EKS 포드 정의의 작업 역할 파라미터에 이 IAM 역할을 제공해야 합니다.

   JSON

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

9. 와 통합된 컨테이너를 참조 AWS Marketplace 하고 7단계에서 생성한 IAM 역할을 참조하는 Amazon ECS 작업 또는 Amazon EKS 포드 정의를 생성합니다. AWS CloudTrail 로깅을 보려면 작업 정의에서 로깅을 활성화해야 합니다.
10. 작업 또는 포드를 실행할 Amazon ECS 또는 Amazon EKS 클러스터를 생성합니다. Amazon ECS 클러스터 생성에 대한 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 클러스터 생성을 참조하세요. Kubernetes 버전 1.1.3.x 이상을 사용하여 Amazon EKS 클러스터를 생성하는 방법에 대한 자세한 내용은 Amazon EKS 클러스터 생성을 참조하세요.
11. Amazon ECS 또는 Amazon EKS 클러스터를 구성하고 us-east-1에서 생성한 Amazon ECS 작업 정의 또는 Amazon EKS 포드를 시작합니다 AWS 리전. 제품이 라이브 상태가 되기 전에 이 테스트 프로세스 중에만 이 리전을 사용해야 합니다.
12. `RegisterUsage`에서 유효한 응답을 받은 후에 컨테이너 제품을 생성할 수 있습니다. 궁금한 점은 AWS Marketplace 판매자 작업 팀에 문의하세요.

RegisterUsage Java 예제

다음 예제에서는 AWS SDK for Java 및 AWS Marketplace 측정 서비스를 사용하여 `RegisterUsage` 작업을 호출합니다. 서명 확인은 선택 사항이지만 서명 확인을 원한다면 필요한 디지털 서명 확인 라이브러리를 추가해야 합니다. 이번 예제는 설명을 돕기 위한 참고용일 뿐입니다.

```
import com.amazonaws.auth.PEM;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMetering;
import com.amazonaws.services.marketplacemetering.AWSMarketplaceMeteringClientBuilder;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageRequest;
import com.amazonaws.services.marketplacemetering.model.RegisterUsageResult;
import com.amazonaws.util.json.Jackson;
import com.fasterxml.jackson.databind.JsonNode;
import com.nimbusds.jose.JWSObject;
import com.nimbusds.jose.JWSVerifier;
import com.nimbusds.jose.crypto.RSASSAVerifier;
import java.io.ByteArrayInputStream;
import java.nio.charset.StandardCharsets;
import java.security.PublicKey;
import java.security.Security;
import java.security.Signature;
import java.security.interfaces.RSAPublicKey;
import java.util.Base64;
import java.util.Optional;
import java.util.UUID;
import org.bouncycastle.jce.provider.BouncyCastleProvider;
```

```
/**
 * Class for making calls out to &MKT; Metering Service.
 */
class RegisterUsage {

    private static final String PRODUCT_CODE = ".......";

    private final AWSMarketplaceMetering registerUsageClient;
    private final SignatureVerifier signatureVerifier;
    private final int publicKeyVersion;

    public RegisterUsage(final SignatureVerifier signatureVerifier) {
        this.signatureVerifier = signatureVerifier;
        this.publicKeyVersion = PublicKeyProvider.PUBLIC_KEY_VERSION;
        this.registerUsageClient =
 AWSMarketplaceMeteringClientBuilder.standard().build();
    }

    /**
     * Shows how to call RegisterUsage client and verify digital signature.
     */
    public void callRegisterUsage() {
        RegisterUsageRequest request = new RegisterUsageRequest()
                .withProductCode(PRODUCT_CODE)
                .withPublicKeyVersion(publicKeyVersion)
                .withNonce(UUID.randomUUID().toString());

        // Execute call to RegisterUsage (only need to call once at container startup)
        RegisterUsageResult result = this.registerUsageClient.registerUsage(request);

        // Verify Digital Signature w/o JWT
        boolean isSignatureValid = this.signatureVerifier.verify(request, result);
        if (!isSignatureValid) {
            throw new RuntimeException("Revoke entitlement, digital signature
 invalid.");
        }
    }
}

/**
 * Signature verification class with both a JWT-library based verification
 * and a non-library based implementation.
 */
class SignatureVerifier {
```

```
    private static BouncyCastleProvider BC = new BouncyCastleProvider();

    private static final String SIGNATURE_ALGORITHM = "SHA256withRSA/PSS";

    private final PublicKey publicKey;

    public SignatureVerifier(PublicKeyProvider publicKeyProvider) {
        this.publicKey = publicKeyProvider.getPublicKey().orElse(null);
        Security.addProvider(BC);
    }

    /**
     * Example signature verification using the NimbusJOSEJWT library to verify the JWT
 Token.
     *
     * @param request RegisterUsage Request.
     * @param result  RegisterUsage Result.
     * @return true if the token matches.
     */
    public boolean verifyUsingNimbusJOSEJWT(final RegisterUsageRequest request, final
 RegisterUsageResult result) {
        if (!getPublicKey().isPresent()) {
            return false;
        }

        try {
            JWSVerifier verifier = new RSASSAVerifier((RSAPublicKey)
 getPublicKey().get());
            JWSObject jwsObject = JWSObject.parse(result.getSignature());
            return jwsObject.verify(verifier) &&
 validatePayload(jwsObject.getPayload().toString(), request, result);
        } catch (Exception e) {
            // log error
            return false;
        }
    }

    /**
     * Example signature verification without any JWT library support.
     *
     * @param request RegisterUsage Request.
     * @param result  RegisterUsage Result.
     * @return true if the token matches.
     */
```

```
   public boolean verify(final RegisterUsageRequest request, final RegisterUsageResult
result) {
       if (!getPublicKey().isPresent()) {
           return false;
       }
       try {
           String[] jwtParts = result.getSignature().split("\\.");
           String header = jwtParts[0];
           String payload = jwtParts[1];
           String payloadSignature = jwtParts[2];

           Signature signature = Signature.getInstance(SIGNATURE_ALGORITHM, BC);
           signature.initVerify(getPublicKey().get());
           signature.update(String.format("%s.%s", header,
payload).getBytes(StandardCharsets.UTF_8));
           boolean verified = signature.verify(Base64.getUrlDecoder()
                   .decode(payloadSignature.getBytes(StandardCharsets.UTF_8)));

           String decodedPayload = new String(Base64.getUrlDecoder().decode(payload));
           return verified && validatePayload(decodedPayload, request, result);
       } catch (Exception e) {
           // log error
           return false;
       }
   }

   /**
    * Validate each value in the returned payload matches values originally
    * supplied in the request to RegisterUsage. TimeToLiveInMillis and
    * PublicKeyExpirationTimestamp will have the values in the payload compared
    * to values in the signature
    */
   private boolean validatePayload(final String payload, final RegisterUsageRequest
request,
                                   final RegisterUsageResult result) {
       try {
           JsonNode payloadJson = Jackson.getObjectMapper().readTree(payload);
           boolean matches = payloadJson.get("productCode")
                   .asText()
                   .equals(request.getProductCode());
           matches = matches && payloadJson.get("nonce")
                   .asText()
                   .equals(request.getNonce());
           return matches = matches && payloadJson.get("publicKeyVersion")
```

```
                    .asText()
                    .equals(String.valueOf(request.getPublicKeyVersion()));

        } catch (Exception ex) {
            // log error
            return false;
        }
    }

    private Optional<PublicKey> getPublicKey() {
        return Optional.ofNullable(this.publicKey);
    }
}

/**
 * Public key provider taking advantage of the &AWS; PEM Utility.
 */
class PublicKeyProvider {
    // Replace with your public key. Ensure there are new-lines ("\n") in the
    // string after "-----BEGIN PUBLIC KEY-----\n" and before "\n-----END PUBLIC
 KEY-----".
    private static final String PUBLIC_KEY =
            "-----BEGIN PUBLIC KEY-----\n"
                    + "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDdlatRjRjogo3WojgGHFHYLugd
\n"
                    + "UWAY9iR3fy4arWNA1KoS8kVw33cJibXr8bvwUAUparCwlvdbH6dvEOfou0/gCFQs
\n"
                    + "HUfQrSDv+MuSUMAe8jzKE4qW+jK+xQU9a03GUnKHkkle+Q0pX/g6jXZ7r1/xAK5D
\n"
                    + "o2kQ+X5xK9cipRgEKwIDAQAB\n"
                    + "-----END PUBLIC KEY-----";

    public static final int PUBLIC_KEY_VERSION = 1;

    public Optional<PublicKey> getPublicKey() {
        try {
            return Optional.of(PEM.readPublicKey(new ByteArrayInputStream(
                    PUBLIC_KEY.getBytes(StandardCharsets.UTF_8))));
        } catch (Exception e) {
            // log error
            return Optional.empty();
        }
    }
```

```
}
```

AWS Marketplace Metering Service를 사용하여 컨테이너 제품의 사용자 지정 측정 구성

> **Note**
>
> Amazon EKS 배포의 경우 소프트웨어는 서비스 계정에 대한 IAM 역할(IRSA)을 사용하여 API 작업에 대한 `MeterUsage` API 호출에 서명해야 합니다. EKS Pod Identity, 노드 역할 또는 장기 액세스 키를 사용하는 것은 지원되지 않습니다.
> Amazon ECS 배포의 경우 소프트웨어는 Amazon ECS 태스크 IAM 역할을 사용하여 API 작업에 대한 `MeterUsage` API 호출에 서명해야 합니다. 노드 역할 또는 장기 액세스 키 사용은 지원되지 않습니다.

AWS Marketplace 컨테이너 제품은 제품당 최대 24개의 다양한 요금 차원에 대해 사용자 지정 측정이 가능합니다. 각 차원에는 연결된 장기 계약 요금이 있을 수 있습니다. 사용자 지정 측정을 활성화하려면 컨테이너 제품을 AWS Marketplace 측정 서비스와 통합합니다. `MeterUsage` API 작업을 사용하여 결제를 위해 해당 사용량 AWS 에 대한 자체 요금 단위 및 사용자 지정 측정을 로 정의할 수 있습니다. 다음 섹션에서는 컨테이너 제품에 대한 사용자 지정 측정 구성 방법을 보여줍니다.

가격 차원은 AWS Marketplace Management Portal (판매자 포털)에서 제품을 생성할 때 한 번, `MeterUsage` 작업을 수행하기 위해 소프트웨어에서 한 번, 두 위치에 정의됩니다. 이 2단계 방법을 통해 후속 제안이 공개되기 전에 의도한 대로 작동하는지 확인할 수 있습니다.

사용자 지정 측정을 설정하려면 사용 범주, 단위 유형 및 요금 차원을 선택해야 합니다.

- 사용 범주 - 사용 범주는 구매자가 제품을 이해하고 사용 방법을 파악하는 데 도움이 됩니다.
- 단위 유형 - 단위 유형은 청구 측정 단위를 정의합니다. GBps 또는 MBps 단위로 측정되는 대역폭, 호스트 수, MB, GB 또는 TB 단위로 측정되는 데이터를 예로 들 수 있습니다.
- 요금 차원 - 요금 차원은 사용자, 스캔, vCPU, 배포된 에이전트 같은 단위당 가격을 설정한 기능 또는 서비스를 나타냅니다. 요금 차원은 공개됩니다. 하지만 여전히 공개 제품에 대한 비공개 및 기존 보유 라이선스 사용(BYOL) 제안을 정의할 수 있습니다. 측정 레코드를 보낼 때 요금은 포함하지 마세요. 단위 수량을 측정하면 제품을 생성할 때 정의한 가격과 함께 사용하여 구매자의 청구서가 계산됩니다.

  제품 요금이 사전 정의된 범주 또는 단위 유형에 맞지 않는 경우 일반 단위 범주를 선택하면 됩니다. 그 후 차원 설명을 사용하여 단위가 무엇인지 설명합니다.

선택 사항으로, 추적하는 속성별로 사용량을 할당량에 분배할 수 있습니다. 할당량은 구매자에게 태그로 표시됩니다. 이러한 태그를 통해 구매자는 태그 값에 따라 사용량으로 분할된 비용을 볼 수 있습니다. 예를 들어 사용자별로 요금을 부과하고 사용자에게 "Department" 속성이 있는 경우 키가 "Department"인 태그와 값마다 하나의 할당을 사용하여 사용량 할당을 생성할 수 있습니다. 이렇게 해도 보고하는 가격, 크기 또는 총 사용량이 변하지는 않지만, 고객은 제품에 적합한 범주별로 비용을 볼 수 있습니다.

1시간마다 측정 기록을 보내는 것이 좋습니다. 하지만 일별 또는 월별로 사용량을 집계할 수도 있습니다. 중단이 발생할 경우 구매자 소프트웨어 사용을 집계하여 이후 시간 측정 시 보낼 수 있습니다. 시간당 두 개 이상의 레코드를 보낼 수 없습니다.

컨테이너 제품의 AWS Marketplace Metering Service API를 사용자 지정 측정 요금과 통합하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 워크숍의 Integrate with custom metering lab을 참조하세요.

> **Important**
>
> 무료 평가판 및 선결제 권한은 시간당 수준에서 추적됩니다. 따라서 이러한 기록을 별도로 보내면 구매자에게 요금이 과다 청구될 수 있습니다.

주제

- 사용자 지정 측정을 위한 사전 조건
- MeterUsage 통합 테스트
- MeterUsage의 오류 처리
- 공급업체 측정 태깅(선택 사항)
- 코드 예제
- 사용자 지정 측정 기능을 사용하여 컨테이너 제품을 AWS Marketplace Metering Service 및와 통합 AWS SDK for Java

사용자 지정 측정을 위한 사전 조건

제품을 게시하기 전에 다음을 수행해야 합니다.

1. 에서 새 컨테이너 제품을 AWS Marketplace Management Portal생성하고 해당 제품 코드를 기록해 둡니다.

2. 를 호출하는 데 필요한 IAM 권한으로 애플리케이션을 실행하는 작업 또는 포드에 AWS Identity and Access Management (IAM) 역할을 사용합니다`MeterUsage`. IAM 관리형 정책 `AWSMarketplaceMeteringRegisterUsage`에 이러한 권한이 있습니다. 정책에 대한 자세한 내용은 관리형 정책 참조의 AWSMarketplaceMeteringFullAccess를 참조하세요. AWS
3. (선택 사항) AWS CloudTrail 로깅을 보려면 작업 또는 포드 정의에서 로깅을 활성화하는 것이 좋습니다.
4. 정의하는 모든 요금 차원에 대한 레코드를 사용하여 `MeterUsage` API 작업 호출을 테스트합니다.

`MeterUsage` 통합 테스트

게시를 위해 이미지를 AWS Marketplace 에 제출하기 전에 `MeterUsage` 작업을 사용하여 통합을 테스트합니다.

제품을 나열하는 데 사용 AWS 계정 중인를 사용하여 Amazon Elastic Container Service(Amazon ECS) 또는 Amazon Elastic Kubernetes Service(Amazon EKS)에서 제품을 실행하여 컨테이너 이미지`MeterUsage`에서를 호출합니다 AWS Marketplace. 측정 통합은 하드 코딩 AWS 리전이 아닌 동적으로를 설정해야 합니다. 그러나 테스트할 때 AWS Marketplace 운영 팀이 해당 리전의 로그 작업을 확인할 수 있도록 미국 동부(버지니아 북부) 리전에서 유료 컨테이너가 포함된 Amazon ECS 작업 또는 Amazon EKS 포드를 하나 이상 시작합니다.

> **Note**
>
> - 제품이 Amazon ECS와 Amazon EKS를 모두 지원하는 경우 판매자는 Amazon EKS에서 시작하기만 하면 됩니다. 그러면 AWS에서 통합을 검증할 수 있습니다.
> - 제품을 일반 출시하기 전과 새 차원을 추가한 후 모든 차원을 테스트합니다. 컨테이너 제품과 연결된 각 차원에 대한 측정 레코드를 보내지 않으면 요청에 실패하는 오류가 발생합니다.

제품과 함께 필요한 모든 메타데이터 및 요금 정보가 게시될 때까지는 통합에 대한 완전한 테스트가 어렵습니다. 요청된 경우 AWS Marketplace 카탈로그 운영 팀은 측정 레코드의 수신을 확인할 수 있습니다.

`MeterUsage`의 오류 처리

컨테이너 이미지가 `MeterUsage` 작업과 통합된 상태에서 컨테이너 시작 시 `ThrottlingException` 외에 다른 예외가 발생하면 컨테이너를 종료하여 무단 사용을 방지해야 합니다.

ThrottlingException 이외의 예외는 초기 MeterUsage 호출에서만 발생합니다. 하지만 이후에 동일한 Amazon ECS 작업 또는 Amazon EKS 포드에서 호출할 때에는 작업 또는 포드가 여전히 실행 중인 동안 고객이 구독을 해지해도 CustomerNotSubscribedException이 throw되지 않습니다. 이러한 고객은 구독을 취소한 후에도 사용량을 추적하여 컨테이너 실행에 대한 요금이 계속 청구됩니다.

MeterUsage의 일반적인 오류에 대한 자세한 설명은 AWS Marketplace Metering Service API 참조의 MeterUsage를 참조하세요. 각 AWS SDK 프로그래밍 언어에는 추가 정보를 위해 참조할 수 있는 오류 처리 지침 세트가 있습니다.

공급업체 측정 태깅(선택 사항)

공급업체 측정 태깅을 사용하면 독립 소프트웨어 개발 판매 회사(ISV)는 구매자에게 소프트웨어 사용에 대한 보다 세밀한 통찰력을 제공하고 구매자의 비용 할당을 도와줄 수 있습니다.

구매자의 소프트웨어 사용에 태그를 지정하는 여러 가지 방법이 있습니다. 한 가지 방법은 먼저 구매자에게 비용 할당에서 보고 싶은 것이 무엇인지 물어보는 것입니다. 그 후 구매자 계정에 대해 추적하는 모든 속성에 사용량을 분할할 수 있습니다. 속성의 예로는 AccountId, Business Unit, Cost Centers 및 기타 제품 관련 메타데이터가 있습니다. 이러한 속성은 구매자에게 태그로 노출됩니다. 구매자는 태그를 사용하여 AWS 결제 콘솔(https://console.aws.amazon.com/costmanagement/)의 태그 값을 기준으로 사용량으로 분할된 비용을 볼 수 있습니다. 공급업체 측정 태깅은 판매자가 신고한 가격, 크기 또는 총 사용량을 변경하지 않습니다. 이를 통해 고객은 제품에 적합한 범주별로 비용을 볼 수 있습니다.

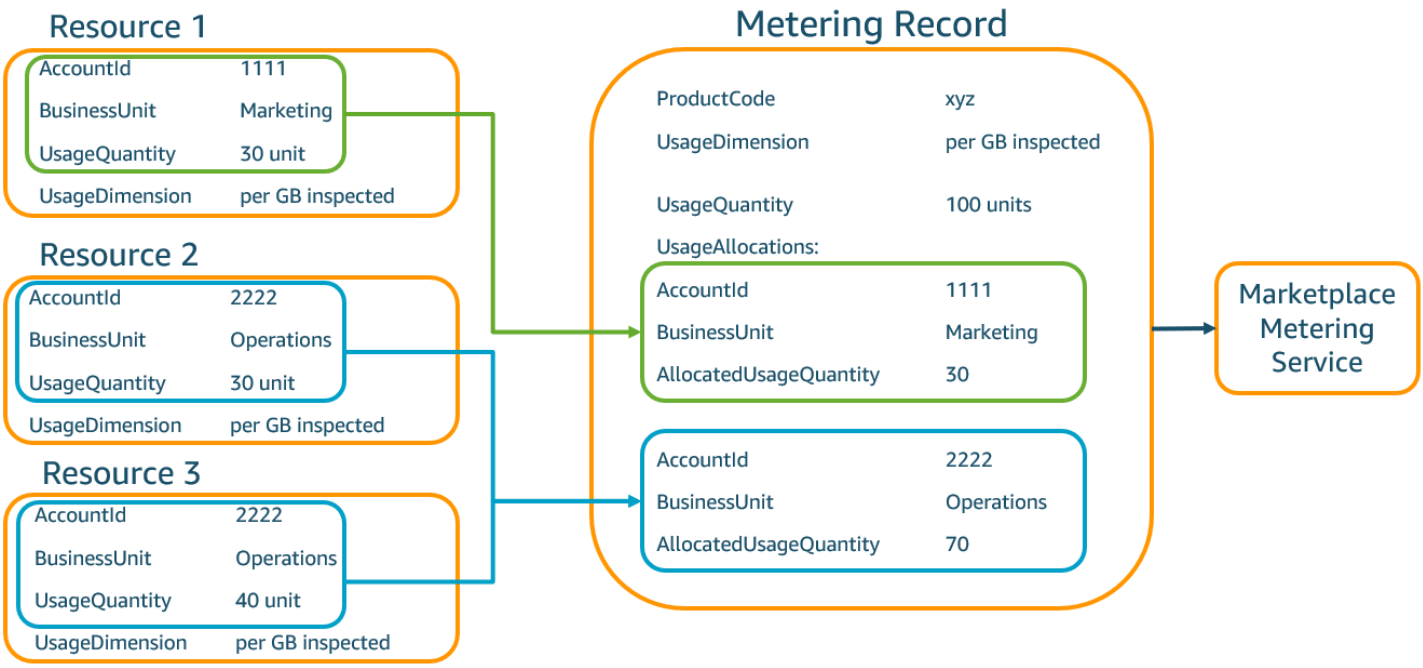
일반적으로 구매자는 AWS 계정하나로 제품을 구독합니다. 또한 구매자의 수많은 사용자가 동일한 제품 구독에 연결됩니다. AccountId 키가 있는 태그를 사용하여 사용량 할당을 생성한 다음, 각 사용자에게 사용량을 할당할 수 있습니다. 이 경우 구매자는 과금 정보 및 비용 관리 콘솔에서 AccountId 태그를 활성화하고 개별 사용자 사용량을 분석할 수 있습니다.

판매자 환경

판매자는 모든 리소스 사용량을 집계하는 대신 동일한 태그 세트를 사용하여 리소스 측정 기록을 집계할 수 있습니다. 예를 들어 판매자는 UsageAllocations의 여러 버킷을 포함하는 측정 기록을 구성할 수 있습니다. 각 버킷은 태그 세트의 UsageQuantity(예: AccountId, BusinessUnit)를 나타냅니다.

다음 다이어그램의 리소스 1은 고유한 AccountId 및 BusinessUnit 태그 세트가 있으며 Metering Record(측정 기록)에 단일 항목으로 표시됩니다.

리소스 2와 리소스 3은 동일한 AccountId, 2222, BusinessUnit 및 Operations 태그를 갖고 있습니다. 따라서 두 리소스는 측정 기록의 단일 UsageAllocations 항목으로 결합됩니다.



판매자는 태그가 없는 리소스를 사용량 수량이 할당된 단일 `UsageAllocation`으로 결합하고 `UsageAllocations`의 항목 중 하나로 전송할 수도 있습니다.

다음과 같은 제한이 있습니다.

- 태그 수 - 5
- `UsageAllocations` 크기(카디널리티) - 2,500

다음 사항을 확인합니다.

- 태그 키 및 값에 허용되는 문자 – a-zA-Z0-9+ -=._:\/@
- `UsageAllocation` 목록의 최대 태그 수 - 5
- 두 `UsageAllocations`의 태그가 같으면(즉, 동일한 태그 키와 값의 조합) 안 됩니다. 같을 경우 둘 다 동일한 `UsageAllocation`을 사용해야 합니다.
- `UsageAllocation`의 `AllocatedUsageQuantity` 합계는 총 사용량인 `UsageQuantity`와 같아야 합니다.

구매자 경험

다음 표는 구매자가 `AccountId` 및 `BusinessUnit` 공급업체 태그를 활성화한 이후의 구매자 경험 예시를 보여줍니다.

이 예에서 구매자는 비용 사용 보고서에서 할당된 사용량을 볼 수 있습니다. 공급업체 측정 태그는 “`aws:marketplace:isv`” 접두사를 사용합니다. 구매자는 과금 정보 및 비용 관리의 비용 할당 태그 아래에 있는 AWS생성 비용 할당 태그에서 태그를 활성화할 수 있습니다.

비용 사용 보고서의 첫 번째 행과 마지막 행은 판매자 환경 예제처럼 판매자가 측정 서비스에 보내는 내용과 관련이 있습니다.

비용 사용 보고서(단순)

ProductCode	구매자	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	네트워크: 검사한 크기(GB)당	70	2222	운영
xyz	111122223333	네트워크: 검사한 크기(GB)당	30	3333	Finance
xyz	111122223333	네트워크: 검사한 크기(GB)당	20	4444	IT
xyz	111122223333	네트워크: 검사한 크기(GB)당	20	5555	마케팅
xyz	111122223333	네트워크: 검사한 크기(GB)당	30	1111	마케팅

코드에 대한 예는 사용량 할당 태그 지정을 사용하는 `MeterUsage` 코드 예제(선택 사항) 단원을 참조하세요.

코드 예제

다음 코드 예제는 컨테이너 제품을 제품 게시 및 유지 관리에 필요한 AWS Marketplace APIs와 통합하는 데 도움이 되도록 제공됩니다.

사용량 할당 태그 지정을 사용하는 **MeterUsage** 코드 예제(선택 사항)

다음 코드 예제는 소비 요금 모델이 적용되는 컨테이너 제품과 관련이 있습니다. Python 예제에서는 적절한 사용량 할당 태그가 달린 측정 레코드를 AWS Marketplace 에 전송하여 고객에게 사용한 만큼만 지불 요금을 청구합니다.

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited

# Import AWS Python SDK
import boto3
import time

usageRecord = [
    {
        "AllocatedUsageQuantity": 2,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "IT" },
                { "Key": "AccountId", "Value": "123456789" },
            ]

    },
    {
        "AllocatedUsageQuantity": 1,
        "Tags":
            [
                { "Key": "BusinessUnit", "Value": "Finance" },
                { "Key": "AccountId", "Value": "987654321" },
            ]

    }
]
```

```
marketplaceClient = boto3.client("meteringmarketplace")

response = marketplaceClient.meter_usage(
    ProductCode="testProduct",
    Timestamp=int(time.time()),
    UsageDimension="Dimension1",
    UsageQuantity=3,
    DryRun=False,
    UsageAllocations=usageRecord
)
```

`MeterUsage`에 대한 자세한 내용은 AWS Marketplace Metering Service API 참조의 MeterUsage를 참조하세요.

응답의 예

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

사용자 지정 측정 기능을 사용하여 컨테이너 제품을 AWS Marketplace Metering Service 및와 통합 AWS SDK for Java

AWS Marketplace 컨테이너 제품은 제품당 최대 24개의 다양한 요금 차원에 대해 사용자 지정 측정이 가능합니다. 사용자 지정 측정을 활성화하려면 컨테이너 제품을 AWS Marketplace Metering Service와 통합합니다. `MeterUsage` API 작업을 사용하여 AWS 대금 청구에 사용할 자체 요금 단위와 사용량에 대한 사용자 지정 측정을 정의할 수 있습니다. 다음 예제에서는를 사용하여 AWS Marketplace 측정 서비스 `MeterUsage` 작업과 AWS SDK for Java 통합하는 구현을 간략하게 설명합니다.

자세한 내용은 `MeterUsage` Java 예제 섹션을 참조하세요. 아래 단계 중 대부분은 언어에 상관없이 적용됩니다.

예: AWS Marketplace Metering Service 통합

1. AWS Marketplace Management Portal에 로그인합니다.
2. 자산에서 컨테이너를 선택하여 새 컨테이너 제품 생성을 시작합니다. 제품을 만들면 제품에 대한 제품 코드가 생성되어 컨테이너 이미지와 통합할 수 있습니다. AWS Identity and Access Management (IAM) 권한 설정에 대한 자세한 내용은 섹션을 참조하세요the section called “AWS Marketplace 측정 및 권한 부여 API 권한”.
3. 퍼블릭 AWS Java SDK를 다운로드합니다.

> **Important**
>
> Amazon Elastic Kubernetes Service(Amazon EKS) 에서 측정 API 작업을 호출하려면 지원되는 AWS SDK를 사용하고 Kubernetes 1.13 이상을 실행하는 Amazon EKS 클러스터에서 실행해야 합니다.

4. 각 차원 사용에 대해 1시간마다 작업 또는 포드에서 `MeterUsage` 작업을 호출합니다. 이 API 작업은 고유한 `Dimension`, `Resource`, `Hour` 조합에 대해 하나의 측정 레코드를 수락합니다. 리소스는 Amazon Elastic Container Service(Amazon ECS) 작업 또는 Amazon EKS 포드입니다.

```
{
    "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**
>
> 에 연결할 때 일시적인 문제가 발생할 수 있습니다 AWS Marketplace Metering Service. AWS Marketplace 단기 중단 또는 네트워크 문제를 방지하기 위해 지수 백오프를 통해 최대 30분 동안 재시도를 구현하는 것이 좋습니다.

5. `MeterUsage` 호출을 포함하는 새 버전의 컨테이너 이미지를 다시 빌드하고, 컨테이너에 태그를 지정하고, 컨테이너를 Amazon Elastic Container Registry(Amazon ECR)처럼 Amazon ECS 또는 Amazon EKS와 호환되는 Docker 레지스트리로 푸시합니다. Amazon ECR을 사용하는 경우에는 Amazon ECS 작업 또는 Amazon EKS 포드를 시작하는 계정에 Amazon ECR 리포지토리에 대한 권한이 있는지 확인합니다. 그렇지 않으면 작업이 실패합니다.

6. 다음 코드 예제에 정의된 대로 컨테이너가 `MeterUsage`를 호출할 수 있는 권한을 부여하는 IAM 역할을 생성합니다. Amazon ECS 태스크 또는 Amazon EKS 포드 정의의 태스크 역할 파라미터에 이 AWS Identity and Access Management (IAM) 역할을 제공해야 합니다.

JSON

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

7. 와 통합된 컨테이너를 참조 AWS Marketplace 하고 6단계에서 생성한 IAM 역할을 참조하는 Amazon ECS 작업 또는 Amazon EKS 포드 정의를 생성합니다. 로깅을 보려면 작업 정의에서 AWS CloudTrail 로깅을 활성화합니다.
8. 작업 또는 포드를 실행할 Amazon ECS 또는 Amazon EKS 클러스터를 생성합니다. Amazon ECS 클러스터 생성에 대한 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 클러스터 생성을 참조하세요. Kubernetes 버전 1.1.3.x 이상을 사용하여 Amazon EKS 클러스터를 생성하는 방법에 대한 자세한 내용은 Amazon EKS 클러스터 생성을 참조하세요.
9. Amazon ECS 또는 Amazon EKS 클러스터를 구성하고 us-east-1 AWS 리전의 8단계에서 생성한 Amazon ECS 작업 정의 또는 Amazon EKS 포드를 시작합니다. 제품이 라이브 상태가 되기 전에 이 테스트 프로세스 중에만 이 리전을 사용해야 합니다.
10. 제품에 대해 게시되는 차원마다 `MeterUsage`에서 유효한 응답을 받으면 컨테이너 제품 생성을 시작할 수 있습니다. 궁금한 점은 AWS Marketplace 판매자 작업 팀에 문의하세요.

`MeterUsage` Java 예제

다음 코드 예제에서는 AWS SDK for Java 및 AWS Marketplace 측정 서비스를 사용하여 `MeterUsage` 작업을 호출합니다.

다음은 `UsageAllocations` 없이 `MeterUsage` 작업을 호출하는 코드 예제입니다.

```
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);
    }
}
```

다음은 `UsageAllocations`를 사용하여 `MeterUsage` 작업을 호출하는 코드 예제입니다.

```
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();
    }
```

를 사용한 컨테이너 제품의 계약 요금 AWS License Manager

계약 요금이 적용되는 컨테이너 기반 제품의 경우 AWS License Manager 를 사용하여 라이선스를 제품과 연결할 수 있습니다. AWS License Manager 는 애플리케이션이 고객이 구매한 라이선스(권한이라고도 함)를 추적하고 업데이트할 수 있는 라이선스 관리 도구입니다. 이 섹션에서는 제품을와 통합하는 방법에 대한 정보를 제공합니다 AWS License Manager. 통합이 완료되면 제품 목록을 AWS Marketplace에 게시할 수 있습니다.

License Manager를 Amazon EKS Anywhere, Amazon ECS Anywhere, Amazon Elastic Compute Cloud(Amazon EC2) 또는 온프레미스 인프라용 AWS Marketplace for Containers Anywhere 제품과 통합하는 경우의 지침을 따르세요AWS Marketplace for Containers Anywhere와 License Manager 통합.

에 대한 자세한 내용은 AWS License Manager 사용 설명서 및 AWS CLI 명령 AWS License Manager 참조의 AWS License Manager 섹션을 참조하세요.

컨테이너 제품과 계약 요금 AWS License Manager 의 통합에 대한 자세한 내용은 AWS Marketplace 판매자 워크숍의 선결제와 통합 실습을 참조하세요.

주제

- 라이선스 모델

- AWS License Manager 통합 사전 조건
- 컨테이너 제품을 License Manager와 통합
- License Manager API 작업
- 라이선스 갱신 및 업그레이드
- AWS Marketplace for Containers Anywhere와 License Manager 통합

라이선스 모델

AWS Marketplace 와의 통합은 두 가지 라이선스 모델을 AWS License Manager 지원합니다.

- 구성 가능한 라이선스 모델
- 계층형 라이선스 모델

구성 가능한 라이선스 모델

구성 가능한 라이선스 모델(정량화 가능한 라이선스 모델이라고도 함)은 구매자가 라이선스를 조달한 후 구매자에게 특정 수량의 리소스를 부여합니다.

판매자는 요금 차원과 단가를 설정합니다. 그러면 구매자는 구매하려는 리소스 수량을 선택할 수 있습니다.

Example 요금 차원 및 단가 예시

판매자는 요금 차원(예: 데이터 백업) 및 단가(예: 30 USD/단위)를 설정할 수 있습니다.

구매자는 5, 10 또는 20단위를 구매하기로 선택할 수 있습니다.

판매자의 제품은 사용량을 추적하고 측정하여 소비된 리소스 양을 측정합니다.

구성 모델의 경우 다음 두 가지 방법 중 하나로 권한이 계산됩니다.

- 드로다운 라이선스
- 플로팅 라이선스

드로다운 라이선스

라이선스를 사용한 만큼 허용된 라이선스 수량 풀에서 차감됩니다. 해당 권한은 영구적으로 체크아웃되며 라이선스 풀로 반환할 수 없습니다.

Example 제한된 양의 데이터 처리 예시

한 사용자에게 500GB의 데이터를 처리할 권한이 있습니다. 사용자가 데이터를 계속 처리하면 500GB 라이센스가 모두 소진될 때까지 500GB 풀에서 수량이 차감됩니다.

드로다운 라이선스의 경우 `CheckoutLicense` API 작업을 사용하여 소비된 라이선스 단위(권한)를 체크아웃할 수 있습니다.

Example 단위 수/년 동안 Amazon S3로 백업

1년 동안 데이터에 대해 최대 1,024단위까지 Amazon Simple Storage Service에 백업할 수 있는 스토리지 제품이 있습니다. Amazon EC2 인스턴스를 여러 개 사용하여 애플리케이션을 시작할 수 있습니다. 애플리케이션에는 데이터를 추적하고 집계하는 메커니즘이 있습니다. 소프트웨어는 백업이 발생할 때마다 또는 고정된 간격으로 제품 ID로 `CheckoutLicense` API 작업을 호출하여 소비량을 업데이트합니다.

이 예시에서는 소프트웨어가 `CheckoutLicense` API 작업을 호출하여 데이터 10단위를 체크아웃합니다. 총 용량이 고객이 구매한 백업 한도에 도달하면 API 직접 호출이 실패합니다.

요청

```
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"
```

응답

```
{"CheckoutType": "PERPETUAL",
"EntitlementsAllowed": [{
"Name": "IntermediateTier",
"Units": "None"
}],
"Expiration": "2021-04-22Tl9:02:36",
"IssuedAt": "2021-04-22Tl8:02:36",
"LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
"LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

플로팅 라이선스

라이선스가 사용 후 허용된 라이선스 수량 풀로 반환됩니다.

플로팅 라이선스의 경우 애플리케이션은 리소스가 사용될 때 `CheckoutLicense` API 작업을 사용하여 권한 풀에서 권한을 체크아웃합니다. `CheckoutLicense` API 작업의 응답에는 체크아웃의 고유 식별자인 라이선스 소비 토큰이 포함되어 있습니다. 라이선스 소비 토큰은 권한을 라이선스 풀로 다시 체크인하거나 체크아웃을 연장하는 등 체크아웃된 권한에 대한 추가 작업을 수행하는 데 사용할 수 있습니다.

리소스가 더 이상 사용되지 않으면 `CheckInLicense` API 작업을 사용하여 권한을 다시 풀에 체크인합니다.

```
aws license-manager check-in-license --license-consumption-token
 "f1603b3c1f574b7284db84..."
```

권한 체크인이 실패하면(예: 애플리케이션 충돌 발생) 60분 후에 자동으로 권한이 다시 풀에 체크인됩니다. 리소스가 60분 넘게 사용 중이면 리소스가 사용되는 동안에는 `ExtendLicenseConsumption` API 작업을 사용하여 풀에서 권한을 체크아웃한 상태로 유지하는 것이 좋습니다.

```
aws license-manager extend-license-consumption --license-consumption-token
 "f1603b3c1f574b7284..."
```

Example 고정된 상한의 사용자 수 예시

사용자는 애플리케이션에서 동시 사용자 500명을 이용할 수 있습니다. 사용자가 로그인하고 로그아웃하면 사용자가 차감된 후 사용자 500명 풀로 반환됩니다. 하지만 고정된 상한은 동시 사용자 500명이기 때문에 애플리케이션이 풀에서 500명을 초과하는 사용자를 차감할 수 없습니다.

플로팅 권한의 경우 `CheckInLicense` API 작업을 사용하여 라이선스 단위를 권한 풀에 반환할 수 있습니다.

Example 1년간 동시 사용자 수 예시

제품 요금은 동시 사용자 수를 기준으로 책정됩니다. 고객이 1년간 사용자 10명 라이선스를 구매합니다. 고객이 AWS Identity and Access Management (IAM) 권한을 제공하여 소프트웨어를 시작합니다. 사용자가 로그인하면 애플리케이션이 `CheckoutLicense` API 작업을 호출하여 수량을 1만큼 줄입니다. 사용자가 로그아웃하면 애플리케이션이 `CheckInLicense` API 작업을 호출하여 해당 라이선스를 풀에 반환합니다. `CheckInLicense`를 호출하지 않으면 1시간 후에 라이선스 단위가 자동으로 체크인됩니다.

> **Note**
>
> 다음 요청에서 `key-fingerprint`는 자리 표시자 값이 아니라 모든 라이선스를 게시하는 데 사용되는 fingerprint의 실제 값입니다.

요청

```
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"
```

응답

```
{
  "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"
}
```

계층형 라이선스 모델

계층형 라이선스 모델은 구매자가 라이선스를 조달한 후 특정 수준 또는 계층의 애플리케이션 기능을 이용할 수 있는 권한을 구매자에게 부여합니다.

판매자가 제품 계층(예: 기본, 중급, 프리미엄)을 생성합니다. 그 후 구매자가 사전 정의된 계층 중 하나를 선택합니다.

애플리케이션은 애플리케이션 사용량을 추적하거나 측정할 필요가 없습니다.

계층형 라이선스 모델의 경우 권한은 계산되지 않고, 대신 고객이 조달한 서비스 계층을 나타냅니다.

번들 기능을 함께 제안하려면 계층을 사용하는 것이 좋습니다.

Example 기본, 중급 및 프리미엄 계층의 예시

고객은 세 가지 소프트웨어 티어인 기본, 중급 또는 프리미엄 중 하나를 계약할 수 있습니다. 티어마다 자체 요금이 책정되어 있습니다. 소프트웨어는 `CheckoutLicense` API 작업을 간접적으로 호출하고 가능한 모든 등급을 요청에서 지정하여 고객이 가입한 등급을 식별할 수 있습니다.

요청 응답에는 고객이 구매한 계층에 해당하는 권한이 포함됩니다. 소프트웨어는 이 정보를 기반으로 적절한 고객 경험을 프로비저닝할 수 있습니다.

요청

```
linux-machine  ~]$ aws  license-manager   checkout-license\
--product-sku  "2205b290-19e6-4c76-9eea-377d6bf71a47"  \
--checkout-type  "PROVISIONAL"  \
--key-fingerprint  "aws:294406891311:AWS/Marketplace:issuer-fingerprint" \
--entitlements  "Name=BasicTier,  Unit=None"   "Name=IntermediateTier,  Unit=None" \
 "Name=PremiumTier, Unit=None"
```

응답

```
{
  "CheckoutType": "PROVISIONAL",
  "EntitlementsAllowed": [
    {
      "Name": "IntermediateTier",
      "Units": "None"
    }
},
  "Expiration": "2021-04-22T19:02:36",
  "IssuedAt": "2021-04-22T18:02:36",
  "LicenseArn": "arn:aws:license-manager::294406891311:license:l-16bf01b...",
  "LicenseConsumptionToken": "AKIAIOSFODNN7EXAMPLE"
}
```

AWS License Manager 통합 사전 조건

제품을 게시하기 전에 다음을 수행해야 합니다.

1. 에서 새 컨테이너 제품을 AWS Marketplace Management Portal생성하고 해당 제품 코드를 기록해 둡니다.

   자세한 내용은 개요: 컨테이너 제품 만들기 단원을 참조하십시오.

2. `CheckoutLicense`, `ExtendLicenseConsumption` 및 `CheckInLicense` API 작업을 호출하는 데 필요한 IAM 권한이 있는 애플리케이션을 실행하는 작업 또는 포드에 IAM 역할을 사용합니다.

   필요한 IAM 권한은 다음 IAM 정책에 자세히 설명되어 있습니다.

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"VisualEditor0",
         "Effect":"Allow",
         "Action":[
            "license-manager:CheckoutLicense",
            "license-manager:GetLicense",
            "license-manager:CheckInLicense",
            "license-manager:ExtendLicenseConsumption",
            "license-manager:ListReceivedLicenses"
         ],
         "Resource":"*"
      }
   ]
}
```

3. 정의하는 모든 요금 차원에 대한 레코드를 사용하여 `RegisterUsage` API 작업 호출을 테스트합니다.

컨테이너 제품을 License Manager와 통합

컨테이너 기반 제품을 License Manager와 통합하는 방법

1. IAM 권한을 설정하여 License Manager를 호출합니다. 자세한 내용은 AWS License Manager 통합 사전 조건 단원을 참조하십시오.
2. AWS SDK를 다운로드합니다.

> **Note**
>
> 소프트웨어 내에서 자격 AWS 증명을 구성하지 마세요. 구매자의 AWS 자격 증명은 컨테이너가 Amazon EC2 인스턴스, Amazon ECS 작업 또는 Amazon EKS 포드 내에서 실행 중일 때 런타임에 자동으로 획득됩니다.

3. 제품에 라이선스 확인을 추가합니다.

   제품은 라이선스 확인을 수행해야 하는 모든 위치에서 `CheckoutLicense` API 작업을 호출할 수 있습니다. 라이선스를 확인하려면 제품이 다음 정보를 알아야 합니다.

   1. 신뢰할 수 있는 라이선스 발급자(AWS Marketplace)
   2. 애플리케이션의 제품 SKU(제품 ID)
   3. 이 애플리케이션을 확인할 수 있는 권한

   API 직접 호출은 설정된 요금 라이선스의 종류에 따라 달라집니다.

4. 제품 목록을에 게시합니다 AWS Marketplace.

License Manager API 작업

고객의 License Manager 계정에 저장된 라이선스를 관리하려면 소프트웨어에서 다음 API 작업을 사용하면 됩니다.

- `GetLicense` - 소프트웨어가 쿼리할 수 있는 API. 구매한 라이선스의 상태(예: 만료 또는 곧 만료 예정)를 검색하고 고객에게 상태 알림을 보냅니다.
- `CheckoutLicense` - 사용자가 구매한 라이선스를 검색합니다. 사용자가 라이선스를 어느 정도 사용한 경우 `CheckoutLicense` API 작업을 사용하여 라이선스 수량을 업데이트할 수도 있습니다. `CheckoutLicense`를 사용하면 고객이 사용한 라이선스의 수량을 계속 체크아웃할 수 있습니다. 고객이 라이선스를 모두 소진하면 이 호출에서 오류가 반환됩니다. `CheckoutLicense` 실행에 권장되는 케이던스에 대한 자세한 내용은 the section called “라이선스 갱신 및 업그레이드” 섹션을 참조하세요.
- `ExtendLicenseConsumption` - 플로팅 차원의 경우 소프트웨어가 라이선스를 체크아웃한 후 60분 후에 자동으로 라이선스가 풀에 반환됩니다. 라이선스가 체크아웃된 상태로 유지되는 시간을 연장하려면 `ExtendLicenseConsumption` API 작업을 사용하여 라이선스를 60분 더 연장합니다.

- `CheckInLicense` - 플로팅 차원의 경우 라이선스를 권한 풀에 반환하려면 `CheckInLicense` API 작업을 사용합니다.
- `ListReceivedLicenses` API - 구매자가 구매한 라이선스를 나열합니다.

라이선스 갱신 및 업그레이드

고객은 AWS Marketplace Management Portal에서 라이선스를 갱신하거나 업그레이드할 수 있습니다. 추가 구매를 한 후는 새 권한을 반영하는 라이선스의 새 버전을 AWS Marketplace 생성합니다. 소프트웨어는 동일한 API 작업을 사용하여 새 권한을 읽습니다. 갱신 및 업그레이드를 처리하기 위해 License Manager 통합과 관련하여 아무 것도 할 필요가 없습니다.

라이선스 갱신, 업그레이드, 취소 등으로 인해 제품이 사용되는 동안 제품에서 정기적으로 `CheckoutLicense` API 작업을 호출하는 것이 좋습니다. `CheckoutLicense` API 작업을 정기적으로 사용하면 제품에서 업그레이드 및 만료와 같은 권한 변경을 감지할 수 있습니다.

15분마다 `CheckoutLicense` API 직접 호출을 수행하는 것이 좋습니다.

AWS Marketplace for Containers Anywhere와 License Manager 통합

AWS Marketplace 판매자는 Amazon EKS Anywhere, Amazon ECS Anywhere, Amazon EC2 또는 온프레미스 인프라를 위한 AWS License Manager AWS Marketplace for Containers Anywhere 제품과 통합할 수 있습니다. Amazon ECS Anywhere 다음 섹션에서는 이 통합에 대한 지침을 제공합니다.

사용 가능한 라이선스 모델을 AWS Marketplace포함하여 License Manager와의 통합에 대한 일반적인 내용은 섹션을 참조하세요를 사용한 컨테이너 제품의 계약 요금 AWS License Manager. AWS License Manager에 대한 자세한 내용은 AWS License Manager 사용 설명서 및 AWS CLI 명령 참조의 AWS License Manager 섹션을 참조하세요.

주제

- AWS Marketplace for Containers Anywhere 제품을 License Manager와 통합
- 로컬로 License Manager 통합 테스트
- Amazon EKS에서 License Manager 통합 테스트
- 플로팅 라이선스 권한과 License Manager
- 온프레미스 배포를 위해 License Manager와 통합하는 모범 사례
- LicenseManagerCredentialsProvider - Java 구현
- LicenseManagerCredentialsProvider - Golang 구현

AWS Marketplace for Containers Anywhere 제품을 License Manager와 통합

다음 지침에 따라 AWS Marketplace for Containers Anywhere 제품을와 통합합니다 AWS License Manager.

AWS Marketplace for Containers Anywhere 제품을 License Manager와 통합하려면

1. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
2. 다음 단계를 수행하여 컨테이너 제품의 제품 ID를 생성합니다. 이후 단계에서 라이선스 확인을 위해 컨테이너 이미지에서 이 ID를 사용하게 됩니다.

   a. 메뉴 모음에서 [Assets(자산)]를 확장하고 [Container(컨테이너)]를 선택합니다.

   b. 고객에게 표시되는 제품 이름을 입력하고 생성을 선택합니다. 이 이름은 나중에 변경할 수 있습니다.

   c. 제품 ID를 적어 둡니다. 제품 요금 세부 정보를 생성하거나 업데이트할 때 필요합니다.

   > **Tip**
   >
   > 제품 ID를 분실한 경우 자산 메뉴에서 컨테이너를 AWS Marketplace Management Portal 선택하여에서 찾을 수 있습니다. 컨테이너 페이지에는 제품 및 연결된 제품 ID 목록이 표시됩니다.

3. 최신 퍼블릭 AWS SDK를 다운로드한 다음 컨테이너 애플리케이션에 설치합니다. AWS에서 빌드할 도구에서 원하는 AWS SDK에 대한 설치 지침을 찾을 수 있습니다.

   > **Note**
   >
   > Amazon EKS Anywhere 또는에서 제공하지 않는 Kubernetes 클러스터에서 License Manager API 작업을 호출하려면 지원되는 AWS SDK를 사용해야 AWS합니다. 지원되는 AWS SDKs. AWS

4. 온프레미스 AWS 뿐만 아니라에 배포된 컨테이너 애플리케이션에 자격 증명을 제공할 수 있도록 사용자 지정 자격 증명 공급자로 AWS License Manager 클라이언트를 생성합니다. 사용자 지정 보안 인증 정보 공급자 `LicenseCredentialProvider`에 대한 전체 소스 코드는 다음 섹션을 참조하세요.

   - `LicenseManagerCredentialsProvider` - Java 구현
   - `LicenseManagerCredentialsProvider` - Golang 구현

LicenseCredentialsProvider는를 추가하여 온프레미스 사용을 위해 AWS SDK의 기본 자격 증명 공급자 체인을 확장합니다LicenseManagerTokenCredentialsProvider. 이렇게 하면 온프레미스 환경에서 License Manager OIDC 발급 ID 토큰을 사용하여 보안 인증 정보가 제공됩니다. LicenseCredentialsProvider의 소스 코드를 애플리케이션 클래스 경로에 포함해야 합니다.

> **Note**
>
> 를 확장DefaultCredentialsProvider하면 온프레미스 환경에서를 실행할 때 AWS 와를 실행할 때 동일한 컨테이너 애플리케이션이 자격 증명을 얻을 수 있습니다. 컨테이너 애플리케이션이 이미 기본값 대신 사용자 지정 보안 인증 정보 공급자 체인을 사용하는 경우 사용자 지정 체인에 LicenseManagerTokenCredentialsProvider를 추가하여 확장할 수도 있습니다.

다음 코드 조각은 Java를 사용하여 AWS License Manager 클라이언트를 생성하는 예제입니다.

```
LicenseManagerClientBuilder clientBuilder =
 LicenseManagerClient.builder().credentialsProvider(LicenseCredentialsProvider.create());
```

5. 제품 제안의 각 유료 컨테이너 이미지에서 aws license-manager checkout-license 명령을 사용하여 CheckoutLicense API 작업을 호출합니다. 그러면 구매자가 애플리케이션의 라이선스를 사용할 권한이 있는지 확인됩니다. 구매자에게 애플리케이션에 대한 권한이 있으면 CheckoutLicense가 성공하고 요청된 권한과 해당 값을 반환합니다. 구매자에게 애플리케이션에 대한 권한이 없으면 CheckoutLicense가 예외를 throw합니다.

   다음은 CheckoutLicense API 작업을 호출할 때 필요한 파라미터입니다.

   - CheckoutType - 유효한 값은 PROVISIONAL 또는 PERPETUAL입니다.
     - 체크아웃된 권한 수량이 풀에서 모두 소진되면 PERPETUAL을 사용합니다.

       예: 구매자에게 500GB의 데이터를 처리할 권한이 있습니다. 구매자가 데이터를 계속 처리하다 보면 수량이 감소하고 500GB 풀이 모두 소진됩니다.
     - 풀에서 권한을 체크아웃하고 사용 후 반환하는 플로팅 라이선스 권한에 PROVISIONAL을 사용합니다.

예: 사용자는 애플리케이션에서 동시 사용자 500명을 이용할 수 있습니다. 사용자가 로그인 또는 로그아웃하면 사용자가 차감되거나 사용자 500명 풀로 반환됩니다. 플로팅 라이선스 권한에 대한 자세한 내용은 플로팅 라이선스 권한과 License Manager 섹션을 참조하세요.

- `ClientToken` - 대/소문자를 구분하는 고유의 식별자입니다. 각각의 고유한 요청에는 무작위 UUID를 사용하는 것이 좋습니다.
- `Entitlements` - 체크아웃할 권한 목록입니다.
  - 기능 권한의 경우 다음과 같이 `Name` 및 `Unit` 속성을 제공합니다.

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

  - 집계된 권한의 경우 다음과 같이 `Name`, `Unit` 및 `Count` 속성을 제공합니다.

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

- `KeyFingerprint` - AWS Marketplace 에서 발급한 라이선스의 key fingerprint는 `aws:294406891311:AWS/Marketplace:issuer-fingerprint`입니다. 이 키 지문을 사용하면 신뢰할 수 없는 엔터티가 AWS Marketplace 아닌에서 라이선스를 발급할 수 있습니다.
- `ProductSKU` - 이전 단계에서 AWS Marketplace Management Portal 에 생성된 제품 ID입니다.

다음 코드 조각은 AWS CLI를 사용하여 `CheckoutLicense` API 작업을 호출하는 예입니다.

```
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**
>
> 라이선스를 확인하려면 컨테이너 애플리케이션이 License Manager를 사용할 수 있도록 아웃바운드 네트워크 액세스가 필요합니다. 온프레미스에 배포된 애플리케이션은 아웃바운드 네트워크 액세스가 불안정하거나 느릴 수 있습니다. 이러한 애플리케이션에는 License Manager를 호출할 때 적절한 재시도 기능이 있어야 합니다. 자세한 내용은 온프레미스 배포를 위해 License Manager와 통합하는 모범 사례 단원을 참조하세요.

6. 정기적으로 `CheckoutLicense` API 작업을 호출하여 AWS Marketplace에서 갱신, 업그레이드 또는 취소로 인해 변경된 고객 라이선스 변경 사항을 확인합니다. 주기는 애플리케이션에 따라 다릅니다. 하루에 한 번 라이선스를 확인하여 구매자의 개입 없이 변경 내용을 자동으로 적용하는 것이 좋습니다.

   온프레미스에 배포된 애플리케이션은 정기적으로 라이선스를 확인하기에는 아웃바운드 네트워크 액세스가 불안정할 수 있습니다. 이 경우 애플리케이션에서 충분한 복원력을 확보할 수 있도록 캐시된 라이선스를 사용해야 합니다. 자세한 내용은 온프레미스 배포를 위해 License Manager와 통합하는 모범 사례 단원을 참조하세요.

7. `CheckoutLicense` 호출을 컨테이너 애플리케이션과 통합한 후에는 변경 사항이 적용된 새 버전의 Docker 컨테이너 이미지를 빌드합니다.

8. 애플리케이션의 차트 Helm을 업데이트하여 License Manager API를 통해 라이선스에 액세스하기 위한 구성을 포함하는 선택적 입력으로 Kubernetes 보안 암호를 수락합니다. 구성 보안 암호에는 License Manager에서 발급한 자격 증명 토큰과 앞서 설명한 사용자 지정 자격 증명 공급자가 컨테이너 애플리케이션이 온프레미스에 배포될 때 License Manager APIs 호출하기 위한 AWS 자격 증명을 가져오는 데 사용할 AWS Identity and Access Management 역할이 포함됩니다. 또한 AWS 리전 을 기본값인 `us-east-1`로 하여 입력으로 추가합니다.

   컨테이너 애플리케이션을 온프레미스에 배포하는 구매자는 컨테이너 제품에 대한 AWS Marketplace 구매자 경험을 통해 Kubernetes 보안 암호를 생성할 수 있습니다. `helm install` 명령의 입력으로 Kubernetes 보안 암호 이름을 제공합니다. 구성 보안 암호는 다음 형식으로 구성됩니다.

```
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. 다음을 AWS License Manager 포함하도록와 통합된 컨테이너 이미지의 차트 Helm에서 애플리케이션 배포 템플릿을 업데이트합니다.

   - 포드용 서비스 계정 - Amazon EKS에서 Helm을 배포하려면 서비스 계정이 필요합니다. 컨테이너 이미지의 서비스 계정에 대한 IAM 역할을 설정하여 License Manager API 작업을 호출할 권한을 얻는 데 사용됩니다. 서비스 계정의 IAM 역할에 대한 자세한 내용은 서비스 계정에 대한 IAM 역할을 참조하세요.
   - 온프레미스 배포에 사용할 라이선스 액세스 권한 - 온프레미스 환경에서 Helm 배포를 위해 License Manager API 작업을 호출하는 데 필요한 보안 인증 정보와 적절한 권한을 제공하려면 라이선스 구성 보안 암호가 필요합니다. 구매자는 AWS Marketplace 구매자 경험에서 라이선스 보안 암호를 생성하여 Helm에 제공합니다.

   다음 코드 조각은 서비스 계정, 라이선스 구성 및 이미지 풀 보안 암호가 포함된 샘플 배포 사양입니다.

```
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**
>
> 라이선스 구성 보안 암호는 선택 사항입니다. 구매자는 이 값을 온프레미스 배포에만 사용합니다. AWS 배포의 경우 배포 사양에 License Manager 통합 이미지에 대한 서비스 계정이 포함되어야 합니다.

10. 다음 섹션의 단계를 수행하여 로컬 및 Amazon EKS에서 License Manager 통합을 테스트합니다.
    a. 로컬로 License Manager 통합 테스트
    b. Amazon EKS에서 License Manager 통합 테스트
11. 온프레미스 AWS 와 온프레미스 모두에서 License Manager 통합을 성공적으로 확인한 후의 단계에 따라 컨테이너 제품 목록을 생성할 수 있습니다개요: 컨테이너 제품 만들기.

로컬로 License Manager 통합 테스트

minikube 또는 기타 설정을 사용하여 아무 Kubernetes 클러스터에서 로컬로 License Manager 통합을 테스트할 수 있습니다. Kubernetes 클러스터에 License Manager API 작업을 호출할 수 있는 아웃바운드 인터넷 액세스 권한이 있어야 합니다.

로컬로 License Manager 통합을 테스트하는 방법

1. 원하는 권한이 있는 테스트 판매자 계정에서 테스트 라이선스를 생성합니다. 테스트 라이선스를 설정하려면 AWS License Manager API 참조의 CreateLicense를 참조하세요. 또는 다음 스크립트를 사용하여 테스트 라이선스를 생성한 다음, 테스트 구매자 계정에 라이선스를 부여하여 라이선스를 사용할 수 있습니다. 다음 스크립트는 테스트 판매자 계정 보안 인증 정보를 사용합니다.

```
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. 이전에 정의한 보안 암호 형식을 사용하는 라이선스 토큰과 IAM 역할을 사용하여 Kubernetes 보안 암호를 생성합니다. License Manager `CreateToken` API 작업을 사용하여 라이선스 토큰을 생성합니다. 그런 다음, IAM `CreateRole` API 작업을 사용하여 권한과 신뢰 정책이 있는 IAM 역할을 생성합니다. 다음 스크립트의 예시를 살펴보세요. 다음 스크립트는 테스트 구매자 계정 보안 인증 정보를 사용합니다.

```
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. 외부에서 호스팅되는 Kubernetes 클러스터를 설정합니다 AWS. 이를 사용하여 컨테이너 애플리케이션이 이외의 환경에서 AWS License Manager API에 연결할 수 AWS 있고 사용자 지정 자격 증명 공급자가 애플리케이션에 잘 통합되었는지 테스트합니다.
4. 앞에서 생성한 라이선스 토큰과 IAM 역할을 로컬 Kubernetes 클러스터에 배포합니다.

```
kubectl create secret generic "awsmp-license-access-config" \
--from-literal=license_token=${TOKEN} \
--from-literal=iam_role=${ROLE_ARN}
```

5. 보안 암호 이름을 입력으로 사용하여 Helm을 통해 애플리케이션을 배포하고 애플리케이션이 License Manager API 작업을 호출하여 권한 검사를 수행할 수 있는지 확인합니다. Helm 및 배포 사양 변경에 대한 내용은 AWS Marketplace for Containers Anywhere 제품을 License Manager와 통합의 9단계를 참조하세요.

Amazon EKS에서 License Manager 통합 테스트

Amazon EKS에서 License Manager 통합을 테스트할 수도 있습니다. 애플리케이션이 라이선스 구성 보안 암호 없이 License Manager API 작업을 호출할 수 있는지 테스트합니다. 또한 서비스 계정을 사용하여 서비스 계정에 대한 IAM 역할(IRSA)을 설정하고 애플리케이션에 관련 보안 인증 정보를 제공할 수 있는지 확인합니다.

Amazon EKS에서 License Manager 통합을 테스트하는 방법

1. 원하는 권한이 있는 테스트 판매자 계정에서 테스트 라이선스를 생성합니다. 테스트 라이선스를 설정하거나, 다음 스크립트를 사용하여 테스트 라이선스를 생성하고 테스트 구매자 계정에 라이선스를 부여하여 라이선스를 사용하려면 CreateLicense API 참조를 확인하세요. 다음 스크립트는 테스트 판매자 계정 보안 인증 정보를 사용합니다.

```
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. 원하는 구성의 테스트 Amazon EKS 클러스터를 생성하거나, 다음 명령을 실행하여 기본 구성을 사용합니다.

```
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. 기존 클러스터의 서비스 계정을 생성하고 IAM 역할과 연결합니다. 다음 명령은 `AWSLicenseManagerConsumptionPolicy`가 있는 IAM 역할을 생성합니다. 그 후 License Manager 통합 이미지를 배포해야 하는 Amazon EKS 클러스터의 `test_sa` 서비스 계정에 IAM 역할을 연결합니다. 따라서 서비스 계정은 License Manager API 작업을 호출하는 데 필요한 올바른 보안 인증 정보를 얻을 수 있습니다.

```
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. 이전 명령에서 IAM 역할이 연결된 서비스 계정의 Helm을 통해 애플리케이션을 배포합니다. 애플리케이션이 License Manager API 작업을 호출하여 권한 검사를 수행할 수 있는지 확인합니다.

플로팅 라이선스 권한과 License Manager

플로팅 라이선스를 사용하면 사용자가 애플리케이션에 로그인할 때 사용 가능한 라이선스 풀에서 라이선스를 가져옵니다. 사용자가 로그아웃하면 라이선스가 사용 가능한 라이선스 풀에 다시 추가됩니다.

플로팅 라이선스의 경우 애플리케이션은 리소스가 사용될 때 `CheckoutLicense` API 작업을 사용하여 권한 풀에서 권한을 체크아웃합니다. `CheckoutLicense` API 작업의 응답에는 체크아웃의 고유 식

별자인 라이선스 소비 토큰이 포함되어 있습니다. 라이선스 소비 토큰은 권한을 라이선스 풀로 다시 체크인하거나 체크아웃을 연장하는 등 체크아웃된 권한에 대한 추가 작업을 수행할 수 있습니다.

리소스가 더 이상 사용되지 않으면 애플리케이션은 `CheckInLicense` API 작업을 사용하여 권한을 다시 풀에 체크인합니다.

```
aws license-manager check-in-license \
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

라이선스를 다시 풀에 체크인하는 데 실패하면, 예를 들어 작업 중에 애플리케이션이 충돌하면 60분 후에 권한이 자동으로 다시 풀에 체크인됩니다. 따라서 리소스가 60분 넘게 사용 중이면 풀에서 권한을 체크아웃하는 것이 가장 좋습니다. 이렇게 하려면 리소스가 사용되는 동안 `ExtendLicenseConsumption` API 작업을 사용합니다.

```
aws license-manager extend-license-consumption \
--license-consumption-token "f1603b3c1f574b7284db84a9e771ee12"
```

온프레미스 배포를 위해 License Manager와 통합하는 모범 사례

온프레미스 환경에서 컨테이너 애플리케이션을 배포할 때 아웃바운드 네트워크 액세스가 불안정할 수 있습니다. 인터넷 연결 불량으로 인한 잠재적 문제 때문에 구매자에게 제공되는 서비스가 중단되지 않도록 다음 모범 사례에 따라 복원력을 강화합니다.

- 적절한 재시도 - 일시적인 네트워크 문제로 인해 애플리케이션이 연결되지 않을 수 있습니다 AWS License Manager. 지수 백오프를 사용하여 최대 30분 동안 재시도하도록 구현합니다. 이렇게 하면 단기적인 중단이나 네트워크 문제를 방지하는 데 도움이 될 수 있습니다.
- 하드 제한 금지 - 연결된 클러스터에 배포된 애플리케이션은 정기적으로 라이선스를 확인하여 업그레이드 또는 갱신으로 인한 변경 사항을 식별할 수 있습니다. 아웃바운드 액세스가 불안정하면 애플리케이션이 이러한 변경 사항을 식별하지 못할 수 있습니다. 가급적이면 애플리케이션에서 License Manager를 통해 라이선스를 확인할 수 없어 구매자에게 제공되는 서비스가 중단되는 일이 없도록 해야 합니다. 라이선스가 만료되고 라이선스가 유효한지 확인할 수 없는 경우 애플리케이션이 무료 평가판 또는 오픈 소스 환경을 사용할 수 있습니다.
- 고객 알림 - 캐시된 라이선스를 사용하는 경우 라이선스 변경 사항(갱신 또는 업그레이드 포함)이 실행 중인 워크로드에 자동으로 반영되지 않습니다. 애플리케이션이 캐시된 라이선스를 업데이트할 수 있도록 애플리케이션에 대한 아웃바운드 액세스를 일시적으로 다시 허용해야 한다고 고객에게 알려야 합니다. 예를 들어 애플리케이션 자체를 통해 또는 애플리케이션의 설명서를 통해 고객에게 알립니다. 마찬가지로, 기능 하위 세트로 대체하는 경우에도 권한이 모두 소진되었거나 라이선스가 만료되었다고 고객에게 알립니다. 그러면 고객은 업그레이드 또는 갱신을 선택할 수 있습니다.

LicenseManagerCredentialsProvider - Java 구현

LicenseCredentialsProvider는를 추가하여 온프레미스 사용을 위해 AWS SDK의 기본 자격 증명 공급자 체인을 확장합니다LicenseManagerTokenCredentialsProvider.

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

LicenseManagerTokenCredentialsProvider는 온프레미스 환경에서 License Manager OIDC 발급 ID 토큰을 사용하여 보안 인증 정보를 제공합니다. LicenseCredentialsProvider의 소스 코드를 애플리케이션 클래스 경로에 포함해야 합니다.

```
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 구현

LicenseCredentialsProvider

LicenseCredentialsProvider는를 추가하여 온프레미스 사용을 위해 AWS SDK의 기본 자격 증명 공급자 체인을 확장합니다LicenseManagerTokenCredentialsProvider.

```
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`는 온프레미스 환경에서 License Manager OIDC 발급 ID 토큰을 사용하여 보안 인증 정보를 제공합니다. `LicenseCredentialsProvider`의 소스 코드를 애플리케이션 클래스 경로에 포함해야 합니다.

```
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
 }
}
```

컨테이너 제품에 대한 Amazon SNS 알림

알림을 받으려면 제품 생성 중에 AWS Marketplace 제공된에 대한 Amazon Simple Notification Service(Amazon SNS) 주제를 구독하면 됩니다. 이 주제는 고객의 제품 구독 변경 사항에 대한 알림을 제공합니다. 예를 들어 이 알림을 사용하면 고객이 언제 비공개 제안을 수락하는지 알 수 있습니다.

> **Note**
>
> 제품 생성 프로세스 중에 제품에 대한 Amazon SNS 주제가 만들어집니다. 알림을 구독하려면 Amazon SNS 주제(예: `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`)의 Amazon 리소스 이름(ARN)이 필요합니다. 서버 제품의 판매자 포털에서는 ARN을 사용할 수 없습니다. AWS Marketplace 운영 팀에 문의하여 ARN을 요청합니다.

다음은 컨테이너 제품에 사용할 수 있는 Amazon SNS 주제입니다.

- Amazon SNS 주제: `aws-mp-subscription-notification` - 이 주제는 구매자가 제품을 구독하거나 구독 해지할 때 알려줍니다. 시간당 및 장기 시간당을 포함한 시간당 요금 모델에 사용할 수 있습니다.

Amazon SNS 주제: **`aws-mp-subscription-notification`**

`aws-mp-subscription-notification` 주제의 각 메시지는 다음과 같은 형식입니다.

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123"
}
```

*`<action-name>`*은 알림에 따라 달라집니다. 가능한 작업은 다음과 같습니다.

- `subscribe-success`
- `subscribe-fail`
- `unsubscribe-pending`
- `unsubscribe-success`

`offer-identifier`는 작업이 `subscribe-success` 또는 인 경우에만 알림에 포함됩니다`subscribe-fail`. 작업이 `unsubscribe-pending` 또는 일 때는 알림에 포함되지 않습니다`unsubscribe-success`. 2024년 1월 이전에 생성된 제안의 경우이 식별자는 비공개 제안에 대한 알림에만 포함됩니다. 2024년 1월 이후에 생성된 제안의 경우이 식별자는 비공개 제안 및 공개 제안을 포함한 모든 제안에 대한 알림에 포함됩니다.

제안 유형에 대한 자세한 내용은 계약 갱신 대시보드에서 DescribeEntity API의 응답 또는 계약의 제안 가시성을 참조하세요. https://docs.aws.amazon.com/marketplace/latest/userguide/agreements-renewals-dashboard.html

> **Note**
>
> DescribeEntity API의 경우 해당 제안에 대한 대상 지정 규칙의 패싯을 대상으로 하는를 계정 AWS 계정 에서 찾으면 비공개 제안입니다. 해당 제안 AWS 계정 에 대한 대상 지정 규칙의 패싯을 대상으로 하는가 계정에 없는 경우 공개 제안입니다.

Amazon SQS 대기열을 Amazon SNS 주제에 구독

Amazon SQS 대기열을 제공된 SNS 주제에 구독하는 것이 좋습니다. SQS 대기열을 생성하고 대기열을 주제에 구독하는 방법에 대한 자세한 지침은 Amazon Simple Notification Service 개발자 안내서의 Amazon SQS 대기열을 Amazon SNS 주제에 구독을 참조하세요.

> **Note**
>
> 제품을 판매하는 데 AWS 계정 사용되는 에서만 AWS Marketplace SNS 주제를 구독할 수 있습니다. 하지만 메시지를 다른 계정에 전달할 수 있습니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 다른 계정의 Amazon SQS 대기열로 Amazon SNS 메시지 전송을 참조하세요.

알림을 위한 SQS 대기열 폴링

SQS 대기열을 SNS 주제에 구독하면 메시지가 SQS에 저장됩니다. 대기열을 지속적으로 폴링하는 서비스를 정의하고, 메시지를 찾고, 그에 따라 메시지를 처리해야 합니다.

의 기계 학습 제품 AWS Marketplace

AWS Marketplace 판매자는 구매자가 배포할 수 있는 기계 학습(ML) 알고리즘과 모델을 생성할 수 있습니다 AWS. 이 주제에서는에 나열된 Amazon SageMaker AI 제품 유형에 대한 정보를 제공합니다 AWS Marketplace.

에는 두 가지 유형의 SageMaker AI 제품이 나열되어 있습니다 AWS Marketplace.

모델 패키지

예측에 사용할 수 있도록 사전 학습된 모델로, 구매자가 추가로 학습할 필요가 없습니다.

알고리즘

구매자가 학습 데이터를 제공해야만 예측을 수행하는 하는 모델입니다. 학습 알고리즘이 포함되어 있습니다.

이러한 제품은 Amazon SageMaker AI 콘솔 또는를 통해 구매자에게 제공됩니다 AWS Marketplace. 구매자는 제품 설명, 설명서, 고객 리뷰, 요금, 지원 정보 등을 살펴볼 수 있습니다. 모델 패키지 제품 또는 알고리즘 제품을 구독하면 SageMaker AI 콘솔의 제품 목록에 추가됩니다. 또한 구매자는 AWS SDKs, AWS Command Line Interface (AWS CLI) 또는 SageMaker AI 콘솔을 사용하여 완전 관리형 REST 추론 엔드포인트를 생성하거나 데이터 배치에 대한 추론을 수행할 수 있습니다.

Amazon SageMaker AI를 사용하여 기계 학습 제품을 생성하는 방법에 대한 지원은 AWS Marketplace 판매자 운영 팀에 문의하세요.

기계 학습 제품 이해

AWS Marketplace 는 Amazon SageMaker AI를 사용하는 두 가지 기계 학습 제품 유형을 지원합니다. 모델 패키지 제품과 알고리즘 제품인데, 두 가지 유형 모두 예측을 수행하는 배포 가능한 추론 모델을 생성합니다.

SageMaker AI 모델 패키지

Amazon SageMaker AI 모델 패키지 제품에는 사전 훈련된 모델이 포함되어 있습니다. 사전 훈련된 모델을 SageMaker AI에 배포하여 실시간으로 또는 배치로 추론하거나 예측할 수 있습니다. 이 제품에는 모델 아티팩트가 포함된(있는 경우) 학습된 추론 구성 요소가 포함되어 있습니다. 판매자는 SageMaker AI를 사용하여 모델을 훈련하거나 자체 모델을 가져올 수 있습니다.

SageMaker AI 알고리즘

구매자는 SageMaker AI 알고리즘 제품을 사용하여 전체 기계 학습 워크로드를 수행할 수 있습니다. 알고리즘 제품은 학습과 추론이라는 두 가지 논리 구성 요소를 갖고 있습니다. SageMaker AI에서 구매자는 자체 데이터 세트를 사용하여 훈련 구성 요소로 훈련 작업을 생성합니다. 학습 구성 요소의 알고리즘이 완료되면 해당 알고리즘은 기계 학습 모델의 모델 아티팩트를 생성합니다. SageMaker AI는 구매자의 Amazon Simple Storage Service(Amazon S3) 버킷에 모델 아티팩트를 저장합니다. 그러면 SageMaker 구매자는 생성된 모델 아티팩트와 함께 추론 구성 요소를 배포하여 실시간으로 또는 배치로 추론(또는 예측)을 수행할 수 있습니다.

추론 모델 배포

추론 모델을 모델 패키지에서 생성하든 알고리즘에서 생성하든 관계없이 두 가지 방법으로 배포할 수 있습니다.

- 엔드포인트 -이 메서드는 SageMaker AI를 사용하여 모델을 배포하고 API 엔드포인트를 생성합니다. 구매자는 이 엔드포인트를 백엔드 서비스의 일부로 사용하여 애플리케이션을 구동할 수 있습니다. 데이터가 엔드포인트로 전송되면 SageMaker AI는 이를 모델 컨테이너로 전달하고 결과를 API 응답으로 반환합니다. 엔드포인트와 컨테이너는 구매자가 중지할 때까지 계속 실행됩니다.

  > **Note**
  >
  > 에서는 엔드포인트 메서드 AWS Marketplace를 실시간 추론이라고 하고, SageMaker AI 설명서에서는 호스팅 서비스라고 합니다. 자세한 내용은 Amazon SageMaker AI에서 모델 배포를 참조하세요.

- 배치 변환 작업 - 이 방법에서는 구매자가 추론에 사용할 데이터 세트를 Amazon S3에 저장합니다. 배치 변환 작업이 시작되면 SageMaker AI는 모델을 배포하고 S3 버킷의 데이터를 모델의 컨테이너로 전달한 다음 결과를 Amazon S3 버킷에 반환합니다. 작업이 완료되면 SageMaker AI는 작업을 중지합니다. 자세한 정보는 배치 변환 사용을 참조하세요.

  > **Note**
  >
  > SageMaker AI는 모델에 데이터를 전달하고 구매자에게 결과를 반환하기 때문에 두 방법 모두 모델에 대해 투명합니다.

기계 학습 제품 수명 주기

의 기계 학습 제품은 하나 이상의 소프트웨어 버전 및 관련 메타데이터로 AWS Marketplace 구성됩니다. 제품 구성에는 이름, 설명, 사용 지침, 요금, 분류 및 검색 키워드와 같은 필수 속성이 포함됩니다.

기계 학습 제품 생성 프로세스

에 기계 학습 제품을 나열하려면 다음을 완료해야 AWS Marketplace합니다.

1. the section called “SageMaker AI에서 제품 준비”
2. the section called “에 제품 나열 AWS Marketplace”

기계 학습 제품을 생성한 후에는 제품을 편집하고 관리할 수 있습니다. 자세한 내용은 the section called “제품 관리” 단원을 참조하십시오.

기계 학습 제품 상태

처음에는 신제품의 가시성이 제한되어 있고, 허용 목록에 있는 계정과 제품 생성자만 액세스할 수 있습니다. 테스트 및 검증 후 제품을 게시하여 모든 구매자가 AWS Marketplace 카탈로그에서 사용할 수 있도록 할 수 있습니다. 의 제품은 다음과 같은 상태 값을 가질 AWS Marketplace 수 있습니다.

상태 표시기	정의
Staging	이 상태는 아직 정보를 추가하고 있는 불완전한 제품을 나타냅니다. 셀프 서비스 환경을 처음 저장하고 종료한 후는 완료된 단계의 정보가 포함된 게시되지 않은 제품을 AWS Marketplace 생성합니다. 이 상태에서 정보를 계속 추가하거나 제출된 세부 정보를 수정할 수 있습니다.
Limited	A product attains this status after it's submitted to AWS Marketplace and passes all validation checks. At this point, the product has a detail page accessible only to your account and allowlisted entities. You can conduct product testing through this detail page.
Public	When you're prepared to make your product visible to buyers for subscription, update

상태 표시기	정의
	the product visibility in the console. Once processed, the product transitions from Limited to Public status. For information about AWS guidelines, see the section called “요구 사항 및 모범 사례”.
Restricted	새 사용자가 제품을 구독하지 못하도록 가시성 설정을 업데이트하여 제품을 제한할 수 있습니다. 제한된 상태는 기존 허용 목록에 있는 사용자가 제품을 계속 사용할 수 있도록 허용하지만 더 이상 일반에 공개되거나 새 사용자에게 제공되지 않습니다.

자세한 내용이나 지원은 AWS Marketplace 판매자 운영 팀에 문의하세요.

에 대한 기계 학습 제품 요금 AWS Marketplace

Amazon SageMaker AI 제품에 사용할 수 있는 여러 요금 모델 중에서 선택할 수 있습니다 AWS Marketplace. 제품을 구독하는 구매자는 자체 내에서 SageMaker AI에서 제품을 실행합니다 AWS 계정. 구매자에게 제시되는 가격은 구매자의 AWS 계정 에서 실행되는 리소스의 인프라 비용과 판매자가 설정한 제품 가격을 합한 금액입니다. 다음 섹션에서는에서 SageMaker AI 제품의 요금 모델에 대한 정보를 제공합니다. AWS Marketplace

주제

- 인프라 요금
- 소프트웨어 요금

인프라 요금

구매자는 제품을 사용하는 동안 SageMaker AI의 모든 인프라 비용을 부담해야 합니다. 이러한 비용은 에서 설정 AWS 하며 Amazon SageMaker AI 요금 페이지에서 확인할 수 있습니다.

소프트웨어 요금

제품 사용에 대해 구매자에게 AWS Marketplace 요금을 부과하는 소프트웨어 가격을 결정합니다. 요금 및 약관은 판매자가 AWS Marketplace에 기계 학습 제품을 추가할 때 설정합니다.

인스턴스 유형당 모든 인프라 및 소프트웨어 가격은 구매자가 구독하기 AWS Marketplace 전에의 제품 목록 페이지에 나와 있습니다.

주제

- 무료
- 시간당 요금
- 추론 요금
- 무료 평가판

무료

제품을 무료로 제공하도록 선택할 수 있습니다. 이 경우 구매자는 인프라 비용만 지불합니다.

시간당 요금

SageMaker AI에서 실행되는 소프트웨어의 인스턴스당 시간당 요금으로 제품을 제공할 수 있습니다. 소프트웨어가 실행되는 인스턴스 유형마다 다른 시간당 요금을 청구할 수 있습니다. 구매자가 소프트웨어를 실행하는 동안는 사용량을 AWS Marketplace 추적한 다음 그에 따라 구매자에게 요금을 청구합니다. 사용량은 분 단위로 계산됩니다.

모델 패키지 제품의 경우 구매자는 두 가지 방법으로 소프트웨어를 실행할 수 있습니다. 엔드포인트를 지속적으로 호스팅하여 실시간 추론을 수행하거나 데이터 세트에서 배치 변환 작업을 실행할 수 있습니다. 구매자가 소프트웨어를 실행할 수 있는 두 가지 방법의 요금을 다르게 책정할 수 있습니다.

알고리즘 제품의 경우 앞에서 설명한 추론 수행 요금뿐 아니라 학습 작업에 대한 시간당 요금도 결정합니다. 학습 이미지가 지원하는 인스턴스 유형마다 다른 시간당 요금을 청구할 수 있습니다.

추론 요금

구매자가 엔드포인트를 호스팅하여 실시간 추론을 지속적으로 수행하는 방식으로 소프트웨어를 실행하는 경우 추론당 가격을 설정할 수 있습니다.

> **Note**
>
> 배치 변환 프로세스에는 항상 시간당 요금이 적용됩니다. 알고리즘 제품의 학습 작업에도 항상 시간당 요금이 적용됩니다. 이러한 요금은 추론 요금과 독립적으로 설정할 수 있으며, 서로 다르게 설정할 수도 있습니다.

기본적으로 추론 요금에 따라는 엔드포인트를 호출할 때마다 구매자에게 AWS Marketplace 요금을 청구합니다. 하지만 소프트웨어가 단일 간접 호출(미니 배치라고도 함)에서 배치 추론을 처리하는 경우도 있습니다. 엔드포인트 배포의 경우 AWS Marketplace 에서 구매자에게 단일 간접 호출 요금을 청구해야 하는 추론 횟수를 사용자 지정할 수 있습니다. 이렇게 하려면 다음 예제와 같이 간접 호출의 HTTP 응답 헤더에 사용자 지정 측정 헤더를 포함해야 합니다. 이 예제에서는 구매자에게 3회 추론 요금을 청구하는 간접 호출을 보여줍니다.

```
X-Amzn-Inference-Metering: {"Dimension": "inference.count", "ConsumedUnits": 3}
```

> **Note**
> 추론 요금의 경우는 HTTP 응답 코드가 인 요청에 대해서만 구매자에게 AWS Marketplace 요금을 부과합니다2XX.

무료 평가판

원하는 경우 제품의 무료 평가판을 생성하고 무료 평가판 사용 일수를 정의할 수 있습니다. 무료 평가판은 5~31일일 수 있습니다. 무료 평가판 기간 동안 구매자는 원하는 만큼 소프트웨어를 실행할 수 있으며 소프트웨어 요금이 청구되지 않습니다. 무료 평가판 기간 동안 발생하는 인프라 비용은 구매자에게 청구됩니다. 평가판이 종료되면 인프라 비용과 함께 일반 소프트웨어 요금이 청구됩니다.

구매자가 무료 평가판이 제공되는 제품을 구독하면 환영 이메일 메시지를 받게 됩니다. 이 메시지에는 무료 평가판 기간, 계산된 만료 날짜, 구독 해지에 대한 세부 정보가 포함되어 있습니다. 미리 알림 이메일 메시지는 만료 날짜 3일 전에 전송됩니다.

에서 제품에 대한 무료 평가판을 제공하는 경우 무료 평가판에 대한 특정 환불 정책에 AWS Marketplace동의하는 것입니다.

> **Note**
> 기계 학습을 위한 비공개 제안에 대한 자세한 내용은 비공개 제안을 참조하세요.

의 기계 학습 제품에 대한 서비스 제한 및 할당량 AWS Marketplace

이 섹션에서는 AWS Marketplace의 기계 학습(ML) 제품에 적용되는 제한 및 할당량을 설명합니다.

주제

- 네트워크 격리
- 이미지 크기
- 스토리지 크기
- 인스턴스 크기
- 추론의 페이로드 크기
- 추론 처리 시간
- Service quotas
- 비동기 추론
- 서버리스 추론
- 관리형 스팟 훈련
- 도커 이미지 및 AWS 계정
- 기본 제공 알고리즘 또는에서 모델 패키지 게시 AWS Marketplace
- 게시 AWS 리전 지원

네트워크 격리

보안상의 이유로, 구매자가 컨테이너화된 제품을 구독하면 Docker 컨테이너는 네트워크에 연결되지 않은 격리된 환경에서 실행됩니다. 컨테이너를 생성할 때 인터넷을 통해 발신 호출을 수행하면 호출이 실패하므로 인터넷을 사용하지 마세요. 에 대한 호출도 실패 AWS 서비스 합니다.

이미지 크기

도커 이미지 크기는 Amazon Elastic Container Registry(Amazon ECR) 서비스 할당량을 통해 관리됩니다. 도커 이미지 크기는 훈련, 배치 변환 및 엔드포인트 생성 작업 동안 시작 시간에 영향을 미칩니다. 성능을 높이려면 도커 이미지 크기를 최적으로 유지하는 것이 좋습니다.

스토리지 크기

엔드포인트를 생성하면 Amazon SageMaker AI는 엔드포인트를 호스팅하는 각 ML 컴퓨팅 인스턴스에 Amazon Elastic Block Store(Amazon EBS) 스토리지 볼륨을 연결합니다. (엔드포인트를 실시간 추론 또는 Amazon SageMaker AI 호스팅 서비스라고도 합니다.) 스토리지 볼륨의 크기는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 Amazon SageMaker AI 개발자 안내서의 호스트 인스턴스 스토리지 볼륨을 참조하세요.

배치 변환은 Amazon SageMaker AI 개발자 안내서의 배치 변환의 스토리지를 참조하세요.

인스턴스 크기

SageMaker AI는 다양한 ML 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공합니다. 인스턴스 유형은 다양한 CPU, GPU, 메모리 및 네트워킹 용량 조합으로 구성됩니다. 인스턴스 유형은 ML 모델을 구축, 훈련, 배포하는 데 적합한 리소스 조합을 선택할 수 있는 유연성을 제공합니다. 자세한 내용은 Amazon SageMaker AI ML 인스턴스 유형을 참조하세요.

추론의 페이로드 크기

엔드포인트의 경우 간접 호출당 입력 데이터의 최대 크기 제한은 6MB입니다. 이 값은 조정할 수 없습니다.

배치 변환의 경우 간접 호출당 입력 데이터의 최대 크기는 100MB입니다. 이 값은 조정할 수 없습니다.

추론 처리 시간

엔드포인트의 경우 간접 호출당 최대 처리 시간은 60초입니다. 이 값은 조정할 수 없습니다.

배치 변환의 경우 호출당 최대 처리 시간은 60초입니다. 이 값은 조정할 수 없습니다.

Service quotas

훈련 및 추론과 관련된 할당량에 대한 자세한 내용은 Amazon SageMaker AI Service Quotas를 참조하세요.

비동기 추론

에 게시된 모델 패키지 및 알고리즘은 Amazon SageMaker AI 비동기 추론을 위해 구성된 엔드포인트에 배포할 AWS Marketplace 수 없습니다. 비동기 추론을 위해 구성된 엔드포인트를 사용하려면 모델이 네트워크에 연결되어야 합니다. 모든 AWS Marketplace 모델은 네트워크 격리 상태에서 작동합니다. 자세한 내용은 네트워크 액세스 없음을 참조하세요.

서버리스 추론

에 게시된 모델 패키지 및 알고리즘은 Amazon SageMaker AI Serverless Inference용으로 구성된 엔드포인트에 배포할 AWS Marketplace 수 없습니다. 서버리스 추론을 위해 구성된 엔드포인트를 사용하려면 모델이 네트워크에 연결되어야 합니다. 모든 AWS Marketplace 모델은 네트워크 격리 상태에서 작동합니다. 자세한 내용은 네트워크 액세스 없음을 참조하세요.

관리형 스팟 훈련

모든 알고리즘의 경우 관리형 스팟 훈련에 대한 체크포인트`MaxWaitTimeInSeconds`가 구현된 경우에도의 AWS Marketplace값은 3,600초(60분)로 설정됩니다. 이 값은 조정할 수 없습니다.

도커 이미지 및 AWS 계정

게시하려면 판매자 AWS 계정 가 소유한 Amazon ECR 리포지토리에 이미지를 저장해야 합니다. 다른 이 소유한 리포지토리에 저장된 이미지는 게시할 수 없습니다 AWS 계정.

기본 제공 알고리즘 또는에서 모델 패키지 게시 AWS Marketplace

Amazon SageMaker AI 기본 제공 알고리즘 또는 AWS Marketplace 구독의 알고리즘을 사용하여 훈련 작업에서 생성된 모델 패키지는 게시할 수 없습니다.

훈련 작업의 모델 아티팩트를 계속 사용해도 되지만, 모델 패키지를 게시하려면 자체 추론 이미지가 필요합니다.

게시 AWS 리전 지원

AWS Marketplace 는 모델 패키지 및 알고리즘 리소스 게시를 지원합니다. AWS 리전 여기서 다음 두 가지가 모두 true입니다.

- Amazon SageMaker AI가 지원하는 리전
- 기본적으로 옵트인되는 사용 가능한 리전(예: describe-regions는 `"OptInStatus": "opt-in-not-required"`를 반환함)

모델 패키지 또는 알고리즘 제품을 게시하는 데 필요한 모든 자산은 게시하려는 리전과 동일한 리전에 저장해야 합니다. 다음 내용이 포함됩니다:

- Amazon SageMaker AI에서 생성된 모델 패키지 및 알고리즘 리소스
- Amazon ECR 리포지토리에 업로드되는 추론 및 훈련 이미지
- Amazon Simple Storage Service에 저장되고 모델 패키지 리소스에 대한 모델 배포 중에 동적으로 로드되는 모델 아티팩트(있는 경우)
- Amazon S3에 저장된 추론 및 훈련 검증을 위한 테스트 데이터

SageMaker AI에서 지원하는 모든 리전에서 제품을 개발하고 훈련할 수 있습니다. 하지만 제품을 게시하려면 먼저 AWS Marketplace 에서 지원하는 게시 소스 리전에 모든 자산을 복사하고 리소스를 다시 생성해야 합니다.

Amazon SageMaker AI를 사용한 보안 및 지적 재산

Amazon SageMaker AI는에서 얻은 모델 및 알고리즘에 대한 지적 재산 및 구매자 데이터를 모두 보호합니다 AWS Marketplace. 다음 섹션에서는 SageMaker AI가 지적 재산과 고객 데이터의 보안을 보호하는 방법에 대한 자세한 정보를 제공합니다.

주제

- 지적 재산 보호
- 네트워크 액세스 없음
- 고객 데이터 보안

지적 재산 보호

제품을 생성하면 코드가 Docker 컨테이너 이미지에 패키징됩니다. 자세한 내용은 이 설명서 후반부의 SageMaker AI에서 제품 준비 섹션을 참조하세요. 컨테이너 이미지를 업로드하면 이미지와 아티팩트가 전송 중에 그리고 저장 시 암호화됩니다. 또한 이미지는 취약점이 있는지 스캔된 후 게시됩니다.

지적 재산을 보호하기 위해 SageMaker AI는 구매자만 AWS 서비스 엔드포인트를 통해 제품에 액세스할 수 있도록 허용합니다. 구매자는 컨테이너 이미지 또는 모델 아티팩트에 직접 액세스하거나 가져올 수 없으며 기본 인프라에도 액세스할 수 없습니다.

네트워크 액세스 없음

구매자가 생성하는 SageMaker AI 모델 및 알고리즘과 달리 구매자가 제품을 시작할 때 네트워크 액세스 없이 AWS Marketplace모델 및 알고리즘이 배포됩니다. SageMaker AI는 네트워크 또는 AWS 서비스 엔드포인트에 액세스할 수 없는 환경에 이미지를 배포합니다. 예를 들어 컨테이너 이미지는 인터넷의 서비스, VPC 엔드포인트 또는 기타 AWS 서비스에 아웃바운드 API 직접 호출을 수행할 수 없습니다.

고객 데이터 보안

제품은 구매자의 내에서 SageMaker AI에서 실행됩니다 AWS 계정. 따라서 구매자가 판매자의 제품을 사용하여 데이터 추론을 수행하면 판매자는 구매자의 데이터에 액세스할 수 없습니다.

알고리즘 제품의 경우 각 훈련 작업 후 훈련 이미지가 모델 아티팩트를 출력합니다. 모델 아티팩트는 구매자의 계정에 저장됩니다. 훈련 작업의 모델 아티팩트는 구매자가 판매자의 추론 이미지로 모델을 배포할 때 사용됩니다. 모델 아티팩트에 들어 있는 모든 지적 재산을 보호하려면 아티팩트를 출력하기 전에 암호화하도록 선택하면 됩니다.

> **Important**
>
> 이러한 보안 모델은 런타임 시 코드가 인터넷에 액세스하지 못하도록 차단합니다. 따라서 구매자의 코드는 인터넷의 리소스나 라이브러리를 사용할 수 없으므로, 종속성을 Docker 컨테이너 이미지에 패키징해야 합니다. 이는 훈련 작업에서 출력된 아티팩트를 암호화하기로 선택할 때 특히 중요합니다. 아티팩트를 암호화 및 암호 해독하는 키는 런타임에 인터넷을 통해 액세스할 수 없습니다. 암호화 키는 이미지와 함께 패키징되어야 합니다.

자세한 내용은 Amazon SageMaker AI의 보안을 참조하세요.

의 기계 학습 보고서 AWS Marketplace

AWS Marketplace 는 구매자, 금융, 사용량 및 세금에 대한 데이터가 포함된 Amazon SageMaker AI 제품에 대한 보고서를 생성합니다. 모든 보고서는 보고서 페이지의 AWS Marketplace Management Portal 에서 사용할 수 있습니다. 자세한 내용은 판매자 보고서를 참조하세요. 다음 섹션에서는 기계 학습 제품의 보고서에 대한 요약 정보를 제공합니다.

주제

- 일별 비즈니스 보고서
- 월별 수익 보고서
- 지출 보고서
- 기타 보고서 및 분석

일별 비즈니스 보고서

일별 비즈니스 보고서는 인스턴스 유형, 사용 시간, 소프트웨어 요금 수익 및 각 구매자와 제품에 대한 기타 세부 정보를 제공합니다. 구매자는 AWS 계정 ID로 식별할 수 있습니다. 자세한 내용은 일별 비즈니스 보고서를 참조하세요.

월별 수익 보고서

월별 수익 보고서는 소프트웨어 사용 요금으로 구매자에게 청구된 월별 수익을 제공합니다. 자세한 내용은 월별 청구 수익 보고서를 참조하세요.

지출 보고서

월별 지출 보고서는 소프트웨어 요금 정산 기간 동안 판매자를 대신하여 결제 받은 모든 자금 내역을 제공합니다. 보고서에 반영된 총 정산 금액은 판매자의 은행 계좌에 입금된 금액과 일치해야 합니다. 자세한 내용은 지출 보고서를 참조하세요.

기타 보고서 및 분석

사용 가능한 다른 보고서는 판매자 보고서를 참조하세요.

AWS Marketplace에서 제공하는 의 판매자 전송 데이터 피드 AWS Marketplace 서비스를 사용자 지정 보고서를 생성할 수도 있습니다.

SageMaker AI에서 제품 준비

에 제품을 게시하려면 먼저 Amazon SageMaker AI에서 제품을 준비 AWS Marketplace해야 합니다. 에는 모델 패키지와 알고리즘 AWS Marketplace이라는 두 가지 유형의 SageMaker AI 제품이 나열되어 있습니다. 자세한 내용은 의 기계 학습 제품 AWS Marketplace 단원을 참조하십시오. 이 주제에서는 제품을 준비하는 데 필요한 세 가지 단계에 대한 개요를 제공합니다.

1. 에서 기계 학습 제품을 위한 이미지로 코드 패키징 AWS Marketplace - 모델 패키지 또는 알고리즘 제품을 준비하려면 제품의 Docker 컨테이너 이미지를 생성해야 합니다.
2. Amazon Elastic Container Registry에 이미지 업로드 - 코드를 컨테이너 이미지로 패키징하고 로컬로 테스트한 후에는 이미지를 업로드하고 알려진 취약성이 있는지 스캔합니다. 모든 취약점을 수정한 후 계속 진행합니다.
3. Amazon SageMaker AI 리소스 생성 - 이미지가 성공적으로 스캔되면 이미지를 사용하여 SageMaker AI에서 모델 패키지 또는 알고리즘 리소스를 생성할 수 있습니다.

에서 기계 학습 제품을 위한 이미지로 코드 패키징 AWS Marketplace

의 기계 학습 제품은 Amazon SageMaker AI를 AWS Marketplace 사용하여 구매자에게 제공하는 기계 학습 로직을 생성하고 실행합니다. SageMaker AI는 로직이 포함된 Docker 컨테이너 이미지를 실행합니다. SageMaker AI는 안전하고 확장 가능한 인프라에서 이러한 컨테이너를 실행합니다. 자세한 내용은 Amazon SageMaker AI를 사용한 보안 및 지적 재산 단원을 참조하십시오. 다음 섹션에서는 SageMaker AI용 Docker 컨테이너 이미지에 코드를 패키징하는 방법에 대한 정보를 제공합니다.

주제

- 어떤 유형의 컨테이너 이미지를 생성할까요?

- 모델 패키지 이미지 만들기
- 알고리즘 이미지 만들기

어떤 유형의 컨테이너 이미지를 생성할까요?

추론 이미지와 훈련 이미지라는 두 가지 유형의 컨테이너 이미지가 있습니다.

모델 패키지 제품을 만들려면 추론 이미지만 있으면 됩니다. 자세한 지침은 모델 패키지 이미지 만들기 섹션을 참조하세요.

알고리즘 제품을 만들려면 훈련 이미지와 추론 이미지가 모두 필요합니다. 자세한 지침은 알고리즘 이미지 만들기 섹션을 참조하세요.

코드를 컨테이너 이미지에 올바르게 패키징하려면 컨테이너가 SageMaker AI 파일 구조를 준수해야 합니다. 서비스가 컨테이너와 데이터를 주고 받을 수 있도록 컨테이너는 올바른 엔드포인트를 노출해야 합니다. 다음 섹션에서는 이 프로세스를 자세히 설명합니다.

> **Important**
>
> 보안상의 이유로, 구매자가 컨테이너화된 제품을 구독하면 Docker 컨테이너는 인터넷에 연결되지 않은 격리된 환경에서 실행됩니다. 컨테이너를 생성할 때 인터넷을 통한 발신 호출을 수행하면 호출이 실패합니다. 에 대한 호출도 실패 AWS 서비스 합니다. 자세한 내용은 Amazon SageMaker AI를 사용한 보안 및 지적 재산(을)를 참조하세요.

원한다면, 추론 및 훈련 이미지를 생성할 때 사용 가능한 딥 러닝 컨테이너 이미지의 컨테이너를 시작점으로 사용합니다. 이 이미지는 이미 다양한 기계 학습 프레임워크와 적절하게 패키징되어 있습니다.

모델 패키지 이미지 만들기

Amazon SageMaker AI 모델 패키지는 예측을 수행하는 사전 훈련된 모델이며 구매자의 추가 훈련이 필요하지 않습니다. SageMaker AI에서 모델 패키지를 생성하고 기계 학습 제품을 게시할 수 있습니다 AWS Marketplace. 다음 섹션에서는 모델 패키지를 생성하는 방법을 설명합니다 AWS Marketplace. 여기에는 컨테이너 이미지 생성과 로컬에서 이미지 구축 및 테스트가 포함됩니다.

주제

- 개요
- 모델 패키지용 추론 이미지 만들기

개요

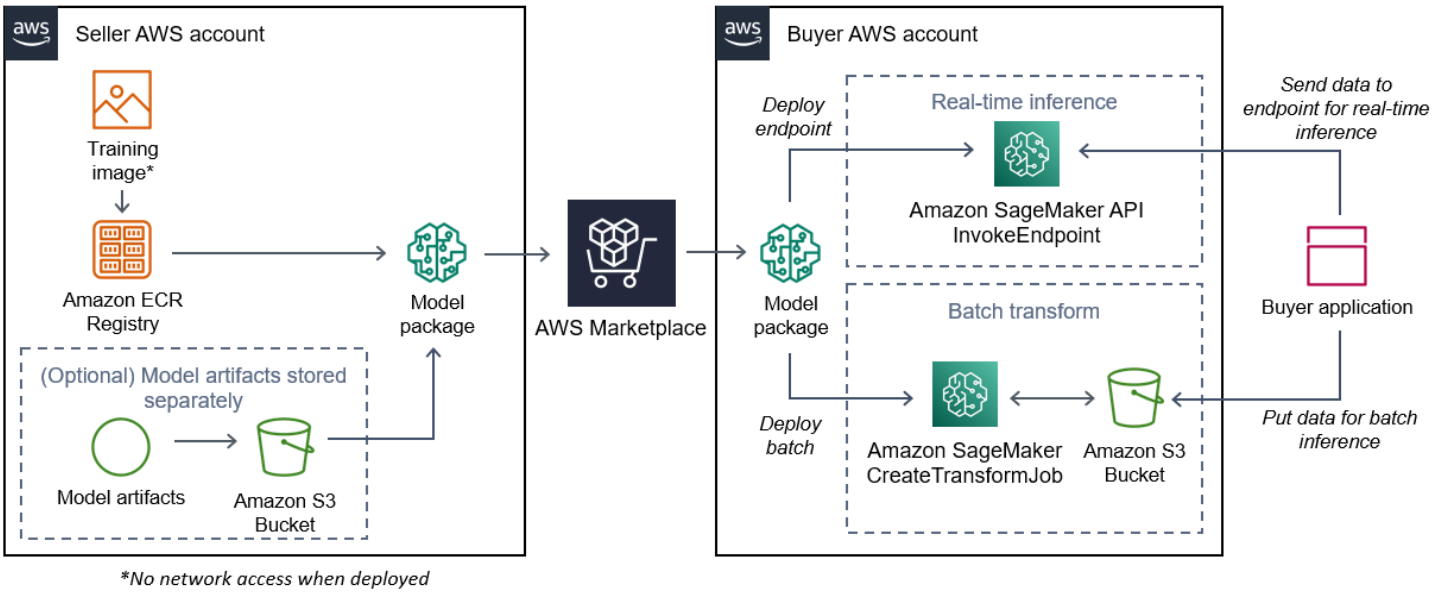
모델 패키지는 다음과 같은 구성 요소로 이루어집니다.

- Amazon Elastic Container Registry(Amazon ECR)에 저장된 추론 이미지
- (선택 사항) Amazon S3에 별도로 저장된 모델 아티팩트

> **Note**
>
> 모델 아티팩트는 모델이 예측에 사용하는 파일이며 일반적으로 자체 훈련 프로세스의 결과입니다. 모델에 필요한 모든 유형의 파일이 아티팩트가 될 수 있지만 반드시 use.tar.gz 압축 파일이어야 합니다. 모델 패키지의 경우 추론 이미지 내에서 번들링하거나 Amazon SageMaker AI에 별도로 저장할 수 있습니다. Amazon S3에 저장된 모델 아티팩트는 런타임에 추론 컨테이너로 로드됩니다. 모델 패키지를 게시할 때 이러한 아티팩트는 게시되어 구매자가 직접 액세스할 수 없는 AWS Marketplace 소유 Amazon S3 버킷에 저장됩니다.

> **Tip**
>
> 추론 모델이 Gluon, Keras, MXNet, PyTorch, TensorFlow, TensorFlow-Lite 또는 ONNX와 같은 딥 러닝 프레임워크로 빌드된 경우 Amazon SageMaker AI Neo를 사용하는 것이 좋습니다. Neo는 `ml.c4`, `ml.p2` 등과 같은 특정 클라우드 인스턴스 유형 제품군에 배포하는 추론 모델을 자동으로 최적화할 수 있습니다. 자세한 내용은 Amazon SageMaker AI 개발자 안내서의 Neo를 사용하여 모델 성능 최적화를 참조하세요.

다음 다이어그램은 모델 패키지 제품을 게시하고 사용하는 워크플로를 보여줍니다.



용 SageMaker AI 모델 패키지를 생성하는 워크플로에는 다음 단계가 AWS Marketplace 포함됩니다.

1. 판매자가 추론 이미지(배포 시 네트워크 액세스 불가)를 생성하여 Amazon ECR 레지스트리에 푸시합니다.

   모델 아티팩트는 추론 이미지에 번들로 제공하거나 S3에 별도로 저장할 수 있습니다.
2. 그런 다음 판매자는 Amazon SageMaker AI에서 모델 패키지 리소스를 생성하고 ML 제품을 게시합니다 AWS Marketplace.
3. 구매자가 ML 제품을 구독하고 모델을 배포합니다.

> **Note**
>
> 모델을 실시간 추론을 위한 엔드포인트로 배포하거나 전체 데이터 세트를 한꺼번에 예측하는 배치 작업으로 배포할 수 있습니다. 자세한 내용은 추론용 모델 배포를 참조하세요.

4. SageMaker AI는 추론 이미지를 실행합니다. 추론 이미지에 번들로 제공되지 않은, 판매자가 제공한 모델 아티팩트가 런타임에 동적으로 로드됩니다.
5. SageMaker AI는 컨테이너의 HTTP 엔드포인트를 사용하여 구매자의 추론 데이터를 컨테이너에 전달하고 예측 결과를 반환합니다.

모델 패키지용 추론 이미지 만들기

이 섹션에서는 추론 코드를 모델 패키지 제품의 추론 이미지로 패키징하는 방법을 안내합니다. 이 프로세스는 다음 단계로 구성됩니다.

단계

- 1단계: 컨테이너 이미지 만들기
- 2단계: 로컬로 이미지 빌드 및 테스트

추론 이미지는 추론 로직을 포함하고 있는 도커 이미지입니다. 런타임 시 컨테이너는 SageMaker AI가 컨테이너와 데이터를 주고받을 수 있도록 HTTP 엔드포인트를 노출합니다.

> **Note**
>
> 다음은 추론 이미지를 위한 패키징 코드의 한 가지 예시입니다. 자세한 내용은 SageMaker AI와 함께 Docker 컨테이너 사용 및 GitHub의 AWS Marketplace SageMaker AI 예제를 참조하세요.
>
> 다음 예제에서는 간단한 설명을 위해 Flask를 사용하는데, 이 웹 서비스는 아직 프로덕션에 사용할 수 없습니다.

1단계: 컨테이너 이미지 만들기

추론 이미지가 SageMaker AI와 호환되려면 Docker 이미지가 HTTP 엔드포인트를 노출해야 합니다. 컨테이너가 실행되는 동안 SageMaker AI는 컨테이너의 HTTP 엔드포인트에 추론을 위해 구매자 입력을 전달합니다. 추론 결과는 HTTP 응답 본문에 반환됩니다.

다음은 Linux Ubuntu 배포판을 사용하는 개발 환경에서 Docker CLI를 사용하는 예제입니다.

- 웹 서버 스크립트 생성
- 컨테이너 실행을 위한 스크립트 생성
- `Dockerfile` 생성
- 모델 아티팩트 패키징 또는 업로드

웹 서버 스크립트 생성

이 예제에서는 Flask라는 Python 서버를 사용하지만, 사용 중인 프레임워크에서 작동하는 아무 웹 서버를 사용해도 됩니다.

> **Note**
> 여기에서는 간단한 설명을 위해 Flask를 사용합니다. Flask는 아직 프로덕션에 사용할 수 없습니다.

SageMaker AI가 사용하는 TCP 포트 8080에서 두 HTTP 엔드포인트를 제공하는 Flask 웹 서버 스크립트를 생성합니다. 두 개의 예상 엔드포인트는 다음과 같습니다.

- `/ping` - SageMaker AI는이 엔드포인트에 HTTP GET 요청을 하여 컨테이너가 준비되었는지 확인합니다. 컨테이너가 준비되었으면 컨테이너는 이 엔드포인트의 HTTP GET 요청에 HTTP 200 응답 코드로 응답합니다.
- `/invocations` - SageMaker AI는 추론을 위해이 엔드포인트에 HTTP POST 요청을 합니다. 추론용 입력 데이터가 요청 본문에 전송됩니다. 사용자가 지정한 콘텐츠 유형이 HTTP 헤더에 전달됩니다. 응답의 본문은 추론 출력입니다. 시간 초과에 대한 자세한 내용은 기계 학습 제품 생성을 위한 요구 사항 및 모범 사례 섹션을 참조하세요.

`./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)
```

이전 예제에서는 실제 추론 로직이 없습니다. 실제 추론 이미지의 경우 웹 앱에 추론 로직을 추가하면 웹 앱이 입력을 처리하고 실제 예측 결과를 반환합니다.

추론 이미지에는 인터넷에 액세스할 수 없고 어떤에 대해서도 호출할 수 없으므로 필요한 종속성이 모두 포함되어야 합니다 AWS 서비스.

> **Note**
>
> 실시간 추론과 배치 추론에서도 이와 동일한 코드가 호출됩니다.

컨테이너 실행을 위한 스크립트 생성

SageMaker AIserve가 Docker 컨테이너 이미지를 실행할 때 실행되는 라는 스크립트를 생성합니다. 다음은 HTTP 웹 서버를 시작하는 스크립트입니다.

./serve

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker
flask run --host 0.0.0.0 --port 8080
```

Dockerfile 생성

빌드 컨텍스트에서 Dockerfile을 생성합니다. 이 예제에서는 Ubuntu 18.04를 사용하지만, 사용 중인 프레임워크에서 작동하는 아무 기본 이미지로 시작해도 됩니다.

./Dockerfile

```
FROM ubuntu:18.04
```

```
# Specify encoding
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8

# Install python-pip
RUN apt-get update \
&& apt-get install -y python3.6 python3-pip \
&& ln -s /usr/bin/python3.6 /usr/bin/python \
&& ln -s /usr/bin/pip3 /usr/bin/pip;

# Install flask server
RUN pip install -U Flask;

# Add a web server script to the image
# Set an environment to tell flask the script to run
COPY /web_app_serve.py /web_app_serve.py
ENV FLASK_APP=/web_app_serve.py

# Add a script that Amazon SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

`Dockerfile`은 이전에 생성한 두 개의 스크립트를 이미지에 추가합니다. `serve` 스크립트의 디렉터리가 PATH에 추가되므로 컨테이너가 실행될 때 스크립트를 실행할 수 있습니다.

모델 아티팩트 패키징 또는 업로드

모델 훈련에서 얻은 모델 아티팩트를 추론 이미지에 제공하는 두 가지 방법은 다음과 같습니다.

- 추론 이미지와 함께 정적으로 패키징됩니다.
- 런타임에 동적으로 로드됩니다. 동적으로 로드되므로 동일한 이미지를 사용하여 다양한 기계 학습 모델을 패키징할 수 있습니다.

모델 아티팩트를 추론 이미지와 함께 패키징하려면 `Dockerfile`에 아티팩트를 포함합니다.

모델 아티팩트를 동적으로 로드하려면 아티팩트를 Amazon S3의 압축 파일(.tar.gz)에 별도로 저장합니다. 모델 패키지를 생성할 때 압축 파일의 위치를 지정하면 SageMaker AI가 컨테이너를 실행할 `/opt/ml/model/` 때 콘텐츠를 추출하여 컨테이너 디렉터리에 복사합니다. 모델 패키지를 게시하면 구

매자가 직접 액세스할 수 없는 AWS Marketplace 소유의 Amazon S3 버킷에 해당 아티팩트가 게시되고 저장됩니다.

2단계: 로컬로 이미지 빌드 및 테스트

빌드 컨텍스트에는 이제 다음과 같은 파일이 있습니다.

- `./Dockerfile`
- `./web_app_serve.py`
- `./serve`
- 추론 로직 및 (선택적) 종속성

다음으로 컨테이너 이미지를 빌드, 실행 및 테스트합니다.

이미지 빌드

빌드 컨텍스트에서 Docker 명령을 실행하여 이미지를 빌드하고 태그를 지정합니다. 이 예제에서는 `my-inference-image` 태그를 사용합니다.

```
sudo docker build --tag my-inference-image ./
```

이 Docker 명령을 실행하여 이미지를 빌드한 후에는 Docker가 `Dockerfile`의 각 라인을 기반으로 이미지를 빌드할 때 출력이 표시됩니다. 모두 완료되면 다음과 비슷한 화면이 표시됩니다.

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```

로컬로 실행

빌드가 완료되면 로컬로 이미지를 테스트할 수 있습니다.

```
sudo docker run \
  --rm \
  --publish 8080:8080/tcp \
  --detach \
  --name my-inference-container \
  my-inference-image \
  serve
```

다음은 명령에 대한 세부 정보입니다.

- `--rm` - 컨테이너가 중지되면 자동으로 컨테이너를 제거합니다.
- `--publish 8080:8080/tcp` - 포트 8080을 노출하여 SageMaker AI가 HTTP 요청을 보내는 포트를 시뮬레이션합니다.
- `--detach` - 컨테이너를 백그라운드에서 실행합니다.
- `--name my-inference-container` - 실행 중인 이 컨테이너의 이름을 지정합니다.
- `my-inference-image` - 빌드된 이미지를 실행합니다.
- `serve` - 컨테이너를 실행할 때 SageMaker AI가 실행하는 것과 동일한 스크립트를 실행합니다.

이 명령을 실행한 후, Docker는 판매자가 생성한 추론 이미지로 컨테이너를 생성하고 백그라운드에서 실행합니다. 컨테이너가 `serve` 스크립트를 실행하고, 그러면 테스트 목적으로 웹 서버가 시작됩니다.

HTTP 엔드포인트를 ping 테스트합니다.

SageMaker AI가 컨테이너를 실행하면 엔드포인트를 주기적으로 ping합니다. 엔드포인트가 상태 코드 200으로 HTTP 응답을 반환하면 컨테이너가 추론할 준비가 되었음을 SageMaker AI에 알립니다. 엔드포인트를 테스트하고 응답 헤더를 포함시키는 다음 명령을 실행하여 이를 테스트할 수 있습니다.

```
curl --include http://127.0.0.1:8080/ping
```

예제 출력 내용은 다음과 같습니다.

```
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
```

추론 HTTP 엔드포인트 테스트

컨테이너가 ping에 200 상태 코드를 반환하여 준비되었음을 나타내면 SageMaker AI는 `POST` 요청을 통해 추론 데이터를 `/invocations` HTTP 엔드포인트에 전달합니다. 다음 명령을 실행하여 추론 포인트를 테스트합니다.

```
curl \
  --request POST \
  --data "hello world" \
```

```
http://127.0.0.1:8080/invocations
```

예제 출력 내용은 다음과 같습니다.

```
{"prediction": "a", "text": "hello world"}
```

이 두 HTTP 엔드포인트가 작동하면 이제 추론 이미지가 SageMaker AI와 호환됩니다.

> **Note**
> 모델 패키지 제품의 모델은 실시간 및 배치의 두 가지 방식으로 배포할 수 있습니다. 두 배포 모두에서 SageMaker AI는 Docker 컨테이너를 실행하는 동안 동일한 HTTP 엔드포인트를 사용합니다.

컨테이너를 중지하려면 다음 명령을 실행합니다.

```
sudo docker container stop my-inference-container
```

추론 이미지가 준비되고 테스트를 마쳤으면 계속해서 Amazon Elastic Container Registry에 이미지 업로드 섹션을 진행할 수 있습니다.

알고리즘 이미지 만들기

Amazon SageMaker AI 알고리즘을 사용하려면 구매자가 예측을 수행하기 전에 훈련할 자체 데이터를 가져와야 합니다. AWS Marketplace 판매자는 SageMaker AI를 사용하여 구매자가 배포할 수 있는 기계 학습(ML) 알고리즘 및 모델을 생성할 수 있습니다 AWS. 다음 섹션에서는 알고리즘 이미지를 생성하는 방법을 설명합니다 AWS Marketplace. 여기에는 알고리즘 훈련을 위한 Docker 훈련 이미지와 추론 로직이 포함된 추론 이미지 만들기가 포함됩니다. 알고리즘 제품을 게시할 때는 훈련 이미지와 추론 이미지가 모두 필요합니다.

주제

- 개요
- 알고리즘에 사용할 훈련 이미지 생성
- 알고리즘용 추론 이미지 생성

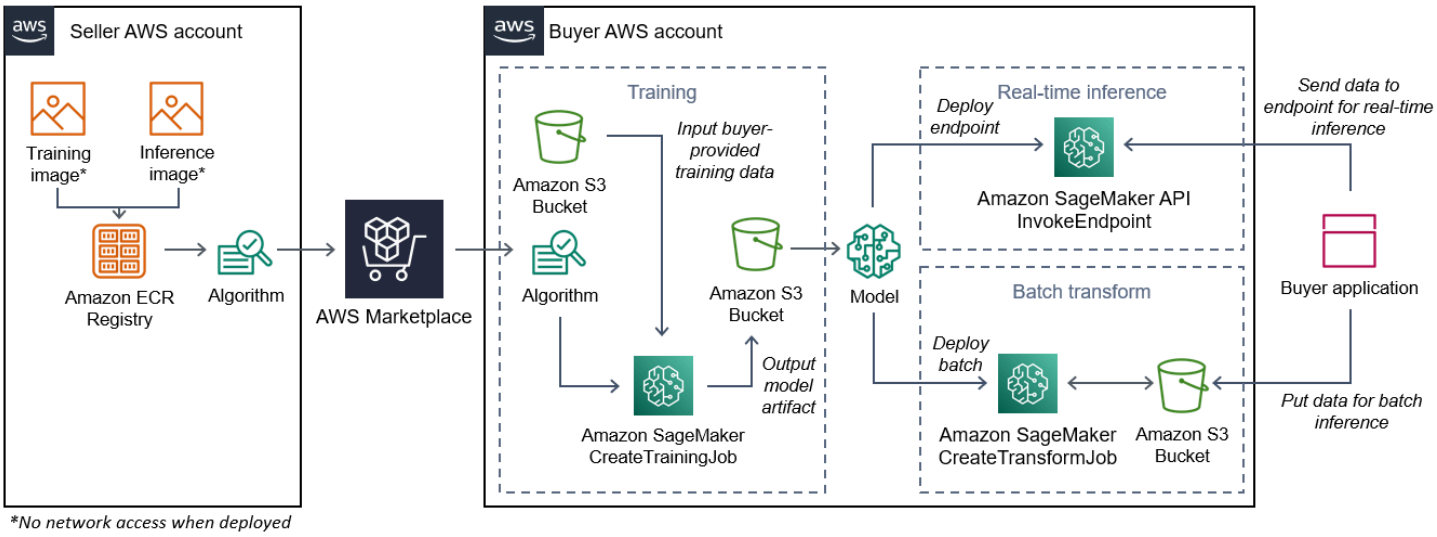
개요

알고리즘의 구성 요소는 다음과 같습니다.

- Amazon ECR에 저장된 훈련 이미지
- Amazon Elastic Container Registry(Amazon ECR)에 저장된 추론 이미지

> **Note**
>
> 알고리즘 제품의 경우 훈련 컨테이너는 모델 배포 시 추론 컨테이너에 로드되는 모델 아티팩트를 생성합니다.

다음 다이어그램은 알고리즘 제품을 게시하고 사용하는 워크플로를 보여줍니다.



용 SageMaker AI 알고리즘을 생성하는 워크플로에는 다음 단계가 AWS Marketplace 포함됩니다.

1. 판매자가 훈련 이미지와 추론 이미지(배포 시 네트워크 액세스 불가)를 생성하여 Amazon ECR 레지스트리에 업로드합니다.
2. 그런 다음 판매자는 Amazon SageMaker AI에서 알고리즘 리소스를 생성하고 ML 제품을 게시합니다 AWS Marketplace.
3. 구매자가 ML 제품을 구독합니다.
4. 구매자가 호환되는 데이터 세트와 적절한 하이퍼파라미터 값을 사용하여 훈련 작업을 생성합니다. SageMaker AI는 훈련 이미지를 실행하고 훈련 데이터와 하이퍼파라미터를 훈련 컨테이너에 로드합니다. 훈련 작업이 완료되면 `/opt/ml/model/`에 있는 모델 아티팩트가 압축되어 구매자의 Amazon S3 버킷에 복사됩니다.
5. 구매자는 Amazon S3에 저장된 훈련의 모델 아티팩트로 모델 패키지를 생성하고 모델을 배포합니다.

6. SageMaker AI`/opt/ml/model/`는 추론 이미지를 실행하고, 압축된 모델 아티팩트를 추출하고, 추론을 제공하는 코드에서 사용하는 추론 컨테이너 디렉터리 경로에 파일을 로드합니다.
7. 모델이 엔드포인트로 배포되든 배치 변환 작업으로 배포되든 SageMaker AI는 구매자를 대신하여 추론용 데이터를 컨테이너의 HTTP 엔드포인트를 통해 컨테이너에 전달하고 예측 결과를 반환합니다.

> **Note**
>
> 자세한 내용은 모델 훈련을 참조하세요.

알고리즘에 사용할 훈련 이미지 생성

이 섹션에서는 훈련 코드를 훈련 이미지로 패키징하는 방법을 안내합니다. 알고리즘 제품을 생성하려면 훈련 이미지가 필요합니다.

훈련 이미지는 훈련 알고리즘을 포함하고 있는 도커 이미지입니다. 컨테이너는 특정 파일 구조를 준수하므로 SageMaker AI가 컨테이너와 데이터를 복사할 수 있습니다.

알고리즘 제품을 게시할 때 훈련 이미지와 추론 이미지가 모두 필요합니다. 훈련 이미지를 생성한 후에는 추론 이미지를 생성해야 합니다. 두 이미지를 하나의 이미지로 결합할 수도 있고 별도의 이미지로 유지할 수도 있습니다. 두 이미지를 결합할 것인지 아니면 별도의 이미지로 유지할 것인지는 사용자가 결정합니다. 일반적으로 추론은 훈련보다 간단하므로, 추론 성능에 도움이 되는 별도의 이미지로 유지하는 것이 더 좋을 수 있습니다.

> **Note**
>
> 다음은 훈련 이미지를 위한 패키징 코드의 한 가지 예시입니다. 자세한 내용은 GitHub에서 자체 알고리즘 및 모델 사용 AWS Marketplace 및 AWS Marketplace SageMaker AI 예제를 참조하세요.

단계

- 1단계: 컨테이너 이미지 생성
- 2단계: 로컬로 이미지 빌드 및 테스트

1단계: 컨테이너 이미지 생성

훈련 이미지가 Amazon SageMaker AI와 호환되려면 SageMaker AI가 훈련 데이터 및 구성 입력을 컨테이너의 특정 경로에 복사할 수 있도록 특정 파일 구조를 준수해야 합니다. 훈련이 완료되면 생성된 모델 아티팩트는 SageMaker AI가 복사하는 컨테이너의 특정 디렉터리 경로에 저장됩니다.

다음은 Linux의 Ubuntu 배포판 개발 환경에 설치된 Docker CLI를 사용하는 예제입니다.

- 구성 입력을 읽을 수 있도록 프로그램 준비
- 데이터 입력을 읽을 수 있도록 프로그램 준비
- 훈련 출력을 작성하도록 프로그램 준비
- 컨테이너 실행을 위한 스크립트 생성
- `Dockerfile` 생성

구성 입력을 읽을 수 있도록 프로그램 준비

훈련 프로그램에 구매자가 제공한 구성 입력이 필요한 경우 실행 시 컨테이너 내부로 복사되는 위치는 다음과 같습니다. 필요한 경우 프로그램이 이러한 특정 파일 경로에서 읽어 와야 합니다.

- `/opt/ml/input/config`는 프로그램 실행 방식을 제어하는 정보가 들어 있는 디렉터리입니다.
  - `hyperparameters.json`은 하이퍼파라미터 이름 및 값의 JSON 형식 사전입니다. 값은 문자열이므로 변환해야 할 수도 있습니다.
  - `resourceConfig.json`은 분산 훈련에 사용되는 네트워크 레이아웃을 설명하는 JSON 형식 파일입니다. 훈련 이미지가 분산 훈련을 지원하지 않는 경우 이 파일을 무시해도 됩니다.

> **Note**
>
> 구성 입력에 대한 자세한 내용은 Amazon SageMaker AI가 훈련 정보를 제공하는 방법을 참조하세요.

데이터 입력을 읽을 수 있도록 프로그램 준비

다음 두 모드 중 하나로 훈련 데이터를 컨테이너에 전달할 수 있습니다. 컨테이너에서 실행되는 훈련 프로그램은 훈련 데이터를 이 두 가지 모드 중 하나로 분해합니다.

파일 모드

- `/opt/ml/input/data/<channel_name>/`에는 해당 채널의 입력 데이터가 포함되어 있습니다. 채널은 `CreateTrainingJob` 작업 호출을 기반으로 생성되지만, 일반적으로 채널이 알고리즘의 예상과 일치하는 것이 중요합니다. Amazon S3 키 구조가 나타내는 트리 구조를 유지하면서 각 채널의 파일이 Amazon S3에서 이 디렉터리로 복사됩니다.

파이프 모드

- `/opt/ml/input/data/<channel_name>_<epoch_number>`는 주어진 에포크에 대한 파이프입니다. 에포크는 0에서 시작하며 읽을 때마다 1씩 증가합니다. 실행할 수 있는 에포크 수에는 제한이 없지만, 다음 에포크를 읽기 전에 각 파이프를 닫아야 합니다.

훈련 출력을 작성하도록 프로그램 준비

훈련의 출력은 다음 컨테이너 디렉터리에 기록됩니다.

- `/opt/ml/model/`는 훈련 알고리즘이 생성하는 모델 또는 모델 아티팩트를 작성하는 디렉터리입니다. 모델 형식은 원하는 대로 무엇이든 가능합니다. 단일 파일도 가능하고 전체 디렉터리 트리도 가능합니다. SageMaker AI는이 디렉터리의 모든 파일을 압축된 파일(.tar.gz)로 패키징합니다. 이 파일은 `DescribeTrainingJob` API 작업에서 반환된 Amazon S3 위치에 있습니다.
- `/opt/ml/output/`는 알고리즘이 작업 실패 이유를 설명하는 `failure` 파일을 작성할 수 있는 디렉터리입니다. 이 파일의 내용은 `DescribeTrainingJob` 결과의 `FailureReason` 필드에 반환됩니다. 작업이 성공하면 이 파일이 무시되므로 이 파일을 작성할 이유가 없습니다.

컨테이너 실행을 위한 스크립트 생성

SageMaker AI가 Docker 컨테이너 이미지를 실행할 때 실행하는 `train`쉘 스크립트를 생성합니다. 훈련이 완료되고 모델 아티팩트가 해당 디렉터리에 기록되면 스크립트를 종료합니다.

`./train`

```
#!/bin/bash

# Run your training program here
#
#
#
#
```

`Dockerfile` 생성

빌드 컨텍스트에서 `Dockerfile`을 생성합니다. 이 예제에서는 Ubuntu 18.04를 기본 이미지로 사용하지만, 사용 중인 프레임워크에서 작동하는 아무 기본 이미지로 시작해도 됩니다.

`./Dockerfile`

```
FROM ubuntu:18.04

# Add training dependencies and programs
#
#
#
#
#
# Add a script that SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /train /opt/program/train
RUN chmod 755 /opt/program/train
ENV PATH=/opt/program:${PATH}
```

`Dockerfile`은 이전에 생성한 `train` 스크립트를 이미지에 추가합니다. 스크립트의 디렉터리가 PATH에 추가되므로 컨테이너가 실행될 때 스크립트를 실행할 수 있습니다.

이전 예제에서는 실제 훈련 로직이 없습니다. 실제 훈련 이미지의 경우 `Dockerfile`에 훈련 종속성을 추가하고, 훈련 입력을 읽는 로직을 추가하여 모델 아티팩트를 훈련하고 생성합니다.

훈련 이미지는 인터넷에 액세스할 수 없으므로 필요한 종속성을 모두 갖고 있어야 합니다.

자세한 내용은 GitHub에서 자체 알고리즘 및 모델 사용 AWS Marketplace 및 AWS Marketplace SageMaker AI 예제를 참조하세요.

2단계: 로컬로 이미지 빌드 및 테스트

빌드 컨텍스트에는 이제 다음과 같은 파일이 있습니다.

- `./Dockerfile`
- `./train`
- 훈련 종속성 및 로직

다음으로 이 컨테이너 이미지를 빌드, 실행 및 테스트할 수 있습니다.

이미지 빌드

빌드 컨텍스트에서 Docker 명령을 실행하여 이미지를 빌드하고 태그를 지정합니다. 이 예제에서는 `my-training-image` 태그를 사용합니다.

```
sudo docker build --tag my-training-image ./
```

이 Docker 명령을 실행하여 이미지를 빌드한 후에는 Docker가 `Dockerfile`의 각 라인을 기반으로 이미지를 빌드할 때 출력이 표시됩니다. 모두 완료되면 다음과 비슷한 화면이 표시됩니다.

```
Successfully built abcdef123456
Successfully tagged my-training-image:latest
```

로컬로 실행

작업이 완료되면 다음 예제와 같이 로컬로 이미지를 테스트합니다.

```
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
```

명령 세부 정보는 다음과 같습니다.

- `--rm` - 컨테이너가 중지되면 자동으로 컨테이너를 제거합니다.
- `--volume '<path_to_input>:/opt/ml/input:ro'` - 테스트 입력 디렉토리를 컨테이너에서 읽기 전용으로 사용할 수 있게 합니다.
- `--volume '<path_to_model>:/opt/ml/model'` - 훈련 테스트가 완료되면 모델 아티팩트가 호스트 머신에 저장되는 경로를 바인드 탑재합니다.
- `--volume '<path_to_output>:/opt/ml/output'` - `failure` 파일의 실패 이유가 호스트 머신에 기록되는 경로를 바인드 탑재합니다.
- `--name my-training-container` - 실행 중인 이 컨테이너의 이름을 지정합니다.

- `my-training-image` - 빌드된 이미지를 실행합니다.
- `train` - 컨테이너를 실행할 때 실행되는 것과 동일한 스크립트 SageMaker AI를 실행합니다.

이 명령을 실행한 후, Docker는 사용자가 빌드한 훈련 이미지로 컨테이너를 생성하고 실행합니다. 컨테이너가 `train` 스크립트를 실행하고, 그러면 훈련 프로그램을 시작됩니다.

훈련 프로그램이 완료되고 컨테이너가 종료되면 출력 모델 아티팩트가 올바른지 확인합니다. 또한 로그 출력을 확인하여 원하지 않는 로그가 생성되지 않는지 확인하고, 훈련 작업에 대한 정보가 충분히 제공되는지 확인합니다.

이로써 알고리즘 제품의 훈련 코드 패키징이 완료되었습니다. 알고리즘 제품에는 추론 이미지도 포함되어 있으므로 알고리즘용 추론 이미지 생성 섹션을 계속 진행하세요.

알고리즘용 추론 이미지 생성

이 섹션에서는 추론 코드를 알고리즘 제품의 추론 이미지로 패키징하는 방법을 안내합니다.

추론 이미지는 추론 로직을 포함하고 있는 도커 이미지입니다. 런타임 시 컨테이너는 SageMaker AI가 컨테이너와 데이터를 주고받을 수 있도록 HTTP 엔드포인트를 노출합니다.

알고리즘 제품을 게시할 때 훈련 이미지와 추론 이미지가 모두 필요합니다. 알고리즘에 사용할 훈련 이미지 생성에 대한 이전 섹션을 아직 살펴보지 않았으면 지금 살펴보세요. 두 이미지를 하나의 이미지로 결합할 수도 있고 별도의 이미지로 유지할 수도 있습니다. 두 이미지를 결합할 것인지 아니면 별도의 이미지로 유지할 것인지는 사용자가 결정합니다. 일반적으로 추론은 훈련보다 간단하므로, 추론 성능에 도움이 되는 별도의 이미지로 유지하는 것이 더 좋을 수 있습니다.

> **Note**
>
> 다음은 추론 이미지를 위한 패키징 코드의 한 가지 예시입니다. 자세한 내용은 GitHub에서 자체 알고리즘 및 모델 사용 AWS Marketplace 및 AWS Marketplace SageMaker AI 예제를 참조하세요.
>
> 다음 예제에서는 간단한 설명을 위해 Flask를 사용하는데, 이 웹 서비스는 아직 프로덕션에 사용할 수 없습니다.

단계

- 1단계: 추론 이미지 생성
- 2단계: 로컬로 이미지 빌드 및 테스트

1단계: 추론 이미지 생성

추론 이미지가 SageMaker AI와 호환되려면 Docker 이미지가 HTTP 엔드포인트를 노출해야 합니다. 컨테이너가 실행되는 동안 SageMaker AI는 구매자가 제공한 추론에 대한 입력을 컨테이너의 HTTP 엔드포인트에 전달합니다. 추론 결과는 HTTP 응답의 본문에 반환됩니다.

다음은 Linux의 Ubuntu 배포판 개발 환경에 설치된 Docker CLI를 사용하는 예제입니다.

- 웹 서버 스크립트 생성
- 컨테이너 실행을 위한 스크립트 생성
- `Dockerfile` 생성
- 모델 아티팩트를 동적으로 로드하도록 프로그램 준비

웹 서버 스크립트 생성

이 예제에서는 Flask라는 Python 서버를 사용하지만, 사용 중인 프레임워크에서 작동하는 아무 웹 서버를 사용해도 됩니다.

> **Note**
>
> 여기에서는 간단한 설명을 위해 Flask를 사용합니다. Flask는 아직 프로덕션에 사용할 수 없습니다.

SageMaker AI가 사용하는 TCP 포트 8080에서 두 HTTP 엔드포인트를 제공하는 Flask 웹 서버 스크립트를 생성합니다. 두 개의 예상 엔드포인트는 다음과 같습니다.

- `/ping` - SageMaker AI는이 엔드포인트에 HTTP GET 요청을 하여 컨테이너가 준비되었는지 확인합니다. 컨테이너가 준비되었으면 컨테이너는 이 엔드포인트의 HTTP GET 요청에 HTTP 200 응답 코드로 응답합니다.
- `/invocations` - SageMaker AI는 추론을 위해이 엔드포인트에 HTTP POST 요청을 합니다. 추론용 입력 데이터가 요청 본문에 전송됩니다. 사용자가 지정한 콘텐츠 유형이 HTTP 헤더에 전달됩니다. 응답의 본문은 추론 출력입니다.

`./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)
```

이전 예제에서는 실제 추론 로직이 없습니다. 실제 추론 이미지의 경우 웹 앱에 추론 로직을 추가하면 웹 앱이 입력을 처리하고 예측 결과를 반환합니다.

추론 이미지는 인터넷에 액세스할 수 없으므로 필요한 종속성을 모두 갖고 있어야 합니다.

컨테이너 실행을 위한 스크립트 생성

SageMaker AI`serve`가 Docker 컨테이너 이미지를 실행할 때 실행되는 라는 스크립트를 생성합니다. 이 스크립트에서 HTTP 웹 서버를 시작합니다.

`./serve`

```
#!/bin/bash

# Run flask server on port 8080 for SageMaker AI
```

```
flask run --host 0.0.0.0 --port 8080
```

Dockerfile 생성

빌드 컨텍스트에서 Dockerfile을 생성합니다. 이 예제에서는 Ubuntu 18.04를 사용하지만, 사용 중인 프레임워크에서 작동하는 아무 기본 이미지로 시작해도 됩니다.

./Dockerfile

```
FROM ubuntu:18.04

# Specify encoding
ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8

# Install python-pip
RUN apt-get update \
&& apt-get install -y python3.6 python3-pip \
&& ln -s /usr/bin/python3.6 /usr/bin/python \
&& ln -s /usr/bin/pip3 /usr/bin/pip;

# Install flask server
RUN pip install -U Flask;

# Add a web server script to the image
# Set an environment to tell flask the script to run
COPY /web_app_serve.py /web_app_serve.py
ENV FLASK_APP=/web_app_serve.py

# Add a script that Amazon SageMaker AI will run
# Set run permissions
# Prepend program directory to $PATH
COPY /serve /opt/program/serve
RUN chmod 755 /opt/program/serve
ENV PATH=/opt/program:${PATH}
```

Dockerfile은 이전에 생성한 두 개의 스크립트를 이미지에 추가합니다. serve 스크립트의 디렉터리가 PATH에 추가되므로 컨테이너가 실행될 때 스크립트를 실행할 수 있습니다.

모델 아티팩트를 동적으로 로드하도록 프로그램 준비

알고리즘 제품의 경우 구매자는 자체 데이터 세트를 훈련 이미지와 함께 사용하여 고유의 모델 아티팩트를 생성합니다. 훈련 프로세스가 완료되면 훈련 컨테이너는 모델 아티팩트를 컨테이너 디렉터리 /

`opt/ml/model/`에 출력합니다. SageMaker AI는 해당 디렉터리의 콘텐츠를 .tar.gz 파일로 압축하여 Amazon S3의 구매자의 AWS 계정 에 저장합니다.

모델이 배포되면 SageMaker AI는 추론 이미지를 실행하고 Amazon S3의 구매자 계정에 저장된 .tar.gz 파일에서 모델 아티팩트를 추출하여 `/opt/ml/model/` 디렉터리의 추론 컨테이너에 로드합니다. 추론 컨테이너 코드는 런타임에 모델 데이터를 사용합니다.

> **Note**
>
> 모델 아티팩트 파일에 들어 있는 모든 지적 재산을 보호하려면 파일을 출력하기 전에 암호화하도록 선택하면 됩니다. 자세한 내용은 Amazon SageMaker AI를 사용한 보안 및 지적 재산 단원을 참조하세요.

2단계: 로컬로 이미지 빌드 및 테스트

빌드 컨텍스트에는 이제 다음과 같은 파일이 있습니다.

- `./Dockerfile`
- `./web_app_serve.py`
- `./serve`

다음으로 이 컨테이너 이미지를 빌드, 실행 및 테스트할 수 있습니다.

이미지 빌드

Docker 명령을 실행하여 이미지를 빌드하고 태그를 지정합니다. 이 예제에서는 `my-inference-image` 태그를 사용합니다.

```
sudo docker build --tag my-inference-image ./
```

이 Docker 명령을 실행하여 이미지를 빌드한 후에는 Docker가 `Dockerfile`의 각 라인을 기반으로 이미지를 빌드할 때 출력이 표시됩니다. 모두 완료되면 다음과 비슷한 화면이 표시됩니다.

```
Successfully built abcdef123456
Successfully tagged my-inference-image:latest
```

로컬로 실행

빌드가 완료되면 로컬로 이미지를 테스트할 수 있습니다.

```
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
```

명령 세부 정보는 다음과 같습니다.

- `--rm` - 컨테이너가 중지되면 자동으로 컨테이너를 제거합니다.
- `--publish 8080:8080/tcp` - 포트 8080을 노출하여 SageMaker AI가 HTTP 요청을 보내는 포트를 시뮬레이션합니다.
- `--volume '<path_to_model>:/opt/ml/model:ro'` - 테스트 모델 아티팩트가 호스트 머신에 읽기 전용으로 저장되는 경로를 바인드 탑재하여 컨테이너의 추론 코드에서 사용할 수 있게 합니다.
- `--detach` - 컨테이너를 백그라운드에서 실행합니다.
- `--name my-inference-container` - 실행 중인 이 컨테이너의 이름을 지정합니다.
- `my-inference-image` - 빌드된 이미지를 실행합니다.
- `serve` - 컨테이너를 실행할 때 실행되는 것과 동일한 스크립트 SageMaker AI를 실행합니다.

이 명령을 실행한 후, Docker는 추론 이미지로 컨테이너를 생성하고 백그라운드에서 실행합니다. 컨테이너가 `serve` 스크립트를 실행하고, 그러면 테스트 목적으로 웹 서버가 시작됩니다.

HTTP 엔드포인트를 ping 테스트합니다.

SageMaker AI가 컨테이너를 실행하면 엔드포인트를 주기적으로 ping합니다. 엔드포인트가 상태 코드 200으로 HTTP 응답을 반환하면 컨테이너가 추론할 준비가 되었음을 SageMaker AI에 알립니다.

다음 명령을 실행하여 엔드포인트를 테스트하고 응답 헤더를 포함시킵니다.

```
curl --include http://127.0.0.1:8080/ping
```

다음은 출력 예시입니다.

```
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
```

추론 HTTP 엔드포인트 테스트

컨테이너가 200 상태 코드를 반환하여 준비가 되었음을 나타내면 SageMaker AI는 POST 요청을 통해 추론 데이터를 `/invocations` HTTP 엔드포인트로 전달합니다.

다음 명령을 실행하여 추론 엔드포인트를 테스트합니다.

```
curl \
  --request POST \
  --data "hello world" \
  http://127.0.0.1:8080/invocations
```

다음은 출력 예시입니다.

```
{"prediction": "a", "text": "hello world"}
```

이 두 HTTP 엔드포인트가 작동하면 이제 추론 이미지가 SageMaker AI와 호환됩니다.

> **Note**
>
> 알고리즘 제품 모델은 실시간 및 배치의 두 가지 방식으로 배포할 수 있습니다. 두 배포 모두에서 SageMaker AI는 Docker 컨테이너를 실행하는 동안 동일한 HTTP 엔드포인트를 사용합니다.

컨테이너를 중지하려면 다음 명령을 실행합니다.

```
sudo docker container stop my-inference-container
```

알고리즘 제품의 훈련 이미지와 추론 이미지를 모두 준비하고 테스트했으면 Amazon Elastic Container Registry에 이미지 업로드 섹션을 계속 진행합니다.

Amazon Elastic Container Registry에 이미지 업로드

추론 및 훈련 이미지를 만든 후 Amazon Elastic Container Registry에 업로드할 수 있습니다. Amazon ECR은 완전 관리형 Docker 레지스트리입니다. Amazon SageMaker AI는 Amazon ECR에서 이미지를

가져와 추론을 위한 모델 패키지 또는 훈련 작업을 위한 알고리즘을 생성합니다. AWS Marketplace 또한는 Amazon ECR에서 이러한 이미지를 검색하여 모델 패키지 및 알고리즘 제품을 게시합니다. 이 주제에서는 Amazon ECR에 추론 및 훈련 이미지를 업로드하는 방법에 대한 연습을 제공합니다.

주제

- 어떤 이미지를 업로드해야 하나요?
- 어떤 IAM 권한이 필요한가요?
- 에 Docker 클라이언트 로그인 AWS
- 리포지토리 생성 및 이미지 업로드
- 업로드한 이미지 스캔

어떤 이미지를 업로드해야 하나요?

모델 패키지를 게시하는 경우 추론 이미지만 업로드합니다. 알고리즘을 게시하는 경우 추론 이미지와 훈련 이미지를 모두 업로드합니다. 추론 이미지와 훈련 이미지가 결합된 경우 결합된 이미지를 한 번만 업로드합니다.

어떤 IAM 권한이 필요한가요?

다음 단계에서는 로컬 시스템에 판매자의 AWS Identity and Access Management (IAM) 역할 또는 사용자에 대한 올바른 AWS 자격 증명이 있다고 가정합니다 AWS 계정. 역할 또는 사용자는 AWS Marketplace 및 Amazon ECR 모두에 대해 올바른 정책을 마련해야 합니다. 예를 들어 다음과 같은 AWS 관리형 정책을 사용할 수 있습니다.

- AWSMarketplaceSellerProductsFullAccess -에 액세스 AWS Marketplace
- AmazonEC2ContainerRegistryFullAccess - Amazon ECR 액세스에 대한 정책

> **Note**
>
> 링크를 클릭하면 AWS 관리형 정책 참조로 이동합니다.

에 Docker 클라이언트 로그인 AWS

게시 AWS 리전 하려는의 변수를 설정합니다(참조게시 AWS 리전 지원). 이 예제의 경우 미국 동부(오하이오) 리전을 사용합니다.

```
region=us-east-2
```

다음 명령을 실행하여 AWS 계정 ID로 변수를 설정합니다. 이 예제에서는 현재 AWS Command Line Interface (AWS CLI) 자격 증명이 판매자의 자격 증명에 속한다고 가정합니다 AWS 계정.

```
account=$(aws sts get-caller-identity --query Account --output text)
```

리전의 AWS 계정 Amazon ECR Docker 레지스트리로 Docker CLI 클라이언트를 인증하려면 다음 명령을 실행합니다.

```
aws ecr get-login-password \
--region ${region} \
| sudo docker login \
--username AWS \
--password-stdin \
${account}.dkr.ecr.${region}.amazonaws.com
```

리포지토리 생성 및 이미지 업로드

업로드된 이미지 태그에 대한 변수와 업로드된 이미지 리포지토리 이름에 대한 또 다른 변수를 설정합니다.

```
image=my-inference-image
repo=my-inference-image
```

> **Note**
>
> 추론 이미지와 훈련 이미지를 빌드한 이 가이드의 이전 섹션에서는 각 이미지에 my-inference-image 및 my-training-image라는 태그를 지정했습니다. 이 예제에서는 추론 이미지를 생성하여 같은 이름의 리포지토리에 업로드합니다.

다음 명령을 실행하여 Amazon ECR에 이미지 리포지토리를 생성합니다.

```
aws ecr --region ${region} create-repository --repository-name "${repo}"
```

Amazon ECR 리포지토리 위치의 전체 이름은 `<account-id>.dkr.ecr.<region>.amazonaws.com/<image-repository-name>` 형식입니다.

이미지를 이 리포지토리로 푸시하려면 리포지토리 위치의 전체 이름을 태그로 지정해야 합니다.

`latest` 태그와 함께 이미지 리포지토리 위치의 전체 이름에 대한 변수를 설정합니다.

```
fullname="${account}.dkr.ecr.${region}.amazonaws.com/${repo}:latest"
```

전체 이름을 이미지의 태그로 지정합니다.

```
sudo docker tag ${image} ${fullname}
```

마지막으로, 추론 이미지를 Amazon ECR의 리포지토리로 푸시합니다.

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

업로드가 완료되면 게시 소스로 사용하려는 리전의 Amazon ECR 콘솔의 리포지토리 목록에 이미지가 나타납니다. 이전 예제에서는 미국 동부(오하이오) 리전의 리포지토리에 이미지를 푸시했습니다.

업로드한 이미지 스캔

Amazon ECR 콘솔에서 게시 AWS 리전 하려는를 선택하고 이미지가 업로드된 리포지토리를 엽니다. 업로드된 이미지를 선택하고 스캔을 시작하여 알려진 취약성을 확인합니다.는 Amazon SageMaker AI 리소스에 사용된 컨테이너 이미지의 Amazon ECR 스캔 결과를 AWS Marketplace 확인한 후 게시합니다. 제품을 생성하려면 심각도가 심각 또는 높음인 취약성이 발견된 컨테이너 이미지를 수정해야 합니다.

이미지를 성공적으로 스캔한 후에는 모델 패키지 또는 알고리즘 리소스를 생성하는 데 사용할 수 있습니다.

제품 스캔 결과에 오탐지 오류가 있다고 생각되면 AWS Marketplace 판매자 운영 팀에 연락하여 오류 정보를 제공하세요.

다음 단계

- 기계 학습 제품 생성을 위한 요구 사항 및 모범 사례의 파일 크기 확인
- 계속해서 Amazon SageMaker AI 리소스 생성 진행

Amazon SageMaker AI 리소스 생성

모델 패키지 또는 알고리즘 제품을 게시하려면 Amazon SageMaker AI에서 해당 모델 패키지 리소스 또는 알고리즘 리소스를 생성해야 합니다. AWS Marketplace 제품용 리소스를 생성할 때에는 검증 단

계를 통해 인증을 받아야 합니다. 검증 단계에서는 모델 패키지 또는 알고리즘 리소스를 테스트할 데이터를 제공해야 하며, 그 후 게시할 수 있습니다. 다음 섹션에서는 모델 패키지 리소스 또는 알고리즘 리소스인 SageMaker AI 리소스를 생성하는 방법을 보여줍니다. 여기에는 SageMaker AI에 검증 수행 방법을 알려주는 검증 사양 설정이 포함됩니다.

> **Note**
>
> 아직 제품 이미지를 생성하여 Amazon Elastic Container Registry(Amazon ECR)에 업로드하지 않은 경우 에서 기계 학습 제품을 위한 이미지로 코드 패키징 AWS Marketplace 및 Amazon Elastic Container Registry에 이미지 업로드에서 자세한 방법을 참조하세요.

주제

- 모델 패키지 생성
- 알고리즘 생성

모델 패키지 생성

AWS Marketplace용 모델 패키지를 생성하려면 다음과 같은 요구 사항을 충족해야 합니다.

- Amazon ECR에 저장된 추론 이미지
- (선택 사항) Amazon S3에 별도로 저장된 모델 아티팩트
- Amazon Simple Storage Service에 저장된 추론에 사용되는 테스트 데이터

> **Note**
>
> 다음은 모델 패키지 제품을 생성하는 방법에 대한 내용입니다. SageMaker AI의 모델 패키지에 대한 자세한 내용은 모델 패키지 리소스 생성을 참조하세요.

모델 패키지 리소스 생성

다음은 모델 패키지 리소스를 생성하는 과정입니다.

1단계: 모델 패키지 리소스 생성

1. Amazon SageMaker AI 콘솔을 엽니다.

2. 페이지 오른쪽 상단을 확인하여 게시하려는 AWS 리전에 있는지 확인합니다. 게시에 대한 내용은 게시 AWS 리전 지원 섹션을 참조하세요. 이전 단계에서 Amazon ECR에 업로드한 추론 이미지가 동일한 리전에 있어야 합니다.
3. 왼쪽 탐색 메뉴에서 모델 패키지를 선택합니다.
4. 모델 패키지 생성을 선택합니다.

패키지를 생성한 후에는 추론 패키지의 사양을 설정해야 합니다.

2단계: 추론 사양 설정

1. 모델 패키지의 이름(예: *my-model-package*)을 입력합니다.
2. 추론 이미지 위치에는 Amazon ECR에 업로드된 추론 이미지의 URI를 입력합니다. Amazon ECR 콘솔에서 이미지를 찾아 URI를 검색할 수 있습니다.
3. 훈련의 모델 아티팩트가 추론 이미지의 로직과 함께 번들로 제공되면 모델 데이터 아티팩트 위치를 비워 둡니다. 그렇지 않으면 모델 아티팩트 압축 파일(.tar.gz)의 전체 Amazon S3 위치를 지정합니다.
4. 드롭다운 상자를 사용하여 실시간 추론(엔드포인트라고도 함) 및 배치 변환 작업을 모두 지원하는 추론 이미지 인스턴스 유형을 선택합니다.
5. 다음을 선택합니다.

모델 패키지를 생성하고 게시하려면 먼저 모델 패키지가 예상대로 작동하는지 확인하는 검증 절차를 거쳐야 합니다. 이를 위해서는 제공한 추론용 테스트 데이터를 사용하여 배치 변환 작업을 실행해야 합니다. 검증 사양은 SageMaker AI에 검증 수행 방법을 알려줍니다.

3단계: 검증 사양 설정

1. AWS Marketplace에서 이 모델 패키지 게시를 예로 설정합니다. 아니요로 설정하면 나중에 이 모델 패키지를 게시할 수 없습니다. 예를 선택하면에 대한 모델 패키지가 인증 AWS Marketplace 되고 검증 단계가 필요합니다.
2. 이 프로세스를 처음으로 완료하는 경우 IAM 역할로 새 역할 생성을 선택합니다. Amazon SageMaker AI는 모델 패키지를 배포할 때이 역할을 사용합니다. 여기에는 Amazon ECR에서 이미지를 가져오고 Amazon S3에서 아티팩트를 가져오는 등의 작업이 포함됩니다. 설정을 검토하고 역할 생성을 선택합니다. 여기서 역할을 생성하면 생성한 역할에 AmazonSageMakerFullAccess IAM 정책에서 설명하는 권한이 부여됩니다.
3. 검증 프로필에서 JSON을 편집합니다. 허용되는 값에 대한 자세한 내용은 TransformJobDefinition을 참조하세요.

1. `TransformInput.DataSource.S3Uri`: 추론용 테스트 데이터가 저장되는 위치로 설정합니다.
2. `TransformInput.ContentType`: 테스트 데이터 콘텐츠 유형(예:, `application/json`, `text/plain`, `image/png` 또는 기타 값)을 지정합니다. SageMaker AI는 실제 입력 데이터를 검증하지 않습니다. 이 값은 `Content-type` 헤더 값의 컨테이너 HTTP 엔드포인트에 전달됩니다.
3. `TransformInput.CompressionType`: Amazon S3의 추론용 테스트 데이터가 압축되지 않았으면 `None`으로 설정합니다.
4. `TransformInput.SplitType`: 추론에 사용할 Amazon S3의 각 객체를 전체적으로 전달하려면 `None`으로 설정합니다.
5. `TransformOutput.S3OutputPath`: 추론 출력이 저장되는 위치로 설정합니다.
6. `TransformOutput.AssembleWith`: Amazon S3에서 각 추론을 별도의 객체로 출력하려면 `None`으로 설정합니다.

4. 모델 패키지 생성을 선택합니다.

SageMaker AI는 Amazon ECR에서 추론 이미지를 가져오고, 모든 아티팩트를 추론 컨테이너에 복사하고, 추론을 위해 테스트 데이터를 사용하여 배치 변환 작업을 실행합니다. 검증이 성공하면 상태가 완료로 변경됩니다.

> **Note**
> 검증 단계에서는 테스트 데이터로 모델의 정확도를 평가하지 않습니다. 검증 단계에서는 컨테이너가 실행되고 예상대로 응답하는지 확인합니다.

모델 제품 리소스 생성을 완료했습니다. 계속해서 에 제품 나열 AWS Marketplace로 이동하세요.

알고리즘 생성

AWS Marketplace용 알고리즘을 생성하려면 다음과 같은 요구 사항을 충족해야 합니다.

- Amazon ECR에 저장된 추론 이미지
- Amazon ECR에 저장된 훈련 이미지
- Amazon S3에 저장된 훈련용 테스트 데이터
- Amazon S3에 저장된 추론용 테스트 데이터

> **Note**
> 다음은 알고리즘 제품을 생성하는 연습입니다. 자세한 내용은 알고리즘 리소스 생성을 참조하세요.

알고리즘 리소스 생성

다음은 알고리즘 패키지에 리소스를 생성하는 과정입니다.

1단계: 알고리즘 리소스 생성

1. Amazon SageMaker AI 콘솔을 엽니다.
2. 페이지 오른쪽 상단을 확인하여 게시하려는 AWS 리전에 있는지 확인합니다(참조게시 AWS 리전 지원). 이전 단계에서 Amazon ECR에 업로드한 추론 이미지가 이 리전과 동일한 리전에 있어야 합니다.
3. 왼쪽 탐색 창에서 알고리즘을 선택합니다.
4. 알고리즘 생성을 선택합니다.

알고리즘 패키지를 생성한 후에는 모델의 훈련 및 튜닝 사양을 설정해야 합니다.

2단계: 훈련 및 튜닝 사양 설정

1. 알고리즘의 이름(예: *my-algorithm*)을 입력합니다.
2. 훈련 이미지의 경우 Amazon ECR에 업로드한 훈련 이미지의 전체 URI 위치를 붙여넣습니다. Amazon ECR 콘솔에서 이미지를 찾아 URI를 검색할 수 있습니다.
3. 드롭다운 상자를 사용하여 훈련 이미지가 지원하는 훈련용 인스턴스 유형을 선택합니다.
4. 채널 사양 섹션에서 알고리즘이 지원하는 각 입력 데이터 세트의 채널을 추가합니다. 입력 소스 채널을 20개까지 추가할 수 있습니다. 자세한 내용은 입력 데이터 구성을 참조하세요.
5. 다음을 선택합니다.
6. 알고리즘이 하이퍼파라미터 및 하이퍼파라미터 튜닝을 지원하는 경우 튜닝 파라미터를 지정해야 합니다.
7. 다음을 선택합니다.

> **Note**
> 알고리즘이 하이퍼파라미터 튜닝을 지원하고 적절한 파라미터를 튜닝 가능하게 만드는 것이 좋습니다. 이렇게 하면 데이터 과학자가 모델을 튜닝하여 최상의 결과를 얻을 수 있습니다.

튜닝 파라미터(있는 경우)를 설정한 후에는 추론 이미지의 사양을 설정해야 합니다.

3단계: 추론 이미지 사양 설정

1. 추론 이미지 위치에는 Amazon ECR에 업로드된 추론 이미지의 URI를 붙여넣습니다. Amazon ECR 콘솔에서 이미지를 찾아 URI를 검색할 수 있습니다.
2. 드롭다운 상자를 사용하여 실시간 추론(엔드포인트라고도 함) 및 배치 변환 작업을 모두 지원하는 추론 이미지의 인스턴스 유형을 선택합니다.
3. 다음을 선택합니다.

알고리즘을 생성하고 게시하려면 먼저 모델 패키지가 예상대로 작동하는지 확인하는 검증 절차를 거쳐야 합니다. 이를 위해서는 훈련용 테스트 데이터로 훈련 작업을 실행하고, 제공한 추론용 테스트 데이터로 배치 변환 작업을 실행해야 합니다. 검증 사양은 SageMaker AI에 검증 수행 방법을 알려줍니다.

4단계: 검증 사양 설정

1. AWS Marketplace에서 이 알고리즘 게시를 예로 설정합니다. 아니요로 설정하면 나중에 이 알고리즘을 게시할 수 없습니다. 예를 선택하면에 대한 알고리즘이 인증 AWS Marketplace 되고 검증 사양이 필요합니다.
2. 기계 학습 패키지를 처음 생성하는 경우 IAM 역할에 대한 새 역할 생성을 AWS Marketplace선택합니다. Amazon SageMaker AI는 알고리즘을 훈련하고 후속 모델 패키지를 배포할 때이 역할을 사용합니다. 여기에는 Amazon ECR에서 이미지를 가져오고 Amazon S3에 아티팩트를 저장하고 Amazon S3에서 훈련 데이터를 복사하는 등의 작업이 포함됩니다. 설정을 검토하고 역할 생성을 선택합니다. 여기서 역할을 생성하면 생성한 역할에 AmazonSageMakerFullAccess IAM 정책에서 설명하는 권한이 부여됩니다.
3. 훈련 작업 정의에 대한 검증 프로필에서 JSON 파일을 편집합니다. 허용되는 값에 대한 자세한 내용은 TrainingJobDefinition을 참조하세요.
    1. `InputDataConfig`: 이 JSON 배열에서, 훈련 사양 단계에서 지정한 각 채널의 채널 객체를 추가합니다. 채널마다 훈련용 테스트 데이터가 저장되는 위치를 지정합니다.

2. `OutputDataConfig`: 훈련이 완료되면 훈련 컨테이너 디렉터리 경로 `/opt/ml/model/`의 모델 아티팩트가 압축되어 Amazon S3에 복사됩니다. 압축 파일(.tar.gz)이 저장되는 Amazon S3 위치를 지정합니다.

4. 변환 작업 정의에 대한 검증 프로필에서 JSON 파일을 편집합니다. 허용되는 값에 대한 자세한 내용은 TransformJobDefinition을 참조하세요.

   1. `TransformInput.DataSource.S3Uri`: 추론용 테스트 데이터가 저장되는 위치로 설정합니다.
   2. `TransformInput.ContentType`: 테스트 데이터 콘텐츠 유형을 지정합니다. 예: `application/json`, `text/plain`, `image/png` 또는 기타 값. Amazon SageMaker AI는 실제 입력 데이터를 검증하지 않습니다. 이 값은 `Content-type` 헤더 값의 컨테이너 HTTP 엔드포인트에 전달됩니다.
   3. `TransformInput.CompressionType`: Amazon S3의 추론용 테스트 데이터가 압축되지 않았으면 `None`으로 설정합니다.
   4. `TransformInput.SplitType`: S3의 객체 분할 방식을 선택합니다. 예를 들어 `None`은 추론에 사용할 Amazon S3의 각 객체를 전체적으로 전달합니다. 자세한 내용은 Amazon SageMaker AI API 참조의 SplitType을 참조하세요.
   5. `TransformOutput.S3OutputPath`: 추론 출력이 저장되는 위치로 설정합니다.
   6. `TransformOutput.AssembleWith`: Amazon S3에서 각 추론을 별도의 객체로 출력하려면 `None`으로 설정합니다.

5. 알고리즘 패키지 생성을 선택합니다.

SageMaker AI는 Amazon ECR에서 훈련 이미지를 가져오고, 데이터를 사용하여 테스트 훈련 작업을 실행하고, 모델 아티팩트를 Amazon S3에 저장합니다. 그 후 Amazon ECR에서 추론 이미지를 가져오고, Amazon S3의 아티팩트를 추론 컨테이너에 복사하고, 제공된 추론용 테스트 데이터를 사용하여 배치 변환 작업을 실행합니다. 검증이 성공하면 상태가 완료로 변경됩니다.

> **Note**
>
> 검증 단계에서는 테스트 데이터로 훈련 또는 모델의 정확도를 평가하지 않습니다. 검증 단계에서는 컨테이너가 실행되고 예상대로 응답하는지 확인합니다.
> 검증 단계에서는 배치 처리만 검증합니다. 실시간 처리가 제품에서 작동하는지 확인하는 것은 판매자의 몫입니다.

알고리즘 제품 리소스 생성을 완료했습니다. 계속해서 에 제품 나열 AWS Marketplace로 이동하세요.

에 제품 나열 AWS Marketplace

코드를 모델 패키지 이미지 또는 알고리즘 이미지로 패키징하고, 이미지를 업로드하고, Amazon SageMaker AI 리소스를 생성한 후에 기계 학습 제품을 게시할 수 있습니다 AWS Marketplace. 다음 섹션에서는 제품 목록 생성, 제품 테스트, 게시 승인 등 게시 프로세스를 안내합니다. 제품이 게시되면 변경 사항을 요청하여 목록을 업데이트합니다. 자세한 내용은 the section called “제품 관리” 단원을 참조하십시오.

주제

- 사전 조건
- 1단계: 새 목록 생성
- 2단계: 제품 정보 제공
- 3단계: 초기 제품 버전 추가
- 4단계: 요금 모델 구성
- 5단계: 환불 정책 구성
- 6단계: EULA 구성
- 7단계: 허용 목록 구성

사전 조건

모델 패키지 또는 알고리즘을에 게시하려면 먼저 다음이 있어야 AWS Marketplace합니다.

- AWS Marketplace 판매자로 AWS 계정 등록된 입니다. AWS Marketplace Management Portal에서 등록할 수 있습니다.
- AWS Marketplace Management Portal의 설정 페이지에서 완성된 판매자 프로필.
- 유료 제품을 게시하려면 세금 인터뷰와 은행 양식을 완료해야 합니다. 무료 제품을 게시할 때는 필수 사항이 아닙니다. 자세한 내용은 판매자 등록 프로세스를 참조하세요.
- AWS Marketplace Management Portal 및 Amazon SageMaker AI에 액세스할 수 있는 권한이 있어야 합니다. 자세한 내용은 필요한 권한 단원을 참조하십시오.

필요한 권한

Amazon SageMaker AI 제품을 게시하려면 AWS Marketplace 서비스 보안 주체와 신뢰 관계가 있는 유효한 IAM 역할 ARN을 지정해야 합니다. 또한 로 로그인한 IAM 사용자 또는 역할에는 필요한 권한이 필요합니다.

로그인 권한 설정

- IAM 역할에 다음 권한을 추가합니다.

  1. sagemaker:DescribeModelPackage — 모델 패키지 나열용
  2. sagemaker:DescribeAlgorithm — 알고리즘 나열용

     JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeModelPackage",
                "sagemaker:DescribeAlgorithm"
            ],
            "Resource": "*"
       }
    ]
}
```

IAM 역할 AddVersion/Create 제품 설정

1. 단계에 따라 사용자 지정 신뢰 정책을 사용하여 역할을 생성합니다. 자세한 내용은 사용자 지정 신뢰 정책을 사용하여 IAM 역할 생성(콘솔)을 참조하세요.
2. 사용자 지정 신뢰 정책 문에 다음을 입력합니다.

   JSON

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

```
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": "assets.marketplace.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

3. 다음 권한 정책을 입력합니다.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:DescribeModelPackage",
                "sagemaker:DescribeAlgorithm"
            ],
            "Resource": "*"
       }
    ]
}
```

4. 요청 시 역할 ARN을 제공합니다. 역할은 형식을 따라야 합니다`arn:aws:iam::`**`<account-id>`**`:role/`**`<role-name>`**.

필요한 AWS Marketplace 권한 또는 판매자 계정 관리에 대한 자세한 내용은 AWS Marketplace 판매자에 대한 정책 및 권한을 참조하세요.

필수 자산

기계 학습 제품 목록을 생성하기 전에, 다음과 같은 필수 자산이 있는지 확인합니다.

- Amazon 리소스 이름(ARN) - 게시 AWS 리전 하려는에서 모델 패키지 또는 알고리즘 리소스의 ARN을 제공합니다(참조게시 AWS 리전 지원).

- 모델 패키지의 ARN은 `arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>` 형식입니다.

  모델 패키지 ARN을 찾으려면 내 마켓플레이스 모델 패키지를 참조하세요.
- 알고리즘의 ARN은 `arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>` 형식입니다.

  알고리즘 리소스 ARN을 찾으려면 내 알고리즘을 참조하세요.
- the section called "사용 정보 요구 사항" - 입력, 출력 및 코드 예제에 대한 세부 정보를 제공합니다.
- the section called "입력 및 출력 요구 사항" - 파일 또는 텍스트를 제공합니다.
- the section called "Jupyter Notebook 요구 사항" - 전체 제품 사용량을 보여줍니다.

1단계: 새 목록 생성

기계 학습 제품을 시작하려면 제품 이름을 설정하고, 조직에 대한 선택적 리소스 태그를 추가하고, 제품 ID를 생성하여 나열 프로세스를 시작합니다. 제품 ID는 제품 수명 주기 전반에 걸쳐 제품을 추적하는 데 사용됩니다.

> **Note**
>
> 목록을 생성하기 전에, 기계 학습 제품 생성을 위한 요구 사항 및 모범 사례에 지정된 필수 리소스가 있는지 확인하세요.

1. 판매자에 로그인 AWS 계정 하고 로 이동합니다AWS Marketplace Management Portal.
2. 상단 메뉴에서 제품으로 이동한 다음 기계 학습을 선택합니다.
3. 기계 학습 제품 생성을 선택합니다.
4. 제품 이름에 제품 목록 페이지 상단과 검색 결과에 구매자에게 표시될 고유한 제품 이름을 입력합니다.
5. (선택 사항) 태그에서 제품과 연결할 태그를 입력합니다. 자세한 내용은 AWS 리소스 태그 지정을 참조하세요.
6. 제품 ID 및 코드에서 제품 ID 및 코드 생성을 선택합니다.
7. 마법사로 계속을 선택합니다. 마법사에서 자세한 제품 정보를 추가하는 프로세스를 시작합니다.

2단계: 제품 정보 제공

기계 학습 제품을에 나열할 때는 포괄적이고 정확한 제품 정보를 AWS Marketplace제공하는 것이 중요합니다. 마법사의 제품 정보 제공 단계를 사용하여 제품 범주 및 지원 정보와 같은 상품에 대한 필수 세부 정보를 캡처합니다.

1. 제품에 대한 정보를 입력합니다.
2. 다음을 선택하여 마법사의 다음 단계로 이동합니다.

3단계: 초기 제품 버전 추가

이 페이지에서는 제품의 초기 버전을 추가하는 방법을 안내합니다. 제품은 수명 주기 동안 여러 버전을 가질 수 있으며 각 버전은 고유한 SageMaker AI ARN으로 식별됩니다.

1. Amazon 리소스 이름(ARNs:
   a. 모델 또는 알고리즘 Amazon SageMaker AI ARN을 입력합니다.
      - 예제 모델 패키지 ARN: `arn:aws:sagemaker:<region>:<account-id>:model-package/<model-package-name>`

        모델 패키지 ARN을 찾으려면 내 마켓플레이스 모델 패키지를 참조하세요.
      - 알고리즘 ARN의 예: `arn:aws:sagemaker:<region>:<account-id>:algorithm/<algorithm-name>`

        알고리즘 리소스 ARN을 찾으려면 내 알고리즘을 참조하세요.
   b. IAM 액세스 역할 ARN을 입력합니다.

      IAM ARN 예: `arn:aws:iam::<account-id>:role/<role-name>`
2. 버전 정보에서 버전 이름과 릴리스 정보를 입력합니다.
3. 모델 입력 세부 정보에서 모델 입력 요약을 입력하고 실시간 및 배치 작업 입력에 대한 샘플 입력 데이터를 제공합니다. 선택적으로 입력 제한을 제공할 수 있습니다.
4. (선택 사항) 입력 파라미터에서 제품에서 지원하는 각 입력 파라미터에 대한 자세한 정보를 제공합니다. 파라미터 이름, 설명, 제약 조건을 제공하고 파라미터가 필요한지 아니면 선택 사항인지 지정할 수 있습니다. 최대 24개의 입력 파라미터를 제공할 수 있습니다.

5. (선택 사항) 사용자 지정 속성에서 제품에서 지원하는 사용자 지정 호출 파라미터를 제공합니다. 각 속성에 대해 이름, 설명, 제약 조건을 제공하고 속성이 필요한지 아니면 선택 사항인지 지정할 수 있습니다.
6. 모델 출력 세부 정보에서 모델 출력 요약을 입력하고 실시간 및 배치 작업 출력에 대한 샘플 출력 데이터를 제공합니다. 선택적으로 출력 제한을 제공할 수 있습니다.
7. (선택 사항) 출력 파라미터에서 제품에서 지원하는 각 출력 파라미터에 대한 자세한 정보를 제공합니다. 파라미터 이름, 설명, 제약 조건을 제공하고 파라미터가 필요한지 아니면 선택 사항인지 지정할 수 있습니다. 최대 24개의 출력 파라미터를 제공할 수 있습니다.
8. 사용 지침에서 모범 사례, 일반적인 엣지 사례 처리 방법 또는 성능 최적화 제안과 같이 모델을 효과적으로 사용하기 위한 명확한 지침을 제공합니다.
9. Git 리포지토리 및 노트북 링크에서 예제 노트북 및 Git 리포지토리에 대한 링크를 제공합니다. 샘플 노트북에는 모델을 호출하는 방법이 포함되어야 합니다. Git 리포지토리에는 노트북, 데이터 파일 및 기타 개발자 도구가 포함되어야 합니다.
10. 권장 인스턴스 유형에서 제품의 권장 인스턴스 유형을 선택합니다.

    모델 패키지의 경우 배치 변환과 실시간 추론 모두에 권장되는 인스턴스 유형을 선택합니다.

    알고리즘 패키지의 경우 훈련 작업에 권장되는 인스턴스 유형을 선택합니다.

    > **Note**
    >
    > 선택할 수 있는 인스턴스 유형은 모델 또는 알고리즘 패키지에서 지원하는 인스턴스 유형으로 제한됩니다. 지원되는 이러한 인스턴스 유형은 Amazon SageMaker AI에서 리소스를 처음 생성할 때 결정되었습니다. 이렇게 하면 제품이 기계 학습 솔루션을 효과적으로 실행할 수 있는 하드웨어 구성에만 연결됩니다.

11. 다음을 선택하여 마법사의 다음 단계로 이동합니다.

4단계: 요금 모델 구성

제품의 요금 모델을 구성할 때 제품을 무료로 제공하거나 사용량 기반 요금을 구현할 수 있습니다. 제품을 게시한 후에는 요금 모델을 변경할 수 없습니다.

1. 요금 모델을 선택합니다. 배치 변환 및 알고리즘 훈련 제품은 시간당 사용량에 대해서만 무료 또는 요금이 청구될 수 있습니다.
   - 제품을 무료로 제공하기로 선택한 경우 다음을 선택하고 마법사를 계속합니다.

- 사용 요금을 선택한 경우 다음 단계를 계속합니다.

2. 사용량에 따라 요금을 청구하기로 선택한 경우 사용 비용을 입력할 수 있습니다. 모든 인스턴스 유형에 적용되는 가격을 입력하거나 보다 세분화된 요금을 위해 인스턴스 유형당 가격을 입력할 수 있습니다.
3. 제품의 무료 평가판을 제공하려면 예, 무료 평가판을 제공합니다를 선택합니다.
4. 다음을 선택하여 마법사의 다음 단계로 이동합니다.

5단계: 환불 정책 구성

환불을 제공할 필요는 없지만 공식 환불 정책을 제출해야 합니다 AWS Marketplace.

1. 환불 정책을 입력합니다.
2. 다음을 선택하여 마법사의 다음 단계로 이동합니다.

6단계: EULA 구성

이 단계에서는 고객이 제품을 사용하는 방법을 규정하는 법적 계약을 선택합니다. AWS의 표준 계약 조건을 선택하거나 사용자 지정 최종 사용자 라이선스 계약(EULA)을 업로드할 수 있습니다.

1. 표준 계약을 선택하거나 사용자 지정 최종 사용자 라이선스 계약을 제공합니다.
2. 다음을 선택하여 마법사의 다음 단계로 이동합니다.

7단계: 허용 목록 구성

제품을 제출하기 전에 제품에 액세스할 AWS 계정 수 있는 항목을 지정해야 합니다. 이 선택적 단계는 제품의 초기 가시성을 제어하여 자체 계정 및 허용 목록에 AWS 계정 추가하는 특별히 승인된 모든에 대한 액세스를 제한합니다.

1. 제품에 액세스하려는 AWS 계정 IDs를 입력합니다.
2. 제출을 선택하여 제품을 제출합니다.

   제품은 제한된 가시성 상태를 가지며 제품을 AWS 계정 생성한 및 기타 허용 목록에 있는 에만 표시됩니다 AWS 계정.

   상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

제한된 가시성에 있는 동안 제품 목록을 보고 테스트할 수 있습니다. 제품의 가시성을 변경할 준비가 되면 섹션을 참조하세요the section called “”.

기계 학습 제품 관리

AWS Marketplace 관리 포털에서 변경 요청을 선택하여의 제품 또는 버전을 수정합니다 AWS Marketplace. 변경 사항을 제출하면 시스템에서 변경 사항을 처리합니다. 처리 시간은 수정 유형에 따라 분마다 다릅니다. AWS Marketplace Management Portal에서 변경 상태를 모니터링할 수 있습니다.

다음을 통해 셀프 서비스를 AWS Marketplace 통해에서 ML 제품을 수정할 수 있습니다.

- 제품 정보 업데이트
- 제품 표시 여부 업데이트
- 허용 목록 업데이트
- 제품 버전 관리
- 제품 요금 업데이트
- 환불 정책 업데이트
- EULA 업데이트
- 제품 제거

> **Note**
>
> AWS Marketplace Management Portal을 통해 변경하는 것 외에도를 사용하여 변경할 수도 있습니다AWS Marketplace Catalog API.

제품 정보 업데이트

기계 학습(ML) 제품을 생성한 후 설명, 하이라이트 AWS Marketplace, 제목, SKU, 범주 및 키워드와 같은의 특정 제품 정보를 수정할 수 있습니다.

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 대상 제품을 선택합니다.
3. 변경 요청을 선택하고 제품 정보 업데이트를 선택합니다.
4. 필요에 따라 필드를 업데이트합니다.

> **Note**
> 로고 사양은 섹션을 참조하세요the section called “회사 및 제품 로고 요구 사항”.

5. 제출을 선택합니다.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

제품 표시 여부 업데이트

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 변경 요청을 선택하고 제품 가시성 업데이트를 선택한 다음 공개 또는 제한을 선택합니다.
4. 변경 사항을 검토하고 제출을 선택합니다.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

허용 목록 업데이트

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 변경 요청을 선택하고 허용 목록 업데이트를 선택합니다.
4. 변경해야 하는 정보를 수정하고 제출을 선택합니다. 자세한 내용은 the section called “7단계: 허용 목록 구성” 단원을 참조하십시오.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

제품 버전 관리

판매자는 기존 버전 정보를 업데이트 AWS Marketplace 하거나 새 버전을 추가하거나 더 이상 지원되지 않는 버전을 제거하여에서 제품 버전을 관리할 수 있습니다. 각 버전에는 구매자가 제품을 평가하고 배포하는 데 사용하는 고유한 SageMaker AI ARN 및 관련 정보가 있습니다.

> **Note**
> 버전을 추가하기 전에 제품 ID를 생성하고 요금을 설정합니다. 자세한 내용은 the section called "1단계: 새 목록 생성" 단원을 참조하십시오.

버전 정보 업데이트

버전을 생성한 후 릴리스 정보, 사용 지침 및 인스턴스 권장 사항과 같은 관련 정보를 수정할 수 있습니다.

> **Note**
> 버전 이름과 ARNs 수정할 수 없습니다. 이러한 변경 사항을 수행하려면 새 버전을 생성해야 합니다.

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 변경 요청을 선택하고 버전 정보 업데이트를 선택합니다.
4. 업데이트할 버전을 선택합니다.
5. 버전 편집을 선택합니다.
6. 필요한 필드를 수정하고 다음을 선택합니다.
7. 요금 정보를 입력하고 제출을 선택합니다. 자세한 내용은 the section called "4단계: 요금 모델 구성" 단원을 참조하십시오.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called "제품 상태".

새 버전 추가

제품의 새 버전을 추가하여 이전 버전에 대한 액세스를 유지하면서 기능, 업데이트 또는 개선 사항을 도입할 수 있습니다.

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 버전을 선택하고 새 버전 추가를 선택합니다.

4. 의 단계에 따라 새 버전에 대한 정보를 입력합니다the section called “3단계: 초기 제품 버전 추가”.
5. 요금 정보를 입력하고 제출을 선택합니다. 자세한 내용은 the section called “4단계: 요금 모델 구성” 단원을 참조하십시오.

새 버전을 성공적으로 추가하면 구매자에게 새 버전을 사용할 수 있다는 이메일 알림이 전송됩니다.

버전 제한

버전이 오래되었거나 가용성을 중단하려는 경우 다른 버전에 대한 액세스를 유지하면서 해당 버전에 대한 구매자 액세스를 제한할 수 있습니다.

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 버전을 선택하고 버전 제한을 선택합니다.

> **Note**
> 항상 하나 이상의 버전을 사용할 수 있어야 합니다.

4. 제출을 선택합니다.

버전을 성공적으로 제한하면 구매자에게 버전이 제한되었다는 이메일 알림이 전송됩니다.

제품 요금 업데이트

요금 모델 자체는 변경할 수 AWS Marketplace없지만에서 기계 학습 제품의 요금 및 무료 평가판 기간을 수정할 수 있습니다. 유료 모델의 경우 가격 인상은 다음 달 1일에 90일 통지 기간 이후에 적용됩니다. 이 알림 기간 동안에는 추가 가격 변경을 수행할 수 없습니다.

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 변경 요청을 선택하고 공개 제안 업데이트를 선택한 다음 제안 정보 편집을 선택합니다.
4. 변경해야 하는 정보를 수정하고 제출을 선택합니다.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

환불 정책 업데이트

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 변경 요청을 선택하고 공개 제안 업데이트를 선택한 다음 환불 정책 업데이트를 선택합니다.
4. 변경해야 하는 정보를 수정하고 제출을 선택합니다.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

EULA 업데이트

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.
2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 변경 요청을 선택하고 공개 제안 업데이트를 선택한 다음 EULA 업데이트를 선택합니다.
4. 변경해야 하는 정보를 수정하고 제출을 선택합니다.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

제품 제거

게시된 제품을에서 제거(일몰)할 수 있습니다 AWS Marketplace. 제거된 신규 고객은 구독할 수 없지만, 최소 90일 동안 기존 고객을 지원해야 합니다.

다음은 제품을 제거하는 조건입니다 AWS Marketplace.

- 검색 AWS Marketplace 및 검색 도구에서 제품이 제거됩니다.
- 구독 기능이 비활성화됩니다.
- 제품 세부 정보 페이지는 직접 URL을 통해 액세스할 수 있습니다.
- 현재 구독자는 구독을 취소할 때까지 액세스 권한을 유지합니다.
- AWS Marketplace 는 현재 구매자에게 제거 사실을 알립니다.

기계 학습 제품을 제거하려면:

1. AWS Marketplace Management Portal에서 판매자 계정에 로그인합니다.

2. 기계 학습 제품 페이지로 이동하여 제품을 선택합니다.
3. 변경 요청을 선택하고 제품 가시성 업데이트를 선택한 다음 제한을 선택합니다.
4. (선택 사항) 대체 제품 ID를 입력합니다.
5. 변경 사항을 검토한 다음 제출을 선택합니다.

기계 학습 제품 페이지의 요청 탭에서 요청을 모니터링할 수 있습니다. 상태에 대한 자세한 내용은 섹션을 참조하세요the section called “제품 상태”.

제거되면 현재 제품 목록에 제품 스프레드시트만 다운로드할 수 있는 제품이 나타납니다. 제품 제거에 대한 질문이 있는 경우 AWS Marketplace 판매자 운영 팀에 문의하세요.

기계 학습 제품에 대한 비공개 제안 생성

기계 학습 제품에 대한 비공개 제안을 협상하고 고객에게 직접 제공할 수 있습니다. 비공개 제안에 대한 자세한 내용은 섹션을 참조하세요비공개 제안 준비.

사전 조건:

- 유료 리스팅이 있어야 합니다 AWS Marketplace.
- AWS Marketplace Management Portal(AMMP)에 액세스할 수 있어야 합니다.

기계 학습 제품에 대한 비공개 제안을 생성하려면:

1. AWS Marketplace Management Portal에 로그인합니다.
2. 제안을 선택한 다음 비공개 제안 생성을 선택합니다.
3. 비공개 제안 생성 페이지에서 비공개 제안을 생성하려는 제품을 선택합니다. 사용 가능한 제품에 대한 제안만 생성할 수 있습니다.
4. 제안 세부 정보 페이지에서:
   a. 제안 이름과 설명을 입력합니다.
   b. 갱신 옵션을 선택합니다.
   c. 제안 만료 날짜를 설정합니다. 제안은 설정된 날짜의 23:59:59 UTC에 만료됩니다.
5. 다음을 두 번 선택합니다
6. 제안 요금 및 기간 구성 페이지에서 다음을 지정합니다.
   - 요금 옵션

(자세한 내용은 the section called "기계 학습 제품 요금" 섹션을 참조하세요.)

- 계약 기간
- 제안 통화
- 제품 차원

> **Note**
> 할부 플랜에 대한 자세한 내용은 섹션을 참조하세요비공개 제안 할부 플랜.

7. 다음을 선택합니다.
8. 구매자 추가 페이지에서 구매자의 AWS 계정 IDs를 입력합니다. 그런 다음 다음을 선택합니다.

> **Important**
> 연결된 계정이 비공개 제안을 활용하는 경우:
>
> - 지급인 AWS 계정 ID를 포함합니다.
> - 지급인 계정은 먼저 비공개 제안의 시간당 조건을 수락해야 합니다.
> - 지급인 계정이 수락한 후 연결된 계정은 비공개 제안을 수락할 수 있습니다.

9. 법률 약관 및 제안 문서 구성 페이지에서 사용자 지정 약관을 추가하고 다음을 선택합니다.

> **Note**
> 최대 5개의 파일(법률 조건, 작업 명세서, 재료표, 요금표 또는 부록)을 추가할 수 있습니다. 시스템은 이를 하나의 문서로 결합합니다.

10. 검토 및 생성 페이지에서 제안 세부 정보를 확인하고 제안 생성을 선택합니다.
11. 비공개 제안 관리 페이지에 제안이 표시되면 작업 메뉴를 열고 제안 URL 복사를 선택한 다음 구매자에게 이메일로 보냅니다.

> **Note**
> 제안을 게시하는 데 시간이 걸릴 수 있습니다. 구매자가 수락할 때까지 비공개 제안 관리 페이지에서 제안을 편집할 수 있습니다.

기계 학습 제품 생성을 위한 요구 사항 및 모범 사례

구매자가 모델 패키지와 알고리즘 제품을 쉽게 테스트할 수 있어야 합니다. 다음 섹션에서는 ML 제품의 모범 사례를 설명합니다. 요구 사항 및 권장 사항의 전체 요약은 ML 제품 목록의 요구 사항 및 권장 사항 요약 섹션을 참조하세요.

> **Note**
>
> 게시된 제품이 요구 사항을 충족하지 않는 경우 AWS Marketplace 담당자가 연락하여 이러한 요구 사항을 충족할 수 있습니다.

주제

- ML 제품의 일반 모범 사례
- 사용 정보 요구 사항
- 입력 및 출력 요구 사항
- Jupyter Notebook 요구 사항
- ML 제품 목록의 요구 사항 및 권장 사항 요약

ML 제품의 일반 모범 사례

기계 학습 제품에 대한 다음 정보를 제공합니다.

- 제품 설명에 다음 정보가 포함되어야 합니다.
  - 모델이 하는 일
  - 대상 고객
  - 가장 중요한 사용 사례
  - 모델을 훈련한 방법 또는 사용된 데이터의 양
  - 성능 지표의 정의 및 사용된 검증 데이터
  - 의료용인 경우 모델이 진단용인지 여부
- 기본적으로 기계 학습 제품은 대중에게 공개되도록 구성됩니다. 그러나 가시성이 제한된 제품을 생성할 수 있습니다. 자세한 내용은 the section called “7단계: 허용 목록 구성” 단원을 참조하십시오.
- (선택 사항) 유료 제품의 경우 고객이 제품을 사용해 볼 수 있도록 14~30일 무료 평가판을 제공합니다. 자세한 내용은 에 대한 기계 학습 제품 요금 AWS Marketplace 단원을 참조하십시오.

사용 정보 요구 사항

제품의 예상 입력 및 출력을 설명하는 명확한 사용 정보(예시 포함)는 긍정적인 구매자 경험을 유도하는 데 매우 중요합니다.

제품 목록에 새 버전의 리소스를 추가할 때마다 사용 정보를 제공해야 합니다.

특정 버전의 기존 사용 정보를 편집하려면 섹션을 참조하세요the section called “버전 정보 업데이트”.

입력 및 출력 요구 사항

지원되는 입력 파라미터와 반환된 출력 파라미터에 대한 명확한 설명과 예제는 구매자가 제품을 이해하고 사용하는 데 도움이 됩니다. 이러한 이해를 통해 구매자는 입력 데이터에 필요한 변환을 수행하여 최상의 추론 결과를 얻을 수 있습니다.

Amazon SageMaker AI 리소스를 제품 목록에 추가할 때 다음을 입력하라는 메시지가 표시됩니다.

추론 입력 및 출력

추론 입력의 경우 제품이 실시간 엔드포인트 및 배치 변환 작업 모두에 대해 기대하는 입력 데이터에 대한 설명을 제공합니다. 필요한 데이터 전처리용 코드 조각을 포함합니다. 해당하는 경우 제한 사항을 포함합니다. GitHub에서 호스팅되는 입력 샘플을 제공합니다.

추론 출력의 경우 제품이 실시간 엔드포인트 및 배치 변환 작업 모두에 대해 반환하는 출력 데이터에 대한 설명을 제공합니다. 해당하는 경우 제한 사항을 포함합니다. GitHub에서 호스팅되는 출력 샘플을 제공합니다.

샘플의 경우 제품에서 작동하는 입력 파일을 제공합니다. 모델에서 멀티클래스 분류를 수행하는 경우 각 클래스에 대해 하나 이상의 샘플 입력 파일을 제공합니다.

훈련 입력

모델 훈련을 위한 정보 섹션에서 입력 데이터 형식과 필요한 데이터 전처리용 코드 조각을 제공합니다. 해당하는 경우 값 및 제한 사항에 대한 설명을 포함합니다. GitHub에서 호스팅되는 입력 샘플을 제공합니다.

구매자가 제공할 수 있는 선택적 기능과 필수 기능을 모두 설명하고, `PIPE` 입력 모드의 지원 여부를 지정합니다. 분산 훈련(CPU/GPU 인스턴스를 2개 이상 사용하는 훈련)이 지원되는 경우 이 옵션을 지정합니다. 튜닝의 경우 권장 하이퍼파라미터를 기재합니다.

Jupyter Notebook 요구 사항

제품 목록에 SageMaker AI 리소스를 추가할 때 구매자에게 데이터를 업로드하거나 찾도록 요청하지 않고 전체 워크플로를 보여주는 GitHub에서 호스팅되는 샘플 Jupyter Notebook에 대한 링크를 제공합니다.

AWS SDK for Python (Boto)를 사용합니다. 잘 개발된 샘플 노트북을 사용하면 구매자가 목록을 더 쉽게 사용해 볼 수 있습니다.

모델 패키지 제품의 경우 샘플 노트북은 입력 데이터 준비, 실시간 추론을 위한 엔드포인트 생성, 배치 변환 작업 수행을 시연합니다. 자세한 내용은 GitHub의 모델 패키지 목록 및 샘플 노트북을 참조하세요. 샘플 노트북은 auto_insurance를 참조하세요. 노트북은 파라미터를 입력 AWS 리전하지 않고 구매자가 샘플 데이터를 찾을 필요 없이 모두 작동합니다.

> **Note**
>
> 여러 가능한 입력 및 데이터 전처리 단계를 보여주지 않는 저개발 샘플 Jupyter Notebook은 구매자가 제품의 가치 제안을 완전히 이해하기 어려울 수 있습니다.

알고리즘 제품의 경우 샘플 노트북은 전체 훈련, 튜닝, 모델 생성, 실시간 추론을 위한 엔드포인트 생성, 배치 변환 작업 수행을 시연합니다. 자세한 내용은 GitHub의 알고리즘 목록 및 샘플 노트북을 참조하세요. 샘플 노트북은 GitHub의 amazon_demo_product 및 automl을 참조하세요. 이러한 샘플 노트북은 파라미터를 입력하지 않아도 모든 리전에서 작동하며 구매자가 샘플 데이터를 찾을 필요가 없습니다.

> **Note**
>
> 예제 훈련 데이터가 부족하면 고객이 Jupyter Notebook을 성공적으로 실행할 수 없습니다. 샘플 노트북이 제대로 개발되지 않으면 구매자가 제품을 사용해 볼 수 없어서 선택을 망설이게 됩니다.

ML 제품 목록의 요구 사항 및 권장 사항 요약

다음 표에는 기계 학습 제품 목록 페이지의 요구 사항과 권장 사항에 대한 요약 정보가 정리되어 있습니다.

세부 정보	모델 패키지 목록	알고리즘 목록
Product descriptions		
Explain in detail what the product does for supported content types (for example, “detects X in images").	Required	Required
Provide compelling and differentiating information about the product (avoid adjectives like "best" or unsubstantiated claims).	Recommended	Recommended
List most important use case(s) for this product.	Required	Required
Describe the data (source and size) it was trained on and list any known limitations.	Required	Not applicable
Describe the core framework that the model was built on.	Recommended	Recommended
Summarize model performan ce metric on validation data (for example, "XX.YY percent accuracy benchmarked using the Z dataset").	Required	Not applicable
Summarize model latency and/or throughput metrics on recommended instance type.	Required	Not applicable
Describe the algorithm category. For example, “This decision forest regressio n algorithm is based on an	Not applicable	Required

세부 정보	모델 패키지 목록	알고리즘 목록
ensemble of tree-structured classifiers that are built using the general technique of bootstrap aggregation and a random choice of features.”		
Usage information		
For inference, provide a description of the expected input format for both the real-time endpoint and batch transform job. Include limitations, if applicable. See 입력 및 출력 요구 사항.	Required	Required
For inference, provide input samples for both the real-time endpoint and batch transform job. Samples must be hosted on GitHub. See 입력 및 출력 요구 사항.	Required	Required
For inference, provide the name and description of each input parameter. Provide details about the its limitations and specify if it is required or optional.	Recommended	Recommended

세부 정보	모델 패키지 목록	알고리즘 목록
For inference, provide details about the output data your product returns for both the real-time endpoint and batch transform job. Include any limitations, if applicable. See 입력 및 출력 요구 사항.	Required	Required
For inference, provide output samples for both the real-time endpoint and batch transform job. Samples must be hosted on GitHub. See 입력 및 출력 요구 사항.	Required	Required
For inference, provide an example of using an endpoint or batch transform job. Include a code example using the AWS Command Line Interface (AWS CLI) commands or using an AWS SDK.	Required	Required
For inference, provide the name and description of each output parameter. Specify if it is always returned.	Recommended	Recommended
For training, provide details about necessary informati on to train the model such as minimum rows of data required. See 입력 및 출력 요구 사항.	Not applicable	Required

세부 정보	모델 패키지 목록	알고리즘 목록
For training, provide input samples hosted on GitHub. See 입력 및 출력 요구 사항.	Not applicable	Required
For training, provide an example of performing training jobs. Describe the supported hyperparameters, their ranges, and their overall impact. Specify if the algorithm supports hyperparameter tuning, distributed training, or GPU instances. Include code example such as AWS CLI commands or using an AWS SDK, for example.	Not applicable	Required
Provide a Jupyter notebook hosted on GitHub demonstrating complete use of your product. See Jupyter Notebook 요구 사항.	Required	Required
Provide technical information related to the usage of the product, including user manuals and sample data.	Recommended	Recommended

기계 학습 제품의 문제 해결

이 섹션에서는 기계 학습 제품의 게시 과정에서 발생할 수 있는 몇 가지 대표적인 오류에 대한 도움말을 제공합니다. 문제가 목록에 없는 경우 AWS Marketplace 판매자 운영 팀에 문의하세요.

일반:에서 모델 패키지 또는 알고리즘의 Amazon 리소스 이름(ARN)을 추가할 때 400 오류가 발생합니다. AWS Marketplace Management Portal

일반적인 원인

SageMaker AI에서 기계 학습 제품을 생성할 때 제품을 게시하기로 선택하지 않았습니다 AWS Marketplace.

해결 방법

Amazon SageMaker AI 콘솔을 사용하여 리소스를 생성한 경우 프로세스의 마지막 페이지에서에 이 모델 패키지 게시 AWS Marketplace에 대해 예를 선택하고 이 알고리즘 게시에 AWS Marketplace 대해 예를 선택해야 합니다. 아니오를 선택하면 나중에 게시할 수 없습니다. 예를 선택해도 모델 패키지 또는 알고리즘이 게시되지는 않습니다. 하지만 모델 패키지 또는 알고리즘 리소스가 생성될 때 검증을 수행하므로 AWS Marketplace에서 사용하려면 필요합니다.

AWS SDK를 사용하여 모델 패키지를 생성하거나 알고리즘을 생성하는 경우 파라미터`CertifyForMarketplace`가 로 설정되어 있는지 확인합니다`true`.

인증 및 검증을 거친 모델 패키지 또는 알고리즘 리소스를 다시 생성한 후에는 AWS Marketplace Management Portal에서 새 ARN을 추가합니다.

일반:에서 모델 패키지 또는 알고리즘의 ARN을 추가할 때 404 오류가 발생합니다. AWS Marketplace Management Portal

일반적인 원인

여러 가지 이유로 이 오류가 발생할 수 있습니다.

- ARN이 유효하지 않을 수 있습니다.
- 모델 패키지 또는 알고리즘 리소스가 판매자 계정 AWS 계정 과 동일한에서 생성되지 않았습니다.
- 게시에 사용하는 사용자 또는 역할에 모델 패키지 또는 알고리즘 리소스에 액세스할 수 있는 올바른 IAM 권한이 없습니다.

해결 방법

1. ARN이 올바른 ARN이고 예상 형식인지 확인합니다.

모델 패키지의 경우 ARN은 `arn:aws:sagemaker:us-east-2:000123456789:model-package/my-model-package-name` 형식입니다.

알고리즘의 경우 ARN은 `arn:aws:sagemaker:us-east-2:000123456789:algorithm/my-algorithm` 형식입니다.

2. 게시에 사용할 모든 리소스와 자산이 게시 소스가 되는 판매자 계정에 있는지 확인합니다.
3. 사용자 또는 역할에 다음 권한이 있는지 확인합니다.

   모델 패키지의 경우 모델 패키지 리소스에 대한 `sagemaker:DescribeModelPackage` 작업을 허용해야 합니다.

   알고리즘의 경우 알고리즘 리소스에 대한 `sagemaker:DescribeAlgorithm` 작업을 허용해야 합니다.

Amazon SageMaker AI: 모델 패키지 또는 알고리즘 리소스를 생성할 때 “클라이언트 오류: 레지스트리에 대한 액세스 거부” 실패 메시지가 표시됨

일반적인 원인

모델 패키지 또는 알고리즘을 생성하는 데 사용되는 이미지를 다른 AWS 계정에 속한 Amazon ECR 리포지토리에 저장하면 이 오류가 발생할 수 있습니다. 모델 패키지 또는 알고리즘 검증에서 교차 계정 이미지를 지원하지 않습니다.

해결 방법

이미지를 게시하는 데 AWS 계정 사용하는가 소유한 Amazon ECR 리포지토리에 복사합니다. 그런 다음, 새 이미지 위치를 사용하여 리소스를 생성합니다.

Amazon SageMaker AI: “시작되지 않음”과 “클라이언트 오류: 스캔 예약되지 않음...”이 표시됩니다. 오류 메시지가 표시됩니다.

일반적인 원인

이 오류는 SageMaker AI가 Amazon ECR에 저장된 Docker 컨테이너 이미지 스캔을 시작하지 못할 때 발생할 수 있습니다.

해결 방법

이 오류가 발생하면 Amazon ECR 콘솔을 열고, 이미지가 업로드된 리포지토리를 찾아서 이미지를 선택한 다음, 스캔을 선택합니다.

의 SaaS 기반 제품 AWS Marketplace

서비스형 소프트웨어(SaaS) 제품을 사용하면 인프라에 AWS 호스팅된 소프트웨어를 배포하고 AWS Marketplace 구매자에게 AWS 환경의 소프트웨어에 대한 액세스 권한을 부여할 수 있습니다. 소프트웨어 내에서 고객 액세스, 계정 생성, 리소스 프로비저닝 및 계정 관리를 관리할 책임이 있습니다. 초기 SaaS 제품을 생성한 후 변경 요청을 제출하고 제품 설명, 국가별 가용성, 요금 등을 AWS Marketplace 포함하여에서 해당 속성을 구성할 수 있습니다. SaaS 구독 및 계약 통합도 구성합니다. 다음 주제에서는 시작하는 데 필요한 정보를 제공합니다.

API를 Amazon API Gateway와 통합하는 방법에 대한 자세한 내용은 Amazon API Gateway 개발자 안내서의 AWS Marketplace를 통해 API Gateway API 판매를 참조하세요.

SaaS 제품에 대한 지원이 필요한 경우 AWS Marketplace 판매자 운영 팀에 문의하세요.

주제

- 에서 SaaS 제품 시작하기 AWS Marketplace
- SaaS 제품 계획
- 에 대한 SaaS 제품 지침 AWS Marketplace
- 의 SaaS 제품 요금 AWS Marketplace
- 에서 SaaS 무료 평가판 제안 생성 AWS Marketplace
- 를 통해 SaaS 제품에 고객 온보딩 AWS Marketplace
- SaaS 제품에 대한 Amazon SNS 알림
- AWS Marketplace 측정 및 권한 부여 서비스 API 액세스
- 에서 SaaS 제품 보고 AWS Marketplace
- SaaS 제품 통합 코드 예제
- 를 사용하여 Amazon VPC를 통해 제품 제공 AWS PrivateLink

에서 SaaS 제품 시작하기 AWS Marketplace

AWS Marketplace 판매자는 서비스형 소프트웨어(SaaS) 제품을에 추가할 수 있습니다 AWS Marketplace. 여기에는 SaaS 제품의 결제 모델에 따라 SaaS 제품을 적절한 AWS Marketplace API 작업과 통합하는 것이 포함됩니다. 이 주제에서는 시작하는 데 필요한 필수 사전 조건부터 시작하여 SaaS 제품을 만들고 구성하는 프로세스에 대한 개요를 제공합니다.

사전 조건

시작하기 전에 다음 사전 조건을 완료해야 합니다.

1. AWS Marketplace Management Portal에 액세스하고 사용합니다. 판매자로 등록하고 판매하는 제품을 관리하는 데 사용하는 도구입니다 AWS Marketplace. 자세한 내용은 AWS Marketplace Management Portal 단원을 참조하십시오.
2. 판매자로 등록하고 세금 및 은행 정보를 제출합니다. 자세한 내용은 AWS Marketplace 판매자로 등록 단원을 참조하십시오.
3. AWS Marketplace에서 SaaS 제품을 생성하고 통합하는 방법을 계획합니다. 자세한 내용은 SaaS 제품 계획 단원을 참조하십시오.

다음 단계

필수 사전 조건을 완료한 후 SaaS 제품을 만들고 구성할 수 있습니다.

주제

- 의 SaaS 제품 수명 주기 AWS Marketplace
- 에서 SaaS 제품 생성 AWS Marketplace
- 에서 초기 SaaS 제품 페이지 생성 AWS Marketplace
- 에서 SaaS 제품 설정 구성 AWS Marketplace
- SaaS 구독 제품을와 통합 AWS Marketplace
- SaaS 계약 제품을와 통합 AWS Marketplace
- SaaS 계약과 pay-as-you-go제 제품 통합 AWS Marketplace
- 서버리스 SaaS 통합 배포

의 SaaS 제품 수명 주기 AWS Marketplace

에서 서비스형 소프트웨어(SaaS) 제품을 생성하면 처음에 AWS Marketplace는 제한된 가시성으로 게시되므로 계정만 액세스할 수 있습니다. 준비가 되면 AWS Marketplace 카탈로그에 게시하여 구매자가 제품을 구독하고 구매할 수 있도록 할 수 있습니다. 다음 주제에서는 SaaS 제품 수명 주기에 대한 정보를 제공합니다. SaaS 제품 만들기에 대한 자세한 내용은 에서 SaaS 제품 생성 AWS Marketplace 섹션을 참조하세요.

SaaS 제품 페이지에서 제품 목록을 볼 수 있습니다. 제품 수명 주기에서 제품의 현재 단계에 따라 제품의 상태는 다음 중 하나입니다.

- 준비 중 - 아직 판매자가 정보를 추가하고 있는 불완전한 제품. 셀프 서비스 경험에서 처음으로 저장 후 종료할 때, 변경 요청이 성공하면 판매자가 제출한 완료된 단계의 정보가 포함된, 게시되지 않은 제품이 생성됩니다. 이 상태에서 제품에 정보를 계속 추가할 수도 있고 변경 요청을 통해 이미 제출한 세부 정보를 변경할 수도 있습니다.
- 제한 - 제품이 시스템에 제출되고 시스템의 모든 검증을 통과하면 완성되었습니다. 그 후 제품이 제한 상태로 릴리스되었습니다. 이제 판매자 계정과 판매자가 허용한 사람만 제품 세부 정보 페이지에 액세스할 수 있습니다. 세부 정보 페이지를 통해 제품을 테스트할 수 있습니다. 자세한 내용이나 도움이 필요하면 AWS Marketplace 판매자 운영 팀에 문의하세요.
- 퍼블릭 - 구매자가 제품을 살펴보고 구독할 수 있도록 제품을 게시할 준비가 되면 표시 여부 업데이트 변경 요청을 사용합니다. 이 요청은 AWS Marketplace 판매자 운영 팀이 AWS 정책을 기준으로 제품을 검토하고 감사할 수 있는 워크플로를 시작합니다. 제품이 승인되고 변경 요청이 처리되면 제품 상태가 제한에서 퍼블릭으로 바뀝니다. AWS 지침에 대한 자세한 내용은 SaaS 제품 지침을 참조하세요.
- 제한 - 신규 사용자가 제품을 구독하지 못하게 하려면 표시 여부 업데이트 변경 요청을 사용하여 제품을 제한하면 됩니다. 제한 상태는 기존 사용자는 제품을 계속 사용할 수 있다는 것을 의미합니다. 하지만 제품이 더 이상 대중에게 표시되거나 신규 사용자에게 제공되지 않습니다.

준비 중, 제한 및 퍼블릭 상태의 제품을 업데이트할 수 있습니다. 자세한 내용은 제품 정보 업데이트를 참조하세요.

에서 SaaS 제품 생성 AWS Marketplace

AWS Marketplace 판매자는 서비스형 소프트웨어(SaaS) 제품을에 추가할 수 있습니다 AWS Marketplace. 여기에는 SaaS 제품을 생성하고 결제 모델에 따라 적절한 AWS Marketplace API 작업과 통합하는 작업이 포함됩니다. 에서 SaaS 제품으로 소프트웨어를 판매하려면 다음과 같은 광범위한 단계를 AWS Marketplace따릅니다.

- 에서 SaaS 제품을 생성합니다 AWS Marketplace.
- SaaS 구독, 계약 또는 사용한 만큼만 지불 계약 제품을 AWS Marketplace와 통합합니다.
- 구독, 계약 또는 사용한 만큼만 지불 계약 제품과 AWS Marketplace의 통합을 테스트합니다.
- 출시할 제품을 제출합니다.

다음 절차에서는에서 SaaS 제품을 생성하는 방법을 보여줍니다 AWS Marketplace.

SaaS 제품 생성

SaaS 제품을 만들려면

1. SaaS 제품을 나열하기로 결정

   판매하려는 SaaS 제품이 있어야 합니다 AWS Marketplace. SaaS 제품 계획 방법을 검토하고 이해합니다.

2. 요금 및 제안 유형 결정

   SaaS 제품은 구독, 계약, 사용한 만큼만 지불 계약의 세 가지 제안 유형을 제공합니다. 제안 유형 선택은 SaaS 제품을 통합하는 방식에 영향을 미칩니다 AWS Marketplace. 자세한 내용은 가격 책정 계획 단원을 참조하십시오.

3. 자산 수집

   제품을 제출하는 데 필요한 자산을 수집합니다. 자산에는 다음이 포함됩니다.

   - 제품 로고 URL - 제공하는 제품의 명확한 로고 이미지가 포함된, 공개적으로 액세스 가능한 Amazon S3 URL입니다.
   - 최종 사용자 라이선스 계약(EULA) URL - PDF 파일로 제공되는 EULA가 제품에 포함되어야 합니다. 고객이 제품의 AWS Marketplace 페이지에서 EULA를 검토할 수 있도록 Amazon S3 버킷 링크를 제공해야 합니다.
   - 제품 등록 URL - 구매자가 AWS Marketplace에서 성공적으로 제품을 구독한 후에 리디렉션되는 URL입니다.
   - 제품에 대한 메타데이터 - AWS Marketplace Management Portal의 제품 생성 마법사에서 메타데이터를 입력합니다.
   - 제품 지원 정보 - 이 정보에는 제품 지원 채널의 이메일 주소와 URL이 포함됩니다.

4. 통합할 제품 제출

   판매자 계정과 AWS Marketplace 관리 포털을 사용하여에 제품을 한정된 제품으로 AWS Marketplace 게시합니다에서 초기 SaaS 제품 페이지 생성 AWS Marketplace. 즉, 통합 및 테스트에만 사용할 수 있습니다. 제품 코드 및 Amazon Simple Notification Service(SNS) 주제는 제품 개요 페이지에서 사용할 수 있습니다.

> **Note**
>
> 사용자와 AWS Marketplace 판매자 운영 팀이 큰 비용을 발생시키지 않고 제품을 테스트할 수 있도록 제품은 할인된 가격으로 유지되어야 합니다. 판매자가 제품 공개 요청을 하면 저희 쪽에서 제품의 실제 가격을 문의합니다.

5. 와 통합 AWS Marketplace

   판매자의 제품은 구독을 검증한 후 액세스 권한을 부여하고 경우에 따라 사용량을 측정하는 등 고객이 제품을 온보딩하고 사용할 수 있도록 지원해야 합니다. 와 통합하는 방법은 제품에 사용 중인 제안 유형에 AWS Marketplace 따라 다릅니다. 제안 유형에 따른 통합에 대한 자세한 내용은 다음 주제를 참조하세요.

   - 구독 통합
   - 계약
   - 사용한 만큼만 지불 계약 통합

   와 제품을 통합하는 마지막 단계는 제품을 테스트하여 통합 AWS Marketplace 이 제대로 작동하는지 확인하는 것입니다.

6. 제품 테스트 및 계약 취소

   통합 프로세스를 완료한 후에는 자체 제품을 구독하여 고객 경험을 평가하고 확인하는 것이 좋습니다. 테스트 구독을 생성하면 다음을 수행할 수 있습니다.

   - 제품 정보 검토
   - 사용 가능한 구매 옵션 검사
   - 테스트 결제 처리
   - 자체 제품 웹 사이트에 대한 링크 확인

   > **Important**
   >
   > 제품의 가시성을 제한에서 퍼블릭으로 변경하기 전에 테스트 구독을 취소합니다. 도움이 필요한 경우 지원 티켓을 제출하여 Marketplace 판매자 운영 팀에 문의하세요.

7. 출시할 제품 제출

통합을 확인하고 제품을 출시할 준비가 되었으면 표시 여부 업데이트를 선택합니다. AWS Marketplace 판매자 운영 팀은 공개로 가시성을 업데이트하기 전에 제품을 검토하고 가격을 업데이트합니다.

> **Note**
>
> AWS Marketplace 판매자 작업은 수동 프로세스를 사용하여 SaaS 제품을 확인하고 업데이트합니다. 이 프로세스는 공개 가시성을 업데이트하는 데 업무일 기준 7~10일이 소요되며, 팀이 오류를 발견하는 경우 더 오래 걸립니다. 타이밍에 대한 자세한 내용은 이 가이드의 타이밍 및 기대치를 참조하세요.

에서 초기 SaaS 제품 페이지 생성 AWS Marketplace

서비스형 소프트웨어(SaaS) 애플리케이션 메타데이터를 사용하여를 사용하여 AWS Marketplace 카탈로그에 초기 SaaS 제품 페이지를 생성할 수 있습니다 AWS Marketplace Management Portal. 제품 정보, 제품 배포 세부 정보 및 공개 제안 세부 정보를 추가할 수도 있습니다. 선택 사항으로, 허용 목록에 계정을 추가하여 제품을 테스트할 수 있습니다. 자세한 내용은 다음 절차를 참조하세요.

초기 SaaS 제품 페이지를 생성하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. SaaS 제품 생성을 선택하고 SaaS 제품을 선택합니다.
3. SaaS 제품 ID 및 코드를 생성합니다. 태그 기반 권한 부여를 지원하는 선택적 태그를 추가할 수도 있습니다.

   > **Note**
   >
   > 태그 기반 권한 부여에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 태그를 사용하여 AWS 리소스에 대한 액세스 제어를 참조하세요.

4. 셀프 서비스 환경을 사용하여 AWS Marketplace 목록을 생성합니다. 제품 정보, 제품 배포 세부 정보 및 공개 제안 세부 정보를 추가합니다. 선택 사항으로, 허용 목록에 계정을 추가하여 제품을 테스트할 수도 있습니다.

> **Note**
>
> 단계를 완료하기 전에 세션을 종료해야 하는 경우 저장 후 종료 옵션을 선택하여 현재 선택 내용을 스테이징 영역에 저장합니다. 이 옵션을 선택하면 입력한 정보를 검증하라는 요청이 생성됩니다. 요청이 검증되는 동안에는 제품을 편집할 수 없습니다. 요청이 성공하면 제품 생성 재개를 선택하여 제품을 계속 생성할 수 있습니다.
> 요청이 성공하지 못하면 검증 오류 때문이며, 검증 오류는 제품 요청 로그에 표시됩니다. 오류를 보려는 요청을 선택하고, 작업에서 새로 복사를 선택하여 오류를 수정하고 요청을 다시 제출합니다. 이전 단계를 업데이트하려면 제품 세부 정보 페이지를 열고 변경 요청을 제출합니다.

> **Note**
>
> 테스트 중에 기본 가격은 차원당 0.001 USD입니다. 이처럼 요금이 저렴하므로 큰 비용을 들이지 않고도 제한 상태에서 제품을 테스트할 수 있습니다. 제품을 공개할 때 실제 요금을 입력해야 합니다.

5. 제출을 선택합니다. 그런 다음, 정보를 AWS Marketplace 검증합니다. 검증에 성공하면는 제품을 제한됨 상태로 AWS Marketplace 릴리스합니다. 검증이 성공하면 판매자는 제품을 미리 보고, 통합하고, 테스트할 수 있습니다.

> **Note**
>
> 검증이 진행 중인 동안에는 제품을 편집할 수 없습니다. 제품이 처음 게시되면 제품 생성에 AWS 계정 사용된 및 AWS Marketplace 판매자 운영 팀의 테스트 계정에만 액세스할 수 있습니다. SaaS 제품 페이지에서 제품을 보는 경우 에서 보기를 AWS Marketplace 선택하여 구매자를 AWS Marketplace 위해에 표시되는 제품 세부 정보를 볼 수 있습니다. 이 세부 정보 목록은 다른 AWS Marketplace 사용자에게 보이지 않습니다.

에서 SaaS 제품 설정 구성 AWS Marketplace

에서 서비스형 소프트웨어(SaaS) 제품을 생성한 후 많은 제품 설정을 수정할 AWS Marketplace수 있습니다. 다음 섹션에서는 변경 요청을 제출하고 요금 세부 정보, 제품 가시성 및 기타 설정 업데이트와 같은 제품 설정을 수정하는 방법을 보여줍니다.

주제

- 변경 요청 관리
- 제품 정보 업데이트
- 아키텍처 세부 정보 업데이트
- ID 허용 목록 업데이트 AWS 계정 IDs
- 제품 표시 여부 업데이트
- 요금 조건 업데이트
- 요금 차원 추가
- 요금 차원 업데이트
- 요금 차원 제한
- 구매자가 제품에 액세스하는 방법 결정
- 빠른 시작 구성
- 국가별 가용성 업데이트
- 제품의 환불 정책 업데이트
- 최종 사용자 라이선스 계약(EULA) 업데이트

변경 요청 관리

셀프 서비스 리스팅에서 변경 요청을 사용하여 제품을 변경합니다. 현재 요청은 요청 탭의 AWS Marketplace Management Portal 에서 찾을 수 있습니다. 탐색 표시줄 아래에 있는 변경 요청 드롭다운 목록을 통해 새 요청을 수행할 수 있습니다.

SaaS 제품에 대한 변경 요청을 생성하는 방법

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. 제품 탭의 드롭다운 목록에서 SaaS를 선택합니다.
3. 요청이 제출되면 처리가 시작됩니다. 변경 요청은 검토 중, 변경 준비 중, 변경 사항 적용 중 상태를 거치게 됩니다.
4. 요청 처리가 완료되면 요청 상태가 다음 값 중 하나로 바뀝니다.
    - 성공 - 이 상태는 요청한 변경이 처리되었으며 변경 사항이 시스템에 반영되었다는 뜻입니다.
    - 실패 - 이 상태는 요청에 문제가 발생하여 변경 사항이 처리되지 않았다는 뜻입니다. 상태가 실패이면 요청을 선택하여 문제 수정 방법에 대한 권장 사항을 제공하는 오류 코드를 찾을 수 있습니다. 오류를 해결하고 변경 요청을 새로 생성할 수 있습니다. 프로세스를 더 빠르게 진행하려면

실패 요청의 세부 정보를 복사하는 새 요청에 복사 기능을 사용하면 됩니다. 필요한 대로 변경한 후 요청을 다시 제출할 수 있습니다.

업데이트로 시작하는 변경 요청은 현재 프로젝트 세부 정보를 로드합니다. 그러면 기존 세부 정보를 덮어쓰는 업데이트를 수행할 수 있습니다. 요청 페어 추가 및 제한은 특히 각 요청이 성공한 후(셀프 서비스 경험에서 저장 후 종료 및 제출 작업을 선택한 후) 프로비저닝되는 업데이트를 위한 것입니다. 즉, 기존 구독자는 구독 또는 계약이 만료될 때까지 제품을 계속 사용할 수 있습니다. 하지만 제한 상태인 제품에 신규 구독자를 추가할 수 없습니다.

제품 정보 업데이트

제품을 생성한 후 AWS Marketplace에서 제품과 관련된 정보를 변경해야 할 때가 있습니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 페이지로 이동한 후 SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 제품 정보 업데이트를 선택합니다.
4. 다음 중 변경하려는 필드를 업데이트합니다.
   - 제품 제목
   - SKU
   - 간단한 설명
   - 자세한 설명
   - 제품 로고 URL
   - 하이라이트
   - 제품 범주
   - 키워드
   - 제품 비디오 URL
   - 리소스
   - 지원 정보

   > **Note**
   > 로고 형식에 대한 자세한 내용은 회사 및 제품 로고 요구 사항을 참조하세요.

5. 제품 정보를 업데이트하려면 제출을 선택합니다.

6. 요청이 요청 탭에 표시되고 상태가 검토 중인지 확인합니다. 새 요청이 보이지 않으면 페이지를 새로 고쳐야 할 수도 있습니다.

아키텍처 세부 정보 업데이트

제품이 배포된 특별 지정을 받으려면 호스팅 패턴을 선택하고 아키텍처 다이어그램을 업로드하여 AWS Marketplace 관리 포털(AMMP)에서 제품의 아키텍처 세부 정보를 AWS업데이트합니다. 에서 배포를 AWS 고려하는 호스팅 패턴 목록은 섹션을 AWS참조하세요지침.

아키텍처 세부 정보를 업데이트하려면

1. AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 SaaS를 선택합니다.
3. SaaS 제품 테이블에서 제품을 선택합니다.
4. 변경 요청에서 아키텍처 세부 정보 업데이트를 선택합니다.
5. 호스팅 패턴을 선택합니다. 에서 배포를 AWS Marketplace 고려하는 호스팅 패턴을 선택하는 경우 다음 두 단계를 추가로 AWS완료합니다. 배포된 것으로 간주되는 패턴에 대한 자세한 내용은이 가이드의 지침앞부분에서 단원을 AWS참조하십시오.

   a. 메시지가 표시되면 아키텍처 다이어그램 섹션에서 파일 선택을 선택하여 아키텍처 다이어그램을 PNG 또는 JPG 형식으로 업로드합니다. 다이어그램에 대한 자세한 내용은 단원을 참조하십시오아키텍처 다이어그램 생성.

   b. 메시지가 표시되면 애플리케이션 영역에서 애플리케이션이 실행되는 위치를 선택합니다. 자세한 내용은 컨트롤 플레인과 애플리케이션 플레인을 참조하세요.
6. 아키텍처 세부 정보 업데이트를 선택합니다.

요청이 완료되면 요청 상태가 성공으로 변경됩니다. 요청 상태를 확인하려면 SaaS 제품 테이블에서 제품을 선택하고 요청 로그 탭을 선택합니다.

평가 결과를 보려면 SaaS 제품 테이블에서 제품을 선택하고 아키텍처 세부 정보 탭을 선택합니다.

ID 허용 목록 업데이트 AWS 계정 IDs

제한된 상태에서 제품을 볼 수 있는 AWS 계정 IDs 목록을 변경할 수 있습니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 페이지로 이동한 후 SaaS 제품 탭에서 수정하려는 제품을 선택합니다.

3. 변경 요청 드롭다운 목록에서 허용 목록 업데이트를 선택합니다. 목록에는 현재 허용 목록에 있는 AWS 계정 IDs가 표시됩니다.
4. 허용 목록에 있는 AWS 계정 필드에 AWS 계정 ID를 입력하고, 쉼표로 ID를 구분합니다.
5. ID의 AWS 계정 IDs 선택합니다.

제품 표시 여부 업데이트

빠른 시작 경험을 볼 수 있는 구매자를 변경하려면 업데이트 가시성을 사용할 AWS Marketplace수 있습니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 페이지에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 표시 여부 업데이트를 선택합니다.

> **Note**
>
> 이 변경 요청을 사용하여 제품을 제한 상태에서 퍼블릭 상태로 전환하라고 요청할 수 있습니다. 그러나 변경 요청은 AWS Marketplace 판매자 운영 팀 승인 프로세스를 거쳐 퍼블릭으로 이동해야 합니다.

4. 대중에게 게시할 때에는 제품의 실제 가격을 입력합니다. 이 가격은 리스팅을 대중에게 공개하도록 승인된 후에 적용됩니다.
5. 검토 요청을 제출하려면 제출을 선택합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

요금 조건 업데이트

SaaS 제품의 차원당 요금을 변경하려면 요금 조건 업데이트를 사용합니다.

> **Note**
>
> 모든 차원의 요금이 인상되면 요금 업데이트 옵션을 최소 향후 90일 동안 사용할 수 없게 됩니다. 요금 인하와 인상을 모두 업데이트하는 경우 요금 인하부터 업데이트하세요.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal 로그인합니다.

2. SaaS 제품 페이지로 이동한 후 SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 공개 제안 업데이트를 선택하고 요금 조건 업데이트를 선택합니다.
4. 필드에 현재 요금이 미리 입력되어 있습니다. 현재 요금을 삭제하고 새 요금을 추가하면 됩니다.
5. 검토 요청을 제출하려면 제출을 선택합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

요금 차원 추가

제품 요금 청구에 사용할 차원을 추가할 수 있습니다. 차원은 구매자에게 제품 사용 요금을 부과하는 기본 측정 단위입니다.

> **Note**
> 기존 요금 차원의 이름 또는 설명을 업데이트하려면 the section called “요금 차원 업데이트” 섹션을 참조하세요.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 요금 차원 업데이트를 선택하고 요금 차원 추가를 선택합니다.
4. 차원 API 식별자, 표시 이름 및 설명을 입력하여 제품에 새 차원을 추가하고 다음을 선택합니다.

   > **Note**
   > API 식별자와 이름은 모든 차원에서 고유해야 합니다. 차원이 생성된 후에는 API 식별자 및 단위를 변경할 수 없습니다.

5. 추가한 각 차원의 요금을 정의한 후 다음을 선택하여 변경 사항을 검토합니다.

   > **Note**
   > 제품에 대해 선택한 요금 모델(예: 계약, 사용량 또는 소비량 계약)의 차원만 추가할 수 있습니다. 제한된 제품의 경우 새로 추가된 차원의 요금이 0.01 USD로 설정됩니다. 제품을 대중에게 공개할 준비가 되면 요금을 업데이트하면 됩니다.

6. 제출을 선택하여 검토 요청을 제출합니다.
7. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 변경됩니다.

요금 차원 업데이트

제품 요금 청구에 사용할 차원을 업데이트할 수 있습니다. 차원은 구매자에게 제품 사용 요금을 부과하는 기본 측정 단위입니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 요금 차원 업데이트를 선택하고 차원 정보 업데이트를 선택합니다.
4. 업데이트할 차원을 찾은 다음, 이름 또는 설명을 선택합니다.
5. 새 이름이나 설명을 입력한 다음, 체크 표시를 선택하여 업데이트를 확인합니다. 차원 이름은 고유해야 합니다.
6. 제출을 선택하여 검토 요청을 제출합니다.
7. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

요금 차원 제한

현재 제품에 나열된 차원을 제한할 수 있습니다. 이 요청은 선택한 차원을 제품에서 제거합니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 요금 차원 업데이트를 선택하고 요금 차원 제안을 선택합니다.
4. 제한된 제품 및 공개 제품의 경우 AWS에 문의 버튼을 사용하여 AWS Marketplace 판매자 운영 팀에 문의하라는 메시지가 표시됩니다. 양식을 사용하여 제품 목록에서 제거할 차원의 세부 정보를 입력합니다.

> **Note**
> 작업이 항상 요금 차원을 제한할 수 있는 것은 아닙니다.

구매자가 제품에 액세스하는 방법 결정

고객이 제품에 액세스하는 방법에 대해 다음 옵션 중 하나를 선택할 수 있습니다.

- the section called “SaaS URL 이행 옵션 업데이트” - 고객은 AWS Marketplace에서 제품에 가입한 후 리디렉션되는 사이트의 URL을 사용합니다.
- the section called “빠른 시작 구성” - 고객은 간소화된 프로세스를 사용하여 제품을 구성하고 시작합니다. 가시성이 제한 또는 공개인 상태로 기존 제품에 대해 이 구성을 완료할 수 있습니다.

SaaS URL 이행 옵션 업데이트

이행 옵션 업데이트 탭을 사용하여 SaaS 제품을 이행하는 데 사용되는 URL을 업데이트할 수 있습니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 페이지의 SaaS 제품 탭에서 수정할 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 이행 옵션 업데이트를 선택한 다음 기본 이행 URL 편집을 선택합니다.
4. 이행 URL 필드에 SaaS 제품 이행 옵션의 새 URL을 입력합니다.
5. 검토 요청을 제출하려면 제출을 선택합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

빠른 시작 구성

에 나열된 SaaS 제품은 구독 구매자의 계정에 AWS 리소스를 배포해야 하는 경우가 AWS Marketplace 많습니다(예: IAM 역할). 빠른 시작을 사용하면 AWS CloudFormation 템플릿을 사용하여 구매자에게 step-by-step 안내 지침 및 리소스 배포를 제공할 수 있습니다. 구매자는 CloudFormation 템플릿을 사용하여 제품을 구성하고 시작합니다. 빠른 시작 구성 프로세스에 대한 자세한 내용은 SaaS 빠른 시작 활성화 랩을 참조하세요.

고객이 SaaS 제품을 시작하는 데 사용할 수 있는 빠른 시작 환경을 구성하려면 이행 옵션 탭을 사용하세요.

1. SaaS 제품 페이지로 이동한 후 SaaS 제품 탭에서 수정하려는 제품을 선택합니다.

> **Note**
> 빠른 시작 환경을 구성하려면 제품 가시성이 제한 또는 공개 상태여야 합니다.

2. 제품 세부 정보 페이지에서 이행 옵션 탭을 선택합니다.
3. 빠른 시작의 경우 활성화 및 구성 버튼을 선택합니다.
4. 계정 로그인 세부 정보에는 구매자가 로그인하거나 계정을 생성할 수 있는 사이트 URL을 제공합니다. 이 URL은 구매자 경험의 새 탭을 엽니다. 그런 다음 구매자는 로그인하거나 계정을 생성하고 로 돌아가 템플릿을 AWS Marketplace 시작합니다.
5. AWS CloudFormation 템플릿을 생성합니다.

> **Tip**
> 구매자 AWS 의 계정에 리소스를 배포하는 CloudFormation 템플릿을 생성할 AWS 때 AWS Well-Architected 프레임워크를 따릅니다. 자세한 내용과 리소스는 AWS Well-Architected를 참조하고 AWS Well-Architected 프레임워크를 참조하십시오.

AWS CloudFormation 템플릿의 경우 AWS CloudFormation 템플릿 추가 버튼을 선택하고 다음 정보를 제공합니다.

- 제목 - CloudFormation 배포의 이름을 입력합니다.
- 설명 - 템플릿에 대한 설명을 제공합니다.
- 스택 이름 - 스택의 이름을 입력합니다. 이 이름은 CloudFormation의 구매자를 위한 스택 이름입니다.
- CloudFormation 템플릿 URL - Amazon Simple Storage Service에서이 템플릿을 검토할 것임을 제공합니다.

> **Note**
> 고객의 시작 프로세스를 단순화하려면 구성 프로세스와 관련된 템플릿 수를 최소화하는 것이 좋습니다. 제품을 사용하는 데 필요한 리소스를 배포하는 템플릿 하나를 사용하는 것이 가장 좋습니다. CloudFormation 템플릿과 관련된 질문은 AWS Marketplace 비즈니스 개발 파트너 또는 AWS Marketplace 판매자 운영 팀에 문의하세요.

- 필수 IAM 권한 - CloudFormation 템플릿을 배포하는 데 필요한 권한을 제공합니다. 구매자를 위해 AWS Secrets Manager에 보안 정보로 저장되는 배포 파라미터를 공유하려는 경우 정책에 다음 작업이 포함되어야 합니다.
  - `secretsManager:ListSecrets`
  - `secretsManager:DescribeSecret`
  - `secretsManager:ReplicateSecretToRegions`
  - `secretsManager:GetSecretValue`

> **Note**
>
> 제품에 판매자가 제공한 CloudFormation 배포 파라미터(예: API 키 및 외부 ID)가 필요한 경우 `PutDeploymentParameter` 작업을 사용하여 파라미터를 고객과 공유합니다. 자세한 내용은 AWS Marketplace 배포 서비스 API 참조의 PutDeploymentParameter를 참조하세요.

6. (선택 사항) 수동 구성 지침의 경우 제품을 수동으로 구성하려는 구매자를 위한 지침을 제공합니다. 제품의 온보딩 가이드 및 설명서로 연결되는 링크를 포함하는 것을 고려해 보세요.
7. 시작 세부 정보를 보려면 구매자가 CloudFormation 스택이 배포된 후 제품에 액세스할 수 있는 URL을 제공합니다.
8. (선택 사항) 빠른 시작을 위해 허용 목록에 등록된 계정의 경우 제한된 가시성으로 빠른 시작 환경을 볼 수 AWS 계정 있는 쉼표로 구분된 목록을 제공합니다.
9. 제출 버튼을 선택합니다. 빠른 시작 환경은 가시성이 제한됩니다. 즉, 사용자 계정과 허용된 계정에서만 볼 수 있습니다. 가시성이 제한된 상태로 제품을 구독하고 계정 설정 버튼을 선택한 후 구성 및 시작 페이지를 사용하여 구성을 테스트할 수 있습니다.
   - 구매자의 관점에서 페이지를 미리 보려면 미리 보기를 선택합니다. 구성 변경 사항은 두 페이지가 모두 열려 있는 동안 미리 보기에 반영됩니다.
10. 준비가 되면 AWS Marketplace 카탈로그에 빠른 시작 환경을 게시할 수 있습니다. 제품 세부 정보 페이지의 이행 옵션 탭에 있는 빠른 시작 가시성 업데이트 버튼을 사용하세요.

    공개로 가시성을 변경하면 AWS Marketplace 판매자 운영 팀이 구성을 검토하고 구매자 테스트를 수행하며 경험을 게시합니다.

> **Note**
> 빠른 시작 환경을 활성화할 때 지원이 필요한 경우 AWS Marketplace 판매자 운영 팀에 문의하세요.

국가별 가용성 업데이트

제품을 제공할 수 있는 국가를 정의할 수 있습니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 공개 제안 업데이트를 선택하고 국가별 가용성 업데이트를 선택합니다.
4. 다음 옵션 중 하나를 선택하세요.
   - 모든 국가 - 지원되는 모든 국가에서 사용할 수 있습니다.
   - 일부 국가를 제외한 모든 국가 - 일부 국가를 제외하고 지원되는 모든 국가에서 사용할 수 있습니다.
   - 허용 목록에 있는 국가만 - 특정 목록의 국가에서만 제품을 사용할 수 있습니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

제품의 환불 정책 업데이트

환불 정책 업데이트를 사용하여 제품의 환불 정책을 업데이트할 수 있습니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 페이지로 이동한 후 SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운 목록에서 공개 제안 업데이트를 선택하고 환불 정책 업데이트를 선택합니다.
4. 현재 환불 정책 세부 정보가 텍스트 상자에 입력됩니다. 세부 정보를 검토하고 원하는 대로 수정합니다. 요청을 제출하면 현재 환불 정책을 덮어씁니다.
5. 검토 요청을 제출하려면 제출을 선택합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공 또는 실패로 업데이트됩니다.

최종 사용자 라이선스 계약(EULA) 업데이트

제품을 구독하는 신규 사용자의 EULA를 업데이트할 수 있습니다.

1. 를 열고 판매자 계정에 AWS Marketplace Management Portal로그인합니다.
2. SaaS 제품 탭에서 수정하려는 제품을 선택합니다.
3. 변경 요청 드롭다운에서 공개 제안 업데이트를 선택하고 EULA 업데이트를 선택합니다.
4. 표준 계약 AWS Marketplace (SCMP)을 선택하거나 사용자 지정 EULA를 제출할 수 있습니다. 사용자 지정 EULA의 경우 계약의 Amazon Simple Storage Service (Amazon S3) URL을 제공해야 합니다. Amazon S3 버킷은 공개적으로 액세스할 수 있어야 합니다.
5. 제출을 선택하여 검토 요청을 제출합니다.
6. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

SaaS 구독 제품을와 통합 AWS Marketplace

제품을와 통합하는 것은의 한 단계 AWS Marketplace 입니다에서 SaaS 제품 생성 AWS Marketplace. 서비스형 소프트웨어(SaaS) 구독 제품을와 통합하려면 코드를 작성하고 여러 고객 시나리오에 성공적으로 대응할 수 있음을 입증해야 AWS Marketplace합니다. 다음 섹션에서는 SaaS 구독 모델을와 통합하는 방법을 보여줍니다 AWS Marketplace.

> **Note**
>
> 시작하기 전에, AWS Marketplace에서 서비스형 소프트웨어(SaaS) 제품에 적합한 요금 모델을 선택했는지 확인합니다. 자세한 내용은 SaaS 제품 계획 단원을 참조하십시오.

주제

- 시나리오: 서비스가 신규 고객 검증
- 시나리오: 사용량 측정
- 시나리오: 사용자 구독 변경 사항 모니터링
- 시나리오: 고객 구독 확인
- SaaS 구독 제품 통합 테스트

시나리오: 서비스가 신규 고객 검증

고객이 제품을 구독하면 고객은 판매자의 등록 URL로 리디렉션되며, 등록 URL은 임시 `x-amzn-marketplace-token` 토큰이 포함된 HTTP POST 요청입니다. 다음과 같은 방법으로 이 요청에 응답합니다.

1. AWS Marketplace Metering Service에서 `ResolveCustomer` API 작업을 호출하여 토큰을 `CustomerIdentifier`, `CustomerAWSAccountId` 및 `ProductCode`로 교환합니다.
2. 향후 호출에 대비하여 시스템에서 `CustomerIdentifier`, `CustomerAWSAccountID` 및 `ProductCode`를 유지합니다. 고객의 구독이 유효한지 여부와 고객에 대해 필요한 정보를 저장해야 합니다.
3. 요청에 대한 응답으로 사용자의 첫 사용 경험(서비스에 해당하는 경우)을 보여주어야 합니다.

시나리오: 사용량 측정

고객이 서비스를 사용하기 시작하면 매시간 측정 기록을 보내야 합니다. 자세한 측정 방법은 SaaS 구독에서 사용할 측정 구성 섹션을 참조하세요.

AWS CloudTrail 를 사용하여 활동을 모니터링하여 결제 정보가 로 전송되는지 확인하는 것이 좋습니다 AWS. 측정 레코드를 전송할 때 다음 사항에 유의하세요.

- 측정 요청은 정시에 중복 제거됩니다.
- 매 시간마다 전송된 레코드는 누적됩니다.
- 지난 시간 동안 레코드가 없더라도 1시간마다 사용량으로 0으로 하여 측정 레코드를 보내는 것이 좋습니다.

시나리오: 사용자 구독 변경 사항 모니터링

Amazon Simple Queue Service(Amazon SQS) 대기열을 설정하고 제품의 Amazon SNS 주제를 구독합니다. SNS 주제 정보는 제품을 생성할 때 AWS Marketplace 판매자 운영 팀에서 받은 이메일 메시지에 포함되어 있습니다. 자세한 내용은 에서 SaaS 제품 생성 AWS Marketplace 단원을 참조하십시오. SNS 주제를 구독하면 특정 고객의 액세스 권한 제공 또는 취소를 포함하여 고객 구독 변경 사항에 대한 알림을 받게 됩니다.

> **Note**
>
> Amazon SNS 주제 Amazon 리소스 이름(ARN)은 `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>` 형식입니다.

응답해야 하는 알림은 다음과 같습니다.

- `subscribe-success` - 고객이 구독하고 있으며, 고객 ID를 대상으로 성공적으로 측정할 수 있습니다.
- `unsubscribe-pending` - 고객이 구독을 해지하는 중입니다. 마지막 측정 기록을 모두 보내야 합니다.
- `unsubscribe-success` - 고객이 구독을 해지했습니다. 고객의 측정 기록은 더 이상 수락되지 않습니다. 방침에 따라 고객 리소스를 종료하고 보존 정책을 준수합니다.
- `subscribe-fail` - 고객 구독이 실패했습니다. 고객 ID를 기준으로 측정하거나 고객을 대신하여 리소스를 생성하면 안 됩니다.

시나리오: 고객 구독 확인

고객을 대신하여 리소스를 생성하기 전에, 고객이 제품에 액세스할 수 있는지 확인합니다. Amazon SQS를 통해 받은 알림에서 고객의 최신 상태를 저장하여 고객에게 액세스 권한이 있는지 확인할 수 있습니다.

SaaS 구독 제품 통합 테스트

SaaS 구독 제품을와 통합한 후에는 심층 테스트를 수행하여 통합이 성공했는지 확인해야 AWS Marketplace합니다. 다음 절차에서는 제품 통합을 확인하는 단계를 간략하게 설명합니다.

> **Note**
>
> 본인의 계정을 사용하여 본인의 제품을 구독하고 통합이 성공적인지 테스트하세요. 가격을 일시적으로 인하하면 계정에 많은 요금이 부과되지 않고 구매 흐름을 테스트할 수 있습니다. 가격을 일시적으로 인하하거나 추가 테스트 계정에서 제품에 액세스할 수 있도록 허용하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 운영 팀에 문의하세요.
> 제품이 출시된 후에도 서비스가 계속해서 신규 고객의 이러한 시나리오에 계속 대응해야 합니다.

1. 허용된 계정을 사용해 제품을 구독하여 고객 경험을 테스트합니다.
2. 허용된 계정으로 구독한 후 계정이 등록 URL로 리디렉션되고 리디렉션이 임시 토큰을 포함하는 POST 요청인지 확인합니다. 향후 호출을 대비하여 애플리케이션에서 고객 ID를 유지합니다. 이것은 시나리오: 서비스가 신규 고객 검증의 일부를 테스트합니다.
3. 이전 단계에서 테스트 계정을 확인한 후 애플리케이션에 계정을 온보딩합니다. 예를 들어 테스트 고객이 양식을 작성하게 하여 새 사용자 계정을 생성할 수 있습니다. 또는 SaaS 애플리케이션에 액세스하기 위한 다음과 같은 다른 단계를 제공합니다. 이것은 시나리오: 서비스가 신규 고객 검증의 일부를 테스트합니다.
4. 테스트 고객이 온보딩된 후에서 `BatchMeterUsage` API 작업을 사용하여 AWS 결제 목적으로에 측정 레코드를 보낼 요청을 합니다 AWS Marketplace Metering Service. 이것은 시나리오: 사용량 측정을 테스트합니다.
5. 구독 변경 사항을 테스트합니다. 가능한 시나리오에는 구독 취소, 성공적인 구독 및 실패한 구독이 포함됩니다. 이것은 시나리오: 사용자 구독 변경 사항 모니터링을 테스트합니다.
6. 성공적으로 구독했는지 확인합니다. 성공적인 구독 메시지와 함께 테스트 계정에 대한 Amazon SNS 알림을 받으면 측정을 시작할 수 있습니다. Amazon SNS 알림을 받기 AWS Marketplace Metering Service 전에 로 전송되는 레코드는 측정되지 않습니다. 이것은 시나리오: 고객 구독 확인을 테스트합니다.

   > **Note**
   >
   > 결제 문제를 방지하려면 고객을 대신하여 리소스를 시작하기 전에 프로그래밍 방식으로 이 알림을 기다리는 것이 좋습니다.

7. 통합 요구 사항을 모두 완료하고 솔루션을 테스트한 후 AWS Marketplace 판매자 운영 팀에 알립니다. `BatchMeterUsage` API 작업과 함께 측정된 레코드를 성공적으로 전송했는지 확인하여 솔루션에 대한 일련의 최종 테스트를 실행합니다.

통합 및 테스트가 완료되면 최종 검토를 수행하고 제품을 공개적으로 나열할 수 있습니다 AWS Marketplace. 자세한 내용은 에서 SaaS 제품 생성 AWS Marketplace 단원을 참조하십시오.

SaaS 계약 제품을와 통합 AWS Marketplace

서비스형 소프트웨어(SaaS) 계약 제품을와 통합하는 것은의 한 단계 AWS Marketplace 입니다에서 SaaS 제품 생성 AWS Marketplace. 서비스형 소프트웨어(SaaS) 계약 제품을 통합하려면 코드를 작성하고 여러 고객 시나리오에 성공적으로 대응할 수 있음을 입증해야 AWS Marketplace합니다. 다음 섹션에서는 이러한 시나리오와 대응 방법을 설명하고, 통합 테스트의 개요를 제공합니다.

> **Note**
>
> 시작하기 전에, AWS Marketplace에서 서비스형 소프트웨어(SaaS) 제품에 적합한 요금 모델을 선택했는지 확인합니다. 자세한 내용은 SaaS 제품 계획 단원을 참조하십시오.

주제

- 시나리오: 서비스가 신규 고객 검증
- 시나리오: 서비스가 고객 요청 처리
- 시나리오: 사용자 구독 변경 사항 모니터링
- SaaS 계약 제품 통합 테스트

시나리오: 서비스가 신규 고객 검증

고객이 제품을 구독하면 고객은 판매자의 등록 URL로 리디렉션되며, 등록 URL은 임시 `x-amzn-marketplace-token` 토큰이 포함된 HTTP POST 요청입니다. 다음과 같은 방법으로 이 요청에 응답합니다.

1. AWS Marketplace Metering Service에서 `ResolveCustomer` API 작업을 호출하여 토큰을 `CustomerIdentifier`, `CustomerAWSAccountId` 및 `ProductCode`로 교환합니다.
2. AWS Marketplace Entitlement Service에서 `GetEntitlements` API 작업을 호출하여 고객이 액세스할 수 있는 구독 및 수량(해당하는 경우)을 확인합니다.
3. 향후 호출에 대비하여 시스템에서 `CustomerIdentifier`, `CustomerAWSAccountId` 및 `ProductCode`를 유지합니다. 고객의 구독이 유효한지 여부와 고객에 대해 필요한 정보를 저장합니다.
4. 요청에 대한 응답으로 사용자의 첫 사용 경험(서비스에 해당하는 경우)을 보여주어야 합니다.

시나리오: 서비스가 고객 요청 처리

고객이 서비스에 요청을 하면 적절한 작업 또는 메시지로 다음 시나리오에 대응해야 합니다.

- 시스템에 고객 ID가 없습니다. 아직 제품을 구독하지 않았다는 뜻입니다. 사용자에게 구독 방법을 알려줘야 합니다.
- 고객 ID가 있으며 `GetEntitlements` API 작업에서 적절한 권한을 반환합니다. 이 시나리오에서는 요청을 이행해야 합니다.

- 고객 ID가 있지만 `GetEntitlements` API 작업에서 권한을 반환하지 않거나 요청을 이행하기에는 수량이 충분하지 않습니다. 이 시나리오에서는 액세스를 처리하고 고객 경험을 관리하는 방법을 결정해야 합니다.

시나리오: 사용자 구독 변경 사항 모니터링

Amazon Simple Queue Service(Amazon SQS) 대기열을 설정하고 제품의 Amazon SNS 주제를 구독합니다. SNS 주제 정보는 제품을 생성할 때 AWS Marketplace 판매자 운영 팀으로부터 받은 이메일 메시지에 포함되어 있습니다. 자세한 내용은 에서 SaaS 제품 생성 AWS Marketplace 단원을 참조하세요. SNS 주제를 구독하면 특정 고객의 액세스 권한 제공 또는 취소를 포함하여 고객 권한 변경 사항에 대한 알림을 받게 됩니다.

> **Note**
>
> SNS 주제 Amazon 리소스 이름(ARN)은 `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>` 형식입니다.

응답해야 하는 유일한 알림은 다음과 같습니다.

- `entitlement-updated` - 고객 권한이 변경되었으므로, 새 상태를 보려면 `GetEntitlements` API 작업을 호출해야 합니다. 고객 스토어를 업데이트하고, 해당하는 경우(예: 고객 계약 만료) 방침에 따라 고객 리소스를 종료하고 보존 정책을 준수합니다.

> **Note**
>
> 자세한 내용은 를 사용하여 권한 확인 AWS Marketplace Entitlement Service 섹션을 참조하세요.

SaaS 계약 제품 통합 테스트

SaaS 계약 제품을와 통합한 후에는 심층 테스트를 수행하여 통합이 성공했는지 확인해야 AWS Marketplace합니다. 다음 절차에서는 제품 통합을 확인하는 단계를 간략하게 설명합니다.

> **Note**
>
> 본인의 계정을 사용하여 본인의 제품을 구독하고 통합이 성공적인지 테스트하세요. 가격을 일시적으로 인하하면 계정에 많은 요금이 부과되지 않고 구매 흐름을 테스트할 수 있습니다. 가격을 일시적으로 인하하거나 추가 테스트 계정에서 제품에 액세스할 수 있도록 허용하는 방법에 대한 자세한 내용은 AWS에 문의하세요.
> 제품이 출시된 후에도 서비스가 계속해서 신규 고객의 이러한 시나리오에 계속 대응해야 합니다.

1. 허용된 계정을 사용해 제품에 대한 계약을 가져와서 고객 경험을 테스트합니다.
2. 계정에 계약이 있으면 계정이 등록 URL로 리디렉션되고 리디렉션이 임시 토큰을 포함하는 POST 요청인지 확인합니다. 애플리케이션이 향후 호출을 위해 고객 ID를 유지하고 고객이 보유한 권한을 올바르게 처리하는지 확인합니다. 이것은 시나리오: 서비스가 신규 고객 검증의 일부를 테스트합니다.
3. 이전 단계에서 테스트 계정을 확인한 후 애플리케이션에 계정을 온보딩합니다. 예를 들어 테스트 고객이 양식을 작성하게 하여 새 사용자 계정을 생성할 수 있습니다. 또는 SaaS 애플리케이션에 액세스하기 위한 다음과 같은 다른 단계를 제공합니다. 이것은 시나리오: 서비스가 신규 고객 검증의 일부를 테스트합니다.
4. 온보딩 또는 진행 중인 검증 과정 중에 `GetEntitlements` API 작업에서 권한이 반환되지 않으면 애플리케이션에서 권한 없는 사용자의 액세스 권한과 경험을 올바르게 관리해야 합니다. 이것은 시나리오: 서비스가 고객 요청 처리을 테스트합니다.
5. 구독 변경 사항을 테스트합니다. 애플리케이션이 구독 해지, 구독 성공 및 구독 실패 시나리오를 제대로 처리하는지 확인합니다. 이것은 시나리오: 사용자 구독 변경 사항 모니터링을 테스트합니다.
6. 모든 통합 요구 사항을 완료하고 솔루션을 테스트한 후 AWS Marketplace 운영 팀에 알립니다. 그러면 이 팀은 판매자가 `GetEntitlements` API 작업을 성공적으로 호출하고 신규 고객을 충분히 온보딩했는지 확인하여 솔루션을 테스트합니다.

통합 및 테스트가 완료되면 최종 검토를 수행하고 제품을 공개적으로 나열할 수 있습니다 AWS Marketplace. 자세한 내용은 에서 SaaS 제품 생성 AWS Marketplace 단원을 참조하십시오. 환급 요청 양식을 작성하여 테스트 구독을 취소할 수도 있습니다. 구독 취소에 대한 자세한 내용은 the section called “제품 환불 요청” 섹션을 참조하세요.

SaaS 계약과 pay-as-you-go제 제품 통합 AWS Marketplace

제품을와 통합하는 것은의 한 단계 AWS Marketplace 입니다에서 SaaS 제품 생성 AWS Marketplace. 서비스형 소프트웨어(SaaS) 계약 제품을 통합하려면 코드를 작성하고 여러 고객 시나리오에 성공적으로 대응할 수 있음을 입증해야 AWS Marketplace합니다. 다음 섹션에서는 이러한 시나리오와 대응 방법을 설명하고, 통합 테스트의 개요를 제공합니다.

> **Note**
>
> 시작하기 전에, AWS Marketplace에서 서비스형 소프트웨어(SaaS) 제품에 적합한 요금 모델을 선택했는지 확인합니다. 자세한 내용은 SaaS 제품 계획 단원을 참조하십시오.

주제

- 시나리오: 서비스가 신규 고객 검증
- 시나리오: 서비스가 고객 요청 처리
- 시나리오: 사용량 측정
- 시나리오: 사용자 권한 변경 모니터링
- pay-as-you-go 통합으로 SaaS 계약 테스트

시나리오: 서비스가 신규 고객 검증

고객이 제품을 구독하면 고객은 판매자의 등록 URL로 리디렉션되며, 등록 URL은 임시 `x-amzn-marketplace-token` 토큰이 포함된 HTTP POST 요청입니다. 다음과 같은 방법으로 이 요청에 응답합니다.

1. AWS Marketplace Metering Service에서 `ResolveCustomer` API 작업을 호출하여 토큰을 `CustomerIdentifier`, `CustomerAWSAccountId` 및 `ProductCode`로 교환합니다.
2. AWS Marketplace Entitlement Service에서 `GetEntitlements` 작업을 호출하여 고객이 액세스할 수 있는 구독 및 수량(해당하는 경우)을 확인합니다.
3. 향후 호출에 대비하여 시스템에서 `CustomerIdentifier`, `CustomerAWSAccountId` 및 `ProductCode`를 유지합니다. 고객의 구독이 유효한지 여부와 고객에 대해 필요한 정보를 저장합니다.
4. 요청에 대한 응답으로 사용자의 첫 사용 경험(서비스에 해당하는 경우)을 보여주어야 합니다.

시나리오: 서비스가 고객 요청 처리

고객이 서비스에 요청을 하면 적절한 작업 또는 메시지로 다음 시나리오에 대응해야 합니다.

- 시스템에 고객 ID가 없습니다. 아직 제품을 구독하지 않았다는 뜻입니다. 구독 방법을 설명하는 메시지를 보내야 합니다.
- 고객 ID가 있으며 `GetEntitlements` API 작업에서 적절한 권한을 반환합니다. 이 시나리오에서는 요청을 이행해야 합니다.
- 고객 ID가 있지만 `GetEntitlements` API 작업에서 권한을 반환하지 않거나 요청을 이행하기에는 수량이 충분하지 않습니다. 이 시나리오에서는 액세스를 처리하고 고객 경험을 관리하는 방법을 결정해야 합니다.

시나리오: 사용량 측정

고객이 서비스를 사용하기 시작하면 매시간 측정 기록을 보내야 합니다. 자세한 측정 방법은 SaaS 구독에서 사용할 측정 구성 섹션을 참조하세요.

AWS CloudTrail 를 사용하여 활동을 모니터링하여 결제 정보가 전송되는지 확인하는 것이 좋습니다 AWS. 측정 레코드를 전송할 때 다음 사항에 유의하세요.

- 측정 요청은 정시에 중복 제거됩니다.
- 매 시간마다 전송된 레코드는 누적됩니다.
- 지난 시간 동안 레코드가 없더라도 1시간마다 사용량으로 0으로 하여 측정 레코드를 보내는 것이 좋습니다.

시나리오: 사용자 권한 변경 모니터링

Amazon Simple Queue Service(Amazon SQS) 대기열을 설정하고 제품의 Amazon SNS 주제를 구독합니다. 두 가지 SNS 주제가 있는데, 하나는 권한 변경에 대한 것이고 다른 하나는 구독 변경에 대한 것입니다. 제품 생성 시 AWS Marketplace 판매자 운영 팀으로부터 받은 이메일 메시지에 주제 정보가 포함되었습니다. 자세한 내용은 에서 SaaS 제품 생성 AWS Marketplace 단원을 참조하십시오. SNS 주제를 구독하면 특정 고객의 액세스 권한 제공 또는 취소를 포함하여 고객 구독 변경 사항에 대한 알림을 받게 됩니다.

> **Note**
>
> 구독 변경을 위한 SNS 주제 Amazon 리소스 이름(ARN)은 `arn:aws:sns:us-east-1:<account id>:aws-mp-subscription-notification-<product code>`

형식입니다. 권한 변경을 위한 SNS 주제 ARN은 `arn:aws:sns:us-east-1:<account id>:aws-mp-entitlement-notification-<product code>` 형식입니다.

응답해야 하는 알림은 다음과 같습니다.

- 자격 SNS 주제의 `entitlement-updated` - 고객 권한이 변경되었으므로, 새 상태를 보려면 `GetEntitlements` API 작업을 호출해야 합니다. 고객 스토어를 업데이트하고, 해당하는 경우(예: 고객 계약 만료) 방침에 따라 고객 리소스를 종료하고 보존 정책을 준수합니다.
- 구독 SNS 주제의 `subscribe-success` - 고객이 구독하고 있으며, 고객 ID를 대상으로 성공적으로 측정할 수 있습니다.
- 구독 SNS 주제의 `unsubscribe-pending` - 고객이 구독을 해지하는 중입니다. 마지막 측정 기록을 모두 보내야 합니다.
- 구독 SNS 주제의 `unsubscribe-success` - 고객이 구독을 해지했습니다. 고객의 측정 기록은 더 이상 수락되지 않습니다. 방침에 따라 고객 리소스를 종료하고 보존 정책을 준수합니다.
- 구독 SNS 주제의 `subscribe-fail` - 고객 구독이 실패했습니다. 고객 ID를 기준으로 측정하거나 고객을 대신하여 리소스를 활성화하면 안 됩니다.

> **Note**
>
> 자세한 내용은 [를 사용하여 권한 확인 AWS Marketplace Entitlement Service](#) 섹션을 참조하세요.

pay-as-you-go 통합으로 SaaS 계약 테스트

계약을 pay-as-you-go 제품과 통합한 후에는 심층 테스트를 수행하여 통합이 성공적인지 확인해야 AWS Marketplace합니다. 다음 절차에서는 제품 통합을 확인하는 단계를 간략하게 설명합니다.

> **Note**
>
> 본인의 계정을 사용하여 본인의 제품을 구독하고 통합이 성공적인지 테스트하세요. 가격을 일시적으로 인하하면 계정에 많은 요금이 부과되지 않고 구매 흐름을 테스트할 수 있습니다. 가격을 일시적으로 인하하거나 추가 테스트 계정에서 제품에 액세스할 수 있도록 허용하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 운영 팀에 문의하세요.

제품이 출시된 후에도 서비스가 계속해서 신규 고객의 이러한 시나리오에 계속 대응해야 합니다.

1. 허용된 계정을 사용해 제품에 대한 계약을 가져와서 고객 경험을 테스트합니다.
2. 계정에 계약이 있으면 계정이 등록 URL로 리디렉션되고 리디렉션이 임시 토큰을 포함하는 POST 요청인지 확인합니다. 애플리케이션이 향후 호출을 위해 고객 ID를 유지하고 고객이 보유한 권한을 올바르게 처리하는지 확인합니다. 이것은 시나리오: 서비스가 신규 고객 검증의 일부를 테스트합니다.
3. 이전 단계에서 테스트 계정을 확인한 후 애플리케이션에 계정을 온보딩합니다. 예를 들어 테스트 고객이 양식을 작성하게 하여 새 사용자 계정을 생성할 수 있습니다. 또는 SaaS 애플리케이션에 액세스하기 위한 다음과 같은 다른 단계를 제공합니다. 이것은 시나리오: 서비스가 신규 고객 검증의 일부를 테스트합니다.
4. 온보딩 또는 진행 중인 검증 과정 중에 `GetEntitlements` API 작업에서 권한이 반환되지 않으면 애플리케이션에서 권한 없는 사용자의 액세스 권한과 경험을 올바르게 관리해야 합니다. 이것은 시나리오: 서비스가 고객 요청 처리을 테스트합니다.
5. 테스트 고객이 온보딩된 후에서 `BatchMeterUsage` API 작업을 사용하여 AWS 결제 목적으로에 측정 레코드를 전송할 요청을 합니다 AWS Marketplace Metering Service. 이것은 시나리오: 사용량 측정을 테스트합니다.
6. 구독 변경 사항을 테스트합니다. 애플리케이션이 구독 해지, 구독 성공 및 구독 실패 시나리오를 제대로 처리하는지 확인합니다. 이것은 시나리오: 사용자 권한 변경 모니터링을 테스트합니다.
7. 모든 통합 요구 사항을 완료하고 솔루션을 테스트한 후 AWS Marketplace 판매자 운영 팀에 알립니다. 그러면 이 팀은 판매자가 `GetEntitlements` API 작업을 성공적으로 호출하고 신규 고객을 충분히 온보딩했는지 확인하여 솔루션을 테스트합니다. 또한 판매자가 `BatchMeterUsage` API 작업을 통해 측정된 레코드를 성공적으로 전송했는지 확인합니다.

통합 및 테스트가 완료되면 최종 검토를 수행하고 공개적으로 제품을 나열할 수 있습니다 AWS Marketplace. 자세한 내용은 에서 SaaS 제품 생성 AWS Marketplace 단원을 참조하십시오.

서버리스 SaaS 통합 배포

SaaS 솔루션을 생성한 후의 해당 목록과 통합합니다 AWS Marketplace. 이렇게 하려면 AWS CloudFormation 템플릿을 사용하여 서버리스 통합을 생성할 수 있습니다. 통합을 통해 고객을 등록하고, 액세스 권한을 부여 및 취소하고, 고객 권한을 업데이트하고, 측정된 사용량을 보고할 수 있습니다.

서버리스 SaaS 통합 생성에 대한 자세한 내용은 AWS CloudFormation 템플릿을 배포하고 관련 작업을 완료하는 방법을 설명하는 랩인 서버리스 SaaS 통합 참조와 Saas 통합을 참조하세요.

SaaS 제품 계획

서비스형 소프트웨어(SaaS) 제품을에 추가 AWS Marketplace하기 전에 먼저 몇 가지 계획을 세워야 합니다. 이 단계는 제품의 성공을 위해 매우 중요합니다. 계획이 없으면 결제 문제가 발생하거나 AWS Marketplace에서 상품을 다시 생성해야 할 수 있습니다. 다음 섹션에서는 SaaS 제품을 계획하는 방법을 보여줍니다.

> **Important**
>
> 대부분의 제품 설정은 구성한 후에 변경할 수 없습니다. 제품을 생성한 후 변경해야 하는 경우 올바른 설정으로 새 제품을 생성해야 할 AWS Marketplace수 있습니다.

주제

- 가격 책정 계획
- 결제 통합 계획
- Amazon SNS 통합 계획
- 고객이 제품에 액세스하는 방법 계획

가격 책정 계획

SaaS 제품에 대한 세 가지 요금 모델이 있습니다 AWS Marketplace. 제품 계획을 수립할 때 가장 중요한 결정은 제품에 적합한 요금 모델을 선택하는 것입니다. 잘못된 요금 모델을 선택하면 일정이 몇 주 뒤로 미뤄질 수 있습니다. 요금 모델에 따라 고객의 결제 옵션과 판매자가 작성, 테스트, 배포해야 하는 결제 통합 코드가 결정됩니다. 다양한 유형의 요금 모델에 대한 자세한 내용은 SaaS 제품 요금을 참조하세요.

> **Note**
>
> 모든 SaaS 요금 모델은 무료 평가판을 지원합니다. 자세한 내용은 SaaS 무료 평가판을 참조하세요.

결제 통합 계획

에서 SaaS 제품을 사용할 때 얻을 수 있는 이점 중 하나는 결제 AWS Marketplace 를 통합하는 것입니다. 이 혜택을 활용하려면 선택한 요금 모델에 따라 AWS Marketplace 측정 서비스 또는 AWS Marketplace Entitlement Service와 통합해야 합니다. 이러한 두 서비스를 통해 결제 및 사용량 보고가 정확한지 확인할 수 있습니다.

통합을 계획한 후에는 실제로 적용되기 전에 제품과의 통합을 테스트해야 합니다. 통합 및 테스트에 대한 자세한 내용은 AWS Marketplace 측정 및 권한 부여 서비스 API 액세스 단원을 참조하세요.

Amazon SNS 통합 계획

SaaS 제품과 관련하여 구독할 수 있는 두 가지 Amazon Simple Notification Service(SNS) 주제가 있습니다. 자세한 내용은 SaaS 알림을 참조하세요. 이러한 메시지는 고객이 또는 고객이 시작한 구독 및 계약의 변경 사항을 프로그래밍 방식으로 처리하는 데 도움이 될 수 AWS 있습니다. Amazon SNS 알림은 고객이 제품 등록 웹 사이트에서 새 계정을 등록할 수 있는 프로그래밍 방식 트리거가 될 수 있습니다. 또한 구독이 만료된 고객의 제품 액세스를 거부할 수도 있습니다. 이러한 알림 처리를 프로그래밍하는 방법에 따라 고객이 알림을 받는 방법을 선택할 수 있습니다.

고객이 제품에 액세스하는 방법 계획

이 섹션에서는 구매자가 제품에 액세스하는 방법에 대해 설명합니다.

SaaS 제품 등록 웹 사이트 계획

SaaS 제품을 구매하는 고객은 해당 제품에 액세스해야 합니다. 고객이 제품에 액세스하는 방법을 계획하고 구현해야 합니다. SaaS 제품은 다음 액세스 옵션을 지원합니다.

- 빠른 시작
- AWS PrivateLink
- 자체 제품 웹 사이트

등록 웹 사이트를 사용하여 AWS Marketplace 고객을 검증하려면 SaaS 고객 온보딩을 참조하세요.

고객이 제품에 액세스하도록 빠른 시작 사용

빠른 시작 배포 옵션을 사용하면 구매자가 제품을 구성, 배포 및 출시하는 데 필요한 시간과 리소스를 줄일 수 있습니다. 빠른 시작을 사용하면 구매자가 프로세스 중에 방문해야 하는 사이트 수가 줄어듭니다. 자세한 내용은 빠른 시작 구성을 참조하세요.

AWS PrivateLink 고객이 SaaS 제품에 액세스할 수 있도록를 사용합니다.

를 사용하여 Amazon VPC를 통해 제품 제공 AWS PrivateLink을 사용하여 서비스를 Amazon Virtual Private Cloud(VPC) 엔드포인트 서비스로 구성할 수 있습니다. 고객은 VPC 엔드포인트를 생성하여 AWS 클라우드 가상 네트워크를 통해 소프트웨어에 액세스할 수 있습니다. 또는 인터넷을 통해 연결을 생성한 고객의 경우 소유 및 유지 관리하는 웹 사이트를 통해 소프트웨어 제품에 대한 액세스를 제공할 수 있습니다.

고유한 웹 사이트 사용

SaaS 제품은 사용자 환경에서 호스팅되며 웹 사이트와 같이 관리 및 유지하는 퍼블릭 엔드포인트를 통해 인터넷에서 액세스해야 합니다. 일반적으로 고객이 제품에 등록하고, 로그인하여 제품을 사용하고, 제품에 대한 지원을 받기 위해 사용하는 웹 사이트가 있습니다.

에 대한 SaaS 제품 지침 AWS Marketplace

AWS Marketplace 는 고객에게 안전하고 신뢰할 수 있는 플랫폼을 홍보 AWS Marketplace 하기 위해의 모든 서비스형 소프트웨어(SaaS) 제품 및 제품에 대해 다음 지침을 유지합니다. 다음 섹션에서는의 SaaS 제품에 대한 지침을 제공합니다 AWS Marketplace.

제출 시 모든 제품 및 관련 메타데이터를 검토하여 현재 AWS Marketplace 지침을 충족하거나 초과하는지 확인합니다. 이러한 가이드라인을 검토하고 조정하여 진화하는 보안 요구 사항을 충족합니다. 또한는 제품을 AWS Marketplace 지속적으로 검토하여 이러한 지침의 변경 사항을 충족하는지 확인합니다. 제품이 규정을 준수하지 않는 경우 제품을 업데이트해야 할 수 있으며, 경우에 따라 문제가 해결될 때까지 새 구독자가 일시적으로 제품을 사용하지 못할 수 있습니다.

주제

- 제품 설정 가이드라인
- 고객 정보 요구 사항
- 제품 사용 가이드라인
- 아키텍처 가이드라인

제품 설정 가이드라인

모든 SaaS 제품은 다음 제품 설정 지침을 준수해야 합니다.

- 요금 차원은 비공개 제안으로만 제한할 수 없습니다. 구매자는 공개 제품에 대한 요금 차원을 구독할 수 있어야 합니다.

- 적어도 하나의 요금 차원이 0.00 USD보다 커야 합니다.
- 모든 요금 차원은 실제 소프트웨어와 관련되어야 하며 소프트웨어와 무관한 다른 제품이나 서비스는 포함할 수 없습니다.
- AWS GovCloud (US) 리전에서만 제공되는 SaaS 제품은 제품 제목에 `GovCloud`어딘가에 포함되어야 합니다.

고객 정보 요구 사항

모든 SaaS 제품은 다음과 같은 고객 정보 요구 사항을 준수해야 합니다.

- SaaS 제품은 모두 AWS Marketplace에 등록된 차원을 통해 요금이 청구되어야 합니다.
- 신용 카드 및 은행 계좌 정보를 포함하여 SaaS 제품에 대한 고객 결제 정보는 절대 수집할 수 없습니다.
- SaaS 등록 페이지에는 구매자의 이메일 주소에 대한 입력 필드가 포함되어야 합니다. 다음과 같은 추가 필드를 포함할 수 있습니다.
  - 명칭
  - 우편 번호
  - 전화번호
  - 회사 정보
  - 제품 설정 기본 설정

  여러 언어를 사용하려는 경우 등록 페이지의 영어 보기를 제공해야 합니다.

제품 사용 가이드라인

모든 SaaS 제품은 다음 제품 사용 지침을 준수해야 합니다.

- 에서 제품을 구독한 후 AWS Marketplace고객은 SaaS 애플리케이션 내에서 계정을 생성하고 웹 콘솔에 액세스할 수 있어야 합니다. 고객이 애플리케이션에 즉시 액세스할 수 없는 경우 액세스 가능 시점에 대한 구체적인 지침이 포함된 메시지를 제공해야 합니다. 계정이 생성되면 고객에게 계정이 생성되었음을 확인하는 알림과 명확한 다음 단계에 대한 지침을 보내야 합니다.
- 고객이 이미 SaaS 애플리케이션에 계정이 있는 경우 이행 랜딩 페이지에서 로그인할 수 있는 권한이 있어야 합니다.
- 고객은 관련 계약 또는 구독 사용 정보를 포함하여 SaaS 애플리케이션 내에서 구독 상태를 볼 수 있어야 합니다.

- 고객은 애플리케이션 사용, 문제 해결 및 환급금 요청(해당하는 경우)과 같은 문제에 대한 도움을 쉽게 받을 수 있어야 합니다. 이행 랜딩 페이지에 지원 연락처 옵션이 명시되어 있어야 합니다.
- AWS Marketplace에서 사용할 수 없는 다른 클라우드 플랫폼, 추가 제품, 업셀 서비스 또는 무료 평가판 제안으로 사용자를 유도하는 언어가 제품 소프트웨어 및 메타데이터에 포함되면 안 됩니다.

  SaaS 제품의 무료 평가판에 대한 자세한 내용은 에서 SaaS 무료 평가판 제안 생성 AWS Marketplace 섹션을 참조하세요.
- 제품이 다른 제품이나 다른 ISV 제품의 추가 기능인 경우 제품 설명에 해당 제품은 다른 제품의 기능을 확장한다는 내용과 다른 제품이 없으면 해당 제품의 유용성이 매우 제한된다는 내용을 명시해야 합니다. 예: 이 제품은 <제품 이름>의 기능을 확장하며, <제품 이름> 제품이 없으면 이 제품의 유용성이 매우 제한됩니다. 이 목록의 모든 기능을 사용하려면 <제품 이름>의 자체 라이선스가 필요할 수 있습니다.

아키텍처 가이드라인

다음 주제에서는 SaaS 제품의 아키텍처 지침을 나열하고 설명합니다.

주제

- 지침
- 아키텍처 다이어그램 생성

지침

> **Note**
> 다음 지침은 2025년 5월 1일부터 유효합니다.

- 모든 SaaS 아키텍처를 게시할 수 있습니다.
- 에 배포된 제품은 검색 결과 및 제품 세부 정보 페이지에서 특별한 지정 AWS Marketplace 을 AWS 받습니다. 가 제품을 "배포 날짜 AWS"로 간주 AWS Marketplace 하려면 제품이 완전히 실행되어야 합니다 AWS. 여기에는 애플리케이션 및 컨트롤 플레인이 포함됩니다.

  애플리케이션 영역은 판매자의 AWS 계정, 구매자의 AWS 계정 또는 둘 다에서 실행할 수 있습니다. 자세한 내용은 컨트롤 플레인과 애플리케이션 플레인 백서를 참조하세요.

콘텐츠 전송 네트워크(CDNs), 도메인 이름 시스템(DNSs) 및 기업 자격 증명 공급자(IdPs)를 제외하고 제품에서 애플리케이션 데이터를 전송, 저장 또는 처리하는 데 사용되는 타사 서비스도 완전히 실행해야 합니다 AWS.

> **Note**
>
> 애플리케이션 데이터는 구매자에 속하거나 구매자를 위해 생성된 데이터입니다.

보안, 모니터링, 데이터 복제 또는 마이그레이션을 위해 제품에서 사용하는 에이전트 또는 게이트웨이는 온프레미스를 AWS포함하여 외부의 구매자 소유 환경에서 실행할 수 있지만 저장 및 분석을 AWS 위해 로만 데이터를 전송해야 합니다.

검토를 위해 아키텍처 다이어그램을 포함해야 합니다. 다이어그램을 퍼블릭으로 설정할 수 없습니다. 자세한 내용은 다음 섹션아키텍처 다이어그램 생성의 섹션을 참조하세요.

- 판매자는 완전히 실행되지 않는 제품을 게시할 수 있습니다 AWS.
- 구매자의 인프라에 리소스가 필요한 애플리케이션은 다음 지침을 따라야 합니다.
  - 관리형 서비스가 아닌 SaaS 제품으로 간주되려면 SaaS 아키텍처 기본 AWS 백서에 정의된 컨트롤 플레인이 관리하는 인프라에 있어야 합니다. 자세한 내용은 SaaS vs. Managed Service Provider 백서를 참조하세요.
  - 제품 설명에서 고객에게 AWS Marketplace 거래와 별도로 인프라 요금이 발생하는 AWS 경우 해당 요금을 지불해야 한다고 알려야 합니다.
  - AWS 보안 토큰 서비스(AWS STS) 또는 AWS Identity and Access Management(IAM)를 사용하는 등 안전한 방식으로 리소스를 프로비저닝해야 합니다.
  - 애플리케이션에 권한을 부여하는 사용 지침 또는 배포 템플릿을 생성할 때 최소 권한 원칙을 따라야 합니다.
  - 프로비저닝된 AWS 모든 서비스, IAM 정책 설명, IAM 역할 또는 사용자가 고객의 계정에 배포되고 사용되는 방법을 설명하는 추가 설명서를 제공해야 합니다.
  - 구매자가 AWS 계정에 필요한 리소스를 배포할 수 있도록 하는 지침 또는 배포 템플릿을 제공해야 합니다.
  - 구매자의 AWS 계정에 리소스를 배포하기 위한 AWS CloudFormation 템플릿(CFTs)을 제공하는 경우 구매자는AWS Marketplace CFTs 정책을 준수해야 합니다. 구매자에 대해 SaaS 빠른 시작 배포 옵션을 활성화할 때 제공된 방법에 따라 이러한 CFTs를 SaaS 목록의 일부로 게시해야 합니다. SaaS Quick Launch를 사용하면 구매자가 SaaS 솔루션을 더 쉽게 구성할 수 있습니다.

- Amazon Machine Image(AMIs 구매자의 AWS 계정에 배포된 경우AWS Marketplace AMIs에 대한 정책을 준수해야 합니다. AMIs AWS Marketplace Management Portal(판매자 포털)에서 AMI 스캐너를 통과해야 합니다. 제품을 공개하도록 요청할 때 작업에 문의 AWS Marketplace하고 스캔 결과 증명을 제공해야 합니다.
- 컨테이너 이미지가 구매자의 AWS 계정에 배포되는 경우 이미지는 AWS Marketplace 컨테이너에 대한 정책을 준수해야 합니다. 컨테이너 이미지는 외부에서 호스팅할 수 AWS있지만 Amazon Elastic Container Registry(Amazon ECR)에서 스캔해야 하며 심각한 취약성이 없어야 합니다. 제품을 공개하도록 요청할 때는 작업에 문의 AWS Marketplace하고 컨테이너가 스캔을 통과했다는 증거를 제공해야 합니다.

- 공급자로 AWS 계정 등록되어 SaaS 게시 요청을 제출한에서 AWS Marketplace APIs를 성공적으로 호출했습니다. SaaS 요금 모델에 따라 호출해야 할 API가 결정됩니다.
  - SaaS 계약 -의 GetEntitlements AWS Marketplace Entitlement Service.
  - 소비가 포함된 SaaS 계약 -의 GetEntitlements AWS Marketplace Entitlement Service 및 AWS Marketplace 측정 서비스의 BatchMeterUsage.
  - SaaS 구독 - AWS Marketplace 측정 서비스의 BatchMeterUsage.
- AWS GovCloud (US) 리전에서만 독점적으로 제공되는 SaaS 제품은 다른 AWS 리전 리전과 AWS GovCloud (US) 리전 간의 아키텍처 경계, 제품의 사용 사례 및 제품에 권장되지 않는 워크로드를 설명해야 합니다.

SaaS 아키텍처에 대한 자세한 내용은 SaaS 아키텍처 기본 백서를 참조하세요 AWS .

아키텍처 다이어그램 생성

제품이 배포된 특별 지정을 받으려면 AWS Marketplace Management Portal에서 제품의 아키텍처 세부 정보를 AWS업데이트합니다. ??? 에 배포된 호스팅 패턴을 AWS 선택하고가 AWS 검토하는 아키텍처 다이어그램을 업로드합니다. 에서 배포를 고려하는 AWS Marketplace 호스팅 패턴은 이전 섹션지침의 섹션을 AWS참조하세요. 호스팅 패턴이 변경되면 제품의 아키텍처 세부 정보를 업데이트해야 합니다.

> **Note**
>
> - 다이어그램을 공개로 설정할 수 없습니다.
> - 완전히 실행되지 않는 제품을 게시할 수 있습니다 AWS.

다이어그램을 생성할 때 다음 기준을 사용합니다.

- 애플리케이션 영역 또는 컨트롤 플레인의 일부로 구성 요소를 그룹화하고 레이블을 지정합니다.
- 제품 핵심 비즈니스 로직의 일부 AWS 인 외부 구성 요소의 경우 애플리케이션 영역으로 구성 요소를 그룹화합니다.
- 구성 요소는 하위 수준 세부 정보(예: 컴퓨팅 인스턴스 및 네트워크 서브넷) 또는 상위 수준 서비스(예: 데이터 분석 플랫폼)를 나타낼 수 있습니다.
- 구성 요소는 사용되는 AWS 서비스 또는 비AWS 서비스의 이름을 식별할 필요가 없습니다.
- 논리적으로 실행되는 구성 요소를 배치합니다. 예를 들어 판매자의 AWS 계정, 구매자의 AWS 계정 AWS , 판매자의 환경 외 환경 또는 기타 환경.
- 데이터 복제 또는 워크로드 마이그레이션 제품의 경우 지원되는 모든 소스 및 대상 환경을 포함합니다.

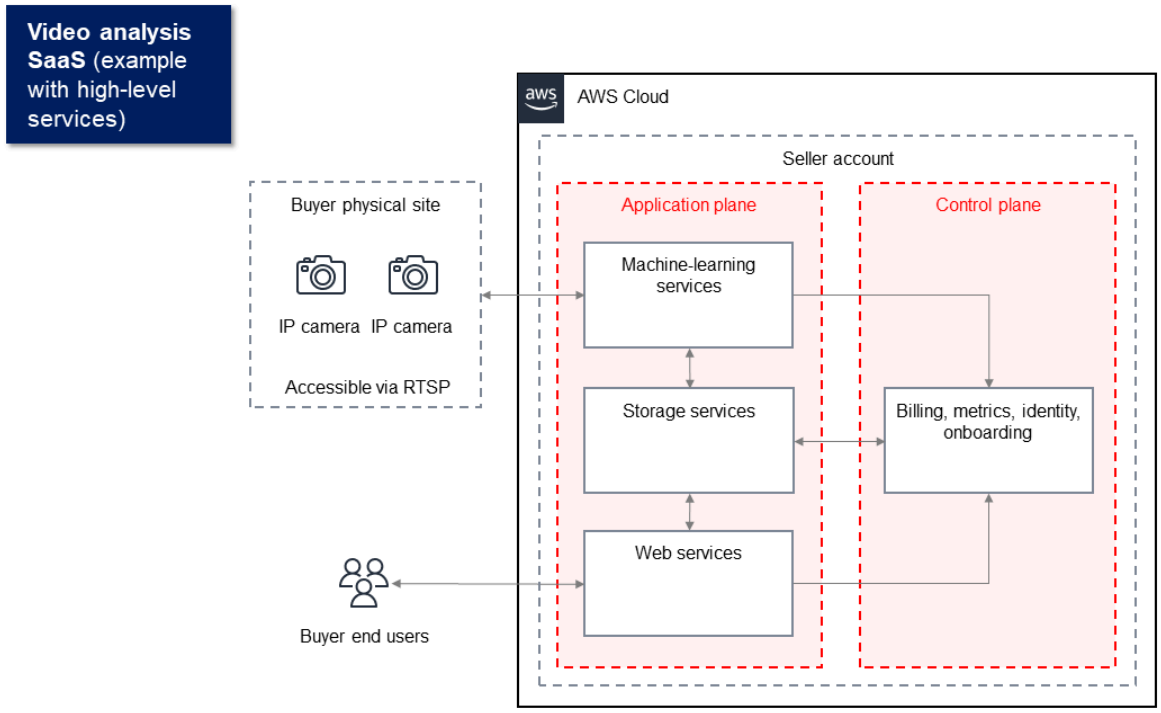
> **Note**
>
> SaaS 제품의 아키텍처 세부 정보를 업데이트하는 데 사용하는 아키텍처 다이어그램은 게시되지 않으며 구매자가 공개적으로 사용할 수 없습니다.

세부 정보의 수준

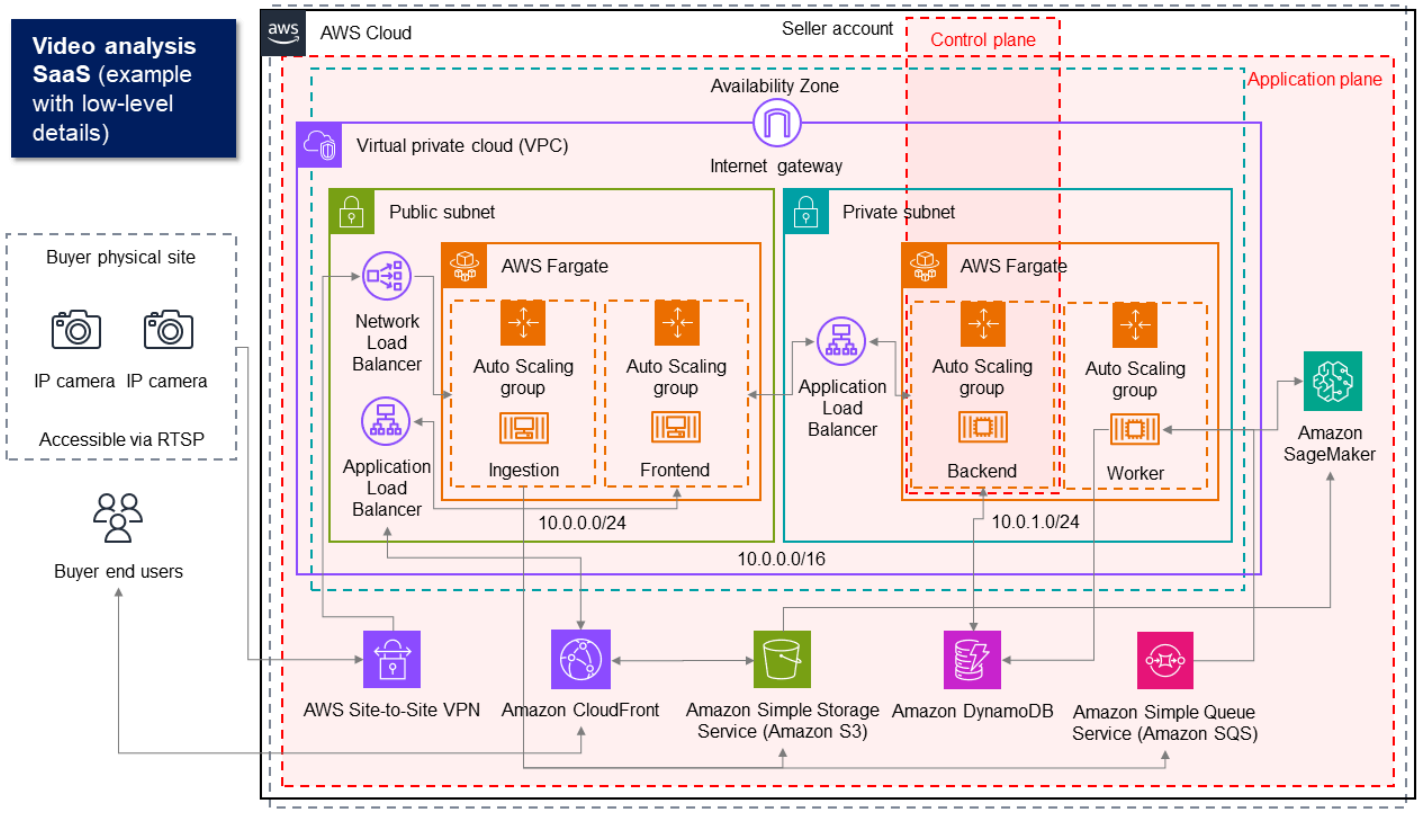
기본 시스템 구성 요소를 보여주고, 기본 데이터 흐름을 포함하고, 애플리케이션 플레인 및 컨트롤 플레인 서비스에 초점을 맞춘 상위 수준 다이어그램을 생성할 수 있습니다. 또는 각 구성 요소를 세분화하고, 특정 연결을 표시하고, 다양한 수준의 세부 정보가 포함된 기술 사양을 포함하는 하위 수준의 세부 다이어그램을 생성할 수 있습니다.

다음 다이어그램은 가상 비디오 분석 SaaS 애플리케이션의 아키텍처를 보여줍니다. 각 에는 서로 다른 수준의 세부 정보가 표시됩니다. 둘 다 허용됩니다. 이를 자체 다이어그램에 포함할 세부 수준 예제로 사용합니다.

다음은 상위 수준 다이어그램의 예입니다.



다음은 하위 수준 세부 다이어그램의 예입니다.



자세한 내용은 아키텍처 다이어그램이란 무엇입니까?를 참조하세요. 다이어그램을 생성한 후 AWS Marketplace Management Portal(AMMP)에서 아키텍처 세부 정보를 업데이트합니다. 자세한 내용은 아키텍처 세부 정보 업데이트 단원을 참조하십시오.

의 SaaS 제품 요금 AWS Marketplace

구매자가 서비스형 소프트웨어(SaaS) 제품을 구매한 후 AWS Marketplace AWS Marketplace 는 결제 식별자를 제공합니다. 판매자는 결제 식별자를 사용하여 AWS Marketplace Entitlement Service 및 AWS Marketplace Metering Service를 호출합니다. 그런 다음 고객은 사용자 AWS 환경에서 또는 사용자가 생성한 Virtual Private Cloud(VPC) 엔드포인트 연결을 통해 제품에 액세스합니다. 이 주제에서는 에 대한 SaaS 요금 모델 목록을 제공합니다 AWS Marketplace.

> **Note**
>
> 모든 SaaS 요금 모델은 무료 평가판을 지원합니다. 자세한 내용은 SaaS 무료 평가판을 참조하세요.

SaaS 요금 모델

요금 모델	설명
SaaS 구독	SaaS 제품의 시간당 사용량에 대해 구매자에게 요금을 청구하는 사용한 만큼만 지불 모델입니다. 자세한 내용은 SaaS 구독 요금 단원을 참조하세요.
SaaS 계약	구매자에게 소프트웨어 사용에 대해 미리 요금을 청구하거나 유연한 결제 일정을 제공할 수 있습니다. 고객은 계약보다 많은 추가 사용에 대한 비용을 결제할 수도 있습니다. 자세한 내용은 SaaS 계약 요금 단원을 참조하세요.
사용한 만큼만 지불 방식의 SaaS 계약	구매자에게 소프트웨어 사용에 대해 미리 요금을 청구하거나 유연한 결제 일정을 제공할 수 있습니다. 또한 구매자에게는 계약 요금 외에도 측정된 사용량에 대한 추가 요금이 청구됩니다. 자

요금 모델	설명
	세한 내용은 SaaS 계약 요금 단원을 참조하십시오.

> **Note**
> 목록을 생성하고 제한적으로 게시한 후에는 요금 모델을 변경할 수 없습니다.

SaaS 제품을에서 사용할 수 있도록 하려면 SaaS 구독 요금 모델을 제공할지 아니면 SaaS 계약 요금 모델을 제공할지 AWS Marketplace결정합니다.

SaaS 구독 요금

서비스형 소프트웨어(SaaS) 구독의 경우는 사용자가 당사에 보내는 측정 레코드를 기반으로 고객에게 AWS Marketplace 요금을 청구합니다. 모든 요금은 고객의 계정에 배포된 소프트웨어에서 매 시간 측정 및 보고되어야 합니다. 그런 다음 모든 사용량은 월별로 계산되고 AMI 기반 AWS Marketplace 제품과 동일한 메커니즘을 사용하여 월별로 청구됩니다. AWS는 고객에게 받는 측정 레코드를 기준으로 고객의 제품 사용 요금을 청구합니다. 사용자는 제품의 측정 레코드가 성공적으로 전송 및 수신되었는지 확인할 책임이 있습니다.

구독 요금이 적용되는 SaaS 제품을 게시하려면 먼저 다음 작업을 수행해야 합니다.

1. 에서 새 SaaS 제품을 생성하고 새 SaaS 구독을 AWS Marketplace Management Portal선택합니다.
2. 일반 탭의 필드에 필요한 정보를 입력합니다. 제품 코드를 적어 둡니다.
3. 요금 탭의 요금 설정에서 제품의 요금을 가장 정확하게 설명하는 범주를 선택합니다. 요금 범주는 AWS Marketplace 웹 사이트에 고객에게 표시됩니다. 대역폭(GBps, MBps), 데이터(GB, MB, TB), 호스트(시간), 요청 또는 사용자(시간) 중에서 선택할 수 있습니다. 미리 정의된 범주 중 요구 사항에 적합한 것이 없는 경우 더 일반적인 단위 범주를 선택할 수 있습니다.

   다음으로, 요금 차원을 정의합니다. 각 요금 차원은 단위 요금당 설정할 수 있는 기능 또는 서비스를 나타냅니다. 차원의 예에는 사용자, 스캔한 호스트 및 수집된 로그 GB가 포함됩니다. 최대 24개 차원을 정의할 수 있습니다. 정의한 각 차원에 대해 다음 정보를 추가해야 합니다.

   - 차원 API 이름 - AWS Marketplace Metering Service에 측정 레코드를 전송할 때 사용되는 API 이름입니다. 이 이름은 고객이 사용한 차원을 나타냅니다. 이 이름은 결제 보고서에 공개됩니다. 사

용자만 보고서에 액세스할 수 있으므로 사람들이 읽기 편한 이름일 필요는 없습니다. 이름을 설정한 후에는 변경할 수 없습니다.

- 차원 설명 - 제품에 대한 차원을 설명하는 고객용 설명문입니다. 설명은 70자를 초과할 수 없으며 사용자에게 친숙해야 합니다. 설명의 예로는 시간당 관리자 수, 프로비저닝된 Mbps당 대역폭 등이 있습니다. 제품이 게시된 후, 이 설명을 변경할 수 없습니다.
- 차원 요금 - 이 제품의 FCP 단위당 소프트웨어 요금(USD)입니다. 이 필드는 소수점 이하 8자리를 지원합니다.

SaaS 구독이 종료되는 경우

고객은 AWS Management Console을 통해 SaaS 구독 제품을 구독 해지할 수 있습니다. SaaS 구독 종료 프로세스의 핵심은 다음과 같습니다.

1. 해당 고객에 대한 Amazon SNS 주제를 통해 `unsubscribe-pending` 알림이 SaaS 제품에 전송됩니다.
2. 1시간 동안 구매자의 나머지 사용량을 측정할 수 있습니다.
3. 이 시간 후 `unsubscribe-success` 알림을 수신합니다. 이때, 이 고객에 대한 측정 레코드를 더 이상 전송할 수 없습니다.

구독 해지한 고객에게 SaaS 제품의 기능을 비활성화하는 방법을 결정하는 것은 판매자의 몫입니다. 예를 들어 제품에서 고객의 기존 작업을 완료하지만 고객이 작업을 생성하지 못하게 할 수 있습니다. 고객에게 고객의 사용량이 비활성화되었다고 알리는 메시지를 표시할 수 있습니다. 고객은를 통해 제품을 다시 구독할 수 있습니다 AWS Marketplace.

SaaS 구독이 취소된 경우

SaaS 구독 취소 프로세스의 핵심은 다음과 같습니다.

1. 고객은 AWS Marketplace 웹 사이트의 Marketplace 소프트웨어 페이지에서 SaaS 구독 제품에 대한 구독을 취소할 수 있습니다.

   해당 고객에 대한 Amazon SNS 주제를 통해 알림이 SaaS 제품에 전송됩니다.
2. 1시간 동안 구매자의 나머지 사용량을 측정할 수 있습니다.
3. 취소가 진행 중이라는 알림을 제품에서 고객에게 보냅니다. 고객이 제품을 통해 취소하기를 원하는 경우 AWS Marketplace로 안내합니다. 향후 요금이 발생하지 않도록 하려면를 사용하여 취소를 확인해야 합니다 AWS Marketplace.

SaaS 계약 요금

서비스형 소프트웨어(SaaS) 계약의 경우 고객이 소프트웨어 구매를 시작하고 판매자와 계약을 체결합니다. 계약에 따라 고객은 지정된 수량의 SaaS 제품을 사용할 수 있습니다. AWS Marketplace는 SaaS 애플리케이션에 이러한 권한 부여를 제공합니다. 이 작업은 AWS Marketplace Entitlement Service에서 이루어집니다. SaaS 계약요금 모델을 사용할 때 애플리케이션은 측정 레코드를 전송하지 않습니다. 대신 AWS Marketplace Entitlement Service를 호출하여 권한을 확인합니다. 사용자는 사용량 범주, 차원 및 계약 기간을 정의합니다.

AWS Marketplace는 사용자와 고객 간의 계약을 기반으로 정의한 선결제 또는 결제 일정에 따라 고객에게 요금을 청구합니다. 그 이후 고객에게 해당 리소스를 사용할 권한이 부여됩니다. 계약 조건을 초과하는 추가 사용량의 경우 소프트웨어에서 사용량을 보고해야 하며 AWS Marketplace는 AWS Marketplace Metering Service를 통해 수신한 측정 기록을 기준으로 고객에게 요금을 청구합니다.

계약 가격이 적용되는 SaaS 제품을 게시하려면 먼저 다음 작업을 수행해야 합니다.

1. AWS Marketplace Management Portal에서 새 SaaS 제품을 생성하고 새 SaaS 계약을 선택합니다.
2. 일반 탭의 필드에 필요한 정보를 입력합니다. 제품 코드를 적어 둡니다.
3. 요금 탭에서 다음을 수행합니다.
   a. 요금 설정에서 고객에게 제안할 계약 기간을 선택합니다. 각 계약 기간에 다른 요금을 입력할 수 있습니다. 월별, 1년, 2년, 3년 옵션 중에 하나 이상을 선택할 수 있습니다. 비공개 제안을 생성하는 경우 사용자 지정 기간을 월 단위로(최대 60개월) 선택할 수 있습니다.
   b. 고객에게 제안할 계약 유형 선택에서 고객이 제품을 구매할 수 있는 옵션을 선택합니다.
      - 구매자는 하나 이상의 옵션 선택 가능 - 고객은 제공되는 각 요금 차원에 대한 수량을 선택할 수 있습니다.
      - 구매자는 하나의 티어만 선택 가능 - 고객은 다양한 기능, 서비스 및 사용량 세트가 포함된 옵션 중에서 하나의 티어를 선택합니다.
   c. 제품의 요금을 가장 정확하게 설명하는 사용 단위 범주를 선택합니다. 요금 범주는 AWS Marketplace 웹 사이트에서 고객에게 표시됩니다. 대역폭(GBps, MBps), 데이터(GB, MB, TB), 호스트(시간), 요청 또는 사용자(시간) 중에서 선택할 수 있습니다. 미리 정의된 범주 중 요구 사항에 적합한 것이 없는 경우 더 일반적인 단위 범주를 선택할 수 있습니다.
4. 범주를 선택한 후 요금 차원을 정의합니다. 각 요금 차원은 단위 요금당 설정할 수 있는 기능 또는 서비스를 나타냅니다. 차원의 예에는 사용자, 검사된 호스트 및 수집된 로그 GB가 있습니다. 정의한 각 차원에 대해 이름, 설명, 요금 및 API 이름을 추가합니다. 이름, 요금 및 설명이 고객에게 표시됩니다. 다음과 같이 AWS Marketplace를 사용하여 추적 및 보고할 때 API 이름을 사용합니다.
   - 고객이 구매한 차원을 검색하기 위해 AWS Marketplace Entitlement Service를 호출할 때

- 고객이 어떤 차원을 사용하는지 표시하기 위해 AWS Marketplace Metering Service를 호출할 때.

계약에 추가하는 각 요금 차원의 경우 고객이 계약 이외의 해당 차원을 추가로 사용하는 만큼 결제하도록 선택할 수 있습니다. 또한 계약 요금 없이 고객이 사용한 만큼 결제하는 차원을 추가할 수도 있습니다.

마법사를 사용하여 SaaS 제품에 대한 계약을 생성할 때 요금 차원에 대해 다음 필드를 정의해야 합니다.

- 차원 API 이름 - 권한 부여 API를 호출할 때 사용되는 이름입니다. 이 이름은 결제 보고서에 표시되며 보고서는 외부용이 아닙니다. API 이름의 최대 길이는 15자입니다. 이름을 설정한 후에는 변경할 수 없습니다.
- 차원 표시 이름: - 차원의 고객용 이름입니다. 이 이름은 고객이 제품의 차원을 이해하는 데 도움이 됩니다. 이름은 사용자에게 친숙해야 하며 최대 길이는 24자입니다. 이 값은 변경할 수 없습니다.
- 차원 설명: - 제품의 차원에 대한 추가 설명을 제공하는 차원의 고객용 설명입니다. 설명의 최대 길이는 70자입니다.
- 차원 - 월별 요금 - 이 차원의 1개월 옵션에 대해 단위당 청구되는 소프트웨어 요금입니다. 이 필드는 소수점 세 자리를 지원합니다.
- 차원 - 1년 요금 - 이 차원의 12개월 옵션에 대해 단위당 청구되는 소프트웨어 요금입니다. 이 필드는 소수점 세 자리를 지원합니다. 월별 요금이 아닙니다. 요금은 12개월의 일회성 요금을 반영해야 합니다.
- 차원 - 2년 요금 - 이 차원의 24개월 옵션에 대해 단위당 청구되는 소프트웨어 요금입니다. 이 필드는 소수점 세 자리를 지원합니다.
- 차원 - 3년 요금 - 이 차원의 36개월 옵션에 대해 단위당 청구되는 소프트웨어 요금입니다. 이 필드는 소수점 세 자리를 지원합니다.

예: 데이터 스토리지 애플리케이션

	월별 요금	12개월 요금	24개월 요금	추가 사용량에 대한 종량 과금제
암호화되지 않은 데이터(GB)	$1.50/GB	$16.00/GB	$30.00/GB	시간당 $0.1/GB
암호화된 데이터(GB)	$1.55/GB	$16.60/GB	$31.20/GB	시간당 $0.11/GB

예제: 로그 모니터링 제품

	월별 요금	12개월 요금	추가 사용량에 대한 종량 과금제
기본(10개의 호스트 모니터링, 5개의 컨테이너 모니터링)	100 USD	1000 USD	
스탠다드(20개의 호스트 모니터링, 10개의 컨테이너 모니터링)	$200	2000 USD	
프로(40개의 호스트 모니터링, 20개의 컨테이너 모니터링)	400 USD	4000 USD	
시간당 추가 호스트 모니터링			$0.1
시간당 추가 컨테이너 모니터링			0.2 USD

> **Note**
>
> 요금은 1개월, 12개월, 24개월, 36개월 기간일 수 있습니다. 제품에 대해 이러한 옵션을 하나 이상 제공하도록 선택할 수 있습니다. 기간은 각 차원에 대해 동일해야 합니다. 예를 들어 `ReadOnlyUsers` 및 `AdminUsers` 차원이 있다고 가정하겠습니다. `ReadOnlyUsers`의 연간 요금을 제안하는 경우 `AdminUsers`의 연간 요금도 제안해야 합니다.

SaaS 계약 업그레이드

고객은 더 긴 기간을 제외하고 더 높은 값 하나에 대해 계약을 업그레이드할 수 있습니다. 예를 들어, 더 높은 수량 또는 더 높은 값 권한으로 업그레이드할 수 있습니다. 기존 계약에 대해 비례 할당으로 계산된 크레딧이 고객에게 제공됩니다. 고객은 기존 계약의 규모를 줄일 수 없습니다. 갱신 크기를 줄이거나 갱신을 취소할 수만 있습니다.

권한은 SaaS 제품에서 AWS Marketplace Entitlement Service를 호출하여 확인됩니다.

자동 갱신

고객이 SaaS 계약을 사용하여 AWS Marketplace를 통해 제품을 구매한 경우 이러한 계약 조건의 자동 갱신에 동의할 수 있습니다. 고객은 매달, 1년마다, 2년마다 또는 3년마다 권한 부여에 대한 요금을 계속 지불합니다. 고객은 언제든지 갱신 설정을 수정할 수 있습니다. 고객은 갱신을 취소하거나 계약의 수량과 기간을 다르게 하여 갱신할 수 있습니다.

SaaS 계약이 종료되는 경우

SaaS 계약 제품에는 계약 만료 조건이 있습니다. 계약이 종료되면 다음과 같은 이벤트가 발생합니다.

1. SaaS 제품은 구매자의 권한이 변경되었음을 알리는 `entitlement-updated` 알림을 받습니다. AWS Marketplace Entitlement Service는 빈 응답을 반환합니다.
2. 1시간 동안 구매자의 나머지 사용량을 측정할 수 있습니다. 이 시간이 경과한 후에는 이 고객에 대한 측정 레코드를 더 이상 전송할 수 없습니다.

SaaS 계약이 취소되는 경우

SaaS 계약 취소 프로세스의 핵심은 다음과 같습니다.

1. 고객은 AWS Support를 통해 SaaS 계약 제품에 대한 취소 및 환급금을 요청할 수 있습니다.

   고객은 AWS Support를 통해 48시간 내에 환급금을 요청해야 합니다.

   전액 또는 비례 배분된 환급금은 일반적으로 영업일 기준 3~5일 내에 제공됩니다.
2. 해당 고객에 대한 Amazon SNS 주제를 통해 알림이 SaaS 제품에 전송됩니다.
3. 추가 사용 요금에 대한 최종 측정 기록을 1시간 이내에 고객에게 보내야 합니다.
4. 취소가 진행 중이라는 알림을 제품에서 고객에게 보냅니다. 고객이 제품을 통해 취소하기를 원하는 경우 AWS Marketplace로 안내합니다. 나중에 요금이 부과되지 않도록 고객은 AWS Marketplace에 취소를 확인해야 합니다.

에서 SaaS 무료 평가판 제안 생성 AWS Marketplace

AWS Marketplace 판매자는 AWS Marketplace Management Portal (AMMP)에서 서비스형 소프트웨어(SaaS) 무료 평가판 제안을 생성할 수 있습니다. 고객은 SaaS 무료 평가판 옵션을 사용하여 소프트웨어 제품을 평가한 후 대규모 구매 결정을 내릴 수 있습니다. 고객이 제품을 구독하면 제품은 유료 고객과 동일한 방식으로 권한 검사를 수행합니다.

각는 SaaS 제품에 대해 한 번만 무료 평가판을 사용할 AWS 계정 수 있습니다. 무료 평가판 중에 부여된 무료 사용량은 AWS 조직의 연결된 계정 간에 공유되지 않습니다. 주 지급인 계정 하나의 서로 다른 연결 계정에서 각각 무료 평가판을 생성할 수 있습니다.

> **Note**
>
> - 판매자 데이터 전송 서비스(SDDS)를 사용하는 경우 Amazon Simple Storage Service 버킷에서 계약 세부 정보 평가판 보고서를 받게 됩니다. 이 보고서에는 구독자 이름 및 ID, 제안 ID, 계약 시작 및 종료 날짜와 같은 계약 세부 정보가 포함됩니다.
> - 또한 판매자는 새 구독이 생성될 때 Amazon Simple Notification Service(Amazon SNS) 알림을 받습니다. Amazon SNS 알림에는 무료 평가판 계약을 식별하는 `isFreeTrialTermPresent` 플래그가 포함됩니다.
> - 또한 무료 평가판을 구독하는 고객은 추가 토큰를 사용하여 등록 URL로 리디렉션됩니다`x-amzn-marketplace-offer-type=free-trial`. 토큰을 사용하여 무료 평가판을 사용하는 고객을 위한 고유한 등록 환경을 만들 수 있습니다.

SaaS 무료 평가판 제안 생성

판매자는 AWS Marketplace Management Portal (AMMP)에서 SaaS 무료 평가판 제안을 생성할 수 있습니다.

SaaS 무료 평가판 제안을 생성하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 에서 다음 중 하나를 AWS Marketplace Management Portal선택합니다.
   - 제안 생성 또는 관리
   - 제안 탭
3. 제안 페이지에서 공개 무료 평가판 탭을 선택하여 모든 SaaS 무료 평가판을 검토합니다.
4. 무료 평가판 제안 생성을 선택합니다. 판매자는 공개 SaaS 제품마다 SaaS 무료 평가판 제안을 하나씩 생성할 수 있습니다.
5. 제안 기본 사항에서 제품을 선택하고 다음을 선택합니다.
6. 무료 평가판 설정에서 다음을 수행합니다.
   a. 무료 평가판 기간(일)을 일 단위로 입력합니다.

무료 평가판 기간은 7~90일입니다.

b. 기존 공개 제안의 제품 차원을 확인합니다.

SaaS 구독 무료 평가판의 제품 차원은 변경할 수 없습니다.

SaaS 계약 무료 평가판의 차원별 수량 제한을 설정하고 차원을 제거 또는 추가할 수 있습니다.

7. 서비스 계약을 확인합니다.

EULA 버전의 경우 AWS Marketplace표준 계약 또는 사용자 지정 EULA를 선택하고 제안 검토를 선택하면 됩니다.

8. 제안에 대한 모든 정보를 확인 및 검토하고 제안 생성을 선택합니다.

SaaS 무료 평가판 제안 취소

판매자는 언제든지 AWS Marketplace Management Portal에서 무료 평가판 제안을 취소할 수 있습니다.

SaaS 무료 평가판 제안을 취소하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 에서 다음 중 하나를 AWS Marketplace Management Portal선택합니다.
   - 제안 생성 또는 관리
   - 제안 탭
3. 제안 페이지에서 제안을 선택합니다.
4. 혜택 보기를 선택합니다.
5. 제안 취소를 선택합니다.

제안이 취소된 후, 이 제안에 대한 활성 계약은 만료될 때까지 유효합니다. 취소된 제안에 대한 새 계약을 생성할 수 없습니다.

를 통해 SaaS 제품에 고객 온보딩 AWS Marketplace

서비스형 소프트웨어(SaaS) 구독 및 SaaS 계약을 통해 고객은를 통해 제품을 구독 AWS Marketplace 하지만 AWS 환경의 제품에 액세스합니다. 제품을 구독하면 고객은 SaaS 제품의 일부로 생성하고 관리하는 웹 사이트로 이동하여 계정을 등록하고 제품을 구성할 수 있습니다.

판매자는 SaaS 제품 리스팅을 생성할 때 등록 랜딩 페이지 URL을 제공합니다. 고객이 구독한 후 해당 URL을 사용하여 고객을 등록 랜딩 페이지로 리디렉션합니다. 소프트웨어의 등록 랜딩 페이지에서 고객의 계정을 생성하는 데 필요한 정보를 수집합니다. 사용량 알림 이메일을 통해 연락하려는 경우 고객의 이메일 주소를 수집하는 것이 좋습니다.

등록 랜딩 페이지는 결제를 위해 고객의 식별자 AWS Marketplace 를 사용하여의 양식 데이터에서 `x-amzn-marketplace-token` 토큰을 식별하고 수락할 수 있어야 합니다. 그런 다음 해당 토큰 값을 AWS Marketplace Metering Service 에 전달하여 고유한 고객 AWS 계정 ID, 고객 식별자 (2025년 12월 31일 이후 사용되지 않음) 및 해당 제품 코드를 확인해야 합니다. 코드에 대한 예는 `ResolveCustomer` 코드 예제 단원을 참조하세요.

> **Note**
>
> 등록 토큰은 특정 구독 고객으로 확인되며 생성된 각 토큰의 만료 기간은 4시간입니다. 호출자가 동일한 토큰으로 API를 호출하는 한, 토큰이 만료될 때까지 계속해서 동일한 응답 값이 반환됩니다.

새 구매자를 적용하도록 SaaS 제품 구성

신규 고객을 수락하고 이를 적절하게 측정하도록 SaaS 소프트웨어를 올바르게 구성할 책임이 있습니다. 다음 프로세스에서는 소프트웨어에 대한 신규 고객의 액세스를 식별, 구현 및 측정하기 위한 권장 방법 중 하나를 간략하게 설명합니다.

1. 고객이 AWS Marketplace 웹 사이트의 제품 페이지를 방문하면 제품을 구독하도록 선택합니다.
2. 고객의 AWS 계정 가 제품을 구독합니다. 즉, 제품에서 전송된 구독 및 측정 레코드가 고객 AWS 청구서의 일부가 됩니다.
3. ID, 고객 식별자 및 제품 코드가 포함된 AWS 계정 등록 토큰이 고객에 대해 생성됩니다.
4. 고객이 소프트웨어의 등록 랜딩 페이지로 리디렉션됩니다. 이 페이지는 고객의 식별자가 포함된 토큰을 수락할 수 있어야 합니다.
5. 고객의 브라우저가 판매자의 소프트웨어 등록 랜딩 페이지 URL로 `POST` 요청을 보냅니다. 이 요청에는 고객의 등록 토큰을 포함하고 있는 `x-amzn-marketplace-token` `POST` 파라미터 하나

가 들어 있습니다. 등록 웹 사이트의 관점에서 고객은 이 파라미터로 양식을 제출합니다. 등록 토큰은 불투명한 문자열입니다. 제안 유형이 무료 평가판이면 값이 `free-trial`인 두 번째 파라미터 `x-amzn-marketplace-offer-type`이 요청에 추가됩니다.

6. 이 등록 토큰을 고객 AWS 계정 ID, 고객 식별자 및 제품 코드로 사용하려면 웹 사이트가 AWS Marketplace 측정 서비스에서 ResolveCustomer를 호출해야 합니다. `ResolveCustomer` 호출의 예는 섹션을 참조하세요`ResolveCustomer` 코드 예제. 고객 식별자는 고객의 AWS 계정 ID가 아니지만 제품 간에 보편적이며 고객 레코드의 일부로 내부 소스에 저장해야 합니다. 제품 코드는가 AWS 제공하는 SaaS 제품의 고유한 문자열입니다. 각 AWS 제품에는 등록 중에 할당되는 고유한 제품 코드가 하나씩 있습니다.
7. 고객은 제품에서 계정을 생성하거나 기존 계정에 로그인하도록 안내를 받습니다.

> **Note**
>
> 제품에서 기존 고객 계정을 설정하거나 연결하려면 팀의 수동 프로세스가 필요한 경우 AWS에 문의 양식을 사용하여 고객의 연락처 정보를 수집할 수 있습니다. 연락처 정보를 수집하고 AWS 계정 ID 및 고유 고객 식별자(6단계에서 확인)를 확인한 후 고객에 대한 알림 메시지를 표시합니다. 알림에서 해당 계정이 설정 중임을 알리고 사용자에게 연락할 때까지 기다리도록 요청합니다. 고객에게 예상 소요 시간과 연락처 정보를 제공합니다. 또한 고객에게 동일한 세부 정보가 포함된 이메일 메시지를 보냅니다.

8. 이제 고객은 해당 SaaS 제품에 고유한 자격 증명을 사용하여 웹 사이트에 로그인됩니다. 계정 데이터베이스에 각 고객에 대한 항목이 있을 수 있습니다. 계정 데이터베이스에는 AWS 계정 ID 열이 있어야 합니다. 시스템의 다른 계정이 AWS 계정 ID를 공유하지 않는지 확인합니다.
9. 판매자 등록 프로세스 중에 판매자는 고객이 제품을 구독하거나 구독 해지하면 알려주는 Amazon SNS 주제를 구독합니다. 다음은 고객의 행동을 알려주는 JSON 형식의 Amazon SNS 알림입니다.

   - 권한 알림 - 계약이 포함된 요금 모델을 사용하는 제품의 경우 구매자가 새 계약을 생성하거나, 계약을 업그레이드하거나, 갱신하거나, 계약이 만료될 때 알림을 받습니다. 계정 데이터베이스에는 구독 상태에 대한 추가 열이 있어야 합니다. 자세한 내용은 Amazon SNS 주제: `aws-mp-entitlement-notification` 단원을 참조하세요.
   - 구독 알림 - 계약 및 구독을 비롯한 요금 모델이 적용되는 제품의 경우 구매자가 제품을 구독하거나 구독 해지할 때 알림을 받습니다. 자세한 내용은 Amazon SNS 주제: `aws-mp-subscription-notification` 단원을 참조하세요.

Amazon Simple Queue Service(Amazon SQS)를 사용하여 이러한 메시지를 캡처하는 것이 좋습니다. `subscribe-success`의 구독 알림을 수신하면 고객 계정이 측정할 준비가 됩니다. 이 알림 전에 전송한 레코드는 측정되지 않습니다. 수행 방법에 대한 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 2단계: Amazon SQS 대기열에 메시지를 전송하도록 Amazon SNS 주제에 권한 부여를 참조하세요.

> **Note**
>
> `subscribe-success` 알림을 받지 않는 한 제품 구독을 활성화하지 마세요.

10. 데이터베이스에 저장된 AWS 계정 ID를 사용하여 AWS Marketplace 측정 서비스를 통해 사용량을 측정하거나를 통해 권한을 확인합니다 AWS Marketplace Entitlement Service.

보안 및 주문

판매자는에서 즉시 반환되는 AWS 계정 IDs AWS 또는 시스템이 서명한 ID를 신뢰해야 합니다. 등록 토큰은 약 1시간 후에 만료될 수 있으므로 즉시 확인하는 것이 좋습니다. 등록 토큰을 확인한 후 등록이 완료될 때까지 ID를 고객의 브라우저 세션에 서명된 속성 AWS 계정 으로 저장합니다.

SaaS 제품에 대한 Amazon SNS 알림

알림을 받으려면 제품 생성 중에 AWS Marketplace 제공되는에 대한 Amazon Simple Notification Service(Amazon SNS) 주제를 구독해야 합니다. 이 주제는 고객의 제품 구독 및 계약 권한 변경 사항에 대한 알림을 제공합니다. 이를 통해 특정 고객에 대한 액세스가 제공 및 취소된 시기를 알 수 있습니다.

> **Note**
>
> 판매자는 제품 생성 프로세스 중에 SNS 주제에 대한 실제 Amazon 리소스 이름(ARN)을 받게 됩니다. 예: `arn:aws:sns:us-east-1:123456789012:aws-mp-subscription-notification-PRODUCTCODE`

서비스형 소프트웨어(SaaS) 제품에 제공되는 Amazon SNS 주제는 다음과 같습니다.

- Amazon SNS 주제: `aws-mp-entitlement-notification` - 이 주제는 구매자가 새 계약을 생성하거나, 기존 계약을 업그레이드 또는 갱신하거나, 계약이 만료될 때 알림을 제공합니다. 계약(SaaS

계약 및 소비가 포함된 SaaS 계약(초과분)이라고도 함)이 포함된 요금 모델을 사용하는 제품에만 제공됩니다.

- Amazon SNS 주제: `aws-mp-subscription-notification` - 이 주제는 구매자가 제품을 구독하거나 구독 해지할 때 알림을 제공하며, 비공개 제안의 `offer-identifier` 및 SaaS 무료 평가판의 무료 평가판 플래그를 포함하고 있습니다. 계약 및 구독(SaaS 구독, SaaS 계약 및 소비가 포함된 SaaS 계약(초과분)이라고도 함)을 포함한 모든 요금 모델에 제공됩니다.

이러한 알림에 응답하는 시나리오에 대한 자세한 내용은 다음 주제를 참조하세요.

- SaaS 구독 제품을와 통합 AWS Marketplace
- SaaS 계약 제품을와 통합 AWS Marketplace
- SaaS 계약과 pay-as-you-go제 제품 통합 AWS Marketplace

Amazon SNS 주제: **`aws-mp-entitlement-notification`**

`aws-mp-entitlement-notification` 주제의 각 메시지는 다음과 같은 형식입니다.

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
}
```

*`<action-name>`*은 항상 `entitlement-updated`입니다.

> **Note**
>
> - 권한 메시지의 경우 작업(신규, 업그레이드, 갱신 또는 만료)에 관계없이 메시지는 동일합니다. 업데이트 내용을 검색하려면 후속 `GetEntitlement` 호출이 필요합니다.
> - 소비가 포함된 SaaS 계약(초과분)의 경우 판매자에게 `aws-mp-subscription-notification` SNS 주제가 제공됩니다. 판매자가 초과 요금을 추가할 때 받게 되는 추가 알림입니다. 판매자는 `entitlement-updated`만 얻는 대신 신규 고객을 확보하면(모든 종류의 조치를 의미할 수 있음) 고객이 신규 고객이라는 것을 알리는 구독 메시지를 받게 됩니다.

- 미래 날짜의 계약(FDA)의 경우 이 주제는 계약 시작 날짜(계약 서명 날짜가 아님)에 시작됩니다. 또한 계약의 취소, 교체, 갱신 또는 만료와 같은 권한에 후속 변경 사항이 발생할 때 시작됩니다.

계약 가격이 적용되는 제품(사용한 만큼만 지불 요금이 적용되는 계약 포함)은 이러한 메시지에 응답해야 합니다. 응답 방법에 대한 자세한 내용은 시나리오: 사용자 구독 변경 사항 모니터링 섹션을 참조하세요.

Amazon SNS 주제: `aws-mp-subscription-notification`

`aws-mp-subscription-notification` 주제의 각 메시지는 다음과 같은 형식입니다.

```
{
    "action": "<action-name>",
    "customer-identifier": " X01EXAMPLEX",
    "product-code": "n0123EXAMPLEXXXXXXXXXXXX",
    "offer-identifier": "offer-abcexample123",
    "isFreeTrialTermPresent":"true"
}
```

`offer-identifier`는 작업이 `subscribe-success` 또는 인 경우에만 알림에 포함됩니다`subscribe-fail`. 작업이 `unsubscribe-pending` 또는 일 때는 알림에 포함되지 않습니다`unsubscribe-success`. 2024년 1월 이전에 생성된 제안의 경우이 식별자는 비공개 제안에 대한 알림에만 포함됩니다. 2024년 1월 이후에 생성된 제안의 경우이 식별자는 비공개 제안 및 공개 제안을 포함한 모든 제안에 대한 알림에 포함됩니다.

제안 유형에 대한 자세한 내용은 계약 갱신 대시보드에서 DescribeEntity API의 응답 또는 계약의 제안 가시성을 참조하세요. https://docs.aws.amazon.com/marketplace/latest/userguide/agreements-renewals-dashboard.html

> **Note**
>
> DescribeEntity API의 경우 해당 제안에 대한 대상 지정 규칙의 패싯을 대상으로 하는를 계정 AWS 계정 에서 찾으면 비공개 제안입니다. 해당 제안 AWS 계정 에 대한 대상 지정 규칙의 패싯을 대상으로 하는가 계정에 없는 경우 공개 제안입니다.

`isFreeTrialTermPresent` 속성 정보는 구매자의 구독이 무료 평가판인지 여부를 나타냅니다. 이 속성의 JSON 값은 부울 데이터 유형이 아닙니다. 그 대신 값이 문자열 데이터 유형으로 변환됩니다. 자세한 내용은 SaaS 무료 평가판을 참조하세요.

*<action-name>*은 알림에 따라 달라집니다. 가능한 작업은 다음과 같습니다.

- `subscribe-success` - 판매자가 측정 기록을 전송할 수 있게 되면 `subscribe-success` 메시지가 그 사실을 알려줍니다. 구매자가 계약 기반 제안을 수락하면 이 메시지가 새 `offer-identifier`와 함께 다시 전송됩니다.
- `subscribe-fail` - `subscribe-fail` 메시지가 생성되면 구매자가 이미에서 AWS Marketplace 판매자의 SaaS 랜딩 페이지로 전환했더라도 결제가 실패했을 수 있습니다. 판매자는 `subscribe-success` 메시지를 기다렸다가 제품 사용을 허용해야 합니다.
- `unsubscribe-pending` - 구매자가 구독을 해지하면 가장 먼저 `unsubscribe-pending` 메시지가 전송됩니다. 구독이 완전히 취소되기 전까지 제한된 시간(약 1시간) 동안 판매자가 최종 측정 기록을 받을 수 있음을 나타냅니다.
- `unsubscribe-success` - `unsubscribe-success`는 취소 완료를 알리는 메시지이며, 취소 완료 후에는 더 이상 측정 기록이 수락되지 않습니다.

> **Note**
>
> - 구매자가 구독을 해지한 후 최종 `unsubscribe-success` 메시지가 전송되기 전에 즉시 다시 구독하면 최종 `unsubscribe-success` 메시지가 전송되지 않고 `subscribe-success` 메시지가 대신 전송됩니다.
> - 미래 날짜의 계약(FDA)의 경우 `subscribe-success` 작업은 계약 시작 날짜(계약 서명 날짜가 아님)에 시작됩니다.

구독 요금이 적용되는 제품(사용한 만큼만 지불 요금이 적용되는 계약 포함)은 이러한 메시지에 응답해야 합니다. 응답 방법에 대한 자세한 내용은 다음 주제를 참조하세요.

- SaaS 구독 제품을와 통합 AWS Marketplace
- SaaS 계약과 pay-as-you-go제 제품 통합 AWS Marketplace

SNS 주제에 대한 SQS 대기열 구독

Amazon SQS 대기열을 제공된 SNS 주제에 구독하는 것이 좋습니다. SQS 대기열을 생성하고 대기열을 주제에 구독하는 방법에 대한 자세한 지침은 Amazon Simple Notification Service 개발자 안내서의 Amazon SQS 대기열을 Amazon SNS 주제에 구독을 참조하세요.

> **Note**
>
> 제품을 판매하는 데 AWS 계정 사용되는 에서만 AWS Marketplace SNS 주제를 구독할 수 있습니다. 하지만 메시지를 다른 계정에 전달할 수 있습니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 다른 계정의 Amazon SQS 대기열로 Amazon SNS 메시지 전송을 참조하세요.

알림을 위한 SQS 대기열 폴링

SQS 대기열을 SNS 주제에 구독하면 메시지가 SQS에 저장됩니다. 대기열을 지속적으로 폴링하는 서비스를 정의하고, 메시지를 찾고, 그에 따라 메시지를 처리해야 합니다.

AWS Marketplace 측정 및 권한 부여 서비스 API 액세스

이 섹션에서는 고객이 사용한 서비스형 소프트웨어(SaaS) 제품의 결제 및 보고가 정확한지 확인하는 데 사용되는 AWS Marketplace Metering Service 또는 AWS Marketplace Entitlement Service와 통합하는 프로세스를 간략하게 설명합니다. 제한된 상태로 게시된 SaaS 구독 제품 또는 SaaS 계약 제품을 제출했다고 가정합니다. 제한된 상태에서 테스트 계정을 사용하여 올바른 구성과 기능을 확인할 수 있지만 제품을 공개적으로 사용할 수는 없습니다.

> **Note**
>
> SaaS 제품이 다른 방식으로 측정(예: Amazon SageMaker Ground Truth 또는 AWS WAF)을 처리하는 다른 AWS 관리형 서비스와 통합되는 경우 AWS Marketplace 측정 서비스와 통합할 필요가 없습니다. 고객에게 요금이 이중으로 청구되지 않도록 제품 측정이 한 시스템에서만 이루어져야 합니다.

주제

- SaaS 구독에서 사용할 측정 구성

- 를 사용하여 권한 확인 AWS Marketplace Entitlement Service
- SaaS 제품 통합 체크리스트

자격 증명과 AWS CLI함께를 설정하는 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 구성을 AWS CLI 참조하세요. AWS Python SDK를 처음 사용하는 경우 Boto 3 Quickstart를 참조하세요.

SaaS 구독에서 사용할 측정 구성

서비스형 소프트웨어(SaaS) 구독의 경우 모든 사용량을 측정하면 사용자가 제공한 측정 레코드에 AWS 따라에서 고객에게 요금이 청구됩니다. SaaS 계약의 경우 고객의 계약 권한을 초과하여 사용량을 측정할 수 있습니다. 다음 섹션에서는 SaaS 제품과 함께 사용하도록 측정을 구성하는 방법에 대한 정보를 제공합니다.

애플리케이션이 고객의 사용량을 측정할 때 애플리케이션은 누적된 사용량 AWS 을 제공합니다. 애플리케이션에서는 전송된 GB 또는 지정된 시간에 스캔한 호스트와 같이 제품을 생성할 때 정의한 요금 차원을 측정합니다. 예를 들어 애플리케이션으로 전송되는 데이터의 양을 기준으로 요금을 청구하는 경우 데이터 양을 측정하고 해당 측정 레코드를 한 시간에 한 번 전송할 수 있습니다.는 제품을 생성할 때 제공한 요금과 함께 측정 데이터를 사용하여 고객의 청구서를 AWS 계산합니다.

> **Note**
>
> 원한다면 추적하는 모든 속성에 사용량을 분할할 수 있습니다. 이러한 속성은 구매자에게 태그로 노출됩니다. 이러한 태그를 통해 구매자는 태그 값에 따라 사용량으로 분할된 비용을 볼 수 있습니다. 예를 들어 사용자별로 요금을 부과하고 사용자에게 `Department` 속성이 있는 경우 키가 `Department`인 태그와 값마다 하나의 할당을 사용하여 사용량 할당을 생성할 수 있습니다. 이렇게 해도 보고하는 가격, 크기 또는 총 사용량이 변하지는 않지만, 고객은 제품에 적합한 범주별로 비용을 볼 수 있습니다. 자세한 내용은 공급업체 측정 태깅(선택 사항) 단원을 참조하세요.

주제

- 시간별로 측정
- 사용량을 측정하도록 제품 구성
- 공급업체 측정 태깅(선택 사항)

시간별로 측정

모든 고객에 대해 한 번에 최대 25개의 배치로 시간당 기준으로 사용량을 AWS 에 보고하는 것이 좋습니다. 이를 통해 고객은 사용량과 비용을 최대한 세밀하게 파악할 수 있습니다. 1시간을 초과하는 기간(예: 하루)의 사용량을 집계하는 경우 다음 고려 사항에 유의합니다.

- AWS 는 사용자의 측정 레코드 수신 시에만 제품 사용에 대한 요금을 고객에게 청구할 수 있습니다. 사용자는 제품의 측정 레코드가 성공적으로 전송 및 수신되었는지 확인할 책임이 있습니다. AWS CloudTrail 를 사용하여 전송하는 레코드가 정확한지 확인할 수 있습니다. 이 정보를 사용하여 시간 경과에 따라 감사를 수행할 수도 있습니다. 자세한 내용은 를 사용하여 AWS Marketplace 측정 API 호출 로깅 AWS CloudTrail 단원을 참조하세요.
- 요금 모델 '구독'('계약' 또는 '소비 계약' 요금 모델 아님)을 사용하는 SaaS인 경우 구매자는 언제든지 구독을 취소할 수 있습니다. 구매자가 이 구독 취소 작업을 시작하면 판매자는 `unsubscribe-pending` 알림을 받고 최종 `unsubscribe-success` 알림 전에 보고되지 않은 모든 사용량을 1시간 동안 보낼 수 있습니다. 두 번째 알림 이후의 모든 사항은 적용되지 않습니다. 다른 두 요금 모델은 구독 시간을 기준으로 기간이 설정되며 구매자는 구독 도중에 구독을 취소할 수 없습니다. 자동 갱신만 끌 수 있습니다. 자동 갱신하지 않으면 해당 기간이 끝날 때 동일한 알림이 전송됩니다.
- 시간당 측정 레코드를 전송하지 않고 애플리케이션 또는 네트워크 중단이 있는 경우 레코드가 더 지연됩니다. 구독이 만료된 후 애플리케이션 또는 네트워크 중단이 복원되면 보고되지 않은 사용량이 발생할 수 있습니다.
- 보고할 사용량이 없더라도 매시간 측정 레코드를 계속 전송하고 해당 시간에 대해 보고할 사용량이 없는 경우 수량이 0으로 기록될 수 있습니다. 구매자의 차원 사용량을 0 이상으로 보고하면 레코드를 수정할 수 없습니다. 따라서 이전 시간의 사용량을 보고하는 것이 좋습니다. 보고된 사용량은 결제 주기 동안 구매자의 AWS Billing and Cost Management 콘솔에 표시됩니다. 콘솔은 차원 가격이 단위당 0달러인 경우에도 해당 결제 주기 또는 지금까지 사용된 총 단위 수와 해당 항목의 총 비용과 함께 개별 항목에 있는 제품의 각 차원을 보여줍니다. 활성화된 경우 AWS 비용 및 사용량 보고서(CUR)는 단위당 요금을 포함하여 이 세부 정보를 표시합니다. 이러한 보고서는 실시간으로 업데이트되지 않습니다.
- 게시 중에 AWS Marketplace 운영 팀은 SaaS SaaS 애플리케이션이 제품을 게시하기 전에 측정 레코드를 성공적으로 전송하는지 테스트합니다. 일반적으로 팀은 SaaS의 모의 가입을 수행하고 측정 레코드가 수신되었는지 확인합니다.

> **Note**
>
> SaaS 제품이 다른 방식으로 측정을 처리하는 다른 AWS 관리형 서비스(예: Amazon SageMaker Ground Truth 또는 AWS WAF)와 통합되는 경우 AWS Marketplace 측정 서비스와

통합할 필요가 없습니다. 고객에게 요금이 이중으로 청구되지 않도록 제품 측정이 한 시스템에서만 이루어져야 합니다. AWS Marketplace 는 현재 새 AWS WAF 제품을 게시하지 않습니다.

사용량을 측정하도록 제품 구성

AWS Marketplace 측정 서비스의 `BatchMeterUsage` 작업을 사용하여 측정 레코드를 AWS에 전달할 수 있습니다. 다음 사항에 유의하세요:

- 판매자는 `BatchMeterUsage` 작업을 사용하여 일괄 처리를 사용해야 합니다.
- 정시에 측정 요청이 중복 제거됩니다.
  - 요청은 제품/고객/시간/차원별로 중복 제거됩니다.
  - 언제든지 요청을 다시 시도할 수 있지만 다른 수량을 측정하는 경우 원래 수량에 대해 요금이 청구됩니다.
  - 동일한 고객, 차원, 시간에 대해 여러 요청을 보내는 경우 기록이 집계되지 않습니다.
- 고객이 제품을 구독하는 경우 판매자는 최대 6시간의 과거 타임스탬프가 포함된 측정 기록을 보낼 수 있습니다. 고객이 구독 해지하면 판매자는 구독 해지 시점으로부터 1시간 이내에 측정 기록을 보내야 합니다.
- `BatchMeterUsage` 페이로드는 1MB를 초과하면 안 됩니다. 페이로드 크기를 초과하지 않도록 `BatchMeterUsage` 요청에 보낼 사용량 기록 수를 선택합니다.
- AWS Marketplace 측정 서비스는 AWS 일반 참조의 엔드포인트 및 할당량에 AWS 리전 나열된에서 사용할 수 있습니다. AWS Marketplace 기본적으로 미국 동부(버지니아 북부) 리전은 구매자가 제품을 요청할 때 SaaS 측정 제품에 사용할 수 있도록 활성화됩니다. 다른 리전을 사용하고 싶을 때는 AWS Marketplace 판매자 작업 팀에게 문의하세요. 자세한 내용은 BatchMeterUsage를 참조하세요.

코드 예제는 SaaS 제품 통합 코드 예제 섹션을 참조하세요.

예: 호스트 스캔

제품은 알려진 보안 취약성을 위해 컴퓨팅 하드웨어를 분석합니다. 고객은 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 스캔을 수동으로 시작하거나 예약합니다. 제품에서 이러한 스캔을 수행할 때 매 시간 스캔된 고유한 호스트 수의 총계를 냅니다. 이 예제에서 제품은 호스트 범주를 사용합니다. 스캔된 호스트 유형에 대해 여러 차원을 선언할 수 있습니다. 예를 들어 소형, 중간 및 대형 호스트에 각각 다른 요금을 부과할 수 있습니다.

예: 로그 분석

SaaS 제품은 고객 제품에서 생성된 로그, 보고 트렌드 및 이상을 이해합니다. 고객이 로그를 제품에 업로드할 때 수신된 데이터의 양(메가바이트, 기가바이트 또는 테라바이트)을 측정합니다. 매시간 10분마다 cron 작업은 이전 시간 동안 각 고객에 대한 이 사용량을 읽습니다. 해당 작업은 배치 보고서를 구축하고 `BatchMeterUsage` 작업을 사용하여 AWS로 전송합니다. 이 예시에서 제품은 데이터 범주를 사용합니다. 제품은 지정된 시간에 저장된 로그 데이터의 양도 측정할 수 있습니다. 이 경우, 제품은 해당 시간에 수신된 데이터와 해당 시간에 저장된 전체 데이터의 두 차원을 측정할 수 있습니다. 고객이 이 데이터를 삭제하거나 데이터가 만료될 때까지 저장된 데이터를 계속 측정할 수 있습니다.

공급업체 측정 태깅(선택 사항)

공급업체 측정 태깅을 사용하면 독립 소프트웨어 개발 판매 회사(ISV)는 구매자에게 소프트웨어 사용에 대한 보다 세밀한 통찰력을 제공하고 구매자의 비용 할당을 도와줄 수 있습니다.

구매자의 소프트웨어 사용에 태그를 지정하는 여러 가지 방법이 있습니다. 한 가지 방법은 먼저 구매자에게 비용 할당에서 보고 싶은 것이 무엇인지 물어보는 것입니다. 그 후 구매자 계정에 대해 추적하는 모든 속성에 사용량을 분할할 수 있습니다. 속성의 예로는 `Account ID`, `Business Unit`, `Cost Centers` 및 기타 제품 관련 메타데이터가 있습니다. 이러한 속성은 구매자에게 태그로 노출됩니다. 구매자는 태그를 사용하여 AWS 결제 콘솔(https://console.aws.amazon.com/costmanagement/)의 태그 값을 기준으로 사용량으로 분할된 비용을 볼 수 있습니다. 공급업체 측정 태깅은 판매자가 신고한 가격, 크기 또는 총 사용량을 변경하지 않습니다. 이를 통해 고객은 제품에 적합한 범주별로 비용을 볼 수 있습니다.

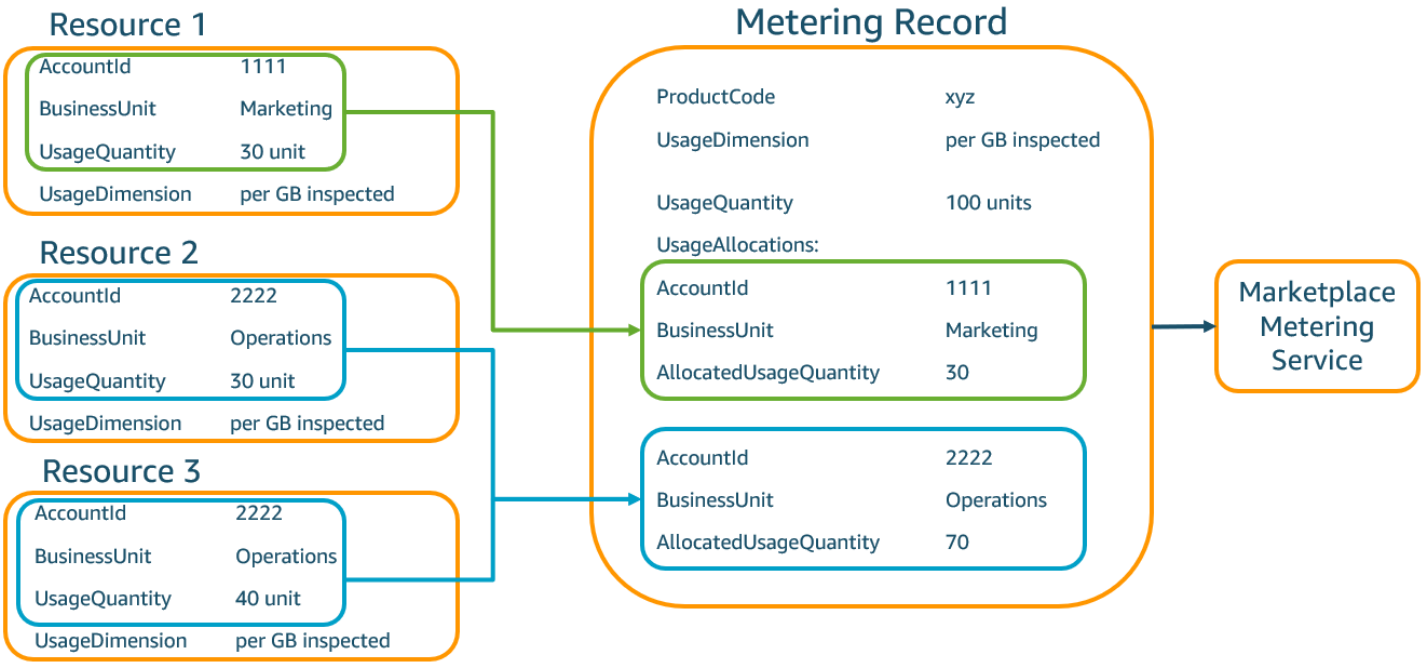
일반적으로 구매자는 AWS 계정하나로 제품을 구독합니다. 또한 구매자의 수많은 사용자가 동일한 제품 구독에 연결됩니다. `Account ID` 키가 있는 태그를 사용하여 사용량 할당을 생성한 다음, 각 사용자에게 사용량을 할당할 수 있습니다. 이 경우 구매자는 과금 정보 및 비용 관리 콘솔에서 `Account ID` 태그를 활성화하고 개별 사용자 사용량을 분석할 수 있습니다.

판매자 환경

판매자는 모든 리소스 사용량을 집계하는 대신 동일한 태그 세트를 사용하여 리소스 측정 기록을 집계할 수 있습니다. 예를 들어 판매자는 `UsageAllocations`의 여러 버킷을 포함하는 측정 기록을 구성할 수 있습니다. 각 버킷은 태그 세트의 `UsageQuantity`(예: `AccountId`, `BusinessUnit`)를 나타냅니다.

다음 다이어그램의 리소스 1은 고유한 `AccountId` 및 `BusinessUnit` 태그 세트가 있으며 Metering Record(측정 기록)에 단일 항목으로 표시됩니다.

리소스 2와 리소스 3은 동일한 `AccountId`, 2222, `BusinessUnit` 및 `Operations` 태그를 갖고 있습니다. 따라서 두 리소스는 측정 기록의 단일 `UsageAllocations` 항목으로 결합됩니다.



판매자는 태그가 없는 리소스를 단일 `UsageAllocation` 항목으로 결합하고 `UsageAllocations`의 항목 중 하나로 전송할 수도 있습니다.

다음과 같은 제한이 있습니다.

- 태그 수 - 5
- `UsageAllocations` 크기(카디널리티) - 2,500
- 최대 요청 크기 – 1MB

다음 사항을 확인합니다.

- 태그 키 및 값에 허용되는 문자 – a-zA-Z0-9+ -=._:\/@
- `UsageAllocation` 목록의 최대 태그 수 - 5
- 두 `UsageAllocations`의 태그가 같으면(즉, 동일한 태그 키와 값의 조합) 안 됩니다. 같을 경우 둘 다 동일한 `UsageAllocation`을 사용해야 합니다.
- `UsageAllocation`의 `AllocatedUsageQuantity` 합계는 총 사용량인 `UsageQuantity`와 같아야 합니다.
- 최대 페이로드 크기가 1MB를 초과하면 안 됩니다. 이 제한은 입력 속성 키(예: `UsageRecords`, `AllocatedUsageQuantity`, 태그)를 포함한 것입니다.

> **Note**
>
> 페이로드 제한을 초과하지 않도록 비즈니스 요구 사항에 따라 최대 크기의 샘플 요청 객체를 만들고, 해당 객체를 JSON 문자열로 변환하고, 크기(바이트)를 확보하세요. 단일 API 직접 호출이 1MB 제한을 넘지 않도록 하세요. 예를 들어 `UsageRecord`가 1개인 요청의 최대 크기가 200KB인 경우 `UsageRecords`를 최대 5개까지만 요청에 포함하여 전송하세요(200KB * 5 = 1MB).

구매자 경험

다음 표는 구매자가 `AccountId` 및 `BusinessUnit` 공급업체 태그를 활성화한 이후의 구매자 경험 예시를 보여줍니다.

이 예에서 구매자는 비용 사용 보고서에서 할당된 사용량을 볼 수 있습니다. 공급업체 측정 태그는 "`aws:marketplace:isv`" 접두사를 사용합니다. 구매자는 과금 정보 및 비용 관리의 비용 할당 태그 아래에 있는 AWS생성 비용 할당 태그에서 태그를 활성화할 수 있습니다.

비용 사용 보고서의 첫 번째 행과 마지막 행은 판매자 환경 예제처럼 판매자가 측정 서비스에 보내는 내용과 관련이 있습니다.

비용 사용 보고서(단순)

ProductCode	구매자	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	네트워크: 검사한 크기(GB)당	70	2222	운영
xyz	111122223333	네트워크: 검사한 크기(GB)당	30	3333	Finance
xyz	111122223333	네트워크: 검사한 크기(GB)당	20	4444	IT

ProductCode	구매자	UsageDimension	UsageQuantity	**aws:marketplace:isv:AccountId**	**aws:marketplace:isv:BusinessUnit**
xyz	111122223333	네트워크: 검사한 크기(GB)당	20	5555	마케팅
xyz	111122223333	네트워크: 검사한 크기(GB)당	30	1111	마케팅

코드에 대한 예는 사용량 할당 태그 지정을 사용하는 `BatchMeterUsage` 코드 예제(선택 사항) 단원을 참조하세요.

를 사용하여 권한 확인 AWS Marketplace Entitlement Service

제품이 SaaS 계약 제품인 경우 해당 제품은 AWS Marketplace Entitlement Service 를 호출하여 GetEntitlements API를 사용하는 고객의 권한을 검색합니다. 제품은 AWS Marketplace 권한 부여 서비스에 대해 해당 계정의 후속 사용을 확인해야 합니다. 예를 들어 고객이 계정에 10명의 사용자를 프로비저닝하는 경우 제품은 AWS Marketplace 권한 부여 서비스에 해당 용량에 대한 권한이 있는지 확인해야 합니다.

제품에 대한 고객의 권한을 확인하려면 AWS Marketplace 권한 부여 서비스에서 `GetEntitlements` 작업을 사용합니다. AWS Marketplace 권한 부여 서비스는 또는 `entitlement.marketplace.us-east-1.amazonaws.com`를 통해 액세스할 수 있는 미국 동부(버지니아 북부) 리전에서만 사용할 수 있습니다`entitlement-marketplace.us-east-1.api.aws`.

`GetEntitlements`는 고객 AWS 계정 ID, 고객 식별자 및 차원을 필터로 허용합니다. `ProductCode`는 필수 파라미터입니다. 해당 작업은 권한의 페이지 매김 목록을 반환합니다. 결과에는 권한이 유효한 최소 기간을 표시하는 `ExpirationDate` 필드가 있습니다. 고객이 자동 갱신을 설정한 경우 ExpirationDate 필드의 날짜가 갱신 날짜입니다.

코드 예제는 SaaS 제품 통합 코드 예제 섹션을 참조하세요.

사용자 작업에 대한 권한 부여 검색

다음 예제는 사용자 작업에 대한 권한 부여 검색 프로세스를 보다 정확하게 이해하는 데 도움이 될 수 있습니다.

예: 사용자 기반 제품

지정된 고객에 대해 다수의 계정을 허용하는 제품을 제공합니다. 고객은 대시보드를 방문하여 새 사용자를 프로비저닝(예: 자격 증명 할당)할 수 있습니다. 고객이 새 사용자를 프로비저닝하면 제품은 용량이 존재하는지 확인하기 위해 `GetEntitlements`를 호출합니다. 그렇지 않은 경우 AWS Marketplace 측정 서비스를 호출하여 추가 사용자에게 요금을 청구할 수 있습니다.

예: 데이터 스토리지 제품

고객이 암호화된 또는 암호화되지 않은 양식으로 특정 양의 데이터를 저장하도록 지원하는 제품을 제공합니다. 고객은 제품에 존재하고 할당된 데이터의 양을 표시하는 대시보드를 볼 수 있습니다. 대시보드는 `GetEntitlements`를 통해 할당량을 검색합니다.

SaaS 제품 통합 체크리스트

서비스형 소프트웨어(SaaS) 제품이 사용되기 전에 필요한 구성을 완료했는지 확인해야 합니다. 다음 SaaS 통합 체크리스트를 사용하여 필요한 구성을 완료했는지 확인할 수 있습니다.

카테고리	요구 사항
액세스	원하는 사용 AWS 계정으로 AWS Marketplace 판매자 등록 양식을 제출했습니다.
액세스	이용 약관, 은행 계좌, W8 또는 W9 세금 신고서를 포함하는 판매자 등록을 완료했습니다.
액세스	등록된 계정에 대해 구성된 교차 AWS Marketplace 계정 역할입니다.
Product	AWS Marketplace Management Portal에서 제품 요청 양식을 완성했습니다.
Product	AMMP의 제품 생성 마법사의 메모 탭에서 테스트할 AWS 계정 IDs를 제공했습니다.

카테고리	요구 사항
Product	제품 탭에 EULA의 URL을 .txt 형식으로 제공했습니다.
Product	에서 제품 코드 및 Amazon SNS 주제 정보를 수신했습니다 AWS Marketplace.
Product	Amazon SNS 주제를 구독하고 Amazon SNS 주제를 구독할 Amazon SQS 대기열을 생성했습니다.
결제 솔루션	각 고객에게 SaaS 구독 제품에 대해 1시간마다 `BatchMeterUsage` 작업의 측정 레코드를 전송할 수 있음을 검증했습니다. SaaS 계약 제품에 대해 각 고객의 추가 사용량과 관련한 측정 레코드를 전송할 수 있습니다.
결제 솔루션	검증된 SaaS SaaS 계약 제품에 대한 AWS Marketplace 권한 부여 서비스에서 고객 권한을 확인할 수 있습니다.
결제 솔루션	테스트 계정용으로 생성한 청구서에 예상대로 비용이 표시되는지 검증했습니다.
결제 솔루션	잘못된 고객 ID, 취소된 구독 등과 같은 상황을 테스트했습니다.
Product	게시를 AWS Marketplace 위해에 제품 요청을 다시 제출했습니다.
등록	HTTP POST 요청을 수락할 수 있는 HTTPS 등록 페이지를 구현했습니다.
등록	새 고객 등록을 수락할 수 있음을 검증했습니다.
등록	쿠키에 등록 토큰이 저장되지 않음을 검증했습니다.

카테고리	요구 사항
등록	AWS 토큰`CustomerIdentifier` 에서 `ProductCode` 및를 가져오는 `ResolveCustomer` 데 사용 중인 것으로 검증되었습니다.
등록	에서 수신한 등록 토큰을 지연 AWS 없이 확인할 수 있음을 검증했습니다.
등록	Gmail 같은 이메일 서비스 주소 등록이 차단되지 않는지 테스트했습니다.
등록	불완전한 등록 또는 여러 번의 등록 시도를 수락할 수 있는지 테스트했습니다.
구독	`unsubscribe-pending` 및 `unsubscribe-success` 메시지를 처리할 수 있음을 테스트합니다.
구독	`unsubscribe-pending` 메시지 수신 후 1 시간 이내에 최종 측정 레코드를 전송하는 것을 검증했습니다.
보안	AWS 루트 계정에 API 키가 없고, 강력한 암호가 있으며, 하드웨어 다중 인증(MFA) 디바이스와 연결되어 있는지 검증했습니다. 모든 관리 액세스는 AWS Identity and Access Management (IAM)로 생성된 자격 증명을 통해 이루어집니다. 공유 계정이 없습니다.
보안	IAM 역할이 모든 프로그래밍 방식 Amazon Elastic Compute Cloud(Amazon EC2) 액세스에 사용되는지 확인했습니다. 스크립트, 헤더 또는 소스 코드에 자격 증명 하드 코딩이 불가능합니다.
보안	포괄적 로깅 및 로그 통합을 유지하는 것을 검증했습니다.

카테고리	요구 사항
보안	애플리케이션 서비스, 데이터베이스 및 파일 시스템 액세스를 격리하는 퍼블릭 및 프라이빗 서브넷 경계가 명확하게 정의되어 있음을 확인했습니다. 민감한 데이터를 구분하고 퍼블릭 및 프라이빗 데이터를 분리하는 데이터 등급 정의가 명확합니다.
보안	키 교체가 예약되어 있는 전송 중 및 미사용 프라이빗 데이터 암호화를 확인했습니다.
보안	보안 인시던트 도구 및 액세스가 마련되어 있으며, 인시던트 대응 연습이 정기적으로 예약되어 있어 적기 조사 및 복구가 가능함을 검증했습니다.
신뢰성	시스템이 필요에 따라 스케일업과 스케일 다운 변화에 맞게 조정되며, 로드 밸런싱을 사용하여 높은 성능을 보장하는 것을 확인했습니다. 시스템이 필요에 따라 엣지 기반 캐싱도 제공합니다.
신뢰성	복구 시간 및 지점 목표가 명시되어 있고, 재해 복구가 정기적으로 예약되어 있음을 검증했습니다. 자동 트리거 및 알림을 통해 구성요소 장애를 자체적으로 복구합니다.

에서 SaaS 제품 보고 AWS Marketplace

AWS Marketplace 는 구독자, 금융, 사용량 및 세금에 대한 데이터가 포함된 서비스형 소프트웨어 (SaaS) 제품에 대한 보고서를 생성합니다. 다음 표에 사용 가능한 보고서에 대한 정보가 나와 있습니다. 보고서를 사용하여 고객 기반에 대한 더 많은 인사이트를 얻고 판매 및 세금을 포함한 재무 상태를 더 잘 이해할 수 있습니다. 자세한 내용은 the section called “판매자 보고서” 단원을 참조하세요. 다음 표에는 SaaS 제품의 재무에 대해 보고하는 방법이 요약되어 있습니다.

보고서	SaaS 콘텐츠
일별 비즈니스 보고서	선결제 계약 요금은 Fees(수수료) 섹션에 나타납니다. 과금된 사용 요금은 Usage(사용량) 섹션에 나타납니다.
월별 수익 보고서	선결제 계약 요금은 연간 구독 섹션에 나타납니다. 측정된 사용 요금은 결제 및 수익 데이터 섹션에 나타납니다.
영업 보상 보고서	선결제 계약 요금 및 월간 추가 사용 요금은 별도의 개별 항목으로 표시됩니다.
고객 구독자 보고서	새 SaaS 계약은 연간 구독 섹션에 나타납니다. 새 SaaS 구독은 시간별/월별 구독 섹션에 나타납니다.

SaaS 제품 통합 코드 예제

다음 코드 예제를 사용하여 서비스형 소프트웨어(SaaS) 제품을 제품 게시 및 유지 관리에 필요한 AWS Marketplace APIs와 통합할 수 있습니다. 자세한 내용은 다음 섹션을 참조하세요.

주제

- ResolveCustomer 코드 예제
- GetEntitlement 코드 예제
- BatchMeterUsage 코드 예제
- 사용량 할당 태그 지정을 사용하는 BatchMeterUsage 코드 예제(선택 사항)

`ResolveCustomer` 코드 예제

다음 코드 예제는 모든 요금 모델과 관련이 있습니다. Python 예제는 `CustomerIdentifier`, `ProductCode` 및 `CustomerAWSAccountId`에 대한 `x-amzn-marketplace-token` 토큰을 교환합

니다. `CustomerAWSAccountId`는 구독과 연결된 AWS 계정 ID입니다. 이 코드는 AWS Marketplace Management Portal에서 리디렉션될 때 등록 웹 사이트의 애플리케이션에서 실행됩니다. 리디렉션은 토큰을 포함하는 POST 요청입니다.

`ResolveCustomer`에 대한 자세한 내용은 AWS Marketplace 측정 서비스 API 참조의 ResolveCustomer를 참조하세요.

```
# 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
```

응답의 예

```
{
    'CustomerIdentifier': 'string',
    'CustomerAWSAccountId':'string',
    'ProductCode': 'string'
}
```

`GetEntitlement` 코드 예제

다음 코드 예제는 계약이 포함된 SaaS 제품과 소비 요금 모델이 포함된 SaaS 계약과 관련이 있습니다. Python 예제는 고객에게 활성 권한이 있는지 확인합니다.

`GetEntitlement`에 대한 자세한 내용은 AWS Marketplace 권한 부여 서비스 API 참조의 GetEntitlement를 참조하세요.

```
# Import AWS Python SDK
import boto3

marketplaceClient = boto3.client('marketplace-entitlement', region_name='us-east-1')

# Filter entitlements for a specific customerID
#
# productCode is supplied after the AWS Marketplace Ops team has published
# the product to limited
#
# customerID is obtained from the ResolveCustomer response
entitlement = marketplaceClient.get_entitlements({
    'ProductCode': 'productCode',
    'Filter' : {
        # Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
        'CUSTOMER_IDENTIFIER': [
            'customerID',
        ]
        # Option 2: Using CustomerAWSAccountID (preferred)
        # 'CUSTOMER_AWS_ACCOUNT_ID': [
        #     'awsAccountID',
        # ]
    },
    'NextToken' : 'string',
    'MaxResults': 123
})

# TODO: Verify the dimension a customer is subscribed to and the quantity,
# if applicable
```

응답의 예

반환된 값은 AWS Marketplace Management Portal에서 제품을 생성할 때 생성된 차원과 일치합니다.

```
{
   "Entitlements": [
      {
         "CustomerIdentifier": "string",
         "CustomerAWSAccountID": "string",
         "Dimension": "string",
         "ExpirationDate": number,
         "ProductCode": "string",
         "Value": {
```

```
            "BooleanValue": boolean,
            "DoubleValue": number,
            "IntegerValue": number,
            "StringValue": "string"
         }
      }
   ],
   "NextToken": "string"
}
```

BatchMeterUsage 코드 예제

다음 코드 예제는 소비 요금 모델이 포함된 SaaS 구독 및 계약과는 관련이 있지만, 소비가 포함되지 않은 SaaS 계약 제품과는 관련이 없습니다. Python 예제에서는 측정 레코드를에 전송 AWS Marketplace 하여 고객에게 pay-as-you-go 요금을 청구합니다.

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited
#
# You can use either:
# - customerID from the ResolveCustomer response (deprecated after Dec 31, 2025)
# - AWS account ID of the buyer

# Import AWS Python SDK
import boto3
from datetime import datetime

# Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
usageRecord = [
    {
        'Timestamp': datetime(2015, 1, 1),
        'CustomerIdentifier': 'customerID',
        'Dimension': 'string',
        'Quantity': 123
    }
]

# Option 2: Using CustomerAWSAccountID (preferred)
# usageRecord = [
```

```
#     {
#         'Timestamp': datetime(2015, 1, 1),
#         'CustomerAWSAccountID': 'awsAccountID',
#         'Dimension': 'string',
#         'Quantity': 123
#     }
# ]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(
    UsageRecords=usageRecord,
    ProductCode='productCode'
)
```

`BatchMeterUsage`에 대한 자세한 내용은 AWS Marketplace 측정 서비스 API 참조의 BatchMeterUsage를 참조하세요.

응답의 예

```
{
    'Results': [
        {
            'UsageRecord': {
                'Timestamp': datetime(2015, 1, 1),
                'CustomerIdentifier': 'string',
                'CustomerAWSAccountID': 'string',
                'Dimension': 'string',
                'Quantity': 123
            },
            'MeteringRecordId': 'string',
            'Status': 'Success' | 'CustomerNotSubscribed' | 'DuplicateRecord'
        },
    ],
    'UnprocessedRecords': [
        {
            'Timestamp': datetime(2015, 1, 1),
            'CustomerIdentifier': 'string',
            'CustomerAWSAccountID': 'string',
            'Dimension': 'string',
            'Quantity': 123
        }
    ]
```

```
}
```

사용량 할당 태그 지정을 사용하는 **BatchMeterUsage** 코드 예제(선택 사항)

다음 코드 예제는 사용 요금 모델이 있는 SaaS 구독 및 계약과 관련이 있지만, 사용하지 않는 SaaS 계약 제품은 관련이 없습니다. Python 예제에서는 적절한 사용량 할당 태그가 달린 측정 레코드를 AWS Marketplace 에 전송하여 고객에게 사용한 만큼만 지불 요금을 청구합니다.

```
# NOTE: Your application will need to aggregate usage for the
#       customer for the hour and set the quantity as seen below.
# AWS Marketplace can only accept records for up to an hour in the past.
#
# productCode is supplied after the AWS Marketplace Ops team has
# published the product to limited
#
# You can use either:
# - customerID from the ResolveCustomer response (deprecated after Dec 31, 2025)
# - AWS account ID of the buyer

# Import AWS Python SDK
import boto3
import time

# Option 1: Using CustomerIdentifier (deprecated after Dec 31, 2025)
usageRecords = [
    {
        "Timestamp": int(time.time()),
        "CustomerIdentifier": "customerID",
        "Dimension": "Dimension1",
        "Quantity": 3,
        "UsageAllocations": [
            {
                "AllocatedUsageQuantity": 2,
                "Tags": [
                    { "Key": "BusinessUnit", "Value": "IT" },
                    { "Key": "AccountId", "Value": "*********" },
                ]
            },
            {
                "AllocatedUsageQuantity": 1,
                "Tags": [
```

```
                    { "Key": "BusinessUnit", "Value": "Finance" },
                    { "Key": "AccountId", "Value": "*********" },
                ]
            },
        ]
    }
]

# Option 2: Using CustomerAWSAccountID (preferred)
# usageRecords = [
#     {
#         "Timestamp": int(time.time()),
#         "CustomerAWSAccountID": "awsAccountID",
#         "Dimension": "Dimension1",
#         "Quantity": 3,
#         "UsageAllocations": [
#             {
#                 "AllocatedUsageQuantity": 2,
#                 "Tags": [
#                     { "Key": "BusinessUnit", "Value": "IT" },
#                     { "Key": "AccountId", "Value": "*********" },
#                 ]
#             },
#             {
#                 "AllocatedUsageQuantity": 1,
#                 "Tags": [
#                     { "Key": "BusinessUnit", "Value": "Finance" },
#                     { "Key": "AccountId", "Value": "*********" },
#                 ]
#             },
#         ]
#     }
# ]

marketplaceClient = boto3.client('meteringmarketplace')

response = marketplaceClient.batch_meter_usage(
    UsageRecords=usageRecords,
    ProductCode="testProduct"
)
```

`BatchMeterUsage`에 대한 자세한 내용은 AWS Marketplace Metering Service API 참조의 BatchMeterUsage를 참조하세요.

응답의 예

```
{
    "Results": [
        {
            "Timestamp": "1634691015",
            "CustomerIdentifier": "customerID",
            "CustomerAWSAccountID": "awsAccountID",
            "Dimension": "Dimension1",
            "Quantity": 3,
            "UsageAllocations": [
                {
                    "AllocatedUsageQuantity": 2,
                    "Tags": [
                        { "Key": "BusinessUnit", "Value": "IT" },
                        { "Key": "AccountId", "Value": "*********" }
                    ]
                },
                {
                    "AllocatedUsageQuantity": 1,
                    "Tags": [
                        { "Key": "BusinessUnit", "Value": "Finance" },
                        { "Key": "AccountId", "Value": "*********" }
                    ]
                }
            ],
            "MeteringRecordId": "8fjef98ejf",
            "Status": "Success"
        }
    ],
    "UnprocessedRecords": [
        {
            "Timestamp": "1634691015",
            "CustomerIdentifier": "customerID",
            "CustomerAWSAccountID": "awsAccountID",
            "Dimension": "Dimension1",
            "Quantity": 3,
            "UsageAllocations": []
        }
    ]
}
```

를 사용하여 Amazon VPC를 통해 제품 제공 AWS PrivateLink

AWS Marketplace AWS PrivateLink는 Amazon 네트워크를 사용하여 구매자에게 판매하는 제품에 대한 액세스 권한을 제공할 수 AWS 서비스 있는를 지원합니다 AWS Marketplace. 다음 섹션에서는 AWS PrivateLink 기술을 사용하여 Amazon Virtual Private Cloud(VPC) 엔드포인트를 통해 제품을 구성하고 제공하는 프로세스를 간략하게 설명합니다.

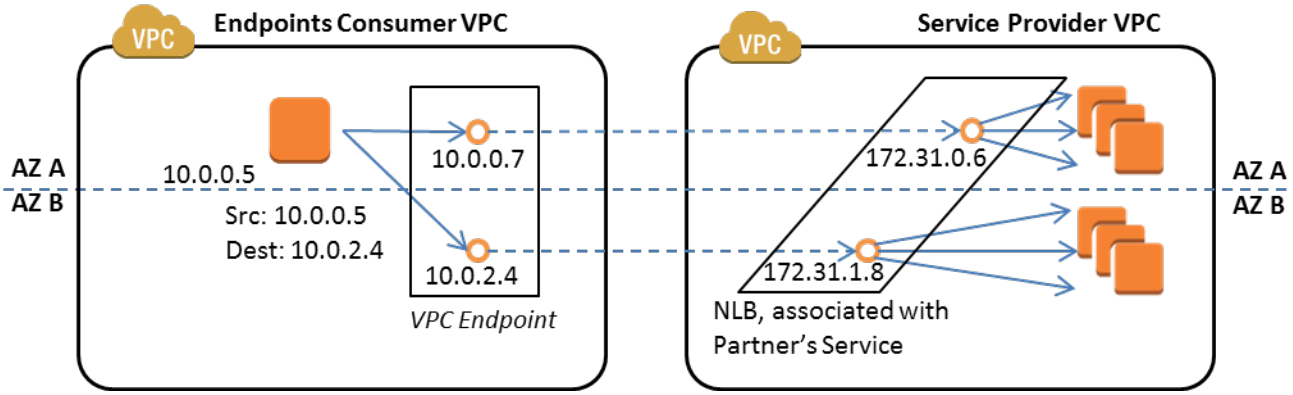
이 주제에서는 여러 AWS 서비스와 AWS Marketplace 환경에 대한 실무 지식이 있다고 가정합니다.

주제

- 소개
- 제품 구성
- 에 제품 제출 AWS Marketplace
- VPC 엔드포인트에 대한 구매자 액세스
- 부록: 체크리스트

소개

AWS Marketplace 판매자는 구매자에게 Amazon VPC 엔드포인트를 통해 서비스에 대한 액세스 권한을 제공할 수 있습니다. 이러한 접근 방식을 통해 구매자는 AWS PrivateLink 기술을 사용하여 Amazon 네트워크에서 귀사의 서비스에 액세스할 수 있습니다. AWS Marketplace 를 사용하여이 상품을 생성하고 제공하는 경우 구매자는에서 서비스를 검색할 수 있습니다 AWS Marketplace. VPC 엔드포인트 생성에 사용할 수 있는 서비스 목록에서도 구매자가 귀사의 제품을 검색할 수 있습니다.



VPC 엔드포인트는 AWS 고객이 인터넷, NAT 디바이스, VPN 연결 또는를 통해 액세스할 필요 없이 VPC와 다른 AWS 서비스 간에 프라이빗 연결을 생성할 수 있는 가상 디바이스입니다 AWS Direct

Connect. 를 통해 구매자가이 기술을 사용하여 서비스에 연결할 수 AWS Marketplace 있는 엔드포인트 서비스를 생성할 수 있습니다. 구매자가 인터넷보다는 Amazon 프라이빗 네트워크를 통해 귀사의 서비스에 액세스하기 때문에 이러한 연결 방법이 구매자에게 더욱 안전합니다.

서비스를 제공할 각 리전마다 리소스를 생성하거나 기존 리소스를 사용하여 VPC를 구성하고, 서비스 인스턴스를 설정하고, 네트워크 로드 밸런서를 설정하고, 서비스 엔드포인트를 생성하여 네트워크 로드 밸런서를 통해 서비스를 등록합니다. 이러한 단계를 완료하고 상품을 테스트한 후 AWS Marketplace 판매자 작업 팀에 구성 정보를 제공합니다.

AWS 에서는 구매자가 VPC 엔드포인트를 생성할 때 사용할 수 있는 프라이빗 DNS 이름을 제공할 것을 권장합니다.

구매자가 VPC 엔드포인트를 생성할 때 프라이빗 DNS 이름을 활성화하는 옵션이 있습니다. 이 옵션을 선택하면 구매자의 VPC 서비스에서 프라이빗 호스팅 영역이 구성됩니다. 프라이빗 DNS 이름을 제공하면 구매자는 VPC 엔드포인트 구성 시 이를 사용하여 귀사의 서비스에 연결할 수 있습니다. 구매자의 프라이빗 호스팅 영역에서 프라이빗 DNS 이름(api.example.com)은 엔드포인트 서비스용으로 생성되는 무작위 생성 DNS 이름(vpce-11111111111111111-yyyyyyyy.api.vpce.example.com)을 가리키게 됩니다. 구매자의 EC2 인스턴스는 서로 다른 VPC에서 동일한 통합 DNS 이름(api.example.com)을 호출합니다. 그리고 퍼블릭 및 프라이빗 DNS 이름이 동일할 경우 구매자는 VPC 내부 또는 외부에서 귀사의 서비스에 액세스할 때 동일한 퍼블릭 이름을 사용할 수 있습니다.

를 통해 서비스를 제공하는 데 도움이 필요하면 AWS Marketplace 판매자 운영 팀에 문의 AWS Marketplace하세요. AWS Marketplace 구매자가 서비스를 구독하고 VPC 엔드포인트를 생성하면 서비스가 AWS Marketplace 서비스 아래에 표시됩니다. AWS Marketplace 판매자 운영 팀은 VPC 엔드포인트를 생성할 때 서비스를 쉽게 검색할 수 있도록 사용자에게 친숙한 DNS 이름을 사용합니다.

제품은 SaaS(Software-as-a-Service) 제품으로 생성됩니다. 측정 및 결제는 다른 AWS Marketplace SaaS 제품과 동일합니다.

제품 구성

Amazon VPC 엔드포인트를 통해 사용할 수 있도록 제품을 구성하려면

1. 새로 생성하거나 기존 Amazon VPC를 사용합니다.
2. 제품용 Amazon EC2 인스턴스를 생성합니다(또는 기존 것을 사용합니다).
3. 제품을 제공하는 각 리전에 네트워크 로드 밸런서를 생성합니다. AWS 에서는 리전에 대한 모든 가용 영역(AZ)을 포함할 것을 권장합니다.
4. Amazon VPC 콘솔, CLI 또는 지원되는 SDK를 사용하여 VPC 엔드포인트 서비스를 생성합니다.

5. 네트워크 로드 밸런서를 통해 서비스에 액세스할 수 있는지 확인합니다.
6. 사용자에게 친숙한 DNS 이름에 대한 인증서를 AWS Certificate Manager (ACM)에 요청합니다. ACM은 인증서 요청 시 인증서를 발급하기 전에 귀사가 도메인 이름을 소유하거나 관리 권한을 보유하고 있는지 검증합니다.
7. api.vpce.example.com 같은 사용자에게 친숙한 DNS 이름의 하위 도메인을 AWS Marketplace 판매자 운영 팀이 제공한 이름 서버에 위임합니다. DNS 시스템에서 이름 서버(NS) 리소스 레코드를 생성하여이 하위 도메인이 AWS Marketplace 판매자 작업 팀이 제공하는 Amazon Route 53 이름 서버를 가리키도록 해야 DNS 이름(예: vpce-0ac6c347a78c90f8.api.vpce.example.com)을 공개적으로 확인할 수 있습니다.
8. 구매자의 AWS 계정에 대한 액세스를 허용합니다.

   참고: 지원되는 SDK 또는 이 CLI 명령을 사용하여 계정에 대한 액세스를 자동화할 수 있습니다(aws vpcev2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-0123456789abcdef1 --add-allowed-principals arn:aws:iam::111111111111:root arn:aws:iam::222222222222:root).

에 제품 제출 AWS Marketplace

서비스를에 게시하는 과정에서 AWS Marketplace 판매자 운영 팀과 AWS Marketplace협력합니다. PrivateLink 지원 제품을 제출하려면

1. AWS Marketplace 판매자 작업 팀에 다음 정보를 이메일로 보냅니다.
   a. 엔드포인트를 생성하는 데 사용되는 엔드포인트 및 AWS 계정입니다. 엔드포인트는 com.amazonaws.vpce.us-east-1.vpce-svc-0daa010345a21646과 유사합니다.
   b. 사용자에게 친숙한 서비스 DNS 이름. AWS Marketplace 구매자가 제품에 액세스하는 데 사용하는 DNS 이름입니다.
   c. 인증서를 요청하는 데 사용한 AWS 계정과 구매자가 VPC 엔드포인트에 액세스하는 데 사용하는 프라이빗 DNS 이름입니다.

      AWS Marketplace 판매자 운영 팀은 등록 중인 서비스에 사용할 회사의 자격 증명과 DNS 이름을 확인합니다(예: api.vpce.example.com). 확인 후, DNS 이름은 기본 엔드포인트 DNS 이름을 재정의합니다.

VPC 엔드포인트에 대한 구매자 액세스

AWS Marketplace VPC 엔드포인트를 생성하는 구매자는 다음과 같은 상황에서 서비스를 검색할 수 있습니다.

- 이 페이지의 앞부분에서 설명한 판매자 프로세스를 수행하여 제품을 생성하거나 기존 제품을 사용한 경우
- 구매자가 귀사의 서비스를 구독하는 경우
- 허용된 AWS 계정 목록에 구매자의 계정을 추가했습니다.

구매자가 VPC 엔드포인트를 생성할 때 프라이빗 호스팅 영역을 구매자의 VPC와 연결하는 옵션이 있습니다. 호스팅 영역에 고객의 VPC에 있는 엔드포인트 네트워크 인터페이스의 프라이빗 IP 주소를 확인하는 서비스의 기본 프라이빗 DNS 이름으로 설정된 레코드 세트가 있습니다.

AWS Marketplace 서비스를 포함하여 구매자 호스팅 엔드포인트는 모든 계정에 권한을 제공할 수 있습니다("*" 권한). 단, 이 접근 방식을 사용할 때는 서비스 이름으로 검색할 경우에만 Describe(설명) 호출 또는 콘솔에 서비스가 포함됩니다. 호출 설명에 서비스를 표시하려면 서비스에서 구매자의 AWS 계정을 허용 목록에 명시적으로 추가해야 합니다.

구매자는 다음을 수행하여 귀사의 서비스에 액세스할 수 있습니다.

1. 에서 서비스를 검색하고 구독합니다 AWS Marketplace.
2. AWS Command Line Interface (AWS CLI), API 또는 Amazon VPC 콘솔을 사용하여 서비스를 검색한 다음 VPC 엔드포인트를 설정하여 사용하는 서브넷 및 AZs에서 서비스에 연결합니다. 엔드포인트가 서브넷에서 탄력적 네트워크 인터페이스로 표시됩니다. 로컬 IP 주소와 리전 및 영역 DNS 이름이 엔드포인트에 할당됩니다.

클라이언트 측 DNS 이름	이름
리전	Vpce<0dc9a211a78c90f8>.api.vpce.example.com
IAD2(1a)	us-east-1a-Vpce<0dc9a211a78c90f8>.api.vpce.example.com
IAD2(1b)	us-east-1b-Vpce<0dc9a211a78c90f8>.api.vpce.example.com

기본 프라이빗 DNS 이름이 제공된 상태에서 구매자가 VPC 엔드포인트를 생성할 때 (프라이빗 호스팅 영역과 연결된) 프라이빗 DNS 이름 활성화를 선택하면 판매자의 서비스에 연결할 리전의 기본 프라이빗 DNS 이름이 구매자에게 표시됩니다.

이름	별칭	별칭 호스팅 영역 ID	(참고)
api.example.com	vpce<0dc9a211a78c90f8>. api.vpce.example.com	Z00AABBCCDD	IAD1 IAD2

부록: 체크리스트

AWS Marketplace 판매자 운영 팀에 제품을 제출하기 전에 다음 체크리스트를 사용하여 제품을 구성하고 테스트해야 합니다.

제품 생성 체크리스트

- VPC를 생성한 다음(또는 기존 것을 사용) 구성합니다.
- VPC 안에 네트워크 로드 밸런서를 생성하고 구성합니다.
- VPC 엔드포인트 서비스를 생성할 때 네트워크 로드 밸런서로 서비스를 등록합니다.
- VPC 엔드포인트를 구성하는 데 사용한 AWS 계정 ID를 AWS Marketplace 판매자 운영 팀에 제공합니다.
- AWS Marketplace 판매자 운영 팀에 기본 엔드포인트 서비스 이름(예: com.amazonaws.vpce.us-east-1.vpce-svc-0bbb070044a2164)을 제공합니다.
- 무작위로 생성되는 서비스 DNS 이름에 우선할 사용자에게 친숙한 DNS 이름(필수)을 제공합니다. 사용자에 친숙한 서비스 DNS 이름에 사용되는 하위 도메인의 SSL 인증서를 ACM에 요청합니다. 이러한 인증서와 이를 요청하는 데 사용한 AWS 계정 ID를 AWS Marketplace 판매자 운영 팀에 제공합니다.
- 권장: 프라이빗 DNS 이름을 제공합니다.
- AWS Marketplace 구매자가 AWS PrivateLink 기술을 사용하여 서비스에 연결할 수 있는 옵션을 알리고 허용하는 프로세스를 생성합니다. 허용되는 AWS 계정 목록에 구매자의 계정 IDs를 추가합니다.

제품 테스트

- 귀사의 서비스가 구성되어 검색 가능한지 확인합니다.
- 네트워크 로드 밸런서를 통해 귀사의 서비스를 검색할 수 있는지 확인합니다.

- 구매자가 VPC 엔드포인트를 생성하여 귀사의 서비스에 액세스할 수 있는지 확인합니다. 서비스를 설정하는 데 사용한 AWS 계정이 아닌 소유한 계정을 사용합니다.

의 전문 서비스 제품 AWS Marketplace

판매자는 AWS Marketplace 구매자에게 전문 서비스를 제공할 수 있습니다. 전문 서비스는 AWS 서비스 및 AWS Marketplace의 제품을 평가, 마이그레이션, 지원, 관리하고 다른 사람들에게 사용 방법을 교육하는 서비스입니다. 판매자는 제공하는 서비스에 대해 설명하는 제품을 생성하고, 고객과 협상하여 조건에 대한 계약을 작성한 다음, AWS Marketplace를 통해 서비스에 대한 사용자 지정 제안을 생성합니다.

> **Note**
>
> 독립 소프트웨어 개발 판매 회사(ISV), 채널 파트너 또는 컨설팅 파트너는 채널 파트너 비공개 제안을 사용하여 전문 서비스 제품을 재판매할 수 있는 권한을 다른 파트너에게 부여할 수도 있습니다. 자세한 내용은 ISV로서 채널 파트너를 위한 재판매 기회 만들기를 참조하세요.

구매자는 범주에서 전문 서비스를 선택하고, 제공 방법에서 전문 서비스를 선택하고, 게시자, 요금 모델 및 요금 단위별로 검색을 구체화하여 AWS Marketplace 카탈로그에서 전문 서비스 제품을 찾을 수 있습니다. AWS 청구서의 서비스에 대한 요금이 청구됩니다. 와 같은 도구를 사용하여 결제 AWS Cost Explorer 를 중앙 집중화하고 비용을 관리할 수 있습니다.

전문 서비스 제품에 대한 자세한 내용은 다음을 참조하세요.

- 전문 서비스 제품 시작
- 전문 서비스 제품에 대한 세부 정보 제공
- 전문 서비스 제품에 대한 요구 사항
- ISV로서 채널 파트너를 위한 재판매 기회 만들기

다음 비디오에서는 AWS Marketplace에서 전문 서비스 제품을 관리하는 방법에 대해 설명합니다.

에서 전문 서비스 제품을 관리합니다 AWS Marketplace.

도움말 가져오기

전문 서비스 제품에 대한 도움이 필요하면의 비즈니스 개발 파트너 AWS Marketplace 또는 AWS Marketplace 판매자 운영 팀에 문의하십시오.

에서 전문 서비스 제품 시작하기 AWS Marketplace

판매자는 AWS Marketplace 구매자에게 전문 서비스를 제공할 수 있습니다. 전문 서비스는 AWS 서비스 및 AWS Marketplace의 제품을 평가, 마이그레이션, 지원, 관리하고 다른 사람들에게 사용 방법을 교육하는 서비스입니다. 제품 정의는 구매자에게 제공하는 서비스에 대해 알려주고, 고객이 서비스를 위해 판매자의 회사를 선택해야 하는 이유를 설명합니다. AWS Marketplace 를 통해 구매자는 판매자에게 연락할 수 있습니다. 판매자는 계약에 동의한 다음, 구매자가 고정 요금으로 서비스를 구매할 수 있는 비공개 제안을 만듭니다. 다음 섹션에서는 전문 서비스 제품을 시작하는 방법을 보여 주고, 첫 번째 제품을 만들고 고객에게 제공하는 데 필요한 단계를 안내합니다.

다음 비디오에서는 AWS Marketplace에서 전문 서비스 제품을 리스팅하는 방법에 대해 자세히 설명합니다.

주제

- 사전 조건
- 전문 서비스 제품 만들기
- 비공개 제안 생성
- 제품 정보 편집
- 제품 요금 편집
- 제품 가시성 편집
- 전문 서비스 제품 제거

사전 조건

에서 전문 서비스를 판매하려면 다음 사전 조건을 완료해야 AWS Marketplace합니다.

- AWS Marketplace Management Portal에 대한 액세스 권한이 있습니다. 판매자로 등록하고 판매하는 제품을 관리하는 데 사용하는 도구입니다 AWS Marketplace. 에 액세스하는 방법에 대한 자세한 내용은 섹션을 AWS Marketplace Management Portal참조하세요AWS Marketplace 판매자를 위한 정책 및 권한.
- AWS Marketplace 판매자로 등록하고 세금 및 은행 정보를 제출합니다. 지급을 받으려면 영국 외 은행 계좌를 제공해야 합니다. 판매자가 되는 방법에 대해 자세히 알아보려면 AWS Marketplace 판매자로 시작하기 섹션을 참조하세요.
- 서비스 또는 하나 이상의 퍼블릭 제품과 관련된 전문 AWS 서비스 제품이 있어야 합니다 AWS Marketplace. 판매자의 제품이 이러한 제품을 직접 지원하거나, 이러한 제품에 대한 구독을 유도하는 서비스를 제공해야 합니다.

> **Note**
>
> 판매자의 제품이 기본 범주인 평가, 구현, 관리형 서비스, 프리미엄 지원 또는 훈련 중 하나 이상에 나열되어야 합니다.
> 전문 서비스 제품 지침에 대한 자세한 내용은 의 전문 서비스 제품에 대한 요구 사항 AWS Marketplace 섹션을 참조하세요.

전문 서비스 제품 만들기

다음 절차에서는 AWS Marketplace Management Portal에서 새 전문 서비스 제품을 생성하는 방법을 설명합니다.

전문 서비스 제품을 생성하는 방법

1. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 전문 서비스를 선택합니다. 이 페이지에는 이미 생성한 모든 전문 서비스 제품과 이러한 제품을 생성하거나 수정하기 위해 수행한 모든 요청이 표시됩니다.
3. 전문 서비스 제품 탭에서 전문 서비스 제품 생성을 선택합니다.
4. 제품 생성 페이지에서 제품 정보를 입력하고 제출을 선택합니다. 입력해야 하는 세부 정보에 대한 자세한 내용은 에서 전문 서비스 제품에 대한 세부 정보 제공 AWS Marketplace 섹션을 참조하세요.
5. (선택 사항)의 제품 메뉴에서 전문 서비스를 AWS Marketplace Management Portal선택한 다음 요청 탭을 선택합니다. 제품 요청의 제품 제목이 올바르게 표시되고 요청 상태가 검토 중인지 확인합니다. 몇 분 내에 제품이 제한된 미리 보기 모드로 생성됩니다.

> **Note**
>
> 언제든지 전문 서비스 페이지의 요청 탭으로 돌아가서 요청 상태를 확인할 수 있습니다. 생성 프로세스에서 발생한 모든 오류가 여기에 표시됩니다. 요청을 선택하면 요청 세부 정보를 보거나 오류를 수정할 수 있습니다.

제품이 처음 생성되면 AWS 계정 (제품을 생성할 때 사용한 계정)만 제품에 액세스할 수 있습니다. 전문 서비스 페이지에서 제품을 보는 경우 에서 보기를 AWS Marketplace 선택하여 구매자를 AWS

Marketplace 위해에 표시된 제품 세부 정보를 볼 수 있습니다. 비공개 제안을 제공하지 않는 한 다른 AWS Marketplace 사용자는이 세부 정보 목록을 사용할 수 없습니다.

제품을 공개적으로 제공하는 방법을 알아보려면 제품 가시성 편집 섹션을 참조하세요.

비공개 제안 생성

잠재적 구매자가 제품을 볼 때 AWS Marketplace제품을 직접 구매할 수 없습니다. 구독을 시도하면 비공개 제안을 요청하도록 리디렉션됩니다.는 AWS Marketplace 판매자 계정 루트 사용자 이메일 주소로 이메일 메시지를 AWS Marketplace 보내 고객이 비공개 제안을 요청했음을 알립니다. 다음 절차에서는 이 요청에 응답하는 방법을 설명합니다.

> **Note**
>
> 를 통해 250,000 USD 이상의 비공개 제안을 생성하는 경우 AWS Marketplace Management Portal추가 승인이 필요할 수 있습니다. 자세한 내용은 AWS Marketplace Management Portal를 통해 Private Offer Success Team(POST), Vendor Finance Success(VFS) 또는 Seller Reporting 팀에 문의하세요.

전문 서비스 제품에 대한 비공개 제안을 생성하는 방법

1. 요청과 관련하여 궁금한 점은 고객에게 문의하세요. AWS Marketplace에서 비공개 제안을 생성하기 전에 제안의 약관에 동의합니다. 구매자는 판매자의 상품을 반드시 구매할 의무가 없으므로 제안을 생성하기 전에 동의해도 괜찮습니다.
2. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
3. 메뉴에서 제안을 선택하고 비공개 제안 생성을 선택합니다.
4. 비공개 제안 생성 페이지에서 비공개 제안을 생성하려는 제품을 선택합니다. 현재 사용 가능한 제품만 포함됩니다.
5. 비공개 제안을 제시하려는 구매자 계정 ID를 입력합니다. 비공개 제안 하나에 최대 25개의 계정을 포함할 수 있습니다. 구매자가 제안 요청 기능을 사용한 경우 판매자가 받는 이메일 메시지에는 요청한 계정의 구매자 계정 ID가 포함됩니다.
6. 구매자가 제품을 할부로 결제할 수 있도록 허용할지 여부를 선택합니다. 일반적으로 단기 계약은 일시불로 결제합니다. 장기 계약의 경우 할부 결제 옵션을 제공할 수 있지만, 이는 판매자가 구매자와 체결하는 계약에 달렸습니다. 다음을 선택합니다.
7. 다음 정보를 비롯한 제안 세부 정보를 작성합니다.

- 사용자 지정 제안 이름 - 판매자와 고객이 제안을 알아보기에 충분한 세부 정보를 제공합니다. 회사 또는 제품 이름과 제품 설명을 기재합니다. 이름, 전화번호, 주소 등 개인 식별 정보는 기재하지 않습니다.
- 계약 종료 날짜 - 계약한 서비스가 종료되는 날짜입니다. 예를 들어 1년 동안 지원을 제공하는 경우 서비스가 제공되기 시작하는 날짜로부터 1년 후 날짜를 입력합니다.
- 제품 크기 - 제공하는 서비스의 가격 및 단위로, 다음과 같습니다.
  - 일시불 결제 제안 - 각 차원(예: 실버, 골드, 플래티넘 차원)을 관련 가격과 함께 표시할 수 있습니다. 구매자는 원하는 대로 선택하여 결제할 수 있습니다.
  - 결제 일정이 포함된 제안 - 판매자는 차원 하나를 선택하고 각 결제의 금액 및 날짜가 포함된 결제 일정을 제공해야 합니다.

> **Note**
>
> 0 USD 제안을 생성하려면 0 USD 가격을 활성화하겠습니다를 선택하여 확인해야 합니다. 이 예방 조치는 실수로 무료 제안을 생성하는 것을 방지하는 데 도움이 됩니다.

- 서비스 계약 - 고객과 체결하는 서비스 계약을 정의하는 문서입니다. 판매자가 업로드하는 문서(텍스트 또는 PDF 형식)는 PDF 문서 하나에 함께 첨부되므로, 파일 이름을 몰라도 내용을 이해할 수 있어야 합니다.
- 제안 만료 날짜 - 제안이 만료되는 날짜입니다. 구매자가 제안을 수락해야 하는 기간을 결정하며, 전문 서비스를 사용할 수 있게 되는 날짜와는 무관합니다. 제안이 생성된 후 제안 만료 날짜를 연장할 수 있습니다.

8. 옵션 편집을 마쳤으면 다음을 선택합니다.
9. 제안 검토 페이지에서 제안 세부 정보가 정확한지 확인하고 제안 생성을 선택합니다.

> **Note**
>
> 제안이 게시되는 데 다소 시간이 걸릴 수 있습니다. 제안이 게시된 후에는 제안 관리 페이지에서 제안을 볼 수 있습니다. 아직 수락되지 않은 제안을 수정해야 하는 경우 이 페이지에서 수정할 수 있습니다.

10. 제안이 게시되고 비공개 제안 관리 페이지에서 사용할 수 있게 되면 해당 제안의 작업 메뉴에서 제안 URL 복사를 선택한 다음, 구매자에게 이메일 메시지로 보냅니다.

제품 정보 편집

다음은 AWS Marketplace Management Portal에서 기존 전문 서비스 제품에 대한 제품 정보를 편집하는 절차입니다.

제품 정보를 편집하는 방법

1. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 전문 서비스를 선택합니다. 이 페이지에는 이미 생성한 모든 전문 서비스 제품과 이러한 제품을 생성하거나 수정하기 위해 수행한 모든 미해결 요청이 표시됩니다.
3. 편집하려는 기존 제품을 선택합니다. 변경 요청 메뉴에서 제품 정보 업데이트를 선택합니다.
4. 세부 정보를 변경합니다. 편집 가능한 필드에 대한 자세한 내용은 에서 전문 서비스 제품에 대한 세부 정보 제공 AWS Marketplace 섹션을 참조하세요.
5. 제출을 선택하여 요청을 생성합니다.
6. (선택 사항) 전문 서비스 페이지의 요청 탭에 아직 없는 경우의 제품 메뉴에서 전문 서비스를 AWS Marketplace Management Portal선택한 다음 요청 탭을 선택합니다. 요청의 제품 제목이 올바르게 표시되고 요청 상태가 검토 중인지 확인합니다. 몇 분 내에 판매자가 요청한 변경 사항으로 제품이 업데이트됩니다. 오류가 있는 경우 여기에서 확인하고 오류를 수정한 후 편집 내용을 다시 제출할 수 있습니다.

제품 요금 편집

다음은 AWS Marketplace Management Portal에서 기존 전문 서비스 제품의 요금 정보를 편집하는 절차입니다.

제품 요금을 편집하는 방법

1. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 전문 서비스를 선택합니다. 이 페이지에는 이미 생성한 모든 전문 서비스 제품과 이러한 제품을 생성하거나 수정하기 위해 수행한 모든 요청이 표시됩니다.
3. 편집하려는 기존 제품을 선택하고, 변경 요청 메뉴에서 요금 차원 업데이트를 선택합니다.

> **Note**
>
> 를 통해서만 새 요금 차원을 추가할 수 있습니다 AWS Marketplace Management Portal. 이전에 생성한 차원을 수정하거나 제거하려면 AWS Marketplace 판매자 운영 팀에 요청에

대해 문의하세요. 요청에는 제품 ID와 변경 또는 제거하려는 차원에 대한 세부 정보를 포함해야 합니다.

4. 원하는 새 요금 차원을 추가합니다. 요금 필드에 대한 자세한 내용은 에서 전문 서비스 제품에 대한 세부 정보 제공 AWS Marketplace 섹션을 참조하세요.
5. 제출을 선택하여 요청을 생성합니다.
6. (선택 사항)의 제품 메뉴에서 전문 서비스를 AWS Marketplace Management Portal선택한 다음 요청 탭을 선택합니다. 요청의 제품 제목이 올바르게 표시되고 요청 상태가 검토 중인지 확인합니다. 몇 분 내에 판매자가 요청한 변경 사항으로 제품이 업데이트됩니다. 오류가 있는 경우 여기에서 확인하고 오류를 수정한 후 편집 내용을 다시 제출할 수 있습니다.

제품 가시성 편집

기본적으로 제품은 가시성이 제한된 상태로 생성되며 새 제품은 판매자의 계정에서만 볼 수 있습니다. AWS Marketplace에서 다른 테스트 계정을 추가하거나 제품을 공개적으로 표시할 수 있습니다. 다음은 AWS Marketplace Management Portal에서 기존 전문 서비스 제품의 표시 여부를 편집하는 절차입니다.

제품 표시 여부를 편집하는 방법

1. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 전문 서비스를 선택합니다. 이 페이지에는 이미 만든 모든 전문 서비스 제품이 표시됩니다.
3. 가시성을 편집 또는 업데이트하려는 기존 제품을 선택합니다. 변경 요청 메뉴에서 제품 표시 여부 업데이트를 선택합니다.
4. 옵션을 선택하여 제품 가시성을 업데이트하고 제출을 선택하여 검토 요청을 제출합니다.
5. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다. 상태가 실패인 경우 요청 이름을 선택하여 발견된 문제를 확인합니다.

> **Note**
>
> 제품을 퍼블릭 AWS Marketplace 카탈로그에 표시하려면 AWS Marketplace 판매자 운영 팀의 제품 검토를 통해 제품이 제품 지침을 충족하는지 확인해야 합니다(참조의 전문 서비스 제품에 대한 요구 사항 AWS Marketplace). 요청이 완료될 때까지 며칠 정도 걸릴 수 있습니다.

전문 서비스 제품 제거

다음은 AWS Marketplace Management Portal에서 기존 전문 서비스 제품을 제거하는 절차입니다.

> **Note**
> 전문 서비스 제품을 제거해도 활성 비공개 제안에는 영향을 미치지 않습니다.

제품을 제거하는 방법

1. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다.
2. 제품 메뉴에서 전문 서비스를 선택합니다. 이 페이지에는 이미 만든 모든 전문 서비스 제품이 표시됩니다.
3. 제거하려는 기존 제품을 선택합니다. 변경 요청 메뉴에서 제품 표시 여부 업데이트를 선택합니다.
4. 제품 가시성을 업데이트하려면 가시성 옵션으로 제한됨을 선택하고 제출을 선택합니다.
5. 요청 탭에서 요청 상태가 검토 중으로 표시되는지 확인합니다. 요청이 완료되면 상태가 성공으로 바뀝니다.

> **Note**
> 요청이 완료될 때까지 며칠 정도 걸릴 수 있습니다. 활성 제안이 있는 제품은 마지막 활성 구독 또는 계약이 완료될 때까지 제한된 상태로 이동한 다음에서 제거됩니다 AWS Marketplace. 제한됨 상태는 기존 사용자는 제품을 계속 사용할 수 있다는 의미입니다. 하지만 제품이 더 이상 대중에게 표시되거나 신규 사용자에게 제공되지 않습니다.

에서 전문 서비스 제품에 대한 세부 정보 제공 AWS Marketplace

전문 서비스 제품을 게시할 때는 제품 메타데이터를 제공해야 AWS Marketplace합니다. 제품 메타데이터에는 가시성 및 요금에 대한 제품 설정과 구매자에게 전문 서비스 제품에 대한 자세한 정보를 제공하는 제품 정보가 포함됩니다. 제품 설명, 추가 리소스, 요금 차원 및 제품 가시성을 편집할 수 있습니다. 다음 섹션에서는 제품 세부 정보를 준비하거나 편집할 때 사용할 수 있는 정보를 제공합니다.

> **Note**
>
> 전문 서비스 제품의 지침과 요구 사항에 대한 자세한 내용은 의 전문 서비스 제품에 대한 요구 사항 AWS Marketplace 섹션을 참조하세요.

주제

- 제품 설명
- 추가 리소스
- 지원 정보
- 차원 사용
- 제품 표시 여부

제품 설명

제품 세부 정보의 제품 설명 섹션은 제품의 핵심입니다. 잠재 구매자가 구매 결정을 내릴 수 있도록 제품에 대해 설명하는 섹션입니다. 제품 세부 정보의 제품 설명 섹션에는 다음 데이터가 있습니다.

- 제품 제목 - 제품의 이름입니다. 제품을 식별하는 데 사용되며, 제품 페이지와 검색 결과에 표시됩니다. 의미 있는 제품 이름을 입력하세요. 내에서 고유해야 합니다 AWS Marketplace.
- SKU - (선택 사항) 제품을 추적하는 데 사용됩니다 AWS Marketplace. 판매자가 직접 사용하기 위한 정보이며 구매자에게는 표시되지 않습니다.
- 간단한 설명 - 타일과 제품 카탈로그의 제품 제목 아래에 표시되는 AWS Marketplace 제품에 대한 간결한 설명입니다.
- 상세한 설명 - 구매자에게 상품 세부 정보를 설명하는 길고 체계적인 설명입니다. 제품 기능, 혜택, 사용법 및 제품과 관련된 기타 정보를 제공하세요. 사용 가능한 형식을 사용하여 정보를 보다 쉽게 이해하고 검사할 수 있게 만들어야 합니다.
- 제품 로고 -이 필드는 제품을 나타내는 이미지 파일을 가리키는 퍼블릭 Amazon S3 URL입니다. 파일은 투명 또는 흰색 배경의 .png, .jpg 또는 .gif 형식이어야 하며, 크기는 5MB 미만이고 폭과 높이는 110~10,000픽셀 사이여야 합니다. 로고는 제품 제출 중에 업로드되고에 저장됩니다 AWS Marketplace. URL의 내용을 수정해도 제출 AWS Marketplace 후의 로고는 수정되지 않습니다.

> **Note**
>
> 제공하는 Amazon S3 URL은 공개적으로 사용할 수 있어야 합니다. 파일이 있는 Amazon S3 버킷의 속성입니다. 자세한 내용은 Amazon Simple Storage Service 콘솔 사용 설명서의 Amazon S3 버킷에 대한 퍼블릭 액세스 설정을 편집하려면 어떻게 해야 합니까?를 참조하세요.

- 하이라이트 - 제품의 주요 기능 또는 차별화 요소를 설명하는 1~3개의 짧은 포인트 세트입니다. 최소 1개 이상의 하이라이트가 필요합니다.
- 제품 범주 - 판매자가 제공하는 서비스 유형입니다. 범주를 하나 이상, 최대 3개까지 선택해야 합니다. 선택할 수 있는 여러 범주가 있지만, 전문 서비스 제품에는 다음 중 하나 이상이 포함되어야 합니다.

  평가

  고객의 현재 운영 환경을 평가하여 고객의 조직에 적합한 솔루션을 찾습니다.

  구현

  타사 소프트웨어의 구성, 설정 및 배포를 도와줍니다.

  프리미엄 지원

  고객의 요구 사항에 맞게 설계된 전문가의 지침 및 지원을 이용할 수 있습니다.

  관리형 서비스

  고객 대신 종합적으로 환경을 관리합니다.

  훈련

  고객의 직원들이 모범 사례를 배울 수 있도록 전문가의 맞춤형 워크숍, 프로그램 및 교육 도구를 제공합니다.
- 검색 결과 키워드 - 구매자가 제품을 검색할 때 사용할 수 있는 키워드를 최대 3개까지 제공합니다. 키워드는 쉼표로 구분된 목록으로 나열할 수 있으며, 최대 250자까지 가능합니다.
- 관련 제품 - 선택 사항 - AWS Marketplace 서비스가 작동하거나 지원하는 하나 이상의 퍼블릭 제품을 포함합니다.는 제품의 세부 정보 페이지 또는 해당 제품의 관련 제품에 표시할 제품을 선택할 때 이러한 제품을 입력으로 AWS Marketplace 사용합니다.

추가 리소스

제품 세부 정보의 추가 리소스 섹션에서 고객을 지원하기 위해 생성한 리소스의 링크를 제공할 수 있습니다. 온라인에 저장되는 1~3개의 다운로드 가능한 리소스로 구성되는 선택적 세트입니다. 리소스의 예로는 제품 정보 시트, 백서 또는 제품 설명서가 있습니다. 각 리소스의 이름과 URL을 제공하세요.

지원 정보

이 섹션은 판매자가 서비스에 제공하는 지원에 대해 설명할 수 있는, 서식이 지정된 텍스트 필드입니다.

고객은 서비스 사용, 문제 해결, 환불 요청(해당하는 경우)과 같은 문제에 대한 지원을 기대합니다. 제품 지원을 받으려면 이메일 주소, 전화번호 또는 지원 URL(들) 중 하나 이상을 제공해야 합니다. 구매자가 제품을 구매할 때 기대할 수 있는 지원 수준에 대한 설명을 포함합니다.

차원 사용

전문 서비스의 요금 차원은 판매자가 제공하는 패키지입니다. 예를 들어 실버, 골드 및 플래티넘 지원을 제공할 수 있습니다. 10시간, 20시간 또는 50시간의 컨설팅을 제공할 수도 있습니다. 제공하려는 각 차원(최소 하나, 최대 24개)의 이름과 설명을 지정하세요. 구매자와 직접 협의하여 제품의 비공개 제안을 생성하는 경우 이러한 차원의 실제 요금을 설정합니다.

> **Note**
>
> 요금 차원 사용 방법 및 요금 설정 방법에 대한 자세한 내용은 비공개 제안 생성 섹션을 참조하세요.

제품 표시 여부

출시된 제품은 AWS Marketplace 에서 자신의 계정, 소량의 테스트 계정 또는 모든 계정에만 표시될 수 AWS 있습니다. 기본적으로 제품은 비공개 릴리스로 게시됩니다. 제품 표시 여부를 변경하려면 제품 가시성 편집 섹션을 참조하세요.

의 전문 서비스 제품에 대한 요구 사항 AWS Marketplace

AWS Marketplace 는의 모든 제품 및 제품에 대한 요구 사항을 유지합니다 AWS Marketplace. 이러한 요구 사항은 고객을 위해 안전하고 신뢰할 수 있고 잘 큐레이션된 디지털 카탈로그를 홍보하는 데 도움

이 됩니다. 판매자는 특정 제품의 요구 사항을 충족할 수 있도록 추가 컨트롤 및 프로토콜 구현을 검토하는 것이 좋습니다.

AWS Marketplace 는 제출 시 모든 제품과 관련 메타데이터를 검토하여 현재 AWS Marketplace 요구 사항을 충족하거나 초과하는지 확인합니다. AWS는 보안 요구 사항을 충족하도록 이러한 요건을 검토하고 조정합니다. 또한는 제품을 AWS Marketplace 지속적으로 검토하여 이러한 요구 사항에 대한 변경 사항을 충족하는지 확인합니다. 제품이 규정을 준수하지 않는 경우 AWS Marketplace 에서는 판매자에게 연락하여 제품을 업데이트할 것을 요청합니다. 경우에 따라 문제가 해결될 때까지 새 구독자가 제품을 일시적으로 사용할 수 없게 될 수 있습니다.

주제

- 제품 설정 가이드라인
- 고객 정보 요구 사항
- 제품 사용 가이드라인
- 아키텍처 가이드라인

제품 설정 가이드라인

모든 전문 서비스 제품은 다음 제품 설정 지침을 준수해야 합니다.

- 모든 요금 차원은 AWS Marketplace를 사용하여 제공되고 요금이 청구되는 실제 서비스와 관련이 있어야 합니다.
- 판매자의 제품이 평가, 구현, 관리형 서비스, 프리미엄 지원 또는 훈련 범주 중 하나에 나열되어야 합니다.
- 필요한 전문 서비스 범주 외에도, 제공되는 서비스에 맞는 다른 적절한 범주를 선택하여 제품을 올바르게 분류해야 합니다.
- 제품 로고는 AWS 로고 또는 관련 없는 타사의 로고와 혼동하도록 설계해서는 안 됩니다.
- 제품 세부 정보에 불쾌하거나 노골적인 자료가 포함되면 안 됩니다. https://aws.amazon.com/aup/ 제공되는 AWS 사용 제한 정책을 준수해야 합니다.
- 전문 서비스 제품은에 나열된 타사 소프트웨어 제품과 관련된 서비스를 직접 지원 또는 제공하거나 고객이의 채택 또는 관리와 관련된 특정 결과를 달성하는 데 AWS Marketplace 도움이 되어야 합니다 AWS 클라우드.

고객 정보 요구 사항

모든 전문 서비스 제품은 다음과 같은 고객 정보 요구 사항을 준수해야 합니다.

- 전문 서비스 제품은 나열된 차원을 통해 전액 청구해야 합니다 AWS Marketplace.
- 신용 카드 및 은행 계좌 정보를 포함하여 AWS Marketplace 전문 서비스 제품 목록에 대한 고객 결제 정보를 언제든지 수집할 수 없습니다.
- 전문 서비스 제품과 함께 AWS 에서 제공하는 구독자 또는 잠재 구독자 정보는 AWS Marketplace 판매자 이용 약관에 따라서만 사용해야 합니다.

제품 사용 가이드라인

모든 전문 서비스 제품은 다음 제품 사용 지침을 준수해야 합니다.

- 고객이 전문 서비스 목록을 통해 판매자에게 연락하면 판매자는 2영업일 이내에 고객에게 연락해야 합니다. 고객이 비공개 제안을 수락하면 판매자는 비공개 제안에 달리 명시된 경우를 제외하고 2일 이내에 고객에게 연락하거나 다음 단계를 안내해야 합니다.
- 고객은 서비스 사용, 문제 해결 및 환불 요청(해당하는 경우)과 같은 문제에 대한 도움을 쉽게 받을 수 있어야 합니다. 이행 랜딩 페이지에 지원 연락처 옵션이 명시되어 있어야 합니다. 지원 설명에는 고객이 기대할 수 있는 지원 수준에 대한 설명이 포함되어야 합니다.
- 제품 개요에는 명확한 가치 제안, 주요 기능, 상세 설명서로 연결되는 링크, 제공되는 서비스의 구매 전 및 구매 후 지원에 대한 명확한 정의가 포함되어야 합니다.
- 제품의 서비스 약관은 명확하고 간단해야 합니다.

아키텍처 가이드라인

모든 전문 서비스 제품은 다음 아키텍처 지침을 준수해야 합니다.

- 전문 서비스 제품은 해당 제품이 해당 관련 제품에 대한 서비스를 직접 제공하거나 해당 관련 제품에 더 많은 구독자를 유도하는 서비스를 제공하는 AWS 서비스 또는 하나 이상의 퍼블릭 AWS Marketplace 제품(최대 4개)과 관련이 있어야 합니다.
- 고객 인프라에 추가 리소스가 필요한 서비스는 다음 지침을 따라야 합니다.
  - AWS Security Token Service 또는 AWS Identity and Access Management (IAM)를 사용하는 등 안전한 방식으로 리소스를 프로비저닝합니다.
  - 프로비저닝된 모든 AWS 서비스에 대한 설명, IAM 정책 설명, IAM 역할 또는 사용자가 고객 계정에 배포되고 사용되는 방법을 포함한 추가 설명서를 제공합니다.

- 고객이 AWS Marketplace 거래와 별도로 추가 AWS 인프라 비용을 발생시키는 경우 추가 인프라 요금을 지불할 책임이 있음을 설명하는 알림을 제품 설명에 포함합니다.
- 제품이 에이전트를 배포하는 경우 고객에게 AWS 계정에 에이전트를 배포하는 방법에 대한 지침을 제공합니다.

에서 전문 서비스 제품 요금 책정 AWS Marketplace

AWS Marketplace 판매자는 전문 서비스 제품에 대한 비공개 제안을 수락할 수 있습니다. 비공개 제안은 제품을 구매하는 데 사용되는 협상된 용어입니다 AWS Marketplace. 판매자는 구매자와 협상하여 비공개 제안에 도달합니다. 비공개 제안은에서 사용할 수 있는 요금 모델 중 하나입니다 AWS Marketplace. 판매자 비공개 제안의 경우 다년 및 사용자 지정 기간 계약에 사용할 수 있는 몇 가지 옵션이 있습니다. 이 주제에서는 전문 서비스 제품 요금 및 비공개 제안에 대한 자세한 정보를 제공합니다.

다년 및 사용자 지정 기간 계약에 대한 자세한 내용은 AWS Marketplace 제품에 대한 비공개 제안 준비 및 비공개 제안 할부 플랜 단원을 참조하세요.

제품마다 설정할 수 있는 요금은 한 가지로 제한됩니다. AWS Marketplace 제품 요금에 대한 자세한 내용은 섹션을 참조하세요에 대한 제품 요금 AWS Marketplace.

비공개 제안 작동 방식

AWS Marketplace Management Portal의 제안 페이지에서 비공개 제안을 만들고 관리할 수 있습니다. 고유한 ID 및 URL을 생성하기 위해 제안의 제품을 지정합니다. 비공개 제안에 대한 요금제를 만들고, 법적 조건과 판매 문서를 추가하고, 특정 구매자 AWS 계정에 제안을 확장합니다. 제안은 해당 제안을 만든 계정에만 표시됩니다.

비공개 제안을 만들고 잠재적 구매자에게 알리면 잠재적 구매자는 비공개 제안을 살펴보고 수락할 수 있습니다. 제안을 보려면 구매자는 제안을 받은 AWS 계정 에 로그인해야 합니다.

> **Note**
>
> 해당 제안을 연결 계정 또는 관리 계정에 제시하지 않으면 구매자가 제안을 볼 수 없습니다. 제안에 서비스 제한을 제공할 수 없습니다. 따라서 구매자는 제품에서 제한이 설정되지 않은 한 협상된 가격으로 원하는 만큼 제품을 사용할 수 있습니다.

비공개 제안 생성에 대한 자세한 내용은 Creating and managing private offers를 참조하세요.

비공개 제안은 판매자 보고서에서 추적됩니다. 자세한 내용은 Reporting for private offers 및 Seller reports guide를 참조하세요.

데이터 제품

AWS Data Exchange는 AWS 고객이 AWS 클라우드에서 파일 기반 데이터 세트를 안전하고 쉽게 교환할 수 있는 서비스입니다. 공급자인 AWS Data Exchange는 데이터 전송, 권한 부여 또는 결제 기술을 구축하고 유지 관리할 필요가 없습니다. AWS Data Exchange의 공급자는 AWS 고객에게 연락하고 기존 고객에게 보다 효율적으로 구독 권한을 부여할 수 있는 안전하고 투명하며 신뢰할 수 있는 채널을 보유하고 있습니다. AWS Data Exchange 공급자가 되려면 몇 단계를 거쳐서 자격을 확인해야 합니다.

데이터 제품에는 다음과 같은 부분이 있습니다.

- 제품 세부 정보 - 이 정보는 잠재적 구독자가 제품이 무엇인지 이해하는 데 도움이 됩니다. 여기에는 이름, 설명(짧은 설명 및 긴 설명), 로고 이미지 및 지원 연락처 정보가 포함됩니다. 제품 세부 정보는 공급자가 작성합니다.
- 제품 제안 - AWS Data Exchange에서 제품을 사용할 수 있도록 하려면 공급자가 공개 제안을 정의해야 합니다. 여기에는 요금 및 기간, 데이터 구독 계약, 환불 정책 및 사용자 지정 제안 생성 옵션이 포함됩니다.
- 데이터 세트 - 제품에는 하나 이상의 데이터 세트가 포함될 수 있습니다. 데이터 세트는 파일 기반 동적 데이터 세트 콘텐츠입니다. 데이터 세트는 동적이며 개정을 사용하여 버전이 지정됩니다. 각 개정에는 여러 자산이 포함될 수 있습니다.

자격 요건을 비롯한 자세한 내용은 AWS Data Exchange 사용 설명서의 AWS Data Exchange에서 데이터 제품 제공을 참조하세요.

에 게시할 제품 제출 AWS Marketplace

제품 제출 프로세스를 사용하여 제품을 AWS Marketplace에서 사용할 수 있게 만듭니다. 제품은 가격 구조가 하나인 단일 Amazon Machine Image(AMI)와 같이 간단하거나 AWS CloudFormation 템플릿, 복잡한 요금 옵션 및 결제 일정이 있는 보다 복잡할 수 있습니다. 제품 제공을 정의하고 다음 두 가지 방법 중 하나로 AWS Marketplace Management Portal 를 통해 제출합니다.

- 제품 탭 사용 - 덜 복잡한 제품의 경우 제품 탭을 사용하여 요청을 완전하게 정의한 후 제출합니다.
- 자산 탭 사용 - 제품이 복잡하여 더 많은 정의가 필요한 경우 제품 로드 양식(PLF)을 다운로드하고, 제품 세부 정보를 추가하고, 파일 업로드 옵션을 사용하여 완성된 양식을 업로드합니다.

> **Note**
>
> 데이터 제품 공급자는 AWS Data Exchange 콘솔을 사용하여 제품을 게시해야 합니다. 자세한 내용은 AWS Data Exchange 사용 설명서의 새 제품 게시를 참조하세요.

제품 탭을 사용하여 시작하면서 어느 방법을 사용하지 결정하는 것이 좋습니다. 다음 표에는 몇 가지 구성과 요청을 제출할 때 사용하는 방법이 나와 있습니다. 첫 번째 열은 제품의 요금 모델이고, 다른 세 열은 제품이 고객에게 배포되는 방식을 설명합니다.

요금 모델	단일 노드 AMI를 사용하여 제품 출시	를 사용하여 출시된 제품 AWS CloudFormation	Software as a Service(SaaS) 방식으로 제품 출시
기존 보유 라이선스 사용(BYOL)	제품 탭	자산 탭	
무료	제품 탭	자산 탭	
유료 시간당	제품 탭	자산 탭	
유료 연간 시간당	제품 탭	자산 탭	
월별 결제	제품 탭	자산 탭	
월별 시간당	자산 탭	자산 탭	

요금 모델	단일 노드 AMI를 사용하여 제품 출시	를 사용하여 출시된 제품 AWS CloudFormation	Software as a Service(SaaS) 방식으로 제품 출시
유료 사용량(AWS Marketplace 측정 서비스)	제품 탭	자산 탭	
계약 요금	제품 탭		
SaaS 구독			제품 탭
SaaS 계약			제품 탭
SaaS 레거시			자산 탭

제품을 개별적으로 제출할 수도 있고, 제품 로드 양식을 사용하여 여러 제품 또는 제품 템플릿을 동시에 제출할 수도 있습니다. 제품 탭을 사용할 경우에는 여러 제품을 동시에 제출할 수 없습니다. 제출할 제품과 제출 방법을 잘 모르는 경우 제품 탭을 사용하여 시작하세요. 제출하는 데 문제가 있는 경우 AWS Marketplace 판매자 작업 팀에 문의하세요.

주제

- 제품 탭 사용
- 회사 및 제품 로고 요구 사항
- 유료 리패키징 소프트웨어 제출 요구 사항
- 하드웨어 구성 요소가 있는 제품에 대한 요구 사항
- AWS CloudFormation에서 시작되는 제품(무료 또는 유료) 또는 사용량 기반 유료 AMI 제품
- 제품 변경 및 업데이트
- 시간 및 기대
- AMIs 제출 AWS Marketplace
- 최종 체크리스트

제품 탭 사용

제품 탭에 액세스하려면 AWS Marketplace Management Portal에 로그인합니다. 제품 탭에서 관리하는 제품의 유형에 따라 서버, SaaS 또는 기계 학습을 선택합니다.

해당 제품 유형에 대해 현재 제품을 모두 보여주는 대시보드가 나타납니다. 요청 탭을 선택하면 대시보드에 미처리 요청과 완료된 요청 기록이 표시됩니다. 제품 요청을 만들기 시작하면 진행 중인 작업을 저장하고 필요한 경우 다양한 세션에서 목록을 만들 수 있습니다.

제품 요청을 제출하면 AWS Marketplace 팀이 요청을 검토합니다. 요청한 제품 유형의 제품 페이지에서 요청 상태를 모니터링할 수 있습니다. 새 제품의 경우 요청 게시가 승인되면 판매자가 제출을 미리 보고 승인하는 데 사용하는 제한된 등록 URL이 제공됩니다. 제품 제안은 판매자가 제출을 승인할 때까지 게시되지 않습니다.

제품 업데이트를 요청하면 변경 사항을 검토하고 승인할 필요 없이 해당 업데이트가 게시됩니다. 여기에는 버전 추가 또는 제거와 메타데이터 변경 사항이 포함됩니다.

요청 탭 아래에서 요청의 상태를 추적할 수 있습니다. 탭에는 다음 중 하나가 표시됩니다.

- 초안 – 요청 프로세스를 시작했지만 요청을 제출하지 않았습니다.
- 제출 – 요청을 완료하고 제출했으며 현재 검토 중입니다.
- 조치 필요 - AWS Marketplace 팀이 요청을 검토했으며 추가 정보가 필요합니다.
- 승인 필요 - AWS Marketplace 팀이 제품에 대한 제한된 목록 URL을 생성했습니다. URL을 검토하고 승인 또는 거부해야만 AWS Marketplace 가 제품을 게시합니다. 승인하면 사이트가 게시되는 동안 상태가 게시 보류로 변경됩니다. 거부할 경우 요청을 수정할 수 있도록 상태가 초안 작성으로 돌아갑니다.
- 게시 보류 중 - 요청의 모형을 승인했으며 제품을 AWS Marketplace 게시하고 있습니다.
- 만료 – 요청 프로세스를 시작했지만 6개월 내에 완료하지 않아 요청이 만료되었습니다.

제출 상태의 항목이 있으면 제출을 취소할 수 있습니다. 초안 작성 상태의 항목이 있으면 요청을 삭제할 수 있습니다. 그러면 처음부터 다시 시작할 수 있습니다. 초안 작성 항목을 삭제하면 요청 기록 탭으로 항목이 이동됩니다.

에 제품을 추가하려면 active AWS GovCloud(미국) 계정이 있어야 하며 내보내기 제어 요구 사항을 포함한 요구 사항을 준수해야 AWS GovCloud (US) AWS 리전 AWS GovCloud (US) 합니다.

회사 및 제품 로고 요구 사항

사용자가 AWS Marketplace에서 검색할 때 균일한 경험을 제공할 수 있도록 회사 로고와 제품 로고는 다음 AWS Marketplace 지침을 준수해야 합니다.

제품 로고 사양 - 제품 로고 이미지는 투명 또는 하얀색 배경이어야 하고, 크기는 120~640픽셀, 비율은 1:1 또는 2:1(가로)이어야 합니다.

회사 로고 사양 - 회사 로고 이미지는 투명 배경이어야 하고, 크기는 220 x 220픽셀이어야 하며, 각 측면 내 10픽셀의 여백을 허용합니다.

유료 리패키징 소프트웨어 제출 요구 사항

재포장된 소프트웨어 리스팅을 제출하려면 먼저 다음 요구 사항을 충족해야 합니다. 이 경우 재포장된 소프트웨어에는 Windows를 사용하는 AMI와 같이 다른 공급업체가 생성한 오픈 소스 AMI 또는 소프트웨어가 포함됩니다.

요구 사항

- 제품 제목에 리패키징으로 인해 추가된 가치가 포함되어야 합니다. 제품 제목의 예: 강화된 <제품>, 패키지가 추가된 <제품>, <제품2> 기반 <제품1>.
- 설명서에 없는 다른 용어가 제품 제목에 있으면 안 됩니다. 예를 들어 제공하는 제품의 세부 정보에 인증, 원래 또는 무료라는 용어가 없으면 제품 제목에서 이러한 용어를 사용하면 안 됩니다.
- 간략한 제품 설명에는 상품 요금을 요약하는 명확한 설명이 포함되어야 합니다. 간단한 설명은 이 제품에는 다음과 관련된 요금이 부과됩니다와 같은 문구로 시작해야 합니다. 예를 들어 제품에 제공되는 판매자의 지원이 유료인 경우 제품 설명에 이 제품은 판매자의 지원이 유료라는 내용을 명시해야 합니다.
- 제품 로고는 판매자 등록 과정에서 사용한 회사 로고와 동일해야 합니다. 공식 소프트웨어 로고를 사용하는 경우에만 제품 로고가 회사 로고와 달라도 되고, 이 경우 원래 소프트웨어 공급업체로부터 명시적 허가를 받아야 합니다. 명시적 허가를 받은 경우 변경 요청의 참고 섹션, 또는 제품 로드 양식을 사용하는 경우 파일 업로드 페이지의 간단한 설명 입력 필드에 해당 설명서로 연결되는 링크를 포함해야 합니다.
- AMI 제품의 경우 AMI 이름을 원래 제품에서 재사용하면 안 됩니다. AMI 이름은 판매자 이름으로 시작해야 하며 [판매자 이름] [ami에 지정된 이름] 형식을 따라야 합니다.

  AMI 이름이 이름 지정 규칙을 준수하지 않는 경우 AWS 콘솔에서 AMI를 복사하고 이름을 바꿀 수 있습니다. 자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 Amazon EC2 AMI 복사를 참조하세요.

유료 리스팅이 판매자의 회사에서 만들지 않은 독립형 소프트웨어 제품에 대한 것이고 추가 소프트웨어 라이브러리 번들링이나 특수 구성 추가 등 제품에 지적 재산권이 추가되지 않은 경우, 이전 요구 사항과 함께 다음 요구 사항도 충족해야 합니다.

- 제품 제목에는 앞에서 설명했듯이 추가된 가치와 함께 판매자 이름이 포함되어야 합니다. 판매자 이름은 판매자로 등록할 때 사용된 이름입니다. 예: <판매자>가 유지 관리를 지원하는 <제품>.
- 제품에 대한 긴 설명의 첫 번째 줄은 이 제품은 리패키징 소프트웨어 제품으로 다음과 같은 추가 요금이 부과됩니다라는 문구(또는 오픈 소스인 경우 이 제품은 리패키징 오픈 소스 소프트웨어 제품으로 다음과 같은 추가 요금이 부과됩니다라는 문구)로 시작해야 합니다. 그런 다음, 긴 설명에는 부과되는 요금을 요약하는 명확한 문장과 해당 기능을 설명하는 추가 세부 정보가 포함되어야 합니다. 예를 들어 추가 지원에 비용을 청구하는 오픈 소스 제품의 긴 설명은 이 제품은 리패키징 오픈 소스 소프트웨어 제품으로 {SLA 세부 정보} 지원 시 추가 요금이 적용됩니다라는 문구로 시작할 수 있습니다.

하드웨어 구성 요소가 있는 제품에 대한 요구 사항

하드웨어 제품 판매는 허용되지 않습니다 AWS Marketplace. 하드웨어 구성 요소(예: SIM 카드, 스마트 디바이스, IoT 디바이스 또는 센서)가 필요한 소프트웨어 제품을 제출하는 경우 다음 요구 사항을 충족해야 합니다.

- 하드웨어 구성 요소는 판매할 수 없습니다 AWS Marketplace.
- 하드웨어 구성 요소의 비용을 제품의 리스팅 가격에 포함할 수 없습니다.
- 리스팅의 제품 개요 섹션에는 다음 설명이 포함되어야 합니다. 이 리스팅에 필요할 수 있는 모든 하드웨어는 별도로 구해야 합니다. 자세한 내용은 제품 세부 정보를 참조하세요.

AWS CloudFormation에서 시작되는 제품(무료 또는 유료) 또는 사용량 기반 유료 AMI 제품

> **Note**
>
> 일부 요금 모델은 더 이상이 섹션에 설명된 제품 로드 양식을 사용하여 CloudFormation 제품에 AMI를 게시할 필요가 없습니다. 판매자 포털의 서버 제품 페이지에서 CloudFormation을 사용하여 Amazon Machine Image(AMI) 또는 AMI를 생성하고 제품 로드 양식을 즉시 다운로드하라는 메시지가 표시되지 않는 경우 AMI 기반 제품 생성 및 목록에 CloudFormation 템플릿 추가를 참조하세요.

제품 로드 양식(PLF)을 사용하여 AWS Marketplace 고객이 AWS CloudFormation 템플릿을 사용하여 시작하는 제품을 제출합니다. PLF는 AWS Marketplace Management Portal을 통해 제공됩니다.

다음과 같은 광범위한 단계에 따라 제품을 제출합니다.

- 요금 모델을 선택합니다.
- 제품 로드 양식(PLF) Microsoft Excel 스프레드시트를 다운로드합니다.
- 제품 로드 양식을 작성합니다.
- 양식의 지침에 따라 제품을 제출합니다.

각 단계 완료에 대한 자세한 내용은 섹션을 나열된 순서대로 확장하세요.

요금 모델 선택

제품의 요금 모델을 선택해야 합니다. 선택한 모델은 PLF에 입력하는 요금 정보를 제어합니다. 지원되는 요금 모델의 목록은 이 안내서의 에 대한 AMI 제품 요금 AWS Marketplace 섹션을 참조하세요.

새 제품의 PLF 다운로드

1. AWS Marketplace Management Portal을 시작합니다.
2. 자산 탭의 오른쪽 창에서 CloudFormation 제품 포함 단일 AMI 링크를 선택합니다.

   브라우저에 양식이 나타납니다.
3. 파일 다운로드를 선택한 다음 Excel에서 파일을 편집합니다.

   - 또는 -

   Microsoft OneDrive가 있는 경우 복사본 편집을 선택합니다. 그러면 PLF가 OneDrive에 저장되고 여기에서 편집할 수 있습니다.

> **Note**
>
> 스프레드시트에는 몇 가지 예제 제품이 포함되어 있습니다. 양식을 제출하기 전에 해당 항목을 삭제해야 합니다.

기존 제품의 PLF 다운로드

양식을 다운로드하려면

- AWS Marketplace 대시보드를 시작하고 제품 로드 양식 다운로드를 선택합니다.

스프레드시트에는 제품의 이전 버전에 대한 데이터가 포함되어 있습니다. 해당 데이터를 그대로 두고 다음 빈 행에 새 제품을 추가합니다.

PLF 작성

다음 단계에서는 제품 로드 양식(PLF)을 작성하는 방법을 설명합니다. 단계는 신규 및 기존 제품에 적용됩니다.

양식을 작성하려면

1. SKU부터 환불 및 취소 정책까지의 열에 제품과 관련된 모든 정보를 입력합니다.

   > **Note**
   >
   > 제품 액세스 지침 열에서 상세하고 명확한 사용 지침을 제공해야 합니다. 이 가이드의 에 대한 AMI 및 컨테이너 제품 사용 지침 생성 AWS Marketplace에 나열된 요구 사항을 따릅니다.

2. 입력부터 엔드포인트 URL 상대 URL까지의 열에 AMI에 필요한 정보를 입력합니다.

   > **Important**
   >
   > AMI를 AWS Marketplace와 공유해야 합니다. 이렇게 하려면 이 안내서의 AMIs 제출 AWS Marketplace에 있는 단계를 따릅니다.

3. 최종 사용자 라이선스 계약 URL 열은 표준 AWS Marketplace 계약에 대한 링크를 제공합니다. 해당 계약에 동의하거나 사용하려는 EULA에 대한 링크를 입력할 수 있습니다. 링크를 제공하는 경우 고객이 Amazon S3 버킷의 링크와 같은 EULA를 다운로드할 수 있도록 허용해야 합니다.

   표준 계약에 대한 자세한 내용은 이 가이드의 에서 표준화된 계약 사용 AWS Marketplace 섹션을 참조하세요.

4. us-east-1 가용성에서 향후 모든 인스턴스 유형 열에 사용할 수 있도록 하려면 사용하려는 각 AWS 리전 **FALSE** 아래에 **TRUE** 또는를 입력합니다.

> **Note**
>
> GovCloud 리전에는 추가 요구 사항이 있습니다. 예를 들어 GovCloud 리전을 사용하려면 GovCloud 계정이 있어야 합니다. 자세한 내용은 AWS GovCloud 사용 설명서의 설정을 참조하세요.

5. 권장 인스턴스 유형 열에서 권장 인스턴스 유형을 수락하거나 목록에서 다른 인스턴스 유형을 선택합니다. 사용하려는 리전에서 인스턴스 유형을 사용할 수 있는지 확인합니다.

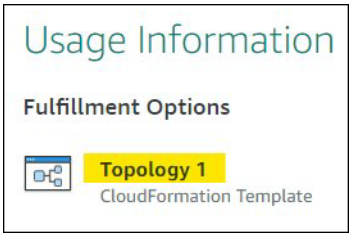
> **Note**
>
> - 대부분의 고객은 권장 인스턴스 유형을 수락합니다.
> - 인스턴스 유형을 제품과 동일한 리전에서 사용할 수 있는지 확인해야 합니다.

6. 권장 인스턴스 유형과 포함할 국가 사이의 열에서 인스턴스 유형 아래에 **TRUE**를 입력합니다. 그러면 인스턴스 유형이 활성화됩니다. 나머지 유형에는 **FALSE**를 입력합니다. 인스턴스 유형에 대한 자세한 내용은 https://aws.amazon.com/ec2/instance-types/를 참조하세요.
7. 포함할 국가 및 제외할 국가 열에서 포함하거나 제외하려는 국가의 2자리 국가 코드(예: US)를 입력합니다.
8. 요금 모델 열에 제품의 요금 모델을 입력합니다.

   다음 목록은 요금 모델과 작성해야 하는 추가 열을 설명합니다.

   - BYOL URL - 라이선스 URL을 입력합니다. 요금 정보는 입력할 필요가 없습니다.
   - 시간당 - TRUE로 설정한 인스턴스 유형의 가격을 입력합니다. 다른 모든 열은 비워 둡니다. 이와 관련된 열은 a1.medium Hourly Price에서 z1d.metal Hourly Price까지의 열입니다.
   - 시간당 연간 요금 - 이전 단계에 나열된 열과 a1.medium 연간 요금부터 z1d.metal 연간 요금까지의 열을 작성합니다. TRUE로 설정한 모든 인스턴스 유형의 가격을 입력합니다. 다른 모든 열은 비워 둘 수 있습니다.
   - 사용량 - FCP 범주부터 FCP Dimension24 요금까지의 열에 사용량 차원과 관련된 정보를 입력합니다.
   - 계약 - 계약 범주부터 계약 Dimension24 36개월 요금까지의 열에 계약 차원과 관련된 정보를 입력합니다.

9. 보안 그룹 규칙 1부터 보안 그룹 규칙 12까지의 열에 제품의 보안 그룹에 대한 정보를 입력합니다. `tcp,#,#,0.0.0.0/0` 형식을 따릅니다. 예를 들어 SSH에는 `tcp,22,22,0.0.0.0/0`을 사용하고 RDP에는 `tcp,3389,3389,0.0.0.0/0`을 사용합니다.
10. 클러스터 및 AWS 리소스 토폴로지 1: 클러스터 및 리소스 토폴로지 3 제목: 아키텍처 다이어그램 URL AWS 열에 제품의 CloudFormationdata를 입력합니다. 다음 데이터를 입력해야 합니다.
    - 토폴로지 제목 - 배포 또는 이행 옵션의 제목입니다. 제목은 이행 옵션 섹션의 제품 세부 정보 페이지에 표시됩니다. 예시:

    - 요금 견적 열에 AWS 계산기에 대한 링크를 값과 함께 입력합니다.
    - 짧은 설명 및 긴 설명 - 배포 옵션에 대한 설명을 입력합니다.
    - 템플릿 URL - CloudFormation 템플릿을 다운로드할 수 있는 링크를 제공합니다.
    - 아키텍처 다이어그램 - CloudFormation의 토폴로지 아키텍처 다이어그램을 다운로드할 수 있는 링크를 제공합니다. 각 배포 옵션에는 스택이 시작되는 항목을 보여주는 고유한 다이어그램이 있어야 합니다. 다이어그램은 아키텍처 다이어그램에 나열된 요구 사항을 따라야 합니다.

PLF 제출

다음 단계에서는 완료된 PLF를 제출하는 방법을 설명합니다.

1. AWS Marketplace Management Portal에 로그인합니다.
2. Assets(자산) 탭에서 File Upload(파일 업로드)를 선택합니다.
3. 파일 업로드 페이지에서 PLF 및 AWS CloudFormation 템플릿을 업로드합니다. 파일 업로더는 보안 전송 메커니즘과 제출된 파일의 기록을 제공합니다. 업로더는 자동으로에 알리고 AWS Marketplace,는 제출에서 정책 및 보안 규정 준수, 소프트웨어 취약성 및 제품 사용성을 검토합니다. 팀에 요청과 관련된 질문이나 문제가 있는 경우 팀에 이메일 메시지가 전송됩니다.

제품 업데이트

제품 로드 양식(PLF)을 사용하여 생성한 제품의 경우 PLF를 사용하여 해당 제품을 변경할 수도 있습니다. 완성한 원래 PLF를 변경할 수도 있고, 원래 PLF를 사용할 수 없는 경우 새 PLF로 시작할 수도 있습

니다. 제품 탭을 사용하는 것처럼 새 버전을 추가하고, 기존 버전을 제거하고, 가격, 인스턴스 유형, 리전 가용성 및 메타데이터를 업데이트할 수 있습니다. 업데이트를 수행하려면 새 제품을 준비할 때와 동일한 방법으로 업데이트된 제품을 준비합니다. 제품 업데이트가 준비되면 다음 단계를 수행합니다.

1. 기존 PLF를 사용하거나 AWS Marketplace Management Portal을 시작하고 자산 탭에서 파일 업로드를 선택합니다. 제품 로드 양식 및 판매자 가이드에서 제품의 제품 로드 양식을 다운로드할 수 있습니다.
2. PLF에서 제품을 업데이트합니다.
3. AWS Marketplace Management Portal의 자산 탭에서 파일 업로드를 선택합니다.
4. 파일 업로드 페이지에서 업데이트된 PLF 및 AWS CloudFormation 템플릿을 업로드합니다. 파일 업로더는 보안 전송 메커니즘과 제출된 파일의 기록을 제공합니다. 업로더는 요청 처리를 시작하도록 AWS Marketplace 팀에 자동으로 알립니다. 제출에 대한 설명(새 버전 추가, 가격 변경, 메타데이터 변경 등)을 포함합니다.

정책 및 보안 컴플라이언스, 소프트웨어 취약성 및 제품 가용성에 대한 제품 제출을 검토합니다. 요청과 관련된 질문이나 문제가 있는 경우 AWS Marketplace 팀이 이메일 메시지를 통해 연락을 드릴 것입니다. 기존 제품 페이지에 대한 업데이트는 추가 검토 없이 바로 처리되고 릴리스됩니다.

제품 변경 및 업데이트

판매자는 언제든지 제품에 대해 변경 사항을 제출할 수 있으며, 앞에서 설명한 대로 처리됩니다. 그러나 일부 변경은 90일 또는 120일마다 수행하거나 보류 중인 변경이 있는 경우에만 수행할 수 있습니다. 가격 변경 및 AWS 리전 또는 인스턴스 유형 변경을 예로 들 수 있습니다. 일반적인 변경 사항은 다음과 같습니다.

- 새 버전 - 새로운 버전의 소프트웨어와 패치 또는 업데이트 롤아웃입니다. 요청 시 AWS Marketplace 콘텐츠를 구독한 고객에게 새 버전의 가용성을 알리거나 사용자를 대신하여 업그레이드 지침을 보낼 수 있습니다.
- 메타데이터 변경 - 제품 정보에 대한 변경 사항(설명, URL 및 사용량 제한).
- 가격 변경 - 가격 금액의 변경. 요청이 완료되면 현재 고객에게 알림이 전송됩니다. 알림이 전송되면 90일 기간이 끝난 후 다음 달 1일에 변경된 가격이 적용됩니다. 예를 들어 3월 16일에 변경하는 경우 90일 후는 대략 6월 16일이지만 변경된 가격은 다음 달 1일부터 적용됩니다. 따라서 실제 변경 날짜는 7월 1일입니다.
- 요금 모델 변경 - 요금 모델에 대한 변경입니다(예: 시간별, 무료, 시간별_연별). 일부 요금 모델 변경은 지원되지 않으며 모든 모델 변경 요청은 AWS Marketplace 팀에서 검토 및 승인해야 합니다. 무료

모델에서 유료 모델로 변경하면 기존 고객에게 상당한 영향을 미칩니다. 대안은 추가 기능이 포함된 새로운 제품을 제안하고 현재 고객의 마이그레이션을 권장하는 것입니다.

- 리전 또는 인스턴스 변경 - 인스턴스 유형 또는 리전을 추가하거나 제거합니다.
- 제품 테이크다운 - 신규 고객이 구독 AWS Marketplace 하지 못하도록에서 제품 페이지를 제거합니다. 요청이 완료되면 현재 고객에게 알림이 전송됩니다.

시간 및 기대

요청을 최대한 빨리 처리하기 위해 노력하지만 요청을 여러 번 반복하고 판매자와 AWS Marketplace 팀이 검토해야 할 수 있습니다. 프로세스를 완료하는 데 필요한 시간을 안내하는 데 다음을 사용합니다.

- 총 요청 시간은 일반적으로 달력 시간을 기준으로 2~4주입니다. 요청 또는 제품이 복잡할수록 제품 메타데이터와 소프트웨어를 여러 번 반복하고 조정하기 때문에 시간이 더 길어질 수 있습니다.
- 요청의 우선 순위를 결정할 수 있도록 계획된 이벤트 또는 릴리스에 앞서 최소 45일 전에 완료된 제품 요청 및 AMI가 필요합니다.

요청에 대한 궁금한 점이 있는 경우 AWS Marketplace 판매자 작업 팀에 문의하세요.

AMIs 제출 AWS Marketplace

빌드되고에 제출된 모든 AMIs 모든 제품 정책을 준수해야 AWS Marketplace 합니다. 제출하기 전에 AMI를 최종적으로 몇 가지 확인하는 것이 좋습니다.

- 시스템에서 모든 사용자 보안 인증 정보(예: 모든 기본 암호, 권한 부여 키, 키 페어, 보안 키 또는 기타 보안 인증 정보)를 제거합니다.
- 루트 로그인이 비활성화되거나 잠겨 있는지 확인합니다. sudo 액세스 계정만 허용됩니다.
- AWS GovCloud (US) 리전에 배포할 AMI를 제출하는 경우, 활성 AWS GovCloud 계정이 있어야 하며 해당하는 수출 통제 요구 사항을 비롯한 AWS GovCloud 요구 사항에 동의해야 합니다.

AMI 셀프 서비스 검사

AWS Marketplace 는 내에서 AMI 스캔을 제공합니다 AWS Marketplace Management Portal. 새 제품의 경우 시스템은 자동으로 스캔을 실행하고 제출 시 결과를 제공합니다. 기존 제품의 새 버전의 경우

'버전 추가' 테스트 기능을 사용하여 스캔을 시작합니다. 이 프로세스는 일반적으로 1시간 이내에 완료되며 하나의 편리한 위치에서 포괄적인 피드백을 제공합니다.

셀프 서비스 검사를 통해 AMI 공유 및 검사를 시작하는 방법

1. https://aws.amazon.com/marketplace/management/products/server 이동합니다.
2. 제품 제목 열에서 제품을 선택합니다.
3. 제품 세부 정보 페이지의 버전 섹션에서 테스트 '버전 추가'를 선택합니다.
4. 테스트 '버전 추가' 페이지에서 필요한 버전 정보와 검증 파라미터를 입력합니다.
5. 테스트 실행을 선택합니다.
6. 스캔 결과를 봅니다.

AMI를 성공적으로 스캔한 후 현재 프로세스에 따라 제품 로드 양식(PLF)을 업로드하여 AWS Marketplace 판매자 운영 팀에 제출할 수 있습니다. 문제가 있는 경우 AWS Marketplace 판매자 작업 팀에 문의하세요.

셀프 서비스 스캔 목록에 AMI를 포함하려면 AMI가 `us-east-1` (버지니아 북부) 리전에 있고 AWS Marketplace 판매자 계정이 소유해야 합니다. 다른 계정에에 대한 액세스 권한을 부여해야 하는 경우 해당 계정을 판매자로 등록 AWS Marketplace Management Portal해야 합니다. 자세한 내용은 AWS Marketplace 판매자로 등록 단원을 참조하십시오.

AMI 복제 및 제품 코드 배정

AMI가 제출되면는 소프트웨어를 사용할 수 있도록 지정한 각 리전에 대해 복제된 AMIs를 AWS Marketplace 생성합니다. 이 복제 및 게시 프로세스 중에 AWS Marketplace 는 복제된 AMI에 제품 코드를 연결합니다. 제품 코드는 액세스 제어 및 사용량 측정에 모두 사용된다. 모든 제출은 이 AMI 복제 프로세스를 거쳐야 한다.

최종 체크리스트

제품 요청을 제출하기 전에 이 체크리스트를 사용하면 제품 게시가 지연되는 것을 방지할 수 있습니다.

제품 사용

- 즉시 프로덕션에 사용할 수 있습니다.
- 시간이나 다른 제한 조건으로 제품 사용을 제한하지 않습니다.
- 1-Click 이행 경험과 호환됩니다.

- 소프트웨어에는 클라이언트 애플리케이션을 포함하여 제품을 사용하는 데 필요한 모든 것이 포함되어 있습니다.
- 기본 사용자가 무작위 암호를 사용하거나 초기 사용자를 생성하려면 구매자가 인스턴스 ID와 같은 인스턴스 고유의 값을 사용하여 인스턴스를 사용할 수 있는 권한이 있는지 확인해야 합니다.

무료 또는 유료 제품

- 제품을 사용하기 위해 추가 라이센스가 필요하지 않음.
- 유료 재패키지 소프트웨어는를 충족합니다 AWS Marketplace 유료 리패키징 소프트웨어 제출 요구사항.
- 구매자는 제품을 사용하기 위해 개인 식별 정보(예: 이메일 주소)를 제공할 필요가 없습니다.

AMI 준비

- 하드웨어 가상 머신(HVM) 가상화 및 64비트 아키텍처를 사용합니다.
- 알려진 취약성, 맬웨어 또는 바이러스가 없습니다.
- 구매자가 AMI에 대한 운영 체제 수준 관리 액세스 권한을 갖습니다.
- 에서 테스트 '버전 추가' AWS Marketplace Management Portal 를 사용하여 AMI를 스캔합니다.

Windows AMI

- Amazon EC2 사용 설명서의 EC2Config 서비스를 사용하여 Windows 인스턴스 구성에 설명된대로 최신 버전의 `Ec2ConfigServicefmf` 사용합니다.
- Amazon EC2 사용 설명서의 EC2Config 서비스를 사용하여 Windows 인스턴스 구성에 설명된 대로 `Ec2SetPassword`, `Ec2WindowsActivate` 및 `Ec2HandleUserData` 플러그인이 활성화됩니다.
- 게스트 계정이나 원격 데스크톱 사용자가 없습니다.

Linux AMI

- 루트 로그인이 잠겼거나 비활성화되었습니다.
- 권한 있는 키, 기본 암호 또는 다른 보안 인증 정보가 들어 있지 않습니다.
- 모든 필수 필드가 완료되었습니다.
- 모든 값이 지정된 문자 제한 내에 있습니다.
- 모든 URL이 오류 없이 로드됩니다.

- 제품 이미지는 너비가 110px 이상이고 비율은 1:1~2:1 사이입니다.
- 활성화된 모든 인스턴스 유형의 요금(시간별, 시간별_월별 및 시간별_연별 요금 모델)이 지정되었습니다.
- 월별 요금(시간별_월별 및 월별 요금 모델)이 지정되었습니다.

자동 AMI 빌드에 대해 궁금한 점이나 의견이 있는 경우 AWS Marketplace 판매자 작업 팀에 문의하세요.

에서 제품 마케팅 AWS Marketplace

에 대한 인식을 높이고의 제품 페이지로 직접 트래픽을 유도 AWS Marketplace 하여 제품의 성공에 기여할 수 있습니다 AWS Marketplace. 다음 섹션은 AWS Marketplace에 등록한 제품을 마케팅하는 데 도움이 되는 정보 및 지원을 제공합니다.

주제

- 180일 GTM 아카데미
- 제품 출시 발표
- AWS Marketplace 메시징
- 에 대한 검토 AWS Marketplace
- 에 연결 AWS Marketplace
- AWS Marketplace 브랜딩 사용
- 판매자 AWS 로를 사용한 구매 이해
- 에서 제품에 직접 연결 AWS Marketplace
- 보도 자료
- AWS Marketplace 상표 사용 지침

180일 GTM 아카데미

180일 GTM 아카데미는 모든 AWS Marketplace 판매자가 사용할 수 있으며에서 상품에 대한 수요 생성 캠페인을 구축, 활성화 및 추적하는 데 도움이 되는 셀프 서비스 go-to-market(GTM) 리소스를 제공합니다 AWS Marketplace. 다음을 할 수 있습니다.

- 목록의 수요를 빠르게 추적합니다.
- AWS 및 AWS Marketplace 가치 제안을 메시징에 통합하여 마케팅 투자 수익률을 개선하고 고객 메시징을 개선합니다.
- AWS Marketplace 시장 진출 프로그램 안내서를 따라 또는 그 안에서 진행할 수 있습니다.

AWS Marketplace Management Portal의 Marketplace Resources(Marketplace 리소스) 섹션에서 180일 GTM 아카데미에 액세스할 수 있습니다.

제품 출시 발표

에서 제품 가용성을 광범위하게 발표하는 것이 좋습니다 AWS Marketplace. 보도 자료, 트윗, 블로그 또는 기타 선호하는 미디어 채널을 활용할 수 있습니다. 당사의 상표 사용 및 보도 자료 발행에 대한 가이드라인 및 지침과 함께 포함할 수 있는 샘플 텍스트를 제공해 드렸습니다.

AWS 메시지 및 브랜드 지침 또는 음성과의 일관성을 보장하기 위해 공개하기 전에 블로그, 트윗 및 기타 보도되지 않는 릴리스 발표를 검토할 것입니다. AWS 계정 관리자에게 검토 요청을 제출합니다. 리뷰를 완료하는 데는 최대 10 영업일이 걸립니다. 트윗, 블로그 또는 보도 자료를 게시할 때 당사에 알려주세요. 그러면 가시성을 높일 수 있도록 최선을 다해 다시 게시하겠습니다.

AWS Marketplace 메시징

고객 통신문에서 AWS Marketplace를 통해 제품을 구매하는 목적, 목표 및 이점을 설명해야 할 수 있습니다. AWS Marketplace를 참조할 때 다음 메시징을 사용하세요.

란 무엇입니까 AWS Marketplace?

AWS Marketplace 는 고객이에서 실행되는 소프트웨어 및 서비스를 쉽게 찾고 비교하고 즉시 사용할 수 있는 온라인 저장소입니다 AWS. 의 방문자는 1-Click 배포를 사용하여 사전 구성된 소프트웨어를 빠르게 시작하고 사용한 만큼만 시간 또는 월 단위로 지불할 AWS Marketplace 수 있습니다.는 결제 및 결제를 AWS 처리하며 소프트웨어 요금은 고객의 AWS 청구서에 표시됩니다.

고객이 쇼핑하는 이유는 무엇인가요 AWS Marketplace?

소프트웨어를 찾고 배포하는 것은 어려울 수 있습니다.는 데이터베이스 및 애플리케이션 서버와 같은 소프트웨어 인프라, IoT 솔루션, 개발자 도구, 비즈니스 애플리케이션 등 인기 있는 판매자의 다양한 상용 및 무료 IT 및 비즈니스 소프트웨어를 AWS Marketplace 제공합니다.를 AWS Marketplace 사용하면 고객이 옵션을 비교하고 리뷰를 읽고 원하는 소프트웨어를 빠르게 찾을 수 있습니다. 그런 다음 1-Click을 사용하거나 AWS Marketplace Management Portal을 사용하여 자체 Amazon Elastic Compute Cloud 인스턴스에 배포할 수 있습니다.

소프트웨어 가격이 웹 사이트에 분명하게 게시되어 있으며 고객은 Amazon Web Services에서 이미 파일에 기재한 결제 수단을 통해 대부분의 소프트웨어를 즉시 구입할 수 있습니다. 소프트웨어 요금은 AWS 인프라 요금과 동일한 월별 청구서에 표시됩니다.

소프트웨어 또는 SaaS 판매자가 판매되는 이유는 무엇입니까 AWS Marketplace?

AWS Marketplace를 사용하면에서 실행되는 오퍼링이 있는 서비스형 소프트웨어(SaaS) 판매자는 고객 인식 향상, 간소화된 배포 및 자동 결제의 이점을 누릴 AWS 수 있습니다.

AWS Marketplace 는에서 실행되는 소프트웨어 및 서비스의 소프트웨어 및 SaaS 판매자가 개별 소프트웨어 개발자부터 대기업에 이르기까지 수십만 명의 고객에게 제품을 노출하여 신규 AWS 고객을 AWS 찾을 수 있도록 지원합니다.

에서 판매 AWS Marketplace 하면 독립 소프트웨어 공급업체(ISVs)가 비용이 많이 드는 코드 변경 없이 소프트웨어에 대한 시간당 청구를 추가할 수 있습니다. Amazon Machine Image(AMI)를에 업로드 AWS 하고 시간당 비용을 제공하기만 하면 됩니다. 결제는에서 관리하므로 판매자는 사용량 측정, 고객 계정 관리 및 결제 처리 책임을 AWS Marketplace지고 소프트웨어 개발자는 훌륭한 소프트웨어 구축에 더 많은 시간을 할애할 수 있습니다.

뿐만 아니라 고객이 미리 구성된 소프트웨어 이미지를 손쉽게 배포할 수 있어 새로운 고객의 온보딩이 간소화된다는 이점이 있습니다.

에 대한 검토 AWS Marketplace

AWS Marketplace 는 고객이 제품에 대한 리뷰를 제출할 수 있는 기능을 제공합니다. 또한 독립적인 제품 리뷰를 큐레이팅하는 B2B 마켓플레이스인 G2와 같은 신디케이트 리뷰어가 AWS Marketplace에서 신디케이트 리뷰를 통합할 수 있는 기능도 제공합니다.

AWS Marketplace 고객 리뷰는 구매자용 사용 설명서에 나열된 리뷰 지침을 충족해야 합니다. 검토 제출은 제출이 검토 기준을 충족하는지 확인하기 위해 검토될 AWS Marketplace 때까지에 릴리스되지 않습니다. 리뷰 지침에 대한 자세한 내용은 제품 리뷰를 참조하세요. 신디케이트 검토 조직은 고유한 프로세스를 사용하여 검토를 검증하며 릴리스 AWS Marketplace 전에에서 검토하지 않습니다. 귀사의 제품에 대한 신디케이트 리뷰가 제품 리뷰 지침을 충족하지 않거나 제품 리뷰에 잘못된 내용이 포함되어 있다고 생각되면 AWS Marketplace 판매자 운영 팀에 문의하세요.

또한 리뷰어는 별 5개 평점 시스템에 따라 제품에 대한 별 평점을 부여할 수 있습니다. 각 평점은 평균되어 제품에 대한 전체 별 평점을 제공합니다. 신디케이트 리뷰에는 별 등급도 포함될 수 있지만 신디케이트 리뷰의 별 등급은 AWS 고객 별 등급과 함께에서 평균화되지 않습니다.

다음은 제품 리뷰 기능에 대해 추가로 알아야 할 주요 사항입니다.

- 에서 제품 리뷰를 제거할 수 없습니다 AWS Marketplace. 하지만 의견이 제품 리뷰에 대한 리뷰 기준을 충족하는 한 리뷰에 대한 의견을 남길 수 있습니다.
- 리뷰가 리뷰 지침을 충족하지 않거나 잘못된 내용이 포함되어 있다고 생각되면 AWS Marketplace 판매자 운영 팀에 연락하여 문제를 설명할 수 있습니다.
- AWS 에서 제품을 검색하는 고객은 등급, 확인된 리뷰 및 외부에서 소싱된 리뷰를 기반으로 결과를 검색하고 필터링할 AWS Marketplace 수 있습니다. AWS 고객은 검색 결과에서 AWS 고객 등급과 함께 외부에서 소싱된 등급을 볼 수 있습니다.

- 제품에 대한 신디케이트 리뷰는 무료로 AWS Marketplace 에 자동으로 추가됩니다. 리뷰가 자동으로 추가되므로 신디케이트 리뷰를 추가하도록 요청을 제출할 필요가 없습니다.
- 제품에 대한 신디케이트 리뷰가 없는 경우 신디케이트 리뷰어에 연락하고 제품 리뷰를 받는 절차를 따를 수 있습니다. 예를 들어 G2의 경우 웹 사이트를 방문하여 제품 페이지를 제공하는 것으로 리뷰 프로세스를 시작할 수 있습니다.

에 연결 AWS Marketplace

제품을 설명하고 홍보하는 회사 웹 서비스가 있다면 제품을에서 실행할 수 AWS 있으며를 사용하여 구매할 수 있음을 강조하는 것이 좋습니다 AWS Marketplace. 고객이 소프트웨어를 검색하고 배포하는 프로세스를 단순화하기 위해 당사는 고객을 제품에 연결하는 지침을 제공해 드렸습니다.

AWS Marketplace 브랜딩 사용

> **Note**
>
> 여기에 나열된 가이드와 자산을 사용하려면 AWS 파트너여야 합니다. AWS 파트너가 되는 방법에 대한 자세한 내용은 AWS 파트너 네트워크 가입을 참조하세요.

AWS Marketplace 는 소프트웨어, 데이터 또는 서비스를 홍보하는 데 도움이 되는 메시징 지침과 브랜딩 자산을 제공합니다. 목록을 승격하려면 다음 리소스를 다운로드할 AWS Marketplace수 있습니다.

- AWS 파트너 크리에이티브 및 메시징 가이드 2023
- 브랜딩 자산에서 사용 가능 AWS Marketplace

리소스를 다운로드하려면

1. AWS Partner Central에 로그인합니다.
2. 브라우저의 주소 표시줄에 다음 URL을 붙여 넣고 AWS Partner Creative and Messaging Guide 2023 페이지를 엽니다.
3. 다운로드를 선택합니다.
4. 브랜딩 자산에서 사용 가능 AWS Marketplace에 대해 2단계와 3단계를 반복합니다.

판매자 AWS 로를 사용한 구매 이해

를 사용하여 구매 AWS하면 AWS Marketplace 판매자가 자체 웹 사이트에서 AWS Marketplace 제품을 표시하고 판매할 수 있습니다. AWS Marketplace 판매자는 CTA(AWS Marketplace call-to-action) 버튼 및 APIs와 통합하여 판매자 소유 웹 사이트에서 AWS기반 소프트웨어 검색 및 조달 경험을 구축할 수 있습니다. 여기에는 독립 소프트웨어 공급업체(ISVs), 채널 파트너(CPs), 배포자(DPs)를 포함하여 AWS 판매자가 소유한 웹 속성이 포함됩니다.

> **Note**
>
> AWS 기능을 사용한 구매에 대한 자세한 내용은 를 사용한 구매를 AWS 참조하세요.

CTA 버튼에는 다음이 포함됩니다.

- 를 사용하여 구매 AWS
- 에서 제안 보기 AWS
- 를 사용하여 무료 체험 AWS
- 비공개 제안 요청
- 데모 요청

를 사용한 구매는 다음과 같은 이점을 제공할 AWS 수 있습니다.

- AWS 웹 사이트에서 공동 브랜딩 및 검색: 웹 사이트에서 기반 공동 브랜딩 환경을 생성합니다 AWS Marketplace. 고객이 웹 사이트에서 큐레이션된 제품 목록을 검색하고, 제품을 필터링하고, 리뷰 AWS Marketplace 에 액세스하고, 사용자 지정 요금을 요청하고, 계정을 사용하여 자신 있게 구매할 수 있습니다 AWS .
- AWS 고객의 조달 속도: 웹 사이트의 로 구매 AWS 버튼을 사용하면 고객이 무료 평가판을 빠르게 구독하거나 AWS 계정을 사용하여 구매할 수 있습니다. 를 사용한 구매는를 통해 고객에게 간소화된 공동 브랜드 구매 경험을 AWS 제공합니다 AWS Marketplace.
- 의 기존 운영 활용 AWS Marketplace: 웹 사이트를 통해 AWS Marketplace 고객에게의 검색 및 조달 이점을 제공하는 동시에의 기존 프로세스를 활용하여 추가 노력 없이 목록, 지출, 지원 및 인사이트를 AWS Marketplace 관리합니다.

AWS 사전 조건을 사용하여 구매

를 사용한 구매를 활성화하려면 다음을 AWS수행해야 합니다.

- 에 판매자로 등록해야 합니다 AWS Marketplace.
- 에서 승인된 판매자 또는 리셀러여야 AWS Marketplace 하며 판매자를 위한 AWS Marketplace 서비스 약관에 동의해야 합니다.
- AWS 상표 지침을 준수합니다.
- AWS 상표를 사용한 구매 라이선스를 준수합니다.
- 하나 이상의 퍼블릭 SaaS 제품이 목록에 있어야 합니다 AWS Marketplace.

> **Note**
>
> 현재를 사용한 구매는 SaaS 제품만 AWS 지원합니다.

- AWS 브랜딩 지침으로 구매 준수:
  - AWS Partner Central 계정에 로그인하여 브랜딩 요소 및 마케팅 자산에 대해 따라야 할 AWS Marketplace 브랜딩 자산 및 사양이 포함된 마케팅 툴킷을 다운로드합니다.
  - 브랜딩 요소 및 마케팅 자산 사용에 대한 지침은 AWS 크리에이티브 및 메시징과 함께 구매 지침을 참조하세요.

AWS로 구매: 판매자로 시작하기

브랜딩 변경 사항 검토

AWS 경험으로 구매 시작하기 전에 브랜드 검토 요청을 제출하려면 다음 단계를 완료하세요.

1. Commercial Marketplace 지원 티켓으로 이동하여 AWS Marketplace Management Portal 제출 https://aws.amazon.com/marketplace/management/contact-us/?
   a. 범주로 를 사용하여 구매 AWS 선택
   b. 브랜딩 검토를 하위 범주로 선택합니다.
2. 티켓에 다음을 입력합니다.
   a. 기본 이메일 주소 및 판매자 ID.
   b. 요청에 대한 세부 정보 제공 섹션에서 다음을 수행합니다.

   i. AWS 통합으로 구입을 구현할 URL입니다.
   ii. AWS Marketplace 운영 팀이 검토할 수 있는 샌드박스, 라이브 사이트 또는 모형에 대한 링크입니다. 이를 포함하지 않으면 티켓을 제출한 후 운영 팀이 이를 요청합니다.
3. AWS Marketplace 판매자 운영 팀은 AWS 브랜드 지침으로 구매와 비교하여 구현을 평가합니다. 규정 준수를 보장하기 위한 피드백을 제공합니다. 시작하기 전에 모든 피드백을 해결하고 승인을 받아야 합니다.

AWS 보고로 구매

AWS 지표를 사용하여 구매하면 판매자에게 변환 퍼널에 대한 포괄적인 보기를 제공합니다. AWS 판매자와 함께 구매 대시보드는 사용자 참여를 측정하기 위한 자세한 인사이트를 제공합니다. 판매자는 를 통해 AWS 지표로 구매에 액세스할 수 있습니다AWS Marketplace Management Portal.

AWS 지표로 구매 대시보드를 보려면:

1. AWS Marketplace Management Portal에 로그인합니다.
2. 인사이트로 이동합니다.
3. 마케팅을 클릭합니다.
4. 로 구매 AWS 클릭

또는 계정을 연결한 판매자는 Partner Central을 통해 AWS 지표에 액세스할 수 있습니다. 이러한 지표를 보려면 AWS Partner Central에 로그인하고 분석 섹션으로 이동합니다.

에서 제품에 직접 연결 AWS Marketplace

웹 사이트 또는 마케팅 자료에 대한 딥 링크를 추가하여 고객을의 특정 제품 페이지로 안내합니다 AWS Marketplace. 다음 형식을 사용합니다.

```
https://aws.amazon.com/marketplace/pp/prodview-prodview ID
```

*prodview ID*를 제품의 고유한 영숫자 코드로 바꿉니다 AWS Marketplace. 예시:

Example

```
https://aws.amazon.com/marketplace/pp/prodview-1234567890abcdef
```

aws.amazon.com/marketplace에서 애플리케이션을 검색할 때 URL에 prodview ID가 표시됩니다. prodview ID를 찾는 데 도움이 필요하면 계정 관리자에게 문의하세요.

> **Note**
>
> 링크를 사용하기 전에 고객이 올바른 페이지로 연결되는지 테스트합니다.

보도 자료

원하는 채널을 AWS Marketplace 통해에서 제품의 가용성을 알리는 것이 좋습니다. 그러나 Amazon 은 게시 또는 발표 전에 참조하는 모든 보도 자료를 검토하고 승인 AWS Marketplace 해야 합니다. 발표할 것을 권장하지만 AWS Marketplace 판매자와의 공동 보도 자료는 지원할 수 없습니다. case-by-case 따옴표가 포함된 보도 자료를 지원합니다 AWS. 견적은에 나열된 새 제품 또는 서비스를 발표 AWS Marketplace 하거나에서 사용하는 고객 참조를 포함하는 여러 조건을 충족해야 합니다 AWS Marketplace.

모든 보도 자료는 직접 기획해야 합니다. [제품 이름 삽입] 이제 사용할 수 있습니다 AWS Marketplace. 일관성을 위해 본 문서에 수록된 메시징을 사용합니다.

보도 자료에서 준수할 사항

- 발표 내용이 Amazon.com과 어떤 관련이 있는지 분명하고 정확하게 설명합니다.
- 고객에 대한 역할과 고객과의 AWS 역할 명료화
- 고객 중심의 입장을 유지하고 고객이 누릴 수 있는 이점을 강조합니다.

보도 자료에서 기피할 사항

- 파트너, 파트너십 또는 제휴라는 용어를 사용하여 관계를 설명하지 마세요. 계약, 같은 팀 또는 관계와 같은 용어가 바람직합니다.
- 사전 동의 없이 Amazon Web Services 임원이 한 말을 인용하지 마세요.
- 매출 전망을 포함하지 마세요. 웹 사이트를 나타내는 것이 아니라면 귀사 소개 문구에 판매자로 .com을 사용하지 마세요.
- 귀사를 Amazon.com 제휴사로 칭하지 마세요. 당사의 온라인 제휴 프로그램인 Amazon Associates와 혼동할 수 있기 때문입니다.
- Amazon.com에 대한 독점 정보를 공개하거나 주식 시세 표시 기호를 참조하지 마세요.

보도 자료를 텍스트 형식으로 담당 계정 관리자에 제출하여 사전에 검토를 받으세요. 또한 상표를 사용하기 전에 Amazon Web Services 상표 지침을 검토하세요. AWS AWS Marketplace 상표 관련 가이드라인은 다음 단원에 나와 있습니다.

AWS Marketplace 상표 사용 지침

이 지침은 Amazon.com, Inc. 및/또는 그 계열사(“Amazon”)가 사전에 승인한 자료에서 AWS Marketplace 로고 및 상표(각각 “상표” 및 총칭하여 “상표”)를 사용하는 데 적용됩니다. 본 지침을 항상 엄격히 준수해야 하며 본 지침을 위반하여 상표를 사용하는 경우 상표 사용과 관련된 모든 라이선스가 자동으로 해지됩니다.

1. Amazon에서 분명히 승인한 목적으로만 상표를 사용할 수 있으며 (i) 상표의 사용과 관련하여 Amazon과 체결한 모든 계약의 최신 버전(“계약”으로 통칭)을 준수하고 (ii) 본 지침의 최신 버전을 준수하며 (iii) 때때로 상표의 사용에 적용하도록 Amazon이 제기할 수 있는 다른 모든 조건이나 정책을 준수하여 상표를 사용해야 합니다.
2. Amazon에서는 귀하가 사용할 수 있도록 승인된 상표를 제공합니다. 상표의 비율, 색상 또는 글꼴을 변경하거나 상표의 요소를 추가 또는 제거하는 등 어떠한 방식으로도 상표를 변경해서는 안 됩니다.
3. 본 계약에 의해 특별히 허가한 대로 상표를 사용하는 것 외에 Amazon의 후원 또는 승인을 암시하는 어떤 방식으로도 상표를 사용해서는 안 됩니다.
4. Amazon의 단독 재량으로 상표에 포함된 Amazon의 영업권을 약화시키거나 다른 식으로 손상시키거나 퇴색시킬 수 있는 방식으로 상표를 사용하여 Amazon, Amazon의 제품 또는 서비스를 훼손해서는 안 됩니다.
5. 상표와 다른 시각적, 그래픽 또는 텍스트 요소 양쪽에 적당한 간격을 두어 상표를 따로 표시해야 합니다. 어떠한 상황에서도 상표의 가독성이나 표시를 방해하는 배경에 상표를 배치해서는 안 됩니다.
6. 상표를 표시하는 모든 자료에 “AWS Marketplace 및 AWS Marketplace 로고는 Amazon.com, Inc. 또는 그 계열사의 상표입니다.
7. 귀하는 상표에 대한 모든 권리가 Amazon의 독점적 소유물이며 귀하가 상표를 사용함으로써 발생하는 모든 영업권이 Amazon의 이익에 적용된다는 사실을 인정합니다. 귀하는 상표에 포함된 Amazon의 권리 또는 소유권에 위배되는 어떠한 행동도 하지 않습니다.

Amazon은 본 지침 및/또는 승인된 상표를 언제든지 수정할 수 있고, 허가를 받지 않은 사용 또는 본 지침에 맞지 않는 사용에 대하여 적절한 조치를 취할 수 있으며, 단독 재량에 따라 행사할 수 있는 권리를 갖습니다. 본 지침에 대한 문의 사항이 있을 경우 trademarks@amazon.com으로 문의하거나 다음 주소를 통해 우편으로 보내 주세요.

Amazon.com, Inc., Attention: Trademarks

PO Box 81226

Seattle, WA 98108-1226

AWS Marketplace 이벤트에 대한 판매자 알림

이메일, Amazon EventBridge 이벤트 및 Amazon Simple Notification Service(Amazon SNS) topics. AWS Marketplace provides를 통해 AWS Marketplace 제품과 관련된 알림을 적시에 받을 수 있습니다. 예를 들어 비공개 제안, 전문 서비스 제품 요청 및 반복 스캔 알림과 관련된 알림을 받을 수 있습니다. 이 주제에서는 사용 가능한 알림 및 이벤트 옵션에 대한 개요를 제공합니다.

- 이메일 알림 - 내에서 AWS Marketplace 판매자, 구매자 및 독립 소프트웨어 공급업체(ISVs)는 이메일 알림을 받을 수 있습니다.는 루트 계정을 AWS Marketplace 사용하여 AWS 계정의 이메일로 자동 이메일을 실시간으로 전송하여 AWS Marketplace 제안 및 계약에 대한 업데이트를 제공합니다. 알림에 대한 사용자 지정 이메일 별칭을 추가하고 이메일 알림에서 수신자 구독을 취소할 수도 있습니다. 자세한 내용은 AWS Marketplace 이벤트에 대한 이메일 알림 관리 단원을 참조하십시오.
- Amazon SNS 주제 - 제품의 고객 구독 변경 사항에 대한 알림을 받으려면 제품 생성 중에 AWS Marketplace 제공되는에 대한 Amazon SNS 주제를 구독하면 됩니다. 예를 들어 고객이 언제 비공개 제안을 수락하는지 알 수 있습니다. 자세한 내용은 제품 유형에 대한 Amazon SNS 주제를 참조하세요.
  - 서비스형 소프트웨어(SaaS) 제품
  - Amazon Machine Image(AMI) 제품
  - 컨테이너 제품
- EventBridge - EventBridge를 사용하여 제안이 생성될 때와 AWS Marketplace같이에서 발생하는 작업에 대한 이벤트를 수신할 수 있습니다. 이벤트에는 ID, 만료 날짜, 제품 세부 정보 등의 정보가 포함되어 있습니다. 자세한 내용은 Amazon EventBridge 이벤트 및 Amazon EventBridge 사용 설명서를 참조하세요.

다음 주제에서는 AWS Marketplace의 알림 및 이벤트에 대한 자세한 정보를 제공합니다.

주제

- AWS Marketplace 이벤트에 대한 이메일 알림 관리
- Amazon EventBridge 이벤트

AWS Marketplace 이벤트에 대한 이메일 알림 관리

AWS Marketplace 는 제안, 계약, 구독, 제품, 보안, 결제 및 결제, Private Marketplace에 업데이트에 대한 이메일 알림을 보냅니다. 독립 소프트웨어 공급업체(ISVs), AWS Marketplace 채널 파트너 및 고객

은 이메일 알림을 받을 수 있습니다. 제안 및 계약 업데이트를 위해 전송된 이메일 알림의 예제와 세부 정보는 이벤트 유형 섹션을 참조하세요.

AWS Marketplace 는 AWS 계정의 루트 사용자와 연결된 이메일 주소로 이메일 알림을 보냅니다. AWS 계정과 연결된 이메일 주소를 업데이트하려면 AWS 계정의 기본 연락처 업데이트를 참조하세요. 알림에 대한 사용자 지정 이메일 별칭을 추가하고 이메일 알림에서 수신자 구독을 취소할 수도 있습니다.

> **Note**
>
> AWS Marketplace 이메일이 누락된 경우 스팸 폴더를 확인하거나 이메일 설정을 조정합니다. 의 이메일 알림 AWS Marketplace 은에서 전송됩니다`no-reply@marketplace.aws`. Google 및 Yahoo와 같은 공급자는 이러한 데이터를 필터링할 수 있습니다. 지침은 유효한 이메일이 스팸(Google)으로 가는 것을 방지 또는 Block and unblock email addresses in Yahoo Mail을 참조하세요.

주제

- 이벤트 유형
- 필드 설명
- 알림 관리

이벤트 유형

다음 이벤트 유형은 기계 학습 제품을 제외한 모든 제품 및 가격 유형에 대한 이메일 알림에서 지원됩니다.

제안

다음 테이블은 제안을 위한 이벤트를 보여줍니다. 제안은 구매자의 제품 사용에 대한 일련의 조건입니다. 자세한 정보는 AWS Marketplace 제품에 대한 비공개 제안 준비 섹션을 참조하세요.

이메일	Event	수신자	Title	필드
제안 게시됨	AWS Marketplace 채널	채널 파트너	AWS Marketplace 채널	고객 AWS 계정 ID, ISV 이름, ISV AWS 계정 ID, 제품 이름, 제품 ID, 제안 ID, 기회 ID, 제안 게시 날짜, 제안 만료 날짜

이메일	Event	수신자	Title	필드
CP 알림 CPPO	파트너 비공개 제안 게시.		파트너 비공개 제안 게시	
제안 게시됨 ISV 알림 CPPO	AWS Marketplace 채널 파트너 비공개 제안 게시.	ISV	AWS Marketplace 채널 파트너 비공개 제안 게시	고객 AWS 계정 IDs, 채널 파트너, 채널 파트너 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 기회 ID, 제안 게시 날짜, 제안 만료 날짜, 도매 가격
OfferPublSVNotificationSCP-1.0	판매자가 AWS Marketplace 채널 파트너 비공개 제안을 게시합니다.	ISV 또는 AWS Marketplace 채널 파트너	비공개 제안 게시	고객 AWS 계정 IDs, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 제안 게시 날짜, 제안 만료 날짜, 총 계약 금액
리셀러 기회 만료됨	AWS Marketplace 채널 파트너 판매 권한 부여가 만료됩니다.	AWS Marketplace 채널 파트너	제안: 만료된 판매 권한 보기	ISV, 제품 이름, 제품 ID

이메일	Event	수신자	Title	필드
리셀러 기회 취소됨	ISV가 AWS Marketplace 채널 파트너 판매 권한 부여를 취소합니다.	AWS Marketplace 채널 파트너	제안: 비활성화된 판매 권한 보기	ISV, 제품 이름, 제품 ID
리셀러 기회 생성됨	ISV는 AWS Marketplace 채널 파트너를 위한 판매 권한 부여 또는 기회를 생성합니다.	AWS Marketplace 채널 파트너	제안: 새 판매 권한을 위한 비공개 제안 생성	ISV, 제품 이름, 제품 ID, 판매 승인 기간

계약

ISVs와 AWS Marketplace 채널 파트너는 구매자가 공개 제안, 비공개 제안 또는 AWS Marketplace 채널 파트너 비공개 제안을 수락하면 이메일 알림을 받습니다. 제안이 수락되면 계약이 만들어집니다. 계약 수명 주기 이벤트에 대한 이메일 알림도 전송됩니다. 여기에는 구매자 수락 후 계약 시작, 취소, 교체 및 실패가 포함됩니다. 다음 테이블에는 계약 이벤트에 대해 전송되는 이메일이 나와 있습니다. 이메일 알림은 현재 일부 제품 유형에 사용할 수 있습니다. 특정 제품 유형에는 아직이 기능이 없을 수 있습니다.

이메일	Event	수신자	Title	필드
제안 수락 - 비공개 또는 공개 제안	AWS Marketplace 고객이 공개 또는 비공개 제안을 수락합니다.	ISV	고객이 AWS Marketplace 제안을 수락했습니다.	고객 회사 이름, 고객 AWS 계정 ID, 제품 이름, 제품 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 계약 수락 날짜, 구매 금액
제안 수락됨 - AWS Marketplace 채널 파트너 비공개 제안입니다.	AWS Marketplace 고객이 AWS Marketplace 채널 파트너 비공개 제안을 수락합니다.	ISV	고객이 AWS Marketplace 채널 파트너 비공개 제안을 수락했습니다.	고객 회사 이름, 고객 AWS 계정 ID, 채널 파트너 이름, 채널 파트너 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 계약 수락 날짜, 도매 가격
제안 수락됨 - AWS Marketplace 채널 파트너 비공개 제안입니다.	AWS Marketplace 고객이 AWS Marketplace 채널 파트너 비공개 제안을 수락합니다.	AWS Marketplace 채널 파트너	고객이 AWS Marketplace 채널 파트너 비공개 제안을 수락했습니다.	고객 회사 이름, 고객 AWS 계정 ID, ISV 이름, ISV AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 계약 수락 날짜, 도매 가격, 마진, 구매 금액

이메일	Event	수신자	Title	필드
계약 시작 - 비공개 제안	계약 또는 구독 제품에 대한 AWS Marketplace 계약은 향후 시작일이 있는 비공개 제안에서 시작되었습니다.	ISV	AWS Marketplace 계약이 시작되었습니다.	고객 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 구매 금액
계약 시작 - AWS Marketplace 채널 파트너 비공개 제안	계약 또는 구독 제품에 대한 AWS Marketplace 계약이 AWS Marketplace 채널 파트너 비공개 제안에서 시작되었습니다.	ISV	AWS Marketplace 계약이 시작되었습니다.	고객 AWS 계정 ID, 채널 파트너 이름, 채널 파트너 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 도매 가격

이메일	Event	수신자	Title	필드
계약 시작됨 - 비공개 제안	AWS Marketplace 채널 파트너 비공개 제안을 수락하면 계약 또는 구독 제품에 대한 계약이 시작됩니다.	AWS Marketplace 채널 파트너	AWS Marketplace 계약이 시작되었습니다.	고객 AWS 계정 ID, ISV 이름, ISV AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 마진, 구매 금액
계약 취소됨 - 비공개 또는 공개 제안	비공개 또는 공개 제안 계약 취소입니다.	ISV	AWS Marketplace 계약이 취소되었습니다.	고객 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜
계약 취소됨 - AWS Marketplace 채널 파트너 비공개 제안	AWS Marketplace 채널 파트너 비공개 제안 계약의 취소.	ISV	AWS Marketplace 계약이 취소되었습니다.	고객 AWS 계정 ID, 채널 파트너, 채널 파트너 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜

이메일	Event	수신자	Title	필드
계약 취소됨 - 채널 파트너 비공개 제안	AWS Marketplace 채널 파트너 비공개 제안 계약의 취소.	AWS Marketplace 채널 파트너	AWS Marketplace 계약이 취소되었습니다.	고객 AWS 계정 ID, ISV, ISV AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID, 계약 시작 날짜, 계약 종료 날짜
규정 준수 실패 - 공식 판매자	고객 결제 실패로 인해 AWS Marketplace 계약이 실패합니다.	제안을 생성한 ISV 또는 AWS Marketplace 채널 파트너	필요한 작업: AWS Marketplace 계약 생성 실패	구독 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID
결제 실패 - 공식 판매자	고객 결제 실패로 인해 AWS Marketplace 계약이 실패합니다.	제안을 생성한 ISV 또는 AWS Marketplace 채널 파트너	AWS Marketplace 계약에 대한 결제 실패	고객 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 계약 ID
계약이 대체됨 - 비공개 제안	AWS Marketplace 계약이 다른 계약으로 대체됩니다.	ISV	AWS Marketplace 계약이 대체되었습니다.	회사 이름, 고객 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 새 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 새 구매 금액

이메일	Event	수신자	Title	필드
계약 대체됨 - 채널 파트너 비공개 제안	AWS Marketplace 계약이 다른 계약으로 대체됩니다.	ISV	AWS Marketplace 계약이 대체되었습니다.	회사 이름, 고객 AWS 계정 ID, 채널 파트너, 채널 파트너 AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 새 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 도매 가격
계약 대체됨 - 채널 파트너 비공개 제안	AWS Marketplace 계약이 다른 계약으로 대체됩니다.	AWS Marketplace 채널 파트너	AWS Marketplace 계약이 대체되었습니다.	회사 이름, 고객 AWS 계정 ID, ISV 이름, ISV AWS 계정 ID, 제품 이름, 제품 ID, 제안 이름, 제안 ID, 새 계약 ID, 계약 시작 날짜, 계약 종료 날짜, 도매 가격, 마진, 새 구매 금액

지급

ISVs와 채널 파트너는 잘못된 은행 계좌 세부 정보로 인해 지급이 일시 중지되면 이메일 알림을 받게 됩니다.

이메일	Event	수신자	Title	필드
지급 일시 중지됨 - 잘못된 은행 계좌 정보	AWS Marketplace 는 잘못된 은행 계좌 세부 정보로 인해 판매자에 대한 지급을 일시	ISV 및 AWS Marketplace 채널 파트너	지급 일시 중지됨	리소스 ARN

이메일	Event	수신자	Title	필드
	중지합니다.			

`Resource ARN` 필드에 잘못된 은행 계좌 Amazon 리소스 번호(ARN)가 표시됩니다. AWS Marketplace Management Portal(AMMP)에서 은행 계좌를 추가하여이 문제를 해결할 수 있습니다.

은행 계좌 세부 정보 추가

AWS Marketplace Management Portal(AMMP)에 은행 계좌 세부 정보를 추가하여 지급에서 잘못된 은행 계좌 ARN 오류를 수정합니다.

은행 계좌 세부 정보를 추가하려면

1. AWS Marketplace 판매자 계정으로 AWS Marketplace Management Portal에 로그인합니다.
2. 메뉴 모음에서 설정을 선택합니다.
3. 결제 정보 탭을 선택합니다.
4. 은행 계좌 섹션에서 은행 계좌 추가를 선택합니다.
5. 은행 계좌 세부 정보를 추가하고 정확성을 검토합니다. 자세한 정보는 3단계: 은행 계좌 정보 제공 섹션을 참조하세요.
6. 구매자로부터 결제를 받으려면 지급 방법에서 지급 방법 추가를 선택합니다.
7. 고객 알기(KYC) 및 은행 계좌를 확인하라는 메시지가 표시되면 확인 단계를 따릅니다.
8. 변경 내용을 저장합니다.

필드 설명

다음 표에는 제안 및 계약 테이블에서 참조되는 필드에 대한 설명이 나와 있습니다.

필드	설명			
고객 회사 이름	구독자의 회사 이름입니다.			

필드	설명
고객 AWS 계정 ID	제품을 구독한 AWS 계정의 ID 입니다.
ISV 이름	판매자 회사 이름입니다.
ISV AWS 계정 ID	판매자 AWS 계정의 ID입니다.
제품 이름	제품의 제목입니다.
제품 ID	소프트웨어 제품의 익숙한 고유 식별자입니다.
제안 이름	제안의 제목입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제안 표시 여부	공개 제안인지, 비공개 제안인지 또는 기업 계약 제안인지 나타냅니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.

필드	설명			
계약 시작 날짜	고객의 제품 구독이 시작되는 날짜로, MM-DD-YYYY 형식입니다. 미래 날짜의 계약인 경우 이 날짜가 수락 날짜와 다를 수 있습니다.			
계약 수락 날짜	고객이 제품을 구독한 날짜로, MM-DD-YYYY 형식입니다.			
계약 종료 날짜	계약이 만료되는 날짜로, MM-DD-YYYY 형식입니다. 과금형/종량제 구독의 경우 이 날짜는 JAN-1-9999 로 설정됩니다.			
계약 종료 날짜	계약이 만료되는 날짜로, MM-DD-YYYY 형식입니다. 과금형/종량제 구독의 경우 이 날짜는 JAN-1-9999 로 설정됩니다.			

필드	설명			
구매 금액	계약의 예상 비용으로, 총 계약 가치라고도 합니다. 이는 SaaS, 전문 서비스 및 서버 제품 유형과 계약 또는 연간 요금 유형에 적용됩니다.			
채널 파트너 회사 이름	고객에게 재판매할 목적으로 ISV로부터 도매로 제품 또는 서비스를 구매한 계정의 이름입니다.			
채널 파트너 AWS 계정 ID	고객에게 재판매하기 위해 ISV에서 제품 또는 서비스를 구매한 AWS Marketplace 채널 파트너의 AWS 계정 ID입니다.			
도매 가격	제품을 AWS Marketplace 채널 파트너에게 재판매하기 위한 ISV의 도매 비용입니다.			

필드	설명
통화 코드	계약의 예상 비용과 관련된 제안 요금 통화입니다.
다음 계약 ID	갱신되거나 업그레이드된 계약의 계약 ID입니다.
제안 게시 날짜	판매자가 제안을 게시한 날짜입니다.
제안 만료 날짜	제안이 만료되는 날짜입니다.
기회 ID	등록된 기회에 대한 고유 식별자입니다.
판매 권한 부여 기간	판매 권한에 지정된 대로 리셀러가 할인을 사용하여 제안을 생성할 수 있는 권한이 부여된 기간입니다.

알림 관리

다음 주제에서는 이벤트에 대한 이메일 알림을 관리하는 방법을 설명합니다.

이메일 주소 추가 또는 업데이트

AWS Marketplace Management Portal을 사용하여 사용자 지정 이메일 알림을 받을 이메일 주소를 10개까지 추가할 수 있습니다.

이메일 주소를 추가 또는 업데이트하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 설정에서 알림 탭을 선택합니다.
3. 사용자 지정 알림을 위한 이메일에서 이메일 주소 추가를 선택합니다.
4. 수신자 세부 정보의 이메일 주소 필드에 사용자 지정 이메일 주소를 입력합니다.
5. (선택 사항) 새 수신자 추가를 선택하여 다른 이메일 주소를 추가합니다(최대 10개).
6. 제출을 선택합니다.

알림에서 수신자 구독 해제

이메일 주소를 제거하여 사용자 지정 이메일 알림에서 수신자를 구독 해제할 수 있습니다.

이벤트 알림에서 수신자를 구독 해제하는 방법

1. AWS Marketplace Management Portal에 로그인합니다.
2. 설정에서 알림 탭을 선택합니다.
3. 사용자 지정 알림을 위한 이메일에서 이메일 주소 업데이트를 선택합니다.
4. 수신자 세부 정보에서 제거를 선택하여 이메일 주소를 제거합니다.
5. 제출을 선택합니다.

> **Note**
> 이메일의 링크를 사용하여 구독 해제할 수도 있습니다.

Amazon EventBridge 이벤트

판매자는 Amazon EventBridge를 사용하여 AWS Marketplace의 이벤트에 대한 알림을 받을 수 있습니다. 예를 들어 제안이 생성될 AWS Marketplace 때에서 이벤트를 수신할 수 있습니다. 이벤트에는 ID, 만료 날짜, 제품 세부 정보 등의 세부 정보가 포함되어 있습니다. EventBridge는 애플리케이션을 다양한 소스의 데이터와 연결하는 데 사용할 수 있는 이벤트 버스 서비스입니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요. 다음 섹션에서는 EventBridge 콘솔의 Marketplace 카탈로그 서비스에 있는 이벤트에 대한 자세한 정보를 제공합니다.

주제

- 새 제안에 대한 이벤트
- 변경 세트에 대한 이벤트
- 보안 요약 보고서 이벤트
- 지급 이벤트

이 주제

판매자 조치	이벤트 수신	관련 주제
독립 소프트웨어 개발 판매 회사(ISV)가 제안을 생성하고 구매할 수 있게 설정	`Offer Released`	the section called “새 제안에 대한 이벤트”
채널 파트너가 ISV의 제품을 사용하여 제안 생성	`Offer Released`	the section called “새 제안에 대한 이벤트”
채널 파트너가 제안 생성	`Offer Released`	the section called “새 제안에 대한 이벤트”
변경 세트 성공	`Change Set Succeeded`	the section called “변경 세트에 대한 이벤트”
변경 세트 실패	`Change Set Failed`	the section called “변경 세트에 대한 이벤트”
변경 세트가 취소됨	`Change Set Cancelled`	the section called “변경 세트에 대한 이벤트”
ISV 제품에서 보안 취약성이 감지됨	`Products Security Report Created`	the section called “보안 요약 보고서 이벤트”

새 제안에 대한 이벤트

판매자가 제안을 생성하고 구매할 수 있게 만들면 `Offer Released`라는 세부 정보 유형의 이벤트를 받을 수 있습니다.

> **Note**
>
> EventBridge 규칙 생성에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 규칙을 참조하세요.

다음은 ISV가 생성한 새 제안에 대한 이벤트 본문 예시입니다.

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

다음은 채널 파트너가 ISV의 제품을 사용하여 제안을 생성할 때의 이벤트 본문 예시입니다.

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

다음은 채널 파트너가 제안을 생성할 때의 이벤트 본문 예시입니다.

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

변경 세트에 대한 이벤트

변경 세트가 완료되면 판매자, 채널 파트너 및 프라이빗 마켓플레이스 관리자가 이벤트를 받을 수 있습니다. AWS Marketplace Catalog API는 변경 세트가 성공, 실패 또는 취소됨 상태로 완료되면 이벤트를 전송합니다. 이러한 이벤트의 소스는 `aws.marketplacecatalog`이며 가능한 세부 정보 유형 값은 `Change Set Succeeded`, `Change Set Failed` 및 `Change Set Cancelled`입니다.

> **Note**
>
> 변경 세트에 대한 자세한 내용은 AWS Marketplace 카탈로그 API 참조의 변경 세트 작업을 참조하세요.

각 이벤트에는 변경 세트 ID, 변경 세트 이름, 이벤트 세부 정보 유형, 실패 코드(실패한 요청의 경우), 요청 시작 및 종료 시간과 같은 변경 요청 세부 정보가 포함됩니다. 이렇게 하면 `DescribeChangeSet` 작업을 지속적으로 쿼리하거나에서 변경 요청의 AWS Marketplace Management Portal 상태를 확인하지 않고도 변경 세트를 모니터링할 수 있습니다.

> **Note**
>
> EventBridge 규칙 생성에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 규칙을 참조하세요.

다음은 `Change Set Succeeded` 세부 정보 유형에 대한 이벤트 본문 예시입니다.

```
{
  "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"
  }
}
```

다음은 `Change Set Failed` 세부 정보 유형에 대한 이벤트 본문 예시입니다.

```
{
  "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"
  }
}
```

다음은 `Change Set Cancelled` 세부 정보 유형에 대한 이벤트 본문 예시입니다.

```
{
  "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"
  }
}
```

보안 요약 보고서 이벤트

판매자의 제품에서 보안 취약성이 감지되면 요약 보고서 이벤트와 미해결 제품 문제에 대한 정기 알림을 받을 수 있습니다. 이러한 이벤트의 소스는 `aws.marketplacecatalog`이고 세부 정보 유형은 `Products Security Report Created`입니다.

각 이벤트에는 감지된 문제가 있는 제품 및 버전 수, 영향을 받는 최신 버전 수, 이러한 제품 또는 버전의 일시적 제한을 방지하기 위해 해결이 필요한 날짜의 요약이 있습니다.

> **Note**
>
> EventBridge 규칙 생성에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 규칙을 참조하세요.
> 보안 이벤트 관리에 대한 자세한 내용은 AWS 블로그의 AWS Marketplace에서 제품 카탈로그의 보안을 개선하는 방법 블로그 게시물을 참조하세요.

다음은 `Products Security Report Created` 세부 정보 유형에 대한 이벤트 본문 예시입니다.

```
{
  "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"
    }
  }
```

지급 이벤트

잘못된 은행 계좌 세부 정보로 인해 판매자 은행 계좌에 대한 지급이 실패하면 AWS Marketplace ISVs 및 채널 파트너가 이벤트를 수신할 수 있습니다.

다음 JSON 이벤트 코드에서 이러한 이벤트의 `source` 값은 이고 `aws.marketplace` `detail-type` 값은 입니다`Disbursement Paused`. `resources` 값은 잘못된 은행 계좌 Amazon 리소스 번호 (ARN)를 보여줍니다.

```
{
"version": "0",
"id": "01234567-0123-0123-0123-0123456789ab",
"detail-type": "Disbursement Paused",
"source": "aws.marketplace",
"account":"<account id of end user>",
"time": "2022-11-01T13:12:22Z",
"region": "us-east-1",
"resources": ["arn:aws:payments:us-east-1:1234567890:paymentinstrument:123"],
"detail": {
"requestId": "3d4c9f9b-b809-4f5e-9fac-a9ae98b05cbb",
"catalog": "AWSMarketplace"
 }
}
```

잘못된 은행 계좌 세부 정보 문제를 해결하려면 AWS Marketplace 관리 포털에 은행 계좌 세부 정보를 추가합니다. 지침은 은행 계좌 세부 정보를 추가하려면 섹션을 참조하세요.

Amazon EventBridge 규칙 생성에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridgeEventBridge의 규칙을 참조하세요.

의 판매자 보고서, 데이터 피드 및 대시보드 AWS Marketplace

AWS Marketplace 판매자는 다음 도구를 사용하여 제품 판매에 대한 정보를 수집하고 분석할 수 있습니다.

- 가 자동으로 AWS Marketplace 생성하고 등록된 모든 AWS Marketplace 판매자가 사용할 수 있도록 하는 보고서입니다.
- API: 이러한 보고서의 섹션에 액세스할 수 있습니다.
- 데이터 피드: 보고서에 나열된 거래에 대한 고객 정보를 식별하는 데 사용할 수 있는 추가 고객 정보를 제공합니다.
- Amazon QuickSight로 구동되는 대시보드: 재무 데이터에 액세스하고 분석하는 데 도움이 되는 차트, 그래프 및 인사이트를 제공합니다.

AWS Marketplace 는 다음을 준수하면서 보고서, 데이터 피드 및 대시보드에서 가능한 많은 데이터를 제공합니다.

- 고객 데이터를 보호하기 위한 Amazon 표준 및 원칙.
- 구매자가 AWS Marketplace에서 제품을 구매할 때 동의하는 이용 약관. 판매자는 구매자 데이터를 안전하게 관리하고 구매자의 요청 시 데이터를 삭제할 계약상의 의무가 있습니다.

> **Note**
>
> 보고서 알림 이메일 수신을 취소하려면 AWS Marketplace Seller Operations 팀에 문의하세요.

의 보고서, 데이터 피드 및 대시보드에 대한 자세한 내용은 다음 주제를 AWS Marketplace참조하세요.

주제

- 의 판매자 전송 데이터 피드 AWS Marketplace
- 의 판매자 보고서 AWS Marketplace
- 보충 보고서
- 판매자 대시보드

의 판매자 전송 데이터 피드 AWS Marketplace

AWS Marketplace 는 AWS Marketplace 시스템에서 판매자 소유 비즈니스 인텔리전스 도구 간의 ETL 용 Amazon S3 버킷(추출, 변환 및 로드)으로 구조화된 up-to-date 제품 및 고객 정보를 보내는 메커니즘으로 데이터 피드를 제공합니다. 데이터 피드는 쉼표로 구분된 값(CSV) 파일을 수집하여 사용자가 제공하는 암호화된 Amazon S3 버킷에 전송합니다. 데이터 피드는 하루 내에 생성되며 전날의 24시간 데이터를 포함합니다. 다음 섹션에서는 데이터 피드에 대한 개요를 제공하고, 데이터 피드에 액세스하여 사용하는 방법을 설명합니다. 후속 단원에서는 각 데이터 피드에 대해 설명합니다.

트랜잭션 데이터는 양방향 시간 구조로 전달 및 추가되므로 판매자는 두 타임라인 및 두 타임라인의 타임스탬프와 함께 데이터를 저장하고 쿼리할 수 있습니다.

- 유효 시간: 실제 세계에서 사실이 발생한 시간(“알게 된 정보”)
- 시스템 시간: 해당 사실이 데이터베이스에 기록된 시간(“알게 된 시간”).

데이터 피드는 전날의 24시간 데이터를 포함하는 전날의 업데이트에 따라 매일 자정(UTC)에 제공됩니다. 업데이트는 고객 구독, 인보이스 발행 중인 고객 또는 결제 AWS 에 의해 정의될 수 있습니다.

주제

- AWS Marketplace 데이터 피드의 저장 및 구조
- 데이터 피드 액세스
- 데이터 피드를 사용하여 데이터 수집 및 분석
- 데이터 피드 테이블 개요
- 데이터 피드 쿼리 예제
- 데이터 피드

AWS Marketplace 데이터 피드의 저장 및 구조

AWS Marketplace 는 판매자 소유의 비즈니스 인텔리전스 도구 간에 ETL(추출, 변환 및 로드)을 위해 AWS Marketplace 시스템에서 판매자 Amazon S3 버킷으로 구조화된 up-to-date 제품 및 고객 정보를 전송하는 메커니즘으로 데이터 피드를 제공합니다. 이 주제에서는 데이터 피드의 구조 및 스토리지에 대한 자세한 정보를 제공합니다.

데이터 피드는 쉼표로 구분된 값(CSV) 파일을 수집하여 사용자가 제공하는 암호화된 Amazon S3 버킷에 전송합니다. 이러한 CSV 파일은 다음과 같은 특성을 갖습니다.

- 4180 표준을 따릅니다.
- 문자 인코딩이 BOM을 사용하지 않는 UTF-8입니다.
- 값 사이의 구분 기호로 쉼표가 사용됩니다.
- 필드는 큰따옴표로 이스케이프됩니다.
- \n은 줄 바꿈 문자입니다.
- 날짜는 UTC 표준 시간대로 보고되고 ISO 8601 날짜 및 시간 형식이며 1초 이내의 오차로 정확합니다.
- 모든 `*_period_start_date` 및 `*_period_end_date` 값은 경계 값을 포함합니다. 따라서 모든 날짜에 가능한 마지막 타임스탬프는 `23:59:59`입니다.
- 모든 금액 필드 앞에는 통화 필드가 있습니다.
- 금액 필드는 마침표(`.`) 문자를 소수 구분 기호로 사용하며 쉼표(,)를 천 단위 구분 기호로 사용하지 않습니다.

데이터 피드는 다음과 같이 생성되고 저장됩니다.

- 데이터 피드는 하루 내에 생성되며 전날의 24시간 데이터를 포함합니다.
- Amazon S3 버킷에서 데이터 피드는 다음 형식을 사용하여 월 단위로 구성됩니다.

  bucket-name/*data-feed-name_version*/year=*YYYY*/month=*MM*/data.csv
- 각 일일 데이터 피드가 생성되면 해당 월의 기존 CSV 파일에 추가됩니다. 새 달이 시작되면 각 데이터 피드에 대해 새 CSV 파일이 생성됩니다.
- 데이터 피드의 정보는 2010년 1월 1일~2020년 4월 30일(포함)의 데이터로 다시 채워지며 `year=2010/month=01` 하위 폴더의 CSV 파일에 있습니다.

  해당 데이터 피드의 이번 달 파일에 열 머리글만 있고 데이터가 없는 상황이 있을 수 있습니다. 이는 이번 달에 해당 피드의 새로운 항목이 없음을 의미합니다. 이 문제는 제품 피드와 같이 자주 업데이트되지 않는 데이터 피드에서 발생할 수 있습니다. 이러한 경우 백필된 폴더에서 데이터를 사용할 수 있습니다.
- Amazon S3에서는 Amazon S3 수명 주기 정책을 생성하여 버킷에 파일을 보관하는 기간을 관리할 수 있습니다.
- 암호화된 Amazon S3 버킷으로 데이터가 전달될 때 알리도록 Amazon SNS를 구성할 수 있습니다. Amazon S3 알림 구성 방법에 대한 내용은 Amazon Simple Notification Service 개발자 안내서의 Amazon SNS 시작하기 섹션을 참조하세요.

데이터에 대한 기록

각 데이터 피드에는 데이터에 대한 기록을 문서화하는 열이 포함됩니다. `valid_to`를 제외하고 이러한 열은 모든 데이터 피드에 공통적입니다. 공통 기록 스키마로 포함되며, 데이터 쿼리에 유용합니다.

열 이름	설명
valid_from	기본 키의 값이 다른 필드의 값과 관련하여 유효한 첫 번째 날짜입니다.
valid_to	이 열은 주소 데이터 피드에만 표시되며 항상 비어 있습니다.
insert_date	레코드가 데이터 피드에 삽입된 날짜입니다.
update_date	레코드가 마지막으로 업데이트된 날짜입니다.
delete_date	이 열은 항상 비어 있습니다.

다음은 이러한 열의 예입니다.

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` 필드와 `update_date` 필드는 함께 양방향 시간 데이터 모델을 구성합니다. `valid_from` 필드는 그 이름처럼 항목이 언제부터 유효한지 알려줍니다. 항목이 편집된 경우 피드에 여러 레코드가 있을 수 있으며, 각 레코드의 `update_date`는 다르지만 `valid_from` 날짜는 같습니다. 예를 들어 항목의 현재 값을 찾으려면 `valid_from` 날짜가 가장 최근인 레코드 목록에서 `update_date`가 가장 최근인 레코드를 찾으면 됩니다.

위 예제의 레코드는 2018년 12월 12에 처음으로 생성되었습니다. 그리고 2019년 3월 29에 변경되었습니다(예: 레코드의 주소 변경). 그 후 2019년 4월 28일에 주소 변경이 수정되었습니다. 따라서 `valid_from`은 변경되지 않았지만 `update_date`는 변경되었습니다. 주소를 수정하면(드문 경우지만) 원래 `valid_from` 날짜의 레코드를 소급하여 변경하므로, 해당 필드는 변경되지 않았습니다. 가장 최근 `valid_from`을 찾는 쿼리는 2개 레코드를 반환하는데, `update_date`가 가장 최근인 레코드는 실제 현재 레코드를 제공합니다.

데이터 피드 액세스

를 사용하면 데이터 피드를 메커니즘으로 사용하여 AWS Marketplace 시스템에서 판매자 소유의 비즈니스 인텔리전스 도구 간에 ETL용 Amazon S3 버킷(추출, 변환 및 로드)을 위한 구조화된 up-to-date 제품 및 고객 정보를 전송할 AWS Marketplace수 있습니다. 암호화된 Amazon S3 버킷에 대한 데이터 피드를 수신하도록 환경을 구성해야 합니다. 이 주제에서는 데이터 피드에 액세스하고 구독을 취소하는 방법을 보여줍니다.

주제

- 데이터 피드 액세스
- 데이터 피드 정책
- 데이터 피드 구독 취소

데이터 피드 액세스

1. SQL 및 ETL(추출, 변환, 로드) 경험을 통해 비즈니스 인텔리전스 또는 데이터 엔지니어를 할당합니다. 또한 이 사람은 API 설정 경험도 필요합니다.
2. Amazon Simple Storage Service 버킷과 데이터 피드 구독을 설정합니다. Marketplace 제품 목록과 연결된 AWS 판매자 계정 ID를 사용합니다. 이렇게 하려면 이 YouTube 동영상을 시청하거나 아래 단계를 따르세요.

   동영상과 단계에서는 구성을 간소화하는 데 도움이 되는 AWS CloudFormation 템플릿을 사용하는 방법을 설명합니다.

   a. 웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인한 다음, 고객 데이터 스토리지 설정으로 이동합니다.

   b. AWS CloudFormation 템플릿을 사용하여 리소스 생성을 선택하여 콘솔의 AWS CloudFormation 다른 창에서 템플릿을 엽니다.

   c. 템플릿에서 다음 필드를 지정하고 다음을 선택합니다.

- 스택 이름 - 데이터 피드에 액세스할 수 있도록 생성하는 리소스 컬렉션입니다.
- Amazon S3 버킷 이름 - 데이터 피드를 저장할 버킷입니다.
- (선택 사항) Amazon SNS 주제 이름 - Amazon Simple Storage Service 버킷에서 알림을 수신하는 주제입니다.

d. 검토 페이지에서 지정한 사항을 확인하고 스택 생성을 선택합니다. 그러면 CloudFormation 상태 및 세부 정보를 제공하는 새 페이지가 열립니다.

e. 리소스 탭에서 CloudFormation 페이지의 다음 리소스에 대한 Amazon 리소스 이름(ARNs)을 AWS Marketplace 고객 데이터 스토리지 설정 페이지의 필드에 복사합니다.

- 데이터 피드를 저장할 Amazon S3 버킷
- AWS KMS Amazon S3 버킷 암호화를 위한 키
- (선택 사항)가 Amazon S3 버킷에 새 데이터를 AWS 전송할 때 알림을 수신하기 위한 Amazon SNS 주제 Amazon S3

f. 고객 데이터 스토리지 설정 페이지에서 제출을 선택합니다.

g. (선택 사항) CloudFormation 템플릿으로 생성한 정책을 편집합니다. 자세한 내용은 데이터 피드 정책 섹션을 참조하세요.

이제 데이터 피드를 구독했습니다. 다음에 데이터 피드가 생성되면 해당 데이터에 액세스할 수 있습니다.

3. ETL(추출, 변환, 로드) 작업을 사용하여 데이터 피드를 데이터 웨어하우스 또는 관계형 데이터베이스에 연결합니다.

> **Note**
>
> 데이터 도구의 기능은 다릅니다. 도구의 기능과 일치하도록 통합을 설정하려면 비즈니스 인텔리전스 엔지니어 또는 데이터 엔지니어가 참여해야 합니다.

4. SQL 쿼리를 실행하거나 만들려면 데이터 도구에서 기본 및 외부 키를 적용하도록 데이터 피드를 구성합니다. 각 데이터 피드는 고유한 테이블을 나타내며 엔터티 관계를 사용하여 데이터 스키마의 모든 데이터 피드를 설정해야 합니다. 테이블 및 엔터티 관계에 대한 자세한 내용은 이 설명서의 데이터 피드 테이블 개요 섹션을 참조하세요.

5. 데이터 웨어하우스 또는 관계형 데이터베이스를 자동으로 새로 고치도록 Amazon Simple Notification Service를 설정합니다. 각 고유 피드의 데이터가 Amazon S3 버킷으로 전달될 때 알림을 보내도록 Amazon SNS 알림을 구성할 수 있습니다. Amazon S3 판매자 데이터 도구가 이 기능을 지원하는 경우, 이러한 알림을 활용하여 데이터 피드를 통해 새 데이터를 수신할 때 판

매자 데이터 웨어하우스를 자동으로 새로 고칠 수 있습니다. 이에 관한 지침은 Amazon Simple Notification Service 개발자 안내서의 Amazon SNS 시작하기를 참조하세요.

알림 예제:

```
{
        "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]"
        }
```

알림의 `executionStatus` 상태는 다음과 같습니다.

- `SKIPPED` - 판매자에게 해당 날짜에 대한 새 데이터가 없습니다.
- `COMPLETED` - 피드에 새 데이터를 제공했습니다.
- `FAILED` – 피드 전송에 문제가 있습니다.

6. SQL 쿼리를 실행하여 설정을 검증합니다. 이 가이드의 샘플 쿼리 또는 GitHub의 쿼리(https://github.com/aws-samples/aws-marketplace-api-samples/tree/main/seller-data-feeds/queries)를 사용할 수 있습니다.

> **Note**
>
> 이 가이드의 샘플 쿼리는 AWS Athena용으로 작성되었습니다. 도구와 함께 사용할 쿼리를 수정해야 할 수 있습니다.

7. 비즈니스 사용자가 데이터를 사용할 위치를 결정합니다. 예를 들어, 다음을 수행할 수 있습니다.

- 데이터 웨어하우스 또는 SQL 데이터베이스에서 .csv 데이터를 내보냅니다.
- 데이터를 PowerBI 또는 Tableau와 같은 시각화 도구에 연결합니다.
- 데이터를 CRM, ERP 또는 Salesforce, Infor 또는 Netsuite와 같은 금융 도구에 매핑합니다.

AWS CloudFormation 템플릿에 대한 자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 템플릿 작업을 참조하세요.

데이터 피드 정책

Amazon S3 버킷이 CloudFormation 템플릿에 의해 생성되면 해당 버킷, AWS KMS 키 및 Amazon SNS 주제에 연결된 액세스에 대한 정책이 생성됩니다. 이 정책은 AWS Marketplace 보고서 서비스가 버킷과 SNS 주제에 데이터 피드 정보를 쓸 수 있도록 허용합니다. 각 정책에는 다음과 같은 섹션이 있습니다. 이 예제는 Amazon S3 버킷에서 가져온 것입니다.

```
        {
            "Sid": "AwsMarketplaceDataFeedsAccess",
            "Effect": "Allow",
            "Principal": {
                "Service": "reports.marketplace.amazonaws.com"
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:PutObject",
                "s3:GetEncryptionConfiguration",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },
```

이 정책에서는 `reports.marketplace.amazonaws.com` 서비스 보안 주체를 AWS Marketplace 사용하여 Amazon S3 버킷으로 데이터를 푸시합니다. CloudFormation 템플릿에서 *`amzn-s3-demo-bucket`*을 지정했습니다.

AWS Marketplace 보고서 서비스가 Amazon S3 AWS KMS또는 Amazon SNS를 호출하면 버킷에 쓰려는 데이터의 ARN을 제공합니다. 버킷에 기록된 데이터만 사용자를 대신하여 기록되게 하려면 정책 조건에서 `aws:SourceArn`을 지정하면 됩니다. 다음 예제에서는 *`account-id`*를 해당 AWS 계정의 ID로 바꿔야 합니다.

```
        {
           "Sid": "AwsMarketplaceDataFeedsAccess",
           "Effect": "Allow",
           "Principal": {
                "Service": "reports.marketplace.amazonaws.com"
```

```
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:PutObject",
                "s3:GetEncryptionConfiguration",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ,
            "Condition": {
                "StringEquals": {
                        "aws:SourceAccount": "account-id",
                        "aws:SourceArn": ["arn:aws:marketplace::account-
id:AWSMarketplace/SellerDataSubscription/DataFeeds_V1",
                        "arn:aws:marketplace::account-id:AWSMarketplace/
SellerDataSubscription/Example-Report"]
                        }
                }
        },
```

데이터 피드 구독 취소

웹 브라우저를 열고 AWS Marketplace Management Portal에 로그인합니다. 그런 다음 문의처 페이지로 이동하여 AWS Marketplace 판매자 운영 팀에 구독 취소 요청을 제출합니다. 구독 취소 요청을 처리하는 데 최대 10일이 걸릴 수 있습니다.

데이터 피드를 사용하여 데이터 수집 및 분석

AWS Marketplace 는 AWS Marketplace 시스템에서 판매자 소유의 비즈니스 인텔리전스 도구 간에 ETL(추출, 변환 및 로드)용 Amazon S3 버킷으로 구조화된 up-to-date 제품 및 고객 정보를 전송하는 메커니즘으로 데이터 피드를 제공합니다. Amazon S3 버킷에 데이터가 있으면 다음과 같은 방법으로 데이터 피드를 사용할 수 있습니다.

- 데이터 피드 액세스에서 생성한 Amazon S3 버킷에서 .CSV 파일을 다운로드하여 스프레드시트에서 데이터를 봅니다.
- ETL(추출, 변환 및 로드), SQL 쿼리 및 비즈니스 분석 도구를 사용하여 데이터를 수집하고 분석합니다.

AWS 서비스를 사용하여 데이터를 수집하고 분석하거나 .CSV 기반 데이터 세트를 분석할 수 있는 타사 도구를 사용할 수 있습니다.

데이터를 수집하고 분석하기 위한 데이터 피드에 대한 자세한 내용은 다음 예제를 참조하세요.

예: AWS 서비스를 사용하여 데이터 수집 및 분석

다음 절차에서는 Amazon S3 버킷에 데이터 피드를 수신하도록 환경을 이미 구성했으며 버킷에 데이터 피드가 포함되어 있다고 가정합니다.

데이터 피드에서 데이터를 수집하고 분석하려면

1. AWS Glue 콘솔에서 크롤러를 생성하여 데이터 피드를 저장하는 Amazon S3 버킷에 연결하고, 원하는 데이터를 추출하고, AWS Glue Data Catalog에서 메타데이터 테이블을 생성합니다.

   에 대한 자세한 내용은 AWS Glue 개발자 안내서를 AWS Glue참조하세요.
2. Athena 콘솔에서 AWS Glue Data Catalog의 데이터에 대해 SQL 쿼리를 실행합니다.

   Athena에 대한 자세한 내용은 Amazon Athena 사용 설명서를 참조하세요.
3. QuickSight 콘솔에서 분석을 생성한 다음 데이터의 시각적 객체를 생성합니다.

   QuickSight에 대한 자세한 내용은 Amazon QuickSight 사용 설명서를 참조하세요.

AWS 서비스를 사용하여 데이터 피드의 데이터를 수집하고 분석하는 방법에 대한 자세한 예는 AWS Marketplace 블로그에서 판매자 데이터 피드 전송 서비스, Amazon Athena 및 QuickSight를 사용하여 판매자 보고서 생성을 참조하세요.

데이터 피드 테이블 개요

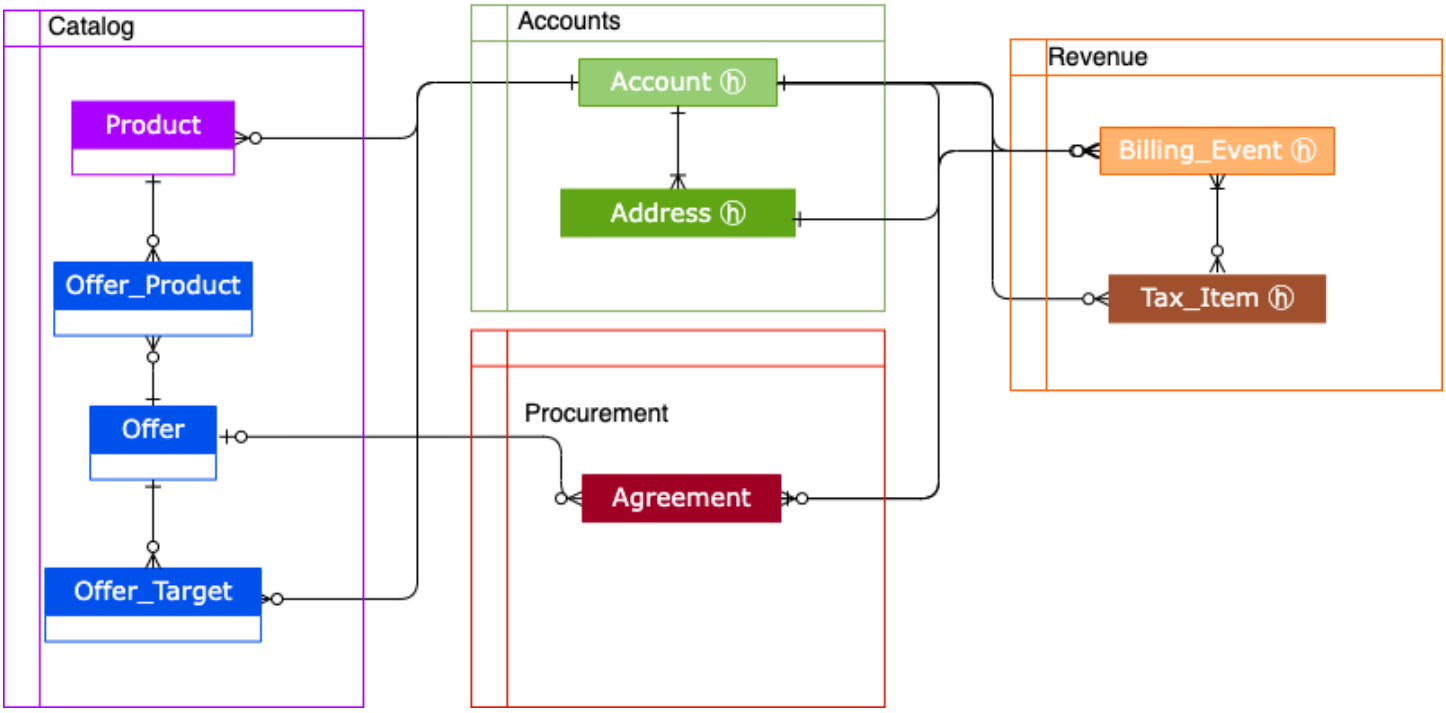
AWS Marketplace 는 쿼리에 대한 더 많은 컨텍스트를 제공하기 위해 함께 조인할 수 있는 테이블 세트로 데이터 피드를 제공합니다.

AWS Marketplace 는 데이터 피드에 다음과 같은 일반 도메인 또는 관심 범주를 제공합니다.

- 카탈로그 - 계정의 제품 및 제안에 대한 정보를 포함하고 있습니다.
- 계정 -에서 제품을 제공하거나 구매하는 계정에 대한 정보를 포함합니다 AWS Marketplace (채널 파트너 또는 구매자와 같이 함께 작업하는 사용자의 계정 또는 사용자의 계정).
- 수익 - 결제, 지급 및 세금에 대한 정보를 포함하고 있습니다.

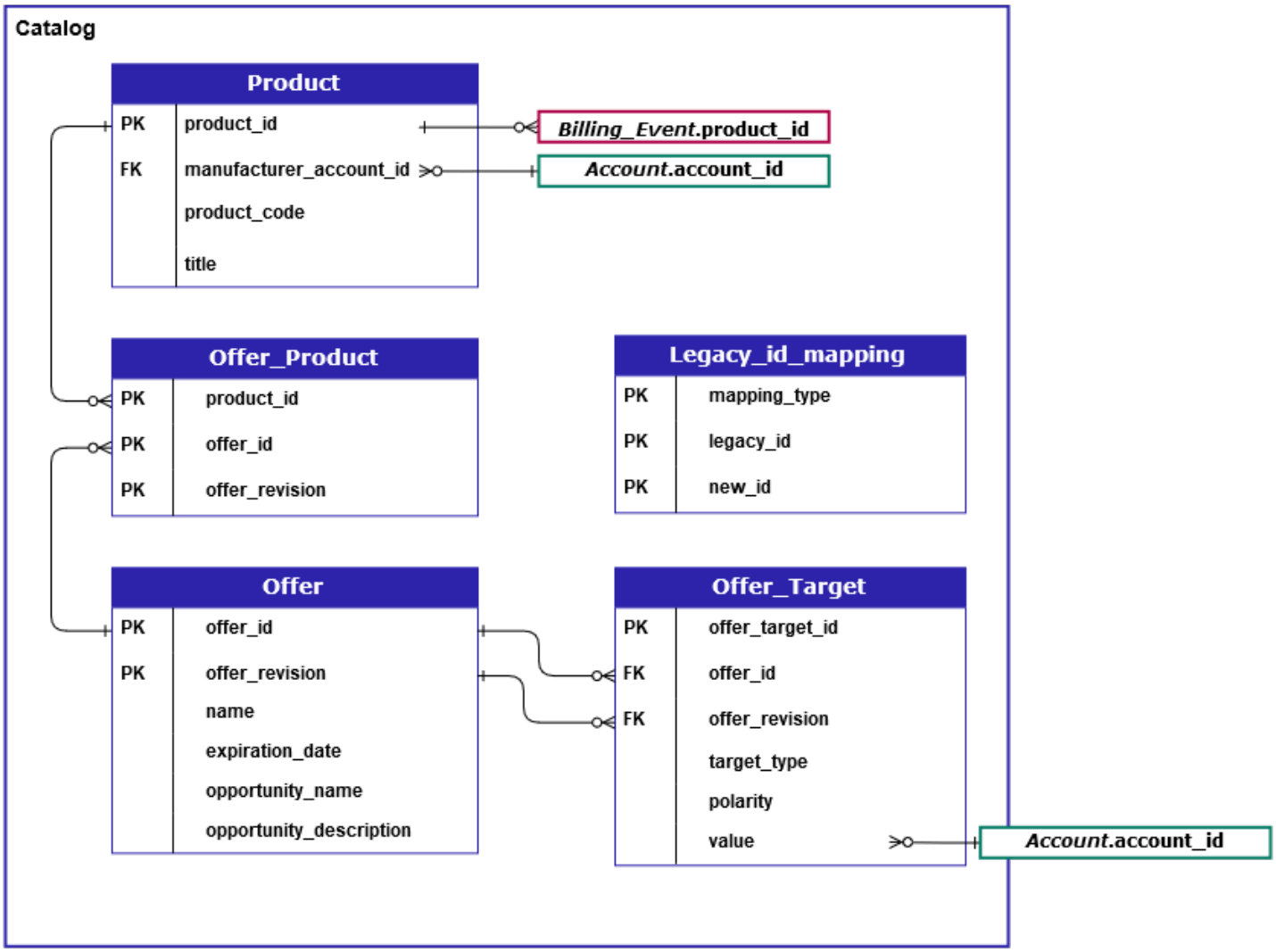
- 조달 - 공식 판매자로서 귀하가 작성한 제품 제공에 대한 계약에 대한 정보를 제공합니다.

이 다이어그램은 카탈로그, 계정 및 수익 도메인의 테이블을 보여줍니다.



카탈로그 관련 테이블

다음 다이어그램은 카탈로그 도메인에 있는 테이블과 테이블 내 필드 간의 관계를 보여줍니다.



`Product`, `Offer_Product`, `Offer`, `Offer_Target` 및 `Legacy_id_mapping__` 테이블은 카탈로그 도메인에 있습니다.

`target_type` 값이 `account`인 경우에만 `Offer_Target` 테이블에는 대상의 `account_id`에 대한 값 필드가 포함됩니다.

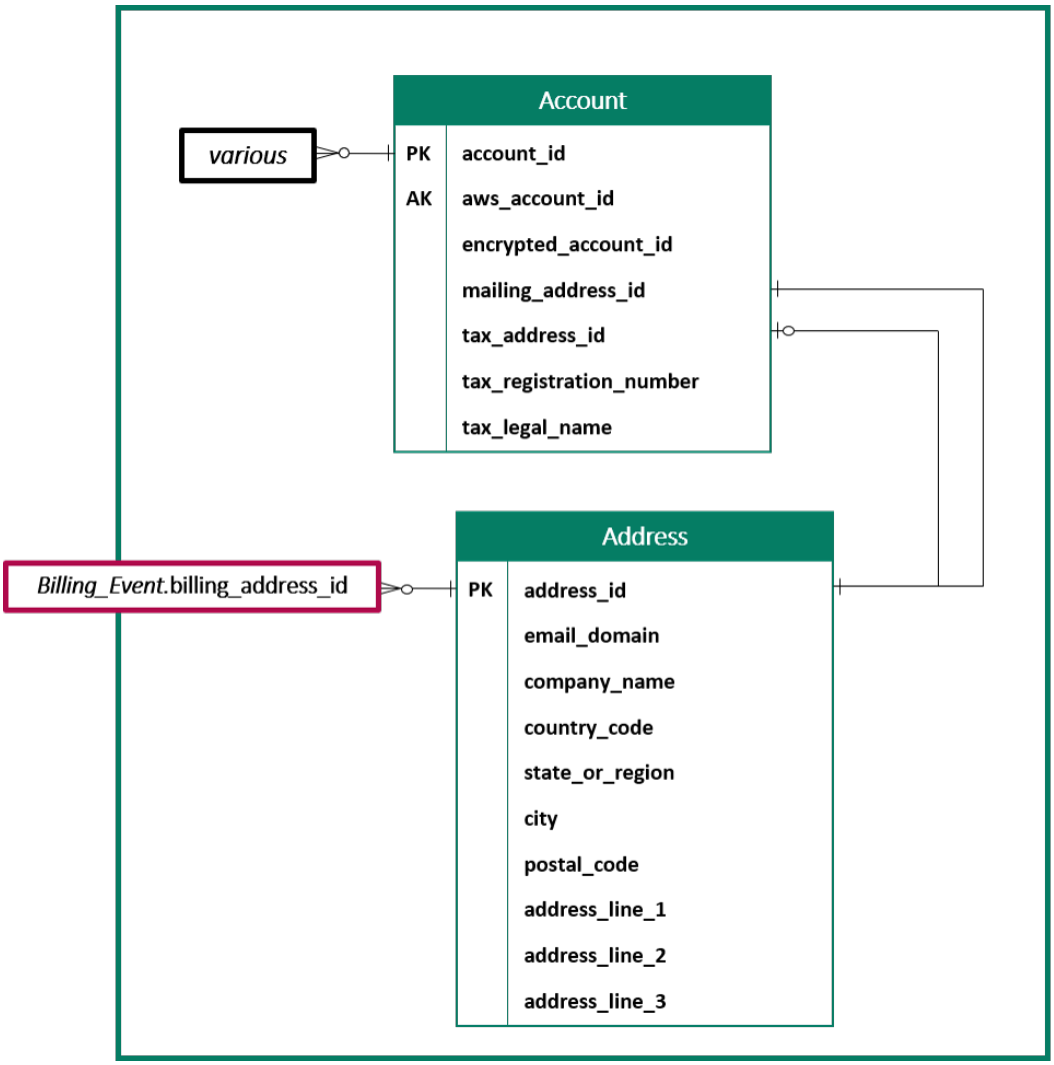
현재 `Legacy_id_mapping` 테이블은 데이터에 사용되지 않습니다.

> **Note**
>
> 테이블의 각 필드에 대한 설명과 생성 가능한 조인을 포함하여 이러한 테이블에 대한 자세한 내용은 다음 주제를 참조하세요.
>
> - 제품 데이터 피드
> - 제안 제품 데이터 피드

- 제안 데이터 피드
- 제안 대상 데이터 피드
- 레거시 매핑 데이터 피드

계정 관련 테이블

다음 다이어그램은 계정 도메인에 있는 `Account` 및 `Address` 테이블과 테이블 내 필드 간의 관계를 보여줍니다.

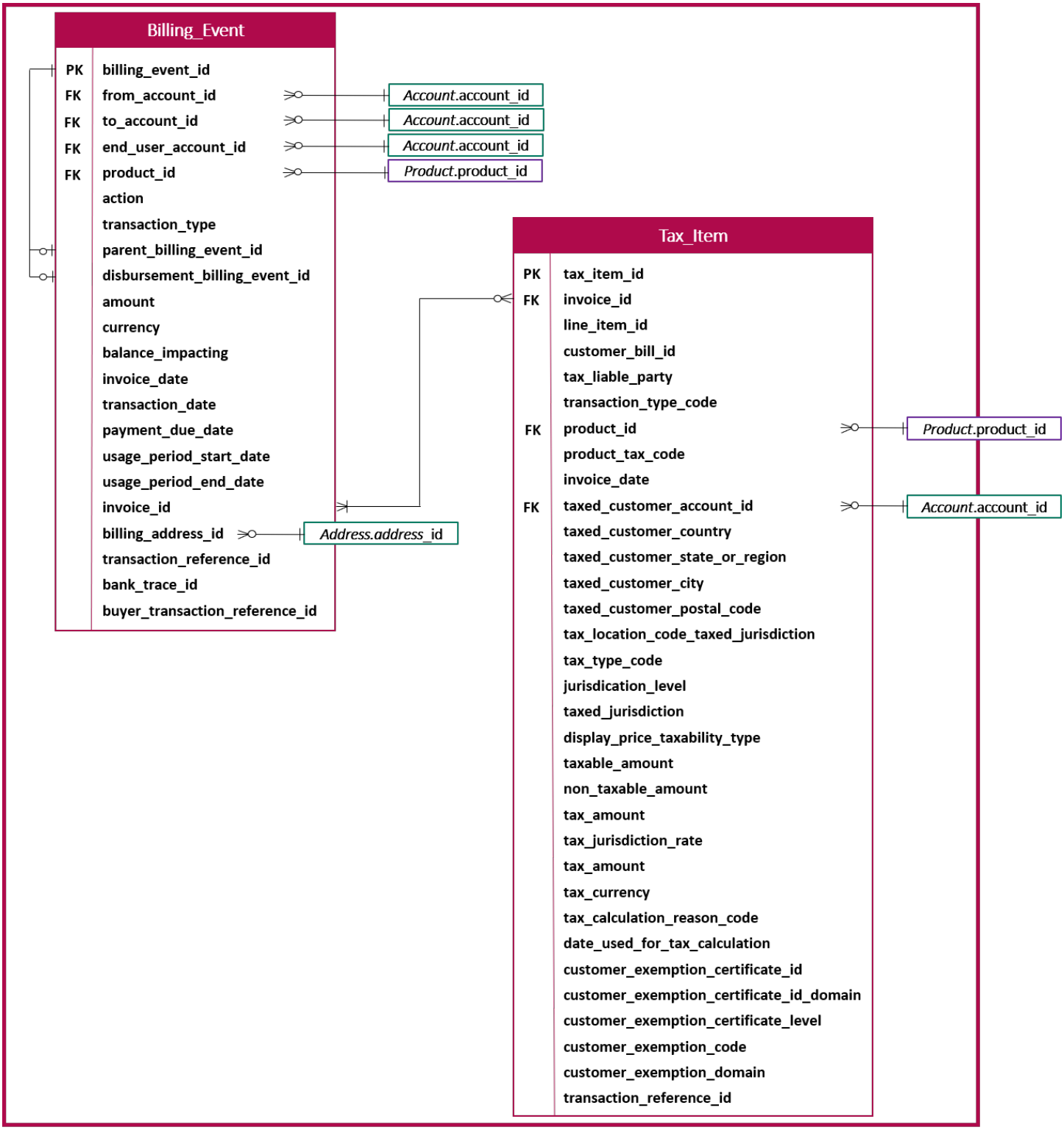


> **Note**
>
> 테이블의 각 필드에 대한 설명과 만들 수 있는 조인을 포함하여 이러한 테이블에 대한 자세한 내용은 다음 주제를 참조하세요.

- 계정 데이터 피드
- 주소 데이터 피드

수익 관련 테이블

다음 다이어그램은 수익 도메인에 있는 `Billing_Event` 및 `Tax_Item` 테이블과 테이블 내 필드 간의 관계를 보여줍니다. `Billing_Event` 테이블에는 지급과 결제 이벤트에 대한 정보가 포함되어 있습니다.


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**
> 테이블의 각 필드에 대한 설명과 만들 수 있는 조인을 포함하여 이러한 테이블에 대한 자세한 내용은 다음 주제를 참조하세요.
>
> - 결제 이벤트 데이터 피드
> - 세금 항목 데이터 피드

조달 관련 테이블

다음 다이어그램은 조달 도메인의 계약 테이블 내의 필드를 보여줍니다.

> **Note**
> 테이블의 각 필드에 대한 설명과 생성 가능한 조인을 포함하여 이러한 테이블에 대한 자세한 내용은 이 안내서의 계약 데이터 피드 섹션을 참조하세요.

다음 섹션에서는 각 도메인의 엔터티 관계(ER) 다이어그램을 제공합니다. 각 ER 다이어그램은 테이블과 각 테이블 내의 필드 그리고 테이블을 조인하는 데 사용할 수 있는 필드를 보여줍니다.

> **Note**
> 이 섹션의 ER 다이어그램에는 모든 데이터 피드의 공통 필드가 나와 있지 않습니다. 공통 필드에 대한 자세한 내용은 AWS Marketplace 데이터 피드의 저장 및 구조 섹션을 참조하세요.

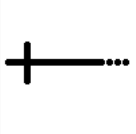
다음 표에는 ER 다이어그램에서 사용되는 기호가 설명되어 있습니다.

Symbol	설명
PK	프라이머리 키 - 테이블의 프라이머리 키입니다. `valid_from` 및 `update_date` 필드와 함께 사용하면 고유한 필드가 됩니다. 이러한 필드를 함께 사용하는 방법에 대한 자세한 내용은 데이터에 대한 기록 섹션을 참조하세요. 여러 필드가

Symbol	설명
	프라이머리 키로 표시되는 경우 해당 필드가 함께 프라이머리 키를 형성합니다.
FK	외래 키 - 다른 테이블의 프라이머리 키를 나타내는 필드입니다. 테이블에서 반드시 고유할 필요는 없습니다. **Note** 현재 테이블의 레코드에 외부 테이블의 해당 레코드가 없는 경우 외래 키가 비어 있기도 합니다.
AK	대체 키 - 테이블에서 키로 사용할 수 있는 키입니다. 프라이머리 키와 동일한 고유성 규칙을 따릅니다.
	커넥터 - 필드 사이의 선은 테이블을 조인하는 데 사용할 수 있는 두 필드인 연결을 나타냅니다. 선의 끝은 연결 유형을 나타냅니다. 이 예제는 일대다 연결을 나타냅니다.

커넥터 유형

다음 표에서는 각 커넥터가 가질 수 있는 끝의 유형을 보여줍니다.

커넥터 유형	설명
	1-n - 이 끝이 있는 커넥터는 조인의 이쪽에 값이 정확히 하나 있는 조인을 나타냅니다.

커넥터 유형	설명
	0 또는 1-n - 이 끝이 있는 커넥터는 조인의 이쪽에 값이 정확히 0개 또는 하나 있는 조인을 나타냅니다.
	0 이상-n - 이 끝이 있는 커넥터는 조인의 이쪽에 값이 0개, 1개 또는 여러 개 있는 조인을 나타냅니다.
	1 이상-n - 이 끝이 있는 커넥터는 조인의 이쪽에 값이 1개 또는 여러 개 있는 조인을 나타냅니다.

데이터 피드 쿼리 예제

이 섹션에서는에서 제공하는 데이터 피드를 사용하는 복잡한 쿼리의 예를 제공합니다 AWS Marketplace. 이러한 예제는 AWS Marketplace Management Portal에서 얻을 수 있는 의 판매자 보고서 AWS Marketplace와 비슷합니다. 이러한 쿼리를 사용자 지정하여 필요한 다른 보고서를 생성할 수 있습니다.

쿼리 예제

- 계약 및 갱신
- 청구 수익
- 미수금 또는 지급된 인보이스
- 과세된 인보이스
- 제품별 지급
- 영업 보상 보고서

계약 및 갱신

계약 및 갱신 데이터를 찾으려면 다음 예제와 같은 쿼리 세트를 실행할 수 있습니다. 쿼리는 서로를 기반으로 구축되어 계약 및 갱신 대시보드, 세분화된 데이터 섹션을 만듭니다. 표시된 예제를 그대로 사용하거나, 자신의 데이터와 사용 사례에 맞게 사용자 지정할 수 있습니다.

쿼리의 주석은 쿼리의 기능과 수정 방법을 설명합니다.

```
    Query currently under development.
```

청구 수익

인보이스 데이터를 찾으려면 다음 예제와 같은 쿼리 세트를 실행할 수 있습니다. 쿼리는 서로를 기반으로 구축되어 청구된 수익 보고서를 만듭니다. 표시된 예제를 그대로 사용하거나, 자신의 데이터와 사용 사례에 맞게 사용자 지정할 수 있습니다.

쿼리의 주석은 쿼리의 기능과 수정 방법을 설명합니다.

```
-- 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)
```

미수금 또는 지급된 인보이스

수집되지 않거나 지불된 인보이스를 찾으려면 다음 예제와 같은 쿼리 집합을 실행할 수 있습니다. 쿼리는 서로를 기반으로 구축되어 수집 및 지급 보고서를 생성합니다. 표시된 예제를 그대로 사용하거나, 자신의 데이터와 사용 사례에 맞게 사용자 지정할 수 있습니다.

쿼리의 주석은 쿼리의 기능과 수정 방법을 설명합니다.

```
-- 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)
```

과세된 인보이스

과세된 인보이스를 찾으려면 다음 예제와 같은 쿼리 세트를 실행할 수 있습니다. 쿼리는 서로를 기반으로 구축되어 과세 보고서를 생성합니다. 표시된 예제를 그대로 사용하거나, 자신의 데이터와 사용 사례에 맞게 사용자 지정할 수 있습니다.

쿼리의 주석은 쿼리의 기능과 수정 방법을 설명합니다.

```
-- 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)
```

제품별 지급

제품별로 지급된 금액을 확인하려면 다음과 같이 쿼리 세트를 실행하면 됩니다. 이 예는 지출 보고서 판매자 보고서와 비슷합니다.

이 예제 쿼리 세트는 서로를 기반으로 지급이 포함된 최종 제품 세부 정보 목록을 생성합니다. 또한 특정 시점의 제품 정보를 얻는 방법도 보여줍니다. 표시된 샘플을 그대로 사용하거나, 자신의 데이터와 사용 사례에 맞게 사용자 지정할 수 있습니다.

쿼리의 주석은 쿼리의 기능과 수정 방법을 설명합니다.

> **Note**
>
> 이 쿼리를 실행할 때 데이터는 `valid_from`과 `update` 열의 두 개의 시간 축을 사용하여 수집된다고 가정합니다. 축에 대한 자세한 내용은 AWS Marketplace 데이터 피드의 저장 및 구조 섹션을 참조하세요.

```
-- 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;
```

영업 보상 보고서

고객별 청구 수익을 확인하려면 다음과 같이 쿼리 세트를 실행하면 됩니다. 이 예는 영업 보상 보고서 판매자 보고서와 비슷합니다. 이 샘플 쿼리는 서로를 기반으로 각 고객에게 소프트웨어 사용 요금으로 청구된 총액이 포함된 최종 고객 세부 정보 목록을 생성합니다. 표시된 쿼리를 그대로 사용하거나 자신의 데이터 및 사용 사례에 맞게 쿼리를 사용자 지정할 수 있습니다.

쿼리의 주석은 쿼리의 기능과 수정 방법을 설명합니다.

> **Note**
>
> 이 쿼리를 실행할 때 데이터는 `valid_from`과 `update` 열의 두 개의 시간 축을 사용하여 수집된다고 가정합니다. 자세한 내용은 AWS Marketplace 데이터 피드의 저장 및 구조 단원을 참조하세요.

```
-- 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;
```

데이터 피드

AWS Marketplace 는 판매자가 제품 판매에 대한 정보를 수집하고 분석하는 데 도움이 되는 여러 데이터 피드를 제공합니다. 데이터 피드는 등록된 모든 AWS Marketplace 판매자가 사용할 수 있습니다. 데이터 피드는 하루 이내에 생성되므로 사용 가능한 최신 데이터를 포함합니다.

다음 단원에서 각 데이터 피드에 대해 설명합니다.

주제

- 계정 데이터 피드
- 주소 데이터 피드
- 계약 데이터 피드
- 결제 이벤트 데이터 피드
- 레거시 매핑 데이터 피드

- 제안 데이터 피드
- 제안 제품 데이터 피드
- 제안 대상 데이터 피드
- 제품 데이터 피드
- 세금 항목 데이터 피드

계정 데이터 피드

이 데이터 피드는 상호 작용하는 모든 계정(판매자 본인 계정, 함께 일하는 채널 파트너, 구매자, 지급인 및 모든 과세 계정)에 대한 정보를 제공합니다.

계정 데이터는 변경할 수 없으며 버전 번호와 연결되어 있지 않습니다. 필드에 대한 변경 사항이 추가되므로 이 데이터 피드는 동일한 `account_id` 및 다른 `valid_from` 값을 가진 여러 행을 가질 수 있습니다. 데이터 기록 필드에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

계정 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다.

열 이름	설명
account_id	계정의 GUID(전역 고유 식별자)입니다. `Product`, `Offer_Target` , `Billing_Event` 및 `Tax_Item` 데이터 피드의 필드에 조인하는 데 사용할 수 있습니다. 조인에 사용할 수 있는 필드에 대한 자세한 내용은 해당 데이터 피드를 참조하세요.
aws_account_id	AWS 파티션별로 고유한 판매자 AWS 계정의 AWS 계정 번호입니다.
encrypted_account_id	애플리케이션의 개별 구매자를 위한 암호화된 고유 ID입니다. `encrypted_account_id` 의 값은 AWS Marketplace 측정 서비스에서 사용됩니다. 예를 들어 `ResolveCustomer` 작업에서 반환되는 `CustomerIdentifier` 의 값으로 사용됩니다.

열 이름	설명
mailing_address_id	이 계정의 우편 주소 참조입니다.
tax_address_id	이 계정의 세금 납부 주소 참조입니다.
tax_registration_number	미국 외 계정의 경우 이 계정의 사업자 등록 번호입니다.
tax_legal_name	미국 외 계정의 경우 법인 회사 이름입니다. 세금 계산서에 사용되는 이름입니다.

계정 데이터 피드의 예

다음은 계정 데이터 피드의 예입니다. 가독성을 위해 데이터 기록 열은 나와 있지 않습니다. 데이터 기록 필드에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

account_id	aws_account_id	encrypted_account_id	mailing_address_id	tax_address_id	tax_registration_number	tax_legal_name
xk0CSmiAm6PQ4QqEog9iiaochIzuPlkMfba7a1oDlZ	444456660000	Zf7oMzheGWpH	25o3k46eN6eViOfFiiqtxwX8e3kaOiPalUiofjyFa3			
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	555567679999	373vuQUqmQ8v	5oJ6vTjSzMrrF2gvh2Vj9HfqiM800MuLEHmyFY5Lr42s8	5oJ6vTjSzMrrF2gvh2Vj9HfqiM800MuLEHmyFY5Lr42s8	SE823935083345	
VIeGa2t9j3MuxioH9wc8lsndXXCgGCGUreeXriocM5	73739998888	8SPxAYmi8MwX	NLUc5UeiMlGFTrDWCoftDPhDUF1oaSd8xgl5QM8Db7	V5NhBYBiYogwy0WMhndGU4AfMggmuoTC2j7Pm8ZKKNNyT	DE469558025	

주소 데이터 피드

> **Important**
>
> AWS Marketplace는 판매자 제공 데이터 피드 서비스를 사용하여 제공되는 PiiFeed를 2023년 12월에 중단할 예정입니다. 주소 데이터 요구 사항에 AddressFeed_V1 데이터 피드를 사용하세요. 궁금한 점이 있거나 지원이 필요하면 AWS Marketplace 판매자 운영 팀에 문의하세요.

이 데이터 피드는 상호 작용하는 모든 계정(판매자 본인 계정, 함께 일하는 채널 파트너, 구매자, 지급인 및 모든 과세 계정)의 연락처 정보를 제공합니다. 새 거래가 발생할 때마다 거래에 해당하는 고객 주소가 검색되고, 이 주소가 데이터 피드에 없는 경우 데이터 피드 파일에 새 항목이 추가됩니다.

주소 데이터는 변경할 수 없습니다.

주소 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다.

열 이름	설명
address_id	주소의 고유 키입니다. `billing_address_id` 필드의 `Billing_Event` 데이터 피드에서 혹은 `mailing_address_id` 또는 `tax_address_id` 필드의 `Account` 데이터 피드에서 조인하는 데 사용할 수 있습니다.
email_domain	이 계정의 파일에 있는 이메일 주소의 도메인입니다.
company_name	이 계정의 파일에 있는 회사 이름입니다.
country_code	이 주소의 파일에 있는 ISO 3166 alpha-2 국가 코드입니다.
state_or_region	이 주소의 파일에 있는 상태 또는 지역입니다.
구/군/시	이 주소의 파일에 있는 도시입니다.

열 이름	설명
postal_code	이 주소의 파일에 있는 우편 번호입니다.
address_line_1	이 주소의 파일에 있는 첫 번째 줄 주소입니다.
address_line_2	이 주소의 파일에 있는 두 번째 줄 주소입니다.
address_line_3	이 주소의 파일에 있는 세 번째 줄 주소입니다.

주소 데이터 피드의 예

다음은 주소 데이터 피드의 예입니다. 데이터 피드에서 이 정보는 단일 테이블에 표시됩니다. 가독성을 위해 여기에는 데이터가 두 개의 표에 나와 있고 데이터 기록 열은 나와 있지 않습니다. 데이터 기록 필드에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

address_id	email_domain	company_name	country_code	state_or_region	구/군/시	postal_code
V5NhBYBiYogwy0WMhndGU4AfMggmuoTC2j7Pm8ZKKNNyT	a.com	Mateo Jackson's Company	DE		Hamburg	67568
G68xdbkZQDVVHzfBGw6yf5yos0A6NiSVWHmH5ViLjf	b.com	Mary Major's Company	미국	OH	Dayton	57684
NLUc5UeiMlGFTrDWCoftDPhDUF1oaSd8xgl5QM8Db7	c.com	Our Seller	미국	NY	New York	89475

address_line_1	address_line_2	address_line_3
	19th Floor	

계약 데이터 피드

이 데이터 피드는 제안자(등록 판매자)와 수락자(AWS 구매자)가 제품 사용을 시작하기 위해 서명한 계약인 계약에 대한 정보를 제공합니다. 이 데이터 피드는 공식 판매자로서 귀하가 작성한 제품 제공에 대한 계약에 대한 정보를 제공합니다.

계약 데이터 피드는 24시간마다 새로 고쳐집니다.

다음 표에서는 데이터 피드의 항목을 나열하고 설명합니다.

열	설명
valid_from	기본 키의 값이 다른 필드의 값과 관련하여 유효한 첫 번째 날짜입니다.
insert_date	레코드가 데이터 피드에 삽입된 날짜입니다.
update_date	레코드가 마지막으로 업데이트된 날짜입니다.
delete_date	이 열은 항상 비어 있습니다.
상태 표시기	계약의 현재 상태입니다. 지원되는 상태는 다음과 같습니다. • 활성 - 계약 조건의 일부 또는 전부가 유효합니다. • 종료됨 - AWS가 시작한 종료 이벤트로 인해 사전 합의된 종료 날짜 이전에 계약이 종료되었습니다. 가장 일반적인 경우는 결제 실패입니다.

열	설명
	• 취소됨 - 수락자가 종료 날짜 이전에 계약을 종료하기로 선택합니다. • 만료됨 - 계약이 사전 합의된 종료일에 종료되었습니다. • 대체됨 - 대체 제안을 사용하여 계약이 대체되었습니다. • 갱신됨 - 자동 갱신과 같은 기능을 사용하여 계약이 새 계약으로 갱신되었습니다. • 아카이브됨 - 계약이 종료되었습니다. 종결, 종료, 취소 또는 만료됨의 정확한 이유가 지정되지 않았습니다. • 롤백 - 오류로 인해 계약 개정이 롤백되었습니다. 이제 이전 개정이 활성화됩니다. 비활성 계약 개정에만 적용됩니다. • 대체됨 - 계약 개정이 더 이상 활성화되지 않고 다른 계약 개정이 이제 활성화되었습니다. 비활성 계약 개정에만 적용됩니다.
estimated_agreement_value	고객이 계약 수명 주기 동안 지불할 가능성이 높은 알려진 총금액입니다.
currency_code	고객이 계약 수명 주기 동안 지불할 가능성이 높은 알려진 총금액의 통화입니다.
agreement_id	계약의 고유 식별자입니다.
proposer_account_id	이 구매 계약을 제안한 판매자, 판매자 계정의 전역 고유 식별자(GUID)로 표현됩니다. 계정 데이터 피드에 연결하는 데 사용할 수 있습니다.
acceptor_account_id	이 구매 계약을 수락한 구매자, 구매자 계정의 전역 고유 식별자(GUID)로 표현됩니다. 계정 데이터 피드에 연결하는 데 사용할 수 있습니다.

열	설명
offer_revision_at_acceptance	이 계약에 해당하는 제안의 표시 ID입니다. 제안 및 제안 대상 데이터 피드에 참여하는 데 사용할 수 있습니다.
start_time	계약이 시작된 날짜와 시간입니다.
end_time	계약이 종료된 날짜와 시간입니다. 종료일이 없는 종량제 계약의 경우 필드는 null입니다.
acceptance_time	제안이 수락되거나 계약이 체결된 날짜와 시간입니다. • 기존 보유 라이선스 사용 계약의 경우 날짜를 이전으로 변경할 수 있습니다. • 계약이 미래 날짜 계약 기능으로 생성된 경우 start_date와 다를 수 있습니다.
의도	계약이 마지막으로 수정되었을 당시 구매자의 의도입니다.
preceding_agreement_id	이전 계약의 계약 ID입니다.
status_reason_code	계약 상태가 변경되는 이유입니다.

계약 데이터 피드 예

vali m	agr _id	pro acc d	acc acc d	offe	offe isio acc e	star e	enc	acc et_t	의도	pre _ag t_id	stat	stat asc	esti _ag t_va	curr cod
202 1 20:: 0	agr 44c 78a 5t	88a xxx xxx xxx x	88a xxx xxx xxx x	offe krx xxx x	1	202 1 20:: 0	202 1 20:: 0	202 1 20:: 0	NE		AC		1,0	USD

결제 이벤트 데이터 피드

이 데이터 피드는 인보이스 발행 및 지불 등 결제 이벤트에 대한 정보를 제공합니다.

예를 들어, 이 데이터 피드를 사용하여 구매자에게 인보이스가 발행되는 시기와 항목을 확인할 수 있습니다. 예제 SQL 쿼리를 사용하여 이 데이터 피드의 데이터를 분석할 수도 있습니다.

이 데이터 피드에는 사용자가 등록 판매자인 결제 이벤트와 관련된 정보가 포함되어 있습니다. 채널 파트너를 통해 체결된 계약의 경우 이 데이터 피드는 제조업자와 등록 판매자 간의 결제 이벤트에 대한 정보를 포함합니다.

결제 이벤트 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

결제 이벤트 데이터는 변경할 수 없습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다.

열 이름	설명
billing_event_id	결제 이벤트의 식별자입니다. 이 ID는 판매자의 환경에서 고유합니다.
from_account_id	결제 이벤트를 시작한 계정입니다. `transaction_type` 이 `SELLER_REV_SHARE` 인 경우 구매자의 지급인 계정입니다. 계정 데이터 피드에 대한 외래 키입니다. `account_id` 필드의 `Account` 데이터 피드에 조인하는 데 사용할 수 있습니다.
to_account_id	제품에 대한 거래 금액을 수취하는 계정입니다. 계정 데이터 피드에 대한 외래 키입니다. `account_id` 필드의 `Account` 데이터 피드에 조인하는 데 사용할 수 있습니다.
end_user_account_id	제품을 사용하는 계정입니다. 이 계정은 구매자 및 지급인 계정과 다를 수 있습니다. `account_id` 필드의 `Account` 데이터 피드에 조인하는 데 사용할 수 있습니다.

열 이름	설명
product_id	제품의 식별자입니다. 제품 데이터 피드에 대한 외래 키입니다. `product_id` 필드의 `Product` 데이터 피드에 조인하는 데 사용할 수 있습니다.
작업	이 이벤트의 작업 유형입니다. 가능한 값은 다음과 같습니다. • `INVOICED` - 구매자에게 해당 금액에 대한 인보이스가 발행되었습니다. • `FORGIVEN` - 구매자에게 해당 금액에 대한 인보이스가 발행되었고 AWS 가 청구를 취소했습니다. • `DISBURSED` - 판매자가 이 금액을 지불했습니다. 한 달의 인보이스를 포함할 수도 있고 온디맨드 지불일 수도 있습니다.

열 이름	설명
transaction_type	거래 유형입니다. 예시는 세금 시나리오 섹션을 참조하세요. 가능한 값은 다음과 같습니다. • SELLER_REV_SHARE - 양수 금액으로, 판매자가 구매자와 계약에서 설정한 가격입니다. • SELLER_TAX_SHARE - 양수 금액으로, 판매자가 부담하는 세금을 충당하기 위해 SELLER_REV_SHARE 에 추가되는 금액입니다. • AWS_REV_SHARE - 음수 금액으로, 등록 수수료입니다. • AWS_TAX_SHARE - 양수 금액입니다.이 금액은 외에 추가로 AWS 징수된 세금 금액입니다SELLER_REV_SHARE . 이 금액은 판매자의 잔액에 영향을 미치지 않습니다. 이 금액은 지불되지 않으며 판매자를 대신하여 구매자에게 청구되고 당국에 송금되는 세금에 대해 판매자에게 알리기 위해 제공됩니다. • *transaction_type* _REFUND - 구매자가 요청한 환불 금액입니다. • *transaction_type* _CREDIT - 구매자에게 AWS 크레딧을 제공하는 금액입니다. • BALANCE_ADJUSTMENT - 인보이스 발행 문제를 해결하기 AWS 위해에서 수행한 조정입니다. • DISBURSEMENT - action의 값이 DISBURSED 이고 balance_impacting 의 값이 1일 때 판매자에게 지급되는 금액입니다. action의 값이 INVOICED인 경우 이 레코드는 parent_billing_event_id 레코드를 전부 또는 일부 무효화합니다. 이 경우 관련 지급 disbursement_billi

열 이름	설명
	ng_event_id 가 표시되고 balance_impacting 의 값은 0입니다. • DISBURSEMENT_FAILURE - 거래를 무효화합니다.

열 이름	설명
parent_billing_event_id	broker_id 가 AWS_INC이고 action이 DISBURSED 또는 FORGIVEN이고 transaction_type 이 DISBURSEMENT 이면 parent_billing_event_id 는 이 결제 이벤트를 시작한 원래 billing_event_id 를 나타냅니다. action에 다른 값이 있는 경우 이 필드는 null입니다. broker_id 가 AWS_EUROPE 이면 parent_billing_event_id 는 다음 시나리오에서 이 결제 이벤트를 시작한 원래 billing_event_id 를 나타냅니다. • action이 FORGIVEN 또는 INVOICED이면 transaction_type 은 AWS_REV_SHARE , AWS_REV_SHARE_REFUND 또는 SELLER_REV_SHARE_REFUND 입니다. • action이 DISBURSED 이면 transaction_type 은 ANY(DISBURSEMENT_FAILURE 제외)입니다. • transaction_type 이 AWS_TAX_SHARE , AWS_TAX_SHARE_REFUND , SELLER_TAX_SHARE 또는 SELLER_TAX_SHARE_REFUND 입니다. • action이 DISBURSED 이고 transaction_type 이 DISBURSEMENT_FAILURE 입니다. broker_id 가 AWS_EUROPE 이면 parent_billing_event_id 는 다음 시나리오에서 이전에 성공한 지급 결제 이벤트의 원래 billing_event_id 를 나타냅니다.

열 이름	설명
	• `action`이 `DISBURSED` 이고 `transaction_type` 이 `DISBURSEMENT_FAILURE` 입니다. `broker_id` 가 `AWS_EUROPE` 이면 나머지 모든 시나리오에서 이 필드는 null입니다.
disbursement_billing_event_id	`action`이 `DISBURSED` 이고 다음 중 하나가 true일 때의 관련 지급입니다. • `transaction_type like ('SELLER%')` • `transaction_type like ('AWS%')` 그 외의 경우에는 이 값이 null입니다.
amount	결제 이벤트 금액입니다.
currency	ISO 639 통화 코드입니다.
balance_impacting	판매자 지불액 계산 시 해당 금액이 고려되는지 여부입니다. 값이 `0`이면 해당 금액이 정보 제공을 위해 표시된 것이며 잔액에는 영향을 주지 않습니다. 값이 `1`이면 판매자 지불액을 결정할 때 이 금액이 고려됩니다.
invoice_date	인보이스가 발행된 날짜입니다.
payment_due_date	`action`의 값이 `INVOICED`일 때 인보이스의 기한입니다.
usage_period_start_date	레코드에 있는 기간 시작 날짜입니다.
usage_period_end_date	레코드에 있는 기간 종료 날짜입니다.
invoice_id	AWS 인보이스 ID입니다.

열 이름	설명
billing_address_id	주소 데이터 피드에서 지급인의 청구지 주소 참조입니다. `address_id` 필드의 `Address` 데이터 피드에 조인하는 데 사용할 수 있습니다.
transaction_reference_id	다음 보고서의 데이터를 상호 참조할 수 있도록 하는 식별자입니다. • 지출 보고서 • 월별 청구 수익 보고서 • 영업 보상 보고서
bank_trace_id	지급 거래(`transaction_type` 은 `DISBURSEMENT` 이고 action은 `DISBURSED`)의 경우 은행에서 할당한 추적 ID입니다. 추적 ID를 사용하여 판매자 은행에서 제공한 보고서와 연결할 수 있습니다.
broker_id	거래를 이끈 기업체의 식별자입니다. 가능한 값은 다음과 같습니다. • `AWS_INC` - Inc AWS.의 식별자입니다(미국 기준). • `AWS_EUROPE` - Amazon Web Services EMEA SARL(룩셈부르크 소재)의 식별자입니다. • `AWS_JAPAN` • `AWS_AUSTRALIA` • `AWS_KOREA` • `NULL` - 명시적 `broker_id` 가 없는 이전 거래는 `AWS_INC`에서 진행했습니다.

열 이름	설명
buyer_transaction_reference_id	GROUP BY 또는 SQL의 창 함수 구조를 사용하여 결제 피드의 모든 관련 레코드를 그룹화하는 식별자입니다. 이러한 관련 레코드에는 구매자 인보이스, 판매자 인보이스, 등록 수수료의 부가가치세(VAT)가 포함될 수 있습니다.

세금 시나리오

구매자와 판매자의 국가와 지방 정부의 과세 모델에 따라 세금을 수금하고 송금하는 방식이 달라집니다. 가능한 시나리오는 다음과 같습니다.

- 세금은에서 징수하고 송금합니다 AWS. 이 경우 `transaction_type`은 `AWS_TAX_SHARE`입니다.
- 세금은 판매자가 징수하고 AWS판매자에게 지급하며 판매자가 세무 당국에 송금합니다. 이 경우 `transaction_type`은 `SELLER_TAX_SHARE`입니다.
- 세금은에서 징수하지 않습니다 AWS. 판매자가 세금을 계산하여 세무 당국에 송금해야 합니다. 이러한 경우 AWS Marketplace 는 세금 계산을 수행하거나 세금 정보를 받지 않습니다. 판매자가 수익에서 세금을 지불합니다.

결제 이벤트 데이터 피드의 예

이 단원에서는 송장 발행 시점부터 한 달 이후의 결제 이벤트 데이터 기간에 대한 예를 보여줍니다. 이 단원의 모든 표에서 다음 사항에 유의하세요.

- 데이터 피드에서 `billing_event_id` 값은 40자의 영숫자 문자열이지만, 여기서는 가독성을 위해 2자의 문자열로 표시되어 있습니다.
- 데이터 피드에서 이 정보는 단일 테이블에 표시됩니다. 가독성을 위해 여기에는 데이터가 여러 표에 나와 있고 모든 열은 표시되어 있지 않습니다.

이 섹션의 예에서는 다음과 같이 가정합니다.

- Arnav는 구매자입니다.
  - 계정 ID는 `737399998888`입니다.
  - Marketplace Facilitator 법률이 적용되는 프랑스에 거주하고 있습니다. 자세한 내용은 Amazon Web Service 세금 도움말을 참조하세요.

  - `prod-o4grxfafcxxxx`를 구매했고 이 제품의 월 사용에 대해 120.60 USD의 인보이스를 받았습니다.
  - 그 달 안에 인보이스를 지불했습니다.
- Jane은 제조업자입니다.
  - 계정 ID는 `111122223333`입니다.
- Paulo는 등록 판매자입니다.
  - 계정 ID는 `777788889999`입니다.
  - Marketplace Facilitator 법률이 적용되지 않는 미국 캔자스주에 거주하고 있습니다.

등록 판매자에 대한 결제 이벤트 데이터 피드

등록 판매자로서 Paulo는 구매자인 Arnav에게 인보이스를 발행합니다.

다음 표는 Paulo가 Arnav에게 인보이스를 발행할 때 Paulo의 데이터 피드에 포함되는 관련 정보를 보여줍니다.

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

다음 표는 Arnav가 인보이스 대금을 지불한 후 월말에 Paulo의 데이터 피드에 포함되는 관련 정보를 보여줍니다.

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		

제조업자에 대한 결제 이벤트 데이터 피드

다음 표는 Paulo가 Arnav에게 인보이스를 발행할 때 Jane의 데이터 피드에 포함되는 관련 정보를 보여줍니다.

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

다음 표는 인보이스 대금이 지불된 후 월말에 Jane의 데이터 피드에 포함되는 관련 정보를 보여줍니다.

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
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		

쿼리 예제

데이터 피드를 사용하여 데이터 수집 및 분석에 설명된 대로 Athena를 사용하여 관리형 Amazon S3 버킷에 데이터 피드로 수집 및 저장된 데이터에 대한 쿼리를 실행할 수 있습니다. 이 단원에서는 이 작업을 수행하는 일반적인 방법의 몇 가지 예제를 제공합니다. 모든 예제에서는 단일 통화가 사용된다고 가정합니다.

예제 1: 세금을 포함한 인보이스 금액

세금을 포함하여 구매자에게 발행된 인보이스 금액을 확인하려면 다음 예제처럼 쿼리를 실행하면 됩니다.

```
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'
  );
```

예제 2: 판매자를 대신하여 구매자에게 발행된 인보이스 금액

판매자를 대신하여 구매자에게 발행된 인보이스 금액을 확인하려면 다음 예제처럼 쿼리를 실행하면 됩니다.

```
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'
;
```

예시 3: 판매자를 대신하여가 수금 AWS 할 수 있는 금액

가 판매자를 대신하여 수금할 AWS 수 있는 금액에서 환급금, 크레딧 및 포기 계정을 뺀 금액을 확인하려면 다음 예제와 같이 쿼리를 실행할 수 있습니다.

```
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')
;
```

예제 4: 판매자가 수금할 수 있는 금액

판매자가 수금할 수 있는 금액을 확인하려면 다음 예제처럼 쿼리를 실행하면 됩니다. 이 예제에서는가 AWS 징수하는 등록 수수료 및 세금을 제거하고 예외적인 잔액 조정을 추가합니다.

```
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')
;
```

또한 다음 예제처럼 다음 쿼리를 사용하여 동일한 정보를 수집할 수도 있습니다.

```
SELECT sum(amount) FROM billing_event
WHERE
  balance_impacting = 1
  and action in ('INVOICED','FORGIVEN')
;
```

다음 예제에서는 동일한 정보를 보여 주지만 2018년 거래로 제한되며 모든 구매자가 인보이스 대금을 지불했다고 가정합니다.

```
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')
;
```

예제 5: 지불 금액

지불된 금액을 확인하려면 다음 예제처럼 쿼리를 실행하면 됩니다.

```
select sum(amount) FROM billing_event
WHERE
  action ='DISBURSED'
  and transaction_type like 'DISBURSEMENT%'
;
```

예제 6: 지불 보류 금액

지급 보류 중인 금액을 확인하려면 다음 예제처럼 쿼리를 실행하면 됩니다. 이 쿼리는 이미 지불된 금액을 제외합니다.

```
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
      )
    )
;
```

동일한 정보를 얻는 또 다른 방법은 다음 예제처럼 쿼리를 실행하여 판매자의 잔액을 구하는 것입니다.

```
SELECT sum(amount) FROM billing_event
WHERE
 balance_impacting = 1
;
```

다음 쿼리는 예제를 확장하여 2018년 거래로 결과를 제한하고 거래에 대한 추가 정보를 반환합니다.

```
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
);
```

예제 7: 인보이스 세트의 잔액

인보이스 세트의 합계를 알아보려면 다음 예제처럼 쿼리를 실행하면 됩니다.

```
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;
```

레거시 매핑 데이터 피드

이 데이터 피드는 제품 ID와 제안 ID가 레거시 GUID(전역 고유 식별자)에 어떻게 매핑되는지를 나열합니다. 레거시 GUID는 이전 보고서에서 사용되었으며 새 ID는 데이터 피드 및 AWS Marketplace API에 사용됩니다.

이 데이터 피드는 등록 판매자로서 생성한 모든 제품과 재판매 권한이 있는 모든 제품에 대한 정보를 제공합니다.

레거시 매핑 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다.

열 이름	설명
mapping_type	제품 ID인지 아니면 제안 ID인지를 나타냅니다.
legacy_id	이 제품 또는 제안의 레거시 ID입니다.
new_id	이 제품 또는 제안의 기억하기 쉬운 ID입니다. 이 ID는 기본 키로, 모든 현재 API 작업에 사용됩니다.

레거시 매핑 데이터 피드의 예

다음은 레거시 매핑 데이터 피드의 예입니다. 가독성을 위해 데이터 기록 열은 나와 있지 않습니다. 데이터 기록 필드에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

mapping_type	legacy_id	new_id
제안	8a806c74-dbd6-403e-9362-bb08f417ff37	offer-dacpxznflfwin
제품	1368541d-890b-4b6c-9bb9-4a55306ab642	prod-o4grxfafcxxxy
제안	558d8382-6b3a-4c75-8345-a627b552f5f1	offer-gszhmle5npzip

제안 데이터 피드

제안 데이터 피드는 등록 판매자로서 생성한 모든 제안에 대한 정보를 제공합니다. 단일 제안에 여러 개정이 있는 경우 모든 개정 버전이 데이터 피드에 포함됩니다.

제안을 개정하여 노출된 필드의 데이터가 변경되면 동일한 프라이머리 키에 대한 새 레코드가 데이터 피드에 생성됩니다(`offer_id` + `offer_revision`). 그러나 `valid_from` 필드의 값은 다릅니다. 데이터 피드 기록 열에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하세요.

제안 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

다음 표에는 데이터 피드 열의 이름과 해당 설명이 나와 있습니다.

열 이름	설명
offer_id	기억하기 쉬운 제안 식별자입니다. Offer_Product 데이터 피드의 offer_id 필드에 조인하는 데 사용할 수 있습니다.
offer_revision	제안 개정. 이 필드와 offer_id 필드가 결합되어 기본 키를 형성합니다. offer_id 필드와 함께, Target_Offer 데이터 피드의 offer_id 및 offer_revision 필드에 조인하는 데 사용할 수 있습니다.
name	판매자가 정의한 제안 이름입니다.
expiration_date	제안이 만료되는 날짜와 시간입니다.
opportunity_name	이 제안에 연결된 모든 기회 데이터입니다. 제안이 AWS 기회에 바인딩된 경우 이 필드가 채워집니다.
opportunity_description	이 제안에 연결된 설명이 포함된 정보입니다. 제안이 AWS 기회에 바인딩된 경우 이 필드가 채워집니다.
seller_account_id	판매자 계정의 전역 고유 식별자(GUID)입니다. 계정 데이터 피드의 account_id 필드에 조인하는 데 사용할 수 있습니다.
opportunity_id	영업 기회 식별자는 리셀러가 귀사의 제품을 판매하는 경우에만 채워집니다. 제품이 동일하면 다른 채널 파트너(또는 판매자)가 생성한 모든 제안의 opportunity_id 도 동일합니다.

제안 데이터 피드의 예

다음은 제안 데이터 피드의 예입니다. 가독성을 위해 데이터 기록 열은 나와 있지 않습니다. 데이터 기록 필드에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하세요.

offer_id	offer_revision	name	expiration_date	opportunity_name	opportunity_description	seller_account_id	opportunity_id
offer-dacpxznflfwin	1	엔터프라이즈 계약 프로그램 제안	9999-01-01T00:00:00Z				
offer-gszhmle5npzip	1	판매자 생성 비공개 제안	2020-10-31T00:00:00Z				
offer-hmzhyle8nphlp	1	엔터프라이즈 계약 프로그램 제안	9999-01-01T00:00:00Z				

제안 제품 데이터 피드

하나의 제안에 여러 제품이 있을 수 있으며 하나의 제품이 여러 제안에 포함될 수 있습니다. 이 데이터 피드는 제안과 제품 간의 관계에 대한 정보를 나열합니다.

이 데이터 피드는 등록 판매자로서 생성한 모든 제품 제안에 대한 정보를 제공합니다.

제안에서 제품을 추가하거나 제거하면 제안 개정이 생성됩니다.

제안 제품 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다. 데이터 피드 기록 열에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

열 이름	설명		
offer_id	이 제안의 기억하기 쉬운 식별자입니다. `Offer` 데이터 피드의 `offer_id` 필드에 조		

열 이름	설명		
	인하는 데 사용할 수 있습니다.		
offer_revision	`offer_id` 필드와 결합하여 제안 개정에 대한 외래 키를 형성합니다.		
product_id	기억하기 쉬운 제품 식별자로, 이 제안이 노출하는 제품에 대한 외래 키입니다. `Product` 데이터 피드의 `product_id` 필드에 조인하는 데 사용할 수 있습니다.		

제안 제품 데이터 피드의 예

다음은 제안 제품 데이터 피드의 예입니다.

offer_id	offer_revision	product_id
offer-dacpxznflfwin	10	prod-o4grxfafcxxxx
offer-gszhmle5npzip	24	prod-o4grxfafcxxxy

제안 대상 데이터 피드

이 데이터 피드는 등록 판매자로서 생성한 모든 제안에 대한 제안 개정의 대상을 나열합니다. 단일 제안에 여러 개정이 있는 경우 모든 개정 버전이 데이터 피드에 포함됩니다.

제안 개정을 생성하여 노출된 필드의 데이터가 변경되면 동일한 기본 키에 대한 새 레코드가 데이터 피드에 생성되지만(`offer_id+offer_revision`) `valid_from` 필드에 다른 값이 적용됩니다.

제안 대상 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다.

열 이름	설명
offer_target_id	피드의 기본 키입니다.
offer_id+offer_revision	제안의 식별자 및 개정입니다. 이러한 두 열은 이 대상과 관련된 제안을 참조합니다. `Target` 데이터 피드의 `offer_id` 및 `offer_revision` 필드에 조인하는 데 사용할 수 있습니다.
target_type	제안 수신자가 `BuyerAccounts` (비공개 제안을 의미)인지 아니면 `ParticipatingPrograms` 인지를 나타냅니다.
polarity	제안이 `target_type` 을 대상으로 하는지 여부를 나타냅니다. 가능한 값은 다음과 같습니다. • `PositiveTargeting` - 제안이 이 `target_type` 에 적용됩니다. • `NegativeTargeting` - 제안이 이 `target_type` 에 적용되지 않습니다.
값	대상을 나타내는 문자열(AWS 계정 ID 또는 제안과 함께 사용할 수 있는 프로그램)입니다. AWS Marketplace 표준 계약(SCMP) 또는 AWS Marketplace Field Demonstration Program(FDP)을 예로 들 수 있습니다.

제안 대상 데이터 피드의 예

다음은 제안 대상 데이터 피드의 예입니다. 가독성을 위해 데이터 기록 열은 나와 있지 않습니다. 데이터 기록 필드에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

offer_target_id	offer_id	offer_revision	target_type	polarity	값
925ddc73f6a373b7d5544ea3210610803b600	offer-dacpxznflfwin	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract
471ff22ae3165278f1fb960d3e14517bcd601	offer-gszhmle5npzip	1	ParticipatingPrograms	PositiveTargeting	FieldDemonstration
511ff22adfj65278f1fb960d3e14517bcd6e602	offer-gszhmle5npzip	1	ParticipatingPrograms	PositiveTargeting	EnterpriseContract

제품 데이터 피드

이 데이터 피드는 등록 판매자로서 생성한 모든 제품과 재판매 권한이 있는 모든 제품에 대한 정보를 제공합니다.

제품 데이터는 변경 가능합니다. 즉, 다음 필드 중 하나의 값을 변경하면 `valid_from` 필드에 다른 값이 적용되어 데이터 피드에 새 레코드가 생성됩니다. 데이터 피드 기록 열에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

제품 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다.

열 이름	설명
product_id	기억하기 쉬운 제품 식별자입니다.

열 이름	설명
	`Account`, `Billing_Event` 및 `Offer_Product` 데이터 피드의 `product_id` 필드에 조인하는 데 사용할 수 있습니다.
manufacturer_account_id	제품 소유자의 식별자입니다. 계정 데이터 피드에 대한 외래 키입니다. `Account` 데이터 피드의 `account_id` 필드에 조인하는 데 사용할 수 있습니다.
product_code	제품의 측정에 사용되는 기존 권한 부여 제품 코드입니다. 이 값은 데이터를 보고서와 조인하거나 AWS Marketplace 측정 서비스에 제공된 내용을 참조하는 데에도 사용됩니다.
title	제품의 제목입니다.

제품 데이터 피드의 예

다음은 제안 대상 데이터 피드의 예입니다. 가독성을 위해 데이터 기록 열은 나와 있지 않습니다. 데이터 기록 필드에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

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

세금 항목 데이터 피드

이 데이터 피드는 고객 인보이스의 세금 계산에 대한 정보를 제공합니다.

주어진 고객 인보이스(`invoice_id`)의 특정 제품(`product_id`)에 여러 항목(`line_item_id`)이 있을 수 있습니다(각 조세 관할 구역에 대해 하나 이상 있을 수 있음). 예를 들어, 다른 AWS 엔터티(예: 미

국 및 아일랜드)에서 서로 다른 AWS 리전 규칙을 사용하는 고객에 대한 사용량 기반 청구서에서 이러한 현상이 발생합니다. 가 판매에 대한 판매세, VAT 또는 GST를 AWS 징수하고 해당 세금을 현지 세무 당국에 송금하는 위치에 대한 자세한 내용은 AWS, Inc. 이름으로 Amazon Web Service Tax Help를 참조하세요.

세금 항목 데이터 피드는 24시간마다 새로 고쳐지므로 매일 새 데이터를 사용할 수 있습니다.

세금 항목 데이터는 변경할 수 없습니다.

다음 표에서는 데이터 피드 열의 이름과 설명을 설명합니다. 데이터 기록 열에 대한 자세한 내용은 데이터에 대한 기록 단원을 참조하십시오.

열 이름	설명
tax_item_id	세금 항목 레코드의 고유 식별자입니다.
invoice_id	AWS 인보이스 ID입니다. 이 값을 `product_id` 값과 함께 사용하여 관련 세금 결제 이벤트를 찾을 수 있습니다.
line_item_id	고객 청구서 항목의 고유 식별자입니다. 환급 거래의 항목 ID는 앞으로의 전달 세금과 동일합니다.
customer_bill_id	고객 청구서의 고유 식별자입니다. 구매자는 이 식별자를 판매자와 공유하여 세금 계산 질문을 식별하고 해결할 수 있습니다.
tax_liable_party	`AWS` 또는 `Seller`입니다. 판매자가 세금 책임 당사자인 경우 세금이 징수됩니다. AWS 가 세금 납부 당사자인 경우에서 판매세를 징수하고 송금합니다 AWS. 자세한 내용은 AWS Marketplace 판매자 및 세금 징수를 참조하세요. 세금이 징수되지 않으면 여기에 값이 표시되지 않습니다. 판매자는 세금 징수에 대한 책임이 있으므로 각 인보이스에 대해 일부 세금이 징수되었는지 여부를 확인해야 합니다.
transaction_type_code	거래 유형입니다. 가능한 값은 다음과 같습니다.

열 이름	설명
	• AWS - 세금 트랜잭션 전달 • REFUND - 전체 또는 일부 환급 • TAXONLYREFUND - 세금만 환급 원래의 선도 거래를 포함한 환불 거래 지분 항목 ID입니다.
product_id	제품에 대한 외래 키입니다. product_id 필드의 Product 데이터 피드에 조인하는 데 사용할 수 있습니다.
product_tax_code	제품의 과세 속성을 식별하는 표준 코드입니다. 판매자는 제품을 생성하거나 수정할 때 속성을 선택합니다.
invoice_date	인보이스가 발행된 날짜입니다.
taxed_customer_account_id	세금이 부과되는 계정 엔터티에 대한 외래 키입니다. account_id 필드의 Account 데이터 피드에 조인하는 데 사용할 수 있습니다.
taxed_customer_country	세금 계산에 사용되는 주소의 ISO 3166 alpha 2 국가 코드입니다.
taxed_customer_state_or_region	세금 계산에 사용되는 주, 지역 또는 지방입니다.
taxed_customer_city	세금 계산에 사용되는 도시입니다.
taxed_customer_postal_code	세금 계산에 사용되는 우편 번호입니다.
tax_location_code_taxed_jurisdiction	과세 지역과 관련된 Vertex 지오코드입니다.

열 이름	설명
tax_type_code	거래에 적용되는 세금 유형입니다. 가능한 값은 `None`, `Sales`, `SellerUse` 입니다.
jurisdiction_level	세금 납부 지역에 사용되는 주소의 관할 구역 수준입니다. 가능한 값은 `State`, `County`, `City` 및 `District`입니다.
taxed_jurisdiction	조세 관할 구역 이름입니다.
display_price_taxability_type	구매자에게 표시된 가격이 세금 포함 가격인지 세금 미포함 가격인지 여부입니다. 모든 AWS Marketplace 상품에는 세금이 포함되어 있지 않습니다.
taxable_amount	이러한 관할 구역 수준에 따라 과세되는 거래 금액입니다.
nontaxable_amount	이러한 관할 구역 수준에 따라 비과세되는 거래 금액입니다.
tax_jurisdiction_rate	관할 구역 수준에서 적용되는 세율입니다.
tax_amount	이 관할 구역 수준에서 부과되는 세액입니다.
tax_currency	위의 금액에 대한 ISO 4217 alpha 3 통화 코드입니다.
tax_calculation_reason_code	관할 구역 수준에서 거래가 과세 대상인지, 비과세 대상인지, 공제인지, 부가가치세가 없는지를 나타냅니다.
date_used_for_tax_calculation	거래의 세금 계산에 사용되는 날짜입니다.
customer_exemption_certificate_id	공제 인증서의 인증서 ID입니다.
customer_exemption_certificate_id_domain	Amazon 시스템에 인증서가 저장되는 위치입니다.

열 이름	설명
customer_exemption_certificate_level	공제를 제공한 관할 구역 수준입니다.
customer_exemption_code	공제를 지정하는 코드입니다(예: RESALE).
customer_exemption_domain	고객 공제 정보(있을 경우)를 캡처하는 데 사용되는 Amazon 시스템입니다.
transaction_reference_id	다음 보고서의 데이터를 상호 참조할 수 있도록 하는 식별자입니다. • 지출 보고서 • 월별 청구 수익 보고서 • 영업 보상 보고서

> **Note**
>
> 2021년 8월 5일부터 AWS Marketplace 판매에 대한 국제 Marketplace Facilitator 세금은 세금 항목 데이터 피드에 항목이 포함됩니다. 즉, 2021년 8월 5일부터 결제 이벤트 데이터 피드의 모든 AWS_TAX_SHARE 및 SELLER_TAX_SHARE 레코드는 세금 항목 데이터 피드에 해당 레코드가 있을 것으로 예상됩니다.

세금 항목 데이터 피드의 예

다음은 세금 항목 데이터 피드의 예입니다. 데이터 피드에서 이 정보는 단일 테이블에 표시됩니다. 가독성을 위해 여기에는 데이터가 여러 표에 나와 있고 모든 열은 표시되어 있지 않습니다.

tax_item_id	invoice_id	line_item_id	customer_bill_id
6p2ni6tu041xagvhbyanbgxl3xameha16txjoav_0001	781216640	71000000000000000000	22100000000000000000

tax_item_id	invoice_id	line_item_id	customer_bill_id
6p2ni6tu041xagvhbyanbgxl3xameha16txjoav_0002	781216640	5300000000000000000000	2210000000000000000000
flr4jobxjzww8czdsrq4noue2uxd56j39wxw0k7_0001	250816266	7640000000000000000000	5720000000000000000000
gfkjjobxjzw56jgkrsrqgjtk52uxd56j39wgj567d_0002	280336288	7640000000000000000000	5724390000000000000000
wwk1qpvb8ran3geiw8e3mp6dgs2qj7wpkuwhgk1_0001	451431024	9930000000000000000000	1230000000000000000000
wwk1qpvb8ran3geiw8e3mp6dgs2qj7wpkuwhgk1_0002	451431024	9930000000000000000000	3120000000000000000000
fnohdid8kwgqq9lvii2k30spn3ftgwihbe8h75x_0001	229987654	9210000000000000000000	6390000000000000000000

tax_liable_party	transaction_type_code	product_id	product_tax_code	invoice_date
판매자	AWS	prod-o4grxfafcxxxx	AWSMP_SOFTWARE_RA	2018-12-31T00:00:00Z
판매자	AWS	prod-o4grxfafcxxxx	AWSMP_SOFTWARE_RA	2018-12-31T00:00:00Z

tax_liable_party	transaction_type_code	product_id	product_tax_code	invoice_date
판매자	AWS	prod-t3grxfafcxxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00:00:00Z
판매자	환급금	prod-t3grxfafcxxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00:00:00Z
판매자	AWS	prod-x8faxxfafcxxy	AWS_REMOTE_ACCESS_SOFTWARE	2018-08-31T00:00:00Z
판매자	세금만 환급	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	미국	GA	밀턴	48573-4839
VIeGa2t9j3MuxioH9wc8lsndXXC	미국	GA	밀턴	48573-4839

taxed_customer_account_id	taxed_customer_country	taxed_customer_state_or_region	taxed_customer_city	taxed_customer_postal_code
gGCGUreeXriocM5				
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	미국	NC	더럼	27517-4834
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	미국	NC	더럼	27517-4834
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	미국	TX	해당 사항 없음	75844-1235
7nyo5jwTRoPlyX81vx9ji04eEwTurO1Ff8biQi88W8	미국	TX	휴스턴	75844-1235
192a0421313e41f069b52962ed7babf716291b688	미국	CT	뉴헤이븐	06002-2948

tax_location_code_taxed_jurisdiction	tax_type_code	jurisdiction_level	taxed_jurisdiction	display_price_taxability_type	taxable_amount	nontaxable_amount
460473664	영업	State	GA	Exclusive	100	0
66301164	영업	카운티	풀턴	Exclusive	0	100
692938178	판매자용	State	NC	Exclusive	58.1	523.8
692938178	판매자용	State	NC	Exclusive	-58.1	523.8
356794387	영업	State	TX	Exclusive	1105.14	0
528887443	영업	구/군/시	휴스턴	Exclusive	-36	0
171248162	영업	State	CT	Exclusive	0	114.55

tax_jurisdication_rate	tax_amount	tax_currency	tax_calculation_reason_code	date_used_for_tax_calculation
0.206	20.6	USD	과세 대상	2018-10-31T00:00:00Z
0	0	USD	비과세 대상	2018-10-31T00:00:00Z
0.1	5.8	USD	과세 대상	2018-07-31T00:00:00Z
0.1	5.8	USD	과세 대상	2018-07-31T00:00:00Z
0.06	66.3	USD	과세 대상	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	비과세 대상	2018-07-31T00:00:00Z
0	0	USD	공제	2019-06-30T00:00:00Z

의 판매자 보고서 AWS Marketplace

> **Important**
>
> 2024년 8월 30일에는 여러 보고서 및 데이터 세트를 중단할 AWS Marketplace 예정입니다.
>
> - Marketplace 레거시 쉼표로 구분된(csv) 판매자 보고서
> - Commerce Analytics Service(CAS) API
> - 연결된 이메일 알림
>
> 보고서 페이지에서 영향을 받는 보고서를 확인하거나 질문이 있는 경우 5/30/2024에 통신이 발행된 AWS 관리자에게 문의하세요.

AWS Marketplace 는 제품 사용, 구매자, 결제 및 결제 정보에 대한 정보가 포함된 보고서를 제공합니다. 보고서는 등록된 모든 AWS Marketplace 판매자가 사용할 수 있습니다.

다음은 보고서 생성에 대한 몇 가지 핵심 사항입니다.

- 보고서는 보고서에 따라 매일, 매주 또는 매월 생성됩니다.
- 보고서는 00:00 UTC에 생성되며 전날 24:00 UTC까지의 정보를 수록합니다.
- 보고서는 .csv 파일로 생성됩니다.
- 암호화된 Amazon S3 버킷으로 데이터가 전달될 때 알리도록 Amazon SNS를 구성할 수 있습니다. Amazon S3 알림을 구성한 후는에 등록한 AWS 계정과 연결된 이메일 주소로 알림을 AWS 보냅니다 AWS Marketplace.

알림 구성 방법에 대한 내용은 Amazon Simple Notification Service 개발자 안내서의 Amazon SNS 시작하기 섹션을 참조하세요.

알림 이메일 수신을 취소하려면 AWS Marketplace 판매자 작업 팀에 문의하세요.

- 각 보고서에 대해 알아보려는 경우 샘플 보고서를 다운로드할 수 있습니다.

보고서 액세스

AWS Marketplace 는 보고서를 구성하는 두 가지 방법을 제공합니다.

- API 인터페이스 사용. AWS Marketplace Commerce Analytics Service를 사용하면 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다.를 사용하면 API 인터페이스를 통해 보고서의 데이터에 자동으로 액세스할 수 있습니다. 정보 수집을 자동화하고, 전체 보고서 대신 일부를 다운로드할 수 있습니다. 서비스는 기존 API와 같이 직접 반환하지 않고 Amazon Simple Storage Service의 파일에 비동기적으로 데이터를 반환합니다. 데이터는 시스템으로 가져오거나 통합할 수 있도록 기계가 판독할 수 있는 형식으로 제공됩니다.
- AWS Marketplace Management Portal에서 보고서 대시보드 사용. 이 대시보드는 이전 보고 기간의 보고서를 제공합니다.

AWS Identity and Access Management (IAM) 권한을 사용하여 보고서에 대한 액세스를 제어할 수 있습니다.

사용 가능한 AWS Marketplace 판매자 보고서

다음 보고서를 사용할 수 있습니다 AWS Marketplace.

- 일별 비즈니스 보고서
- 일별 고객 구독자 보고서
- 지출 보고서
- 월별 청구 수익 보고서
- 영업 보상 보고서

일별 비즈니스 보고서

일일 비즈니스 보고서는 다음에 대한 데이터를 제공합니다.

- AWS 고객이 매일 제품을 사용하는 방법
- 제품 사용에 따른 예상 수익입니다.

관련 정보가 있는 경우에만 이 보고서를 받게 됩니다.

이 보고서를 받지 못했으나 받아야 한다고 판단한 경우, AWS Marketplace Seller Operations 팀에 문의하세요. 보고서에 대한 이메일 알림을 취소하려는 경우에도 해당 팀에 문의할 수 있습니다.

AWS Marketplace Management Portal에서 이 보고서에 액세스할 수 있습니다. the section called "AWS Marketplace 상거래 분석 서비스"에 등록해도 AWS SDK를 사용하여 보고서에 액세스할 수 있습니다.

각 고객에게 고유 식별자를 사용하여 시간이 지나도 보고서 전체에 걸쳐 고객을 식별할 수 있습니다. 이 식별자를 사용하면 고객의 사용 패턴을 추적하여 고객 지출을 예측하고, 무료 평가판 사용 및 연 단위의 사용 동향에 대한 통찰을 얻을 수 있습니다.

게시 일정

이 보고서는 매일 UTC 00:00에 게시되며, 전날 00:00 UTC부터 23:59 UTC까지의 정보입니다. 일정에 대한 예외는 일별 비즈니스 보고서 소개 섹션에 명시되어 있습니다.

주제

- 섹션 1: 인스턴스 유형별 사용량
- 섹션 2: 요금
- 섹션 3: 무료 평가판 변환
- 섹션 4: 새 인스턴스
- 섹션 5: 새 제품 구독자
- 섹션 6: 취소된 제품 구독자

섹션 1: 인스턴스 유형별 사용량

이 섹션에는 고객이 사용하는 각 인스턴스 유형마다 행으로 데이터가 나열됩니다. 예를 들어 고객이 한 인스턴스 유형의 제품과 다른 인스턴스 유형의 동일한 제품을 사용하는 경우, 보고서에는 두 인스턴스 유형 각각에 대한 행이 포함됩니다.

열 이름	설명
고객 레퍼런스 ID	계정 ID가 아닌 고유 식별자. 고객의 사용량, 수익 및 구독을 추적하는 데 효과적입니다. 고객 참조 ID는 2026년 3월 31일에 사용 중단될 예정입니다. 대신 고객 AWS 계정 ID를 사용합니다.
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
사용자의 시/도	제품을 구독한 계정과 연관된 청구지 주소 시/도입니다.
사용자의 국가	제품을 구독한 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
인스턴스 유형	제품 사용량과 연관된 인스턴스 유형(예: t2.micro)입니다.
사용 단위	보고 기간 동안 고객이 사용한 사용량 단위 수입니다.
사용 단위 유형	고객의 사용량을 측정하는 측정 단위입니다. 예: 시간 또는 일수.
제공 설명	제품 제공에 대한 설명입니다. 예를 들어, 제품은 시간 단위 사용, 무료 평가판 사용 또는 연간 사용 형태로 제공됩니다
예상 수익	제품 사용으로 인한 예상 수익입니다. 결제는 월말에 완료됩니다.
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.

열 이름	설명
제안 ID	구매자가 서명한 제안의 식별자입니다.
제공 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
고객 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

섹션 2: 요금

이 섹션에는 제품과 연관된 요금 기반의 거래가 포함됩니다(예: 연간, 월간 및 SaaS 계약 제품 요금, 데이터 제품 구독 요금). 이 섹션의 데이터는 보고서가 생성되기 72시간 전의 24시간에 해당하는 정보입니다. 예를 들어, 보고서가 5월 24일에 생성된 경우, 데이터는 5월 21일 24시간에 해당하는 정보입니다.

열 이름	설명
고객 레퍼런스 ID	계정 ID가 아닌 고유 식별자. 고객의 사용량, 수익 및 구독을 추적하는 데 효과적입니다. 고객 참조 ID는 2026년 3월 31일에 사용 중단될 예정입니다. 대신 고객 AWS 계정 ID를 사용합니다.
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.

열 이름	설명
사용자의 시/도	제품을 구독한 계정과 연관된 청구지 주소 시/도입니다.
사용자의 국가	제품을 구독한 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
Amount	사용료입니다. 환급금이 있는 경우 이 값은 음수입니다. 이 항목이 AWS Marketplace SaaS 계약용인 경우 금액은 전체 계약이 아닌 차원에 대한 요금을 나타냅니다.
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
요금 설명	요금이 발생한 이유: 월별 요금, 연간 요금 또는 환급금 등.
고객 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
고객 시/도	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
고객 구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
고객 우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.

열 이름	설명
고객 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 `liu-jie@example.com` 인 경우 항목은 `example.com` 입니다.
시작일	AWS Marketplace SaaS 계약 또는 데이터 제품 구독의 시작 날짜입니다.
종료일	AWS Marketplace SaaS 계약 또는 데이터 제품 구독의 종료 날짜입니다.
수량	계약이 지정하는 차원의 단위 수입니다.
차원	계약이 지정하는 차원입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

섹션 3: 무료 평가판 변환

이 섹션에는 무료 평가판 시작, 전환 및 취소에 대한 데이터가 수록되어 있으며 이전 24시간 동안의 정보를 다룹니다.

열 이름	설명
제품 제목	제품의 제목입니다.
Product Code	제품을 나타내는 고유 식별자입니다.

열 이름	설명
새 무료 평가판	보고 기간에 시작된 새로운 무료 평가판 개수입니다.
현재 총 무료 평가판	활성 무료 평가판 구독의 총 개수입니다.
변환된 무료 평가판	보고 기간 동안 무료 평가판에서 유료 사용으로 전환한 총 구독자 수입니다.
변환되지 않은 무료 평가판	무료 평가판 기간이 종료되었지만 유료 사용으로 전환하지 않은 총 구독자 수입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.

섹션 4: 새 인스턴스

이 섹션에서는 새 EC2 인스턴스 및 인스턴스 유형에 대한 데이터가 수록되어 있으며, 이전 24시간을 다룹니다.

열 이름	설명
고객 레퍼런스 ID	계정 ID가 아닌 고유 식별자. 고객의 사용량, 수익 및 구독을 추적하는 데 효과적입니다. 고객 참조 ID는 2026년 3월 31일에 사용 중단될 예정입니다. 대신 고객 AWS 계정 ID를 사용합니다.
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
사용자의 시/도	제품을 구독한 계정과 연관된 청구지 주소 시/도입니다.
사용자의 국가	제품을 구독한 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.

열 이름	설명
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
유형	Amazon EC2 인스턴스 유형입니다.
개수	EC2 인스턴스의 개수입니다.
고객 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
고객 시/도	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
고객 구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
고객 우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
고객 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 `liu-jie@example.com` 인 경우 항목은 `example.com` 입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

섹션 5: 새 제품 구독자

이 섹션에는 신규 구매자에 대한 데이터가 나와 있으며, 이전 24시간을 다룹니다.

열 이름	설명
고객 레퍼런스 ID	계정 ID가 아닌 고유 식별자. 고객의 사용량, 수익 및 구독을 추적하는 데 효과적입니다. 고객 참조 ID는 2026년 3월 31일에 사용 중단될 예정입니다. 대신 고객 AWS 계정 ID를 사용합니다.
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
사용자의 시/도	제품을 구독한 계정과 연관된 청구지 주소 시/도입니다.
사용자의 국가	제품을 구독한 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제공 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
고객 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
고객 시/도	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
고객 구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.

열 이름	설명
고객 우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
고객 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 `liu-jie@example.com` 인 경우 항목은 `example.com` 입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정이 아닌 고유 식별자입니다. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

섹션 6: 취소된 제품 구독자

이 섹션에는 구매자 취소에 대한 데이터가 나와 있으며, 이전 24시간을 다룹니다.

열 이름	설명
고객 레퍼런스 ID	계정 ID가 아닌 고유 식별자. 고객의 사용량, 수익 및 구독을 추적하는 데 효과적입니다. 고객 참조 ID는 2026년 3월 31일에 사용 중단될 예정입니다. 대신 고객 AWS 계정 ID를 사용합니다.
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
사용자의 시/도	제품을 구독한 계정과 연관된 청구지 주소 시/도입니다.

열 이름	설명
사용자의 국가	제품을 구독한 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
구독 날짜	구독이 시작된 날짜입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제공 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
고객 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
고객 시/도	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
고객 구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
고객 우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
고객 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 `liu-jie@example.com` 인 경우 항목은 `example.com` 입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.

열 이름	설명
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

일별 고객 구독자 보고서

이 보고서에는 제품을 구매한 고객에 대한 데이터가 나열됩니다. 이 보고서에서는 현재나 과거의 사용이 아니라 고객이 구독하는 제품만 명시합니다. 관련 정보가 있는 경우에만 이 보고서를 받게 됩니다. 이 보고서를 받지 못했으나 받아야 한다고 판단한 경우, AWS Marketplace Seller Operations 팀에 문의하세요. AWS Marketplace 판매자 운영 팀에 문의하여이 보고서에 대한 이메일 알림을 취소할 수도 있습니다.

AWS Marketplace Management Portal에서 이 보고서에 액세스할 수 있습니다. the section called "AWS Marketplace 상거래 분석 서비스"에 등록해도 AWS SDK를 사용하여 보고서에 액세스할 수 있습니다.

보고서에는 두 개의 섹션이 있습니다. 하나는 시간 단위 및 월간 구독이고 다른 하나는 연간 구독입니다. 보고서에는 제품을 구독하는 모든 고객의 AWS 계정 IDs 목록이 포함되어 있습니다.

게시 일정

이 보고서는 매일 UTC 00:00에 게시되며, 전날 00:00 UTC부터 23:59 UTC까지의 정보입니다.

주제

- 섹션 1: 시간별 및 월별 구독
- 섹션 2: 가변 기간 구독

섹션 1: 시간별 및 월별 구독

이 섹션에서는 전날 23:59:59 UTC 기준 모든 사용량 기반 구독에 대한 데이터를 다룹니다.

열 이름	설명
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
제품 제목	제품의 제목입니다.
제품 ID	소프트웨어 제품의 고유 식별자입니다.
Product Code	소프트웨어 제품의 고유 식별자.
구독 시작일	`YYYY-MM-DD` 형식의 구독 시작일입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제공 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
리셀러 계정 ID	채널 파트너 리셀러의 고유 식별자입니다.
리셀러 계정 이름	채널 파트너 리셀러의 이름입니다.

섹션 2: 가변 기간 구독

이 섹션에서는 전날 23:59:59 UTC 기준 모든 요금 기반 구독에 대한 데이터를 다룹니다.

열 이름	설명
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
제품 제목	제품의 제목입니다.

열 이름	설명
제품 ID	소프트웨어 제품의 고유 식별자.
Product Code	소프트웨어 제품의 고유 식별자입니다. 이 정보는 Amazon EC2 인스턴스 메타데이터의 일부로도 제공됩니다.
구독 ID	구독의 ID입니다.
구독 수량	고객기 구입한 총 라이선스 개수입니다.
구독 유형	구독의 유형입니다.
구독 의도	이 제안이 이전 제안의 업그레이드 또는 갱신인지 여부입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
구독 시작일	고객이 제품을 구독한 날짜로 `YYYY-MM-DD` 형식입니다.
이전 제안 ID	업그레이드 또는 갱신 제안 이전에 제출된 제안의 ID입니다(있는 경우).
제공 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
리셀러 계정 ID	채널 파트너 리셀러의 고유 식별자입니다.
리셀러 계정 이름	채널 파트너 리셀러의 이름입니다.

지출 보고서

> **Important**
>
> 이 보고서는 2024년 8월 30일부터 사용할 수 없게 됩니다. 다음 대안을 사용할 수 있습니다.
>
> - 수금 및 지급 대시보드
> - GitHub 의 collections_and_disbursements_report SQL 쿼리입니다.

지출 보고서는 이전 지출 이후로 당사에서 수금하여 귀사의 은행 계좌로 지출된 대금에 대한 정보를 제공합니다. 지출에는 제품 구독에 대한 고객 지불금 또는 환급금, 고객에게 수금했거나 환급된 세금이 포함될 수 있습니다. 고객으로부터 자금을 수금할 때까지는 자금을 지급 받지 못합니다. 고객마다 결제 조건이 다르 AWS므로 수금되지 않은 각 연령 범주의 일부 자금은 고객이 지불하지 않을 수 있습니다.

귀사에서 환급을 승인한 후 고객에게 환급되므로 환급금은 음수로 표시됩니다.

이 보고서는 보고서 탭 AWS Marketplace Management Portal 아래의에서 사용할 수 있습니다. 이 보고서와 유사한 자체 보고서를 생성하려면를 기본 보고서제품별 지급로 AWS Marketplace 의 판매자 전송 데이터 피드 AWS Marketplace포함하여를 사용하여 필요에 맞게 사용자 지정할 수 있습니다.

> **Note**
>
> 이 보고서에 대한 알림 이메일을 취소하려면 AWS Marketplace Seller Operations 팀에 문의하세요.

게시 일정

이 보고서는 은행에 자금을 이체하기 위해 지출이 시작된 후 3~5일이 지나면 게시됩니다. 이것은 일반적으로 월별 케이던스에 대한 지출을 받는 판매자용 보고서입니다. 시작된 지출이 없으면 지출 보고서가 생성되지 않습니다.

주제

- 섹션 1: 제품별 지출 금액
- 섹션 2: 고객 지역별 지출 금액
- 섹션 3: 인스턴스 시간별 지출 금액

- 섹션 4: 미수금 기간
- 섹션 5: 지출대금 기간
- 섹션 6: 기한 초과 대금 기간
- 섹션 7: 미수금 내역

섹션 1: 제품별 지출 금액

이 섹션에는 제품별 지출 데이터가 나와 있습니다.

열 이름	설명
Product	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
판매자의 수익	사용량 또는 제품의 요금으로 고객에게 청구되는 금액입니다.
AWSRefFee	AWS Marketplace 수수료 금액입니다.
판매자의 수익 환급	데이터 적용 기간 중에 환급이 처리된 경우 고객에게 환급되는 구독 비용입니다.
AWSRefFeeRefund	데이터 적용 기간 동안 환급이 처리된 경우 환급되는 AWS Marketplace 수수료 금액입니다.
판매자의 수익 크레딧	가 고객 계정에 AWS Marketplace 부여한 AWS 크레딧입니다.
AWSRefFeeCredit	가 계정에 AWS Marketplace 부여한 AWS 크레딧입니다.
판매 금액	당사에서 귀사에 지급한 총 금액입니다. 이 열은 SellerRev 열에서 AWSRefFee 열을 뺀 값과 같습니다. 고객에게 환급이 이루어지면 이 열은 SellerRevRefund 열에서 AWSRefFeeRefund 열을 뺀 값과 같은 음수입니다.

열 이름	설명
거래 참조 ID	AWS Marketplace 보고서 간에 트랜잭션을 상호 연관시키는 데 도움이 되는 트랜잭션의 고유 식별자입니다.
판매자의 미국 매출세	이 거래를 위해 청구되는 미국 매출세 및 이용세의 총액입니다.
판매자의 미국 매출세 환급	이 거래에서 환급이 처리된 경우, 환급된 총 미국 매출세 및 이용세입니다.
고객 AWS 계정 번호	요금이 청구되는 계정의 ID입니다.
고객 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
고객 시/도	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
고객 구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
고객 우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
고객 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 `liu-jie@example.com` 인 경우 항목은 `example.com` 입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.

열 이름	설명
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

섹션 2: 고객 지역별 지출 금액

이 섹션에는 고객의 지리적 위치별 지출 데이터가 나열됩니다.

열 이름	설명
결산 ID	지출의 고유 식별자입니다.
결산 기간의 시작일	지출 기간의 시작 날짜와 시간입니다.
결산 기간 종료 날짜	지출 기간의 종료 날짜와 시간입니다.
입금 날짜	지출이 발생한 날짜와 시간입니다.
지출한 금액	지출의 총 금액.
국가 코드	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
시/도 또는 지역	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
우편 번호	소프트웨어 요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.

열 이름	설명
세금 납부 지역별 판매 금액	세금 납부 지역별로 판매자에게 지급된 총 자금에서 AWS Marketplace 수수료, 환급금, 미국 매출세 및 이용세를 공제한 값입니다.
세금 납부 지역별 총액	판매자에게 지급된 총 금액을 세금 납부 지역으로 그룹화한 것입니다.
판매자 미국 매출세	판매자를 대신하여 이 거래에 대해 청구되는 미국 매출세 및 이용세의 총액입니다. (즉, 미국 판매 및 세금 보고서의 관련 레코드에는 “tax liable party(세금 책임 당사자)” == “SELLER(판매자)”가 표시됩니다.)
판매자 미국 매출세 환급	판매자를 대신하여 이러한 세금을 처리할 때 이 거래에서 환급이 처리된 경우 환급된 총 미국 매출세 및 이용세입니다. (즉, 미국 판매 및 세금 보고서의 관련 레코드에는 “tax liable party(세금 책임 당사자)” == “SELLER(판매자)”가 표시됩니다.)

섹션 3: 인스턴스 시간별 지출 금액

이 섹션에는 Amazon EC2 인스턴스 시간당 지출 데이터가 나와 있습니다.

열 이름	설명
Product	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
사용 유형의 설명	제공 유형, 리전 및 인스턴스 유형을 포함한 사용에 대한 설명입니다.
Rate	이 제공 유형, 리전 및 인스턴스 유형에 대한 시간당 속도입니다.

열 이름	설명
사용자 개수	이 제공 유형, 지역 및 인스턴스 유형을 사용하는 고유한 고객 수입니다.
인스턴스 시간	인스턴스에서 이 제공 유형, 지역 및 인스턴스 유형별로 사용한 시간 수입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.

섹션 4: 미수금 기간

이 섹션에서는 수금되지 않은 자금의 데이터가 연령별로 나열되어 있습니다. 미수 대금에는 아직 지불되지 않은 금액이 포함될 수 있습니다.

열 이름	설명
미수금(31일 미만 보류 중)	청구되었으나 31일 미만의 기간 동안 수금되지 않은 금액입니다.
미수금(31~60일 보류 중)	청구되었으나 31~60일 동안 수금되지 않은 금액입니다.
미수금(61~90일 보류 중)	청구되었으나 61~90일 동안 수금되지 않은 금액입니다.
미수금(91~120일 보류 중)	청구되었으나 91~120일 동안 수금되지 않은 금액입니다.
미수금(> 120일 보류 중)	청구되었으나 120일 넘게 수금되지 않은 금액입니다.
미수금(전체)	청구되었으나 수금되지 않은 총 금액입니다.

섹션 5: 지출대금 기간

이 섹션에서는 이전 지출 이후 수금된 자금에 대한 데이터를 다룹니다.

열 이름	설명
수금(31일 미만 보류 중)	0~31일 범위 이내에 청구되어 수금된 총 금액입니다.
수금(31~60일 보류 중)	31~60일 범위 이내에 청구되어 수금된 총 금액입니다.
수금(61~90일 보류 중)	61~90일 범위 이내에 청구되어 수금된 총 금액입니다.
수금(91~120일 보류 중)	91~120일 범위 이내에 청구되어 수금된 총 금액입니다.
수금(> 120일 보류 중)	120일 넘는 기간 동안 청구되어 수금된 총 금액입니다.
수금(전체)	모든 수금의 총액.

섹션 6: 기한 초과 대금 기간

이 섹션에서는 고객에 의해 발생했고 지불 가능하지만 고객의 AWS와의 계약에 따라 지불되지 않았던 대금에 대한 데이터를 다룹니다.

열 이름	설명
기한 초과됨(31일 미만)	최근 0~31일 동안 지불 기한이 도래했으나 고객이 지불하지 않은 총 금액입니다.
기한 초과됨(31~60일)	최근 31~60일 동안 지불 기한이 도래했으나 고객이 지불하지 않은 총 금액입니다.
기한 초과됨(61~90일)	최근 61~90일 동안 지불 기한이 도래했으나 고객이 지불하지 않은 총 금액입니다.

열 이름	설명
기한 초과됨(91~120일)	최근 91~120일 동안 지불 기한이 도래했으나 고객이 지불하지 않은 총 금액입니다.
기한 초과됨(120일 초과)	최근 121일 넘는 기간 동안 지불 기한이 도래했으나 고객이 지불하지 않은 총 금액입니다.
기한 초과(전체)	지불 기한이 도래했으나 고객이 지불하지 않은 총 금액입니다.

섹션 7: 미수금 내역

이 섹션에서는 지불 기한 기준으로 정렬된 모든 미수금이 나열되어 있습니다.

열 이름	설명
지급인 AWS 계정 번호	소프트웨어 요금이 청구되는 계정입니다.
Product Code	제품의 고유 식별자입니다.
총 수익	제품 사용 또는 제품 사용에 대한 수수료에 대해 청구된 금액입니다.
AWS 수익 공유	결제 시 청구 금액에서 공제되는 AWS 수수료 금액입니다.
총 환급금	거래에 대한 총 환금액입니다.
AWS 환급 공유	거래에 대해 환급되는 AWS 수수료 부분입니다.
순수익	이 거래에 대해 청구되는 순 금액에서 AWS 수수료, 환급금, 미국 매출세 및 이용세를 뺀 값입니다.
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
AR 기간	거래의 연도와 월(YYYY-MM 형식).

열 이름	설명
거래 참조 ID	AWS Marketplace 보고서 간에 트랜잭션을 상호 연관시키는 데 사용할 수 있는 트랜잭션을 나타내는 고유 식별자입니다.
기회 이름	등록된 기회에 대한 고유 식별자
기회 설명	등록된 기회의 메타데이터입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.
지불 기한	`YYYY-MM-DD` 날짜 형식으로 된 지불 기한.

월별 청구 수익 보고서

> **Important**
>
> 이 보고서의 세 섹션은 2024년 8월 30일부터 사용할 수 없습니다. 섹션 1~3에 다음 대안을 사용할 수 있습니다.
>
> - 청구 수익 대시보드
> - GitHub의 billed_revenue_report SQL 쿼리입니다.
>
> 섹션 4: 유연한 결제 일정이 적용되는 계약은 2024년 8월 30일 이후에도 계속 사용할 수 있습니다.

월별 청구 수익 보고서는 회계 및 기타 재무 보고 목적으로 매월 청구된 수익에 대한 신뢰할 수 있는 정보를 제공합니다. 이 보고서는가 제품의 시간당, 연간 또는 월별 사용량에 대해 고객에게 AWS 청구하는 총 금액을 보여줍니다. 이 보고서는 시간당 사용량과 월별 요금에 대한 청구 금액, 가변 길이 구독, 필드 데모 사용량, 유연한 결제 등 총 4개 섹션으로 구성됩니다.

> **Important**
>
> 이 보고서에 기재된 금액에는 고객에게 청구한 수익만 반영되며 받은 금액은 반영되지 않습니다.

이 보고서는 보고서 탭 AWS Marketplace Management Portal 아래의에서 사용할 수 있습니다. AWS Marketplace 상거래 분석 서비스에 등록한 경우 API 직접 호출을 사용하여이 보고서의 섹션을 풀다운할 수 있습니다. 자세한 내용은 the section called “AWS Marketplace 상거래 분석 서비스” 단원을 참조하십시오.

> **Note**
>
> 이 보고서에 대한 알림 이메일을 취소하려면 AWS Marketplace Seller Operations 팀에 문의하세요.

게시 일정

이 보고서는 매월 15일 00:00 UTC에 매월 게시됩니다.

결제 및 수익 데이터는 다음 달 15일부터 14일까지 적용됩니다.

장기 구독은 매월 1일 00:00 UTC부터 전월 말일 23:59 UTC까지 적용됩니다.

예를 들어 5월 15일에 게시된 보고서는 4월 1일 00:00 UTC에서 4월 30일 23:59 UTC까지의 데이터를 다룹니다.

주제

- 섹션 1: 청구 및 수익 데이터
- 섹션 2: 가변 기간 구독
- 섹션 3: AWS 필드 데모 사용
- 섹션 4: 유연한 결제 일정을 포함하여 생성된 계약

섹션 1: 청구 및 수익 데이터

이 섹션에는 사용량 결제, 환급금, 요금, 수금한 미국 매출세와 이용세에 대한 데이터가 나열됩니다.

열 이름	설명
고객 레퍼런스 ID	계정 ID가 아닌 고유 식별자. 고객의 사용량, 수익 및 구독을 추적하는 데 효과적입니다. 고객 참조 ID는 2025년 12월 31일에 사용 중단될 예정입니다. 대신 고객 AWS 계정 ID를 사용합니다.
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
State	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
고객 청구 금액	사용량 또는 제품의 월간 요금으로 고객에게 청구되는 금액입니다.
AWS 등록 수수료	청구 금액에서 공제할 AWS Marketplace 수수료 금액입니다.
환급 금액	데이터 적용 기간 중에 환급이 처리된 경우 고객에게 환급되는 구독 비용의 총 금액입니다.

열 이름	설명
AWS 요금 환불	데이터 적용 기간 동안 환불이 처리된 경우 환불되는 AWS Marketplace 수수료 부분입니다.
비용	대리점의 상품 비용: 예를 들어 리셀러가 귀사의 제품을 판매할 경우 귀사에 지불하는 금액입니다.
파트너의 수익 금액	거래에 대해 청구되는 총 금액으로, AWS Marketplace 수수료, 환급금, 미국 매출세 및 이용세를 공제한 금액입니다.
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
거래 참조 ID	AWS Marketplace 보고서 간에 트랜잭션을 상호 연관시키는 데 도움이 되는 트랜잭션의 고유 식별자입니다.
미국 매출세 고객 청구 금액	판매자를 대신하여 이 거래에 대해 청구되는 미국 매출세 및 이용세의 총액입니다. (즉, 미국 판매 및 세금 보고서의 관련 레코드에는 “tax liable party(세금 책임 당사자)” == “SELLER(판매자)”가 표시됩니다.)
미국 매출세 환급 금액	판매자를 대신하여 이러한 세금을 처리할 때 이 거래에서 환급이 처리된 경우 환급된 총 미국 매출세 및 이용세입니다. (즉, 미국 판매 및 세금 보고서의 관련 레코드에는 “tax liable party(세금 책임 당사자)” == “SELLER(판매자)”가 표시됩니다.)
제안 ID	구매자가 서명한 제안의 식별자입니다.
제공 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.

열 이름	설명
고객 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어, 이메일 주소가 *liu-jie@example.com* 일 경우 항목은 `example.com` 입니다.
기회 이름	등록된 기회에 대한 고유 식별자
기회 설명	등록된 기회에 대한 메타데이터입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

섹션 2: 가변 기간 구독

이 섹션에서는 요금 기반 비용 청구의 데이터를 다룹니다.

열 이름	설명
고객 레퍼런스 ID	계정 ID가 아닌 고유 식별자. 고객의 사용량, 수익 및 구독을 추적하는 데 효과적입니다. 고객 참조 ID는 2025년 12월 31일에 사용 중단될 예정입니다. 대신 고객 AWS 계정 ID를 사용합니다.
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.

열 이름	설명
국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
State	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
구독 수량	가변 길이 구독 구매의 일부로 지정된 총 라이선스 수입니다.
구독 시작일	가변 길이 구독 구매의 시작일입니다.
구독 종료일	가변 길이 구독 구매의 종료일입니다.
구독의 인스턴스 유형	가변 길이 구독 구매와 연관된 인스턴스 유형입니다.
고객 청구 금액	사용량, 월간 요금 또는 두 항목 모두에 청구된 금액입니다.
AWS 등록 수수료	청구 금액에서 공제되는 AWS Marketplace 수수료 금액입니다.
환급 금액	데이터 적용 기간 중에 환급이 처리된 경우, 고객에게 환급되는 총 금액입니다.
AWS 요금 환불	데이터 적용 기간 동안 환불이 처리된 경우 환불되는 AWS Marketplace 수수료 부분입니다.

열 이름	설명
비용	대리점의 상품 비용: 예를 들어 리셀러가 귀사의 제품을 판매할 경우 귀사에 지불하는 금액입니다.
파트너의 수익 금액	이 거래에 대해 청구되는 총 금액으로, AWS Marketplace 수수료, 환급금, 미국 매출세 및 이용세를 공제한 금액입니다.
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
거래 참조 ID	AWS Marketplace 보고서 간에 트랜잭션을 상호 연관시키는 데 도움이 되는 트랜잭션의 고유 식별자입니다.
미국 매출세 고객 청구 금액	판매자를 대신하여 이 거래에 대해 청구되는 미국 매출세 및 이용세의 총액입니다. (즉, 미국 판매 및 세금 보고서의 관련 레코드에는 “tax liable party(세금 책임 당사자)” == “SELLER(판매자)”가 표시됩니다.)
미국 매출세 환급 금액	판매자를 대신하여 이러한 세금을 처리할 때 이 거래에서 환급이 처리된 경우 환급된 총 미국 매출세 및 이용세입니다. (즉, 미국 판매 및 세금 보고서의 관련 레코드에는 “tax liable party(세금 책임 당사자)” == “SELLER(판매자)”가 표시됩니다.)
고객 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 `liu-jie@example.com` 인 경우 항목은 `example.com` 입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.

열 이름	설명
제공 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
계약 시작일	AWS Marketplace SaaS 계약의 시작 날짜입니다.
계약 종료일	AWS Marketplace SaaS 계약의 종료 날짜입니다.
기회 이름	등록된 기회에 대한 고유 식별자
기회 설명	등록된 기회에 대한 메타데이터입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

섹션 3: AWS 필드 데모 사용

섹션에는 제품의 AWS 필드 데모 사용에 대한 데이터가 나열되어 있습니다. 잠재 고객에게 제품을 시연할 수 있도록 제품을 구성할 수 있습니다. 데모의 사용법은 여기에 나열되어 있습니다.

열 이름	설명
제품 제목	제품의 제목입니다.
Product Code	제품의 고유 식별자입니다.
인스턴스 유형	필드 데모에 관련된 Amazon EC2 인스턴스 유형입니다.

열 이름	설명
사용 단위	이 제품과 관련된 사용 단위의 수입니다.
사용 단위 유형	사용 단위 수와 관련된 사용 단위(예: 시간)입니다.

섹션 4: 유연한 결제 일정을 포함하여 생성된 계약

이 섹션에서는 이전 보고 기간에 유연한 결제 일정을 포함하여 생성된 모든 계약에 대한 데이터를 다룹니다.

열 이름	설명
고객 AWS 계정 번호	요금이 청구되는 AWS 계정 ID입니다.
고객 국가	요금이 청구되는 지급인 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
고객 시/도	요금이 청구되는 지급인 계정과 연결된 청구지 주소 시/도입니다.
고객 구/군/시	요금이 청구되는 지급인 계정과 연결된 청구지 주소 구/군/시입니다.
고객 우편 번호	요금이 청구되는 지급인 계정과 연결된 청구지 주소 우편 번호입니다.
고객 이메일 도메인	요금이 청구되는 지급인 계정과 연결된 이메일 도메인입니다. 예를 들어, 이메일 주소가 *liu-jie@example.com* 일 경우 항목은 `example.com` 입니다.
사용자 참조 ID	요금이 청구되는 지급인 계정의 계정입니다.
사용자 AWS 계정 번호	제품을 구독하는 계정 ID입니다.
제품 ID	제품의 고유 식별자입니다.

열 이름	설명
제품 제목	제품의 제목입니다.
제품 유형	제품의 유형입니다.
AWS Marketplace 제안 ID	구매자가 서명한 제안의 식별자입니다.
계약 생성일	계약 생성일, 즉 계정에서 제공을 구독하는 날짜입니다.
계약 만료 날짜	계약이 만료되는 날짜입니다.
총 계약 금액(USD)	계약의 총액(USD)입니다.
결제 #	계약에 대해 예약된 결제 횟수입니다.
인보이스 날짜	인보이스가 발행된 날짜입니다.
인보이스 금액(USD)	인보이스에 청구된 금액(USD)입니다.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독을 추적하는 데 도움이 됩니다.

영업 보상 보고서

> **Important**
>
> 이 보고서는 2024년 8월 30일부터 사용할 수 없게 됩니다. 다음 대안을 사용할 수 있습니다.
>
> - 청구 수익 대시보드
> - GitHub의 billed_revenue_report SQL 쿼리입니다.

이 보고서에는 표준 the section called “월별 청구 수익 보고서”에 없는 추가 고객 정보와 더불어 월별 청구 수익이 나와 있습니다. 이 보고서는 AWS가 제품을 시간별, 연간 또는 월별로 사용한 고객에게 청구한 총 금액을 나열합니다.

> **Important**
> 이 보고서에 기재된 금액에는 고객에게 청구한 수익만 반영되며 고객으로부터 받은 금액은 반영되지 않습니다.

영업 보상 보고서 및 이 프로그램의 일환으로 공유되는 정보는 귀하와 맺은 비밀 유지 계약에 따라 또는 그러한 계약이 없을 경우 AWS Marketplace 판매자용 이용 약관에 따라 Amazon의 기밀 정보로 간주됩니다. 이 정보는 회사 이름, 지역 및 AWS 계정 ID로 AWS Marketplace 수익을 영업 담당자에게 매핑하여 보상금을 지급할 용도로만 사용할 수 있습니다. 자신에게 지급되는 수수료의 출처를 이해하기 위해 이 정보를 알아야 하는 직원과 이 정보를 공유할 수 있습니다. 해당 정보의 사용과 공유가 AWS Marketplace 판매자용 이용 약관의 3.8항을 포함하되 이에 국한되지 않고 귀하와 맺은 비밀 유지 계약 및 AWS Marketplace 판매자용 이용 약관의 의무를 준수해야 합니다.

이 보고서와 비슷한 보고서를 직접 생성하려면 요구 사항에 맞게 사용자 지정할 수 있는 기본 보고서인 영업 보상 보고서를 포함하여 AWS Marketplace 의 판매자 전송 데이터 피드 AWS Marketplace를 사용하면 됩니다.

게시 일정

이 보고서는 매월 15일 00:00 UTC에 게시됩니다. 이 보고서는 전월의 마지막 역일 00:00 UTC부터 23:59 UTC까지의 기간을 다룹니다. 예를 들어 5월 15일에 게시된 보고서는 4월 1일 00:00 UTC에서 4월 30일 23:59 UTC까지의 데이터를 다룹니다.

청구 수익

이 보고서의 청구 수익 섹션은 이전 달의 사용량 및 수수료 기반 요금을 포함합니다. 다음은 열 이름과 설명입니다.

> **Note**
> 이 보고서에서 등록 수수료는 계층화된 등록 수수료에 따라 결정된 거래 수익금(공인 재판매 제품의 공인 리셀러가 재판매로부터 얻은 수익금 제외)의 백분율입니다.

열 이름	설명
고객 AWS 계정 번호	요금이 청구되는 계정입니다.

열 이름	설명
국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다. 이 보고서는 ISO 3166-1 alpha-2 표준을 사용합니다.
State	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
우편 번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 `liu-jie@example.com` 인 경우 항목은 `example.com` 입니다.
Product Code	제품의 고유 식별자입니다.
제품 제목	제품의 제목입니다.
총 수익	제품 사용 또는 제품 사용에 대한 월 수수료에 대해 청구된 금액.
AWS 수익 배분	결제 시 청구 금액에서 공제되는 AWS 수수료 금액입니다. 이는 the section called “지출 보고서”에 표시됩니다.
총 환급금	거래에 대한 총 환금액입니다.
AWS 환급금 배분	AWS 수수료 중 거래에 대해 환급되는 부분입니다.
순수익	이 거래에 청구되는 순 금액에서 AWS 수수료, 환급금, 미국 매출세 및 이용세를 공제한 값입니다.

열 이름	설명
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
AR 기간	거래의 연도와 월(`YYYY-MM` 형식).
거래 참조 ID	AWS Marketplace 보고서 간의 거래를 연결하는 데 사용할 수 있는 거래를 나타내는 고유 식별자.
기회 이름	등록된 기회에 대한 고유 식별자
기회 설명	등록된 기회의 메타데이터입니다.
솔루션 제목	솔루션의 이름.
솔루션 ID	솔루션의 고유 식별자.
지급인 참조 ID	계정 ID가 아닌 고유 식별자. 수수료가 청구되는 계정과 연결되어 있습니다. 모든 AWS Marketplace 재무 보고서에서 고객의 사용량, 수익 및 구독 수를 추적하는 데 도움이 됩니다.
지급인 주소 ID	고객의 주소를 나타내는 고유 식별자.

보충 보고서

AWS Marketplace 는 의 판매자 전송 데이터 피드 AWS Marketplace 판매자 AWS Marketplace 목록과 연결된 AWS Marketplace 판매자 계정 ID에 연결된 판매자 소유 Amazon S3 계정에를 통해 보충 보고서를 제공합니다. 자세한 내용은 대상 Amazon Simple Storage Service 버킷 생성을 참조하세요.

전날에 새 구독자가 있었으면 매일 16:00 UTC에 보충 보고서가 게시됩니다. 이 보고서는 전날 13:59 UTC부터 다음 날 16:01 UTC까지의 내용을 다룹니다.

계약 세부 정보 보고서

계약 세부 정보 보고서는 서비스형 소프트웨어(SaaS) 계약 무료 평가판을 사용하는 고객을 지원하는 데 도움이 됩니다. 이 보고서에는 구독자 이름, 구독자 ID, 제안 ID, 계약 시작 날짜, 계약 종료 날짜와 같은 계약 세부 정보가 포함됩니다.

관련 정보가 있는 경우에만 이 보고서를 받게 됩니다. 이 보고서를 받아야 하는데 받지 못했다고 생각되면 AWS Marketplace 판매자 운영 팀에 문의하세요.

AWS Marketplace 판매자 계정 ID와 연결된 Amazon S3 버킷을 통해이 보고서에 액세스할 수 있습니다.

다음 표에는 계약 세부 정보 보고서의 열 이름과 설명이 나와 있습니다.

SaaS 계약 무료 평가판 보고서 데이터

명칭	설명
`vendor_display_name`	제품을 판매한 공급업체의 이름입니다.
`vendor_aws_account_id`	제품을 판매한 공급업체의 ID입니다.
`subscriber_aws_account_id`	제품을 구독 AWS 계정 하는와 연결된 식별입니다.
`customer_id`	소프트웨어 제품의 고유 식별자.
`product_title`	제품의 제목입니다.
`offer_id`	구매자가 서명한 제안의 식별자입니다.
`offer_visibility`	제안이 공개 제안, 비공개 제안 또는 엔터프라이즈 계약 제안 중 무엇인지 나타냅니다.
`reseller_name`	채널 파트너 리셀러의 이름입니다.
`reseller_aws_account_id`	채널 파트너 리셀러의 고유 식별자입니다.
`agreement_id`	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.

명칭	설명
agreement_acceptance_date	계약이 체결된 날짜입니다.
agreement_start_date	계약의 시작 날짜입니다.
agreement_end_date	계약의 종료 날짜입니다. 종량제, 사용한 만큼만 지불 및 구독의 경우 이 날짜는 9999년 1월 1일로 설정됩니다.
is_free_trial_offer	제안 또는 계약이 무료 평가판 제안인지 여부를 나타내는 플래그입니다.
is_upgraded_after_free_trial	계약이 유료 계약으로 업그레이드되었는지 여부를 나타내는 플래그입니다.
total_contract_value	계약의 총액입니다.

판매자 대시보드

AWS Marketplace 는 Amazon QuickSight 기반 대시보드에 재무, 영업 및 마케팅 데이터에 액세스하고 분석하는 데 도움이 되는 차트, 그래프 및 인사이트를 제공합니다. 판매자 대시보드는 다음과 같이 구성되어 있습니다.

the section called “재무 운영용 대시보드”

- the section called “청구 수익 대시보드” - 청구 수익에 대한 정보를 회계 및 기타 재무 보고 목적으로 제공합니다.
- the section called “수금 및 지급 대시보드” - 이전 지급 이후 AWS 수집하여 은행 계좌로 지급한 자금에 대한 정보를 제공합니다.
- the section called “과세 대시보드” - 판매자 트랜잭션에 대한 세금 정보를 제공합니다.

영업 운영을 위한 대시보드

- the section called “계약 및 갱신 대시보드” - AWS Marketplace에서 계약 서명 후 24시간 이내의 계약 및 갱신에 대한 정보를 제공합니다.

- the section called “사용 대시보드” - SaaS 및 서버 사용량 기반 제품을 사용하는 고객에게 시각화와 세분화된 데이터를 제공합니다.
- 마케팅용 대시보드 - 마케팅 데이터를 추적하는 데 도움이 되는 여러 대시보드를 제공합니다.

마케팅용 대시보드

- 고객 계약 대시보드 - 제품을 구독하는 계약 및 고객에 대한 데이터를 제공합니다.
- 리스팅 성과 대시보드 - AWS Marketplace 목록의 트래픽 및 사용자 동작에 대한 데이터를 제공합니다.
- 검색 성능 대시보드 - AWS Marketplace 목록에 적용된 키워드에 대한 데이터를 제공합니다.

대시보드는 적절한 권한이 있는 AWS Marketplace 판매자가 사용할 수 있습니다.

대시보드 액세스

기본적으로 판매자 계정의 AWS Marketplace 시스템 관리자는의 인사이트 탭에서 모든 대시보드에 액세스할 수 있습니다 AWS Marketplace Management Portal. 시스템 관리자는 AWS Identity and Access Management (IAM) 정책을 생성하여 판매자 회사의 다른 사용자에게 특정 대시보드에 대한 액세스를 제공할 수 있습니다.

> **Note**
>
> 2023년 9월부터는 레거시 IAM 권한으로 활성화된 판매자 대시보드에 더 이상 액세스할 수 없습니다. 아래 코드 예제의 새로운 Amazon 리소스 이름(ARN) 형식을 사용하여 IAM 권한을 업데이트하세요.

정책 생성에 대한 자세한 내용은 IAM 정책 생성을 참조하세요.

대시보드 정책

다음 정책 중 하나를 사용하여 일부 또는 모든 대시보드에 대한 액세스를 제공합니다.

다음 예제에서는 대시보드 및 보고서를 포함하여 현재 및 향후 AWS Marketplace 리소스에 대한 액세스를 제공합니다. 모든 현재 및 미래 데이터 피드를 사용합니다.

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

```
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

ARNs을 포함하여 하나 이상의 대시보드에 대한 액세스를 제공할 수도 있습니다. 다음 예제에서는 모든 대시보드에 대한 액세스를 제공하는 방법을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/
BillingEvent_V1/Dashboard/BilledRevenue_V1",
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/
BillingEvent_V1/Dashboard/CollectionsAndDisbursements_V1",
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Agreement_V1/
Dashboard/AgreementsAndRenewals_V1"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Usage_V1/
Dashboard/Usage_V1"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/TaxItem_V1/
Dashboard/Tax_V1"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/CustomerAgreements_V1"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/ListingPerformance_V1"
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/SearchPerformance_V1"
    ]
  }]
}
```

대시보드에 대한 액세스를 제거하려면 정책의 `Resource` 섹션에서 대시보드를 삭제합니다.

> **Note**
>
> AWS Identity and Access Management (IAM) 정책 생성에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 정책 생성을 참조하세요.

재무 운영용 대시보드

AWS Marketplace 는 금융 데이터를 추적하는 데 도움이 되는 여러 대시보드를 제공합니다.

주제

- 청구 수익 대시보드
- 수금 및 지급 대시보드
- 과세 대시보드

청구 수익 대시보드

청구 수익 대시보드는 AWS Marketplace의 모든 청구 매출에 대한 정보를 제공합니다. 이 보고서는 시간을 절약할 수 있을 것으로 예상되며 레거시 월별 청구 수익 보고서보다 약 45일 일찍 제공됩니다. 레거시 보고서는 매월 15일에 제공되므로 이전 달의 청구 내역이 늦게 제공됩니다.

이 대시보드를 열려면에 로그인하고 인사이트AWS Marketplace Management Portal, 재무 작업을 선택한 다음 청구 수익 탭을 선택합니다.

주제

- 청구 수익 대시보드의 새로고침 빈도
- 섹션 1: 컨트롤
- 섹션 2: 필터
- 섹션 3: 주요 지표
- 섹션 4: 추세
- 섹션 5: 분석
- 섹션 6: 세분화된 데이터

청구 수익 대시보드의 새로고침 빈도

대시보드는 매일 자정(UTC)에 업데이트됩니다. 인보이스가 5월 10일 자정(UTC) 전에 만들어진 경우 5월 11일 업데이트에 인보이스가 표시됩니다. 인보이스가 5월 10일 자정(UTC) 후에 만들어진 경우 5월 12일 업데이트에 인보이스가 표시됩니다. 업스트림 시스템에서 수신한 최신 인보이스 발행 또는 결제 데이터가 지연되는 경우 최신 데이터가 대시보드에 반영되는 데 1~2일의 지연이 발생할 수 있습니다.

예상 고객 인보이스가 보이지 않는 경우 AWS Marketplace 비즈니스 개발 담당자에게 문의하기 전에 다음 절차를 따르십시오.

청구 수익 대시보드의 고객 인보이스 누락 문제를 해결하는 방법

1. 다음 도구 중 하나 이상을 사용하여 고객이 제안을 수락했는지 확인합니다.
   - 일별 고객 구독자 보고서
   - AWS Marketplace 이벤트에 대한 알림
   - AWS Marketplace Management Portal 계약 탭
2. (비공개 제안의 경우) 고객이 제안을 수락한 것을 확인한 후, 제안 인보이스 일정과 금액을 검토합니다.
   - 비공개 제안의 경우 AWS Marketplace Management Portal 제안 탭을 확인하세요.
   - 채널 파트너 비공개 제안의 경우 AWS Marketplace Management Portal 파트너 탭을 확인하여 파트너 기회 내에서 사용자 지정 결제 일정을 확인합니다.
3. 매월 2일과 3일에 전월 사용량 요금이 청구되는 사용량 요금 모델을 거래에 사용하는 것을 고려해 봅니다.

섹션 1: 컨트롤

대시보드의 이 섹션에서는 청구 수익 대시보드 데이터를 세분화하는 필터를 제공합니다. 예를 들어 AWS Marketplace 이벤트 알림에서 필드의 필터를 선택하여 특정 고객 계정 ID, 구독자 회사 이름 또는 제안 ID에 대한 결제를 확인할 수 있습니다. 시각적 객체에 포함할 날짜 범위와 같은 필터를 분석에 추가할 수도 있습니다. 컨트롤 내에서 선택한 필터는 지표, 추세, 분류 및 세분화된 데이터 섹션에 표시되는 데이터를 업데이트합니다.

필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. QuickSight

컨트롤 설명

컨트롤 이름	설명
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 회사 이름	제품을 구독하는 계정 이름입니다.
제품 제목	제품의 제목입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제안 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.
AWS 레코드 판매자	거래를 이끈 기업체의 식별자입니다. 가능한 값은 다음과 같습니다. • AWS_INC: Inc AWS.(미국 기반)의 식별자입니다. • AWS_EUROPE: AWS EMEA SARL의 식별자입니다(룩셈부르크 기준). • AWS_AUSTRALIA: AWS Australia Pty Ltd의 식별자 • AWS_JAPAN: AWS 일본 G.K의 식별자입니다. • AWS_KOREA: AWS 한국의 식별자입니다.
지급인 AWS 계정 ID	요금이 청구되는 계정의 ID입니다.
지급인 회사 이름	요금이 청구되는 계정의 상호입니다.
리셀러 회사	소프트웨어 제조업체의 제품 판매 허가를 받은 리셀러 계정의 상호입니다.

컨트롤 이름	설명
리셀러 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
재판매 권한 부여 ID	등록된 기회에 대한 고유 식별자
재판매 권한 부여 이름	등록된 기회의 고유한 이름입니다.
구독자 계정	제품을 구독한 계정과 연결된 2자 국가 코드입니다.
구독자 시/도 또는 리전	제품을 구독한 계정과 연결된 청구지 주소 시/도 또는 리전입니다.
거래 참조 ID	AWS Marketplace 레거시 보고서 간에 트랜잭션을 상호 연관시키는 데 도움이 되는 트랜잭션의 고유 식별자입니다.

섹션 2: 필터

대시보드의 이 섹션에서는 제안의 통화와 인보이스 날짜 범위를 기준으로 레코드를 구체화하는 필터를 제공합니다. 인보이스 날짜 범위를 사용하여 청구 날짜가 지정된 날짜 이전인지 이후인지 또는 날짜 범위 내에 있는지 선택할 수 있습니다. 선택한 필터는 지표, 추세, 분석 및 세부 데이터 섹션에 표시되는 데이터를 업데이트합니다. 기본 인보이스 날짜 범위는 지난 6개월의 데이터를 가져오도록 설정됩니다.

섹션 3: 주요 지표

대시보드의 이 섹션에는 선택한 필터를 사용하여 주요 청구 수익 수치를 비교하여 시각화하는 핵심 성과 지표(KPI)가 표시됩니다. 지정된 인보이스 날짜 범위와 통화의 총 수익, 총 환불, 리스팅 요금, 도매 비용(해당하는 경우) 및 판매자 순수익에 대한 KPI가 표시됩니다. 필터 기준을 업데이트하여 다양한 통화 및 날짜를 기준으로 하는 지표를 볼 수 있습니다.

섹션 4: 추세

대시보드의 이 섹션에서는 지정된 날짜 범위의 청구 수익 추세를 제공합니다. 일별, 전월 대비, 전분기 대비, 전년 대비 등과 같이 지정된 날짜 집계별로 추세를 확인하여 청구 수익에 대한 인사이트를 얻을 수 있습니다. 청구 수익 추세 정보는 총계 수준 또는 제안 가시성 유형별로 제공됩니다.

- 결제 추세 - 날짜 집계 필터에서 선택한 날짜 범위의 시간 경과에 따른 총 수익, 판매자 순수익, 도매 비용(해당하는 경우) 및 환불의 개요를 제공합니다.
- 제안 가시성별 결제 추세 - 비공개 제안, 공개 제안(또는 셀프 서비스), 엔터프라이즈 프로그램 전체의 시간 경과에 따른 제안 가시성 유형별 제안 수와 총 수익의 개요를 제공합니다.

섹션 5: 분석

대시보드의 이 섹션에서는 구독자, 등록 판매자, 구독자 지역 및 제품명 전반의 비즈니스에 대한 주요 지표를 제공합니다. 총 수익, 지급인 수, 구독자 수, 총 환불, 리스팅 요금, 판매자 순수익 및 도매 비용으로 필터링할 수 있습니다.

섹션 6: 세분화된 데이터

대시보드의이 섹션에는가 제품의 시간당, 연간 또는 월별 사용량에 대해 고객에게 AWS 청구하는 총 금액을 포함하여 청구된 모든 매출이 표시됩니다.는 다음 세 가지 빈도를 사용하여 고객에게 AWS 청구합니다.

- 구독 수락 날짜(선결제)
- 사용자 지정 결제 일정(유연한 결제 스케줄러로 구축한 비공개 제안)
- 매월 2일과 3일에 전월 사용량을 기준으로 측정된 사용량

> **Note**
>
> 2021년 4월 1일 이전에 생성된 인보이스에는 연결된 계약 ID, 제안 ID, 구독자 AWS 계정 ID 또는 구독자 회사 이름이 없을 수 있습니다.

고객 요금은 고객에게 청구된 지 24시간 후에 이 세분화된 데이터 섹션에 표시됩니다. 예를 들어 고객에게 11월 3일에 요금이 청구된 경우 대시보드에 이 인보이스가 표시되는 것은 11월 4일입니다. QuickSight 테이블에서 데이터를 내보내고 다운로드하는 방법에 대한 자세한 내용은 QuickSight 사용 설명서의 시각적 객체에서 데이터 내보내기를 참조하세요.

세분화된 데이터 설명

열	설명
인보이스 날짜	고객에게 제품 구독 요금이 청구된 날짜.

열	설명
결제 기한	YYYY-MM-DD 날짜 형식으로 된 지불 기한.
결제 조건	고객의 AWS인보이스 결제 조건입니다.
인보이스 ID	요금이 청구된 인 AWS 보이스에 할당된 AWS ID입니다.
리스팅 요금 인보이스 ID	AWS Marketplace 구독이 AWS EMEA SARL, 일본 또는 호주 법인(등록 판매자)을 통해 거래되는 경우 판매 운영자는 판매자에게 판매자 등록 수수료에 대한 부가가치세(VAT)를 부과하고 세금 준수 인보이스를 제공해야 합니다. 해당하는 거래의 경우 리스팅 수수료에 부과된 VAT의 인보이스 ID는 소프트웨어 또는 제품 구독 인보이스 ID와 다릅니다. AWS, Inc.의 거래는 리스팅 요금 인보이스 ID가 적용되지 않으므로 "Missing_listing_fee_invoice_id" 값을 갖습니다.
도매 인보이스 ID	채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매를 나타내는 미지급 인보이스에 할당된 AWS ID입니다. 공개 제안 및 Marketplace 비공개 제안(MPOs 경우이 필드는 '해당 없음'이 됩니다.
구독자 회사 이름	제품을 구독하는 계정 이름입니다.
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 이메일 도메인	제품을 구독한 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 liu-jie@example.com이면 항목은 example.com입니다.
구독자 구/군/시	제품을 구독한 계정과 연결된 청구지 주소 구/군/시입니다.
구독자 시/도 또는 리전	제품을 구독한 계정과 연결된 청구지 주소 시/도입니다.

열	설명
구독자 계정	제품을 구독한 계정과 연결된 청구지 주소 국가입니다.
구독자 우편 번호	제품을 구독한 계정과 연결된 청구지 우편 번호입니다.
제품 제목	제품의 제목입니다.
제안 이름	판매자가 정의한 제안 이름입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제안 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.
계약 시작 날짜	고객의 제품 구독이 시작되는 날짜이며 MM-DD-YYYY 형식입니다. 미래 날짜의 계약인 경우 이 날짜가 수락 날짜와 다를 수 있습니다.
계약 수락 날짜	고객이 제품을 구독한 날짜이며 MM-DD-YYYY 형식입니다.
계약 종료 날짜	계약이 만료되는 날짜이며 MM-DD-YYYY 형식입니다. 종량제/사용한 만큼만 지불 구독의 경우 이 날짜는 9999년 1월 1일로 설정됩니다.
사용 기간 종료 날짜	제품 사용 기간의 종료 날짜입니다.
사용 기간 시작 날짜	제품 사용 기간의 시작 날짜입니다.
지급 상태	AWS 가 은행 계좌에 자금, 부분 또는 전체 금액을 지급하지 않았음을 확인하는 인보이스와 연결된 상태입니다. 가능한 상태는 지급됨, 부분적으로 지급됨, 실패, 지급되지 않음입니다.

열	설명
지급 날짜	판매자의 은행에 대한 지급이 AWS 시작된 날짜입니다.
지급 은행 추적 ID	지급의 경우 은행이 추적 ID를 할당합니다. 지급 은행 추적 ID를 사용하여 판매자 은행에서 제공한 입금 알림 및 보고서를 AWS Marketplace 보고서의 인보이스와 연관시킬 수 있습니다.
총 수익	사용량 또는 제품의 월간 요금으로 고객에게 청구되는 금액입니다.
총 환급금	데이터 적용 기간 중에 환급이 처리된 경우 고객에게 환급되는 구독 비용의 총 금액입니다.
등록 수수료	청구 금액에서 공제되는 AWS Marketplace 수수료 금액입니다.
등록 수수료 환불	데이터 적용 기간 동안 환급이 처리된 경우 환급되는 AWS Marketplace 수수료 부분입니다.
리스팅 수수료 비율	청구 금액에서 공제할 AWS Marketplace 수수료 비율입니다.
판매자 세금	이 거래에 청구되는 미국 매출세 및 이용세의 총액입니다.
판매자 세금 환급	이 거래에서 환급이 처리된 경우 환급된 미국 매출세 및 이용세의 총액입니다.
AWS 세금 분담	판매자를 대신하여 이 거래에 청구되는 미국 매출세 및 이용세의 총액입니다.
AWS 세금 분담 환급	판매자를 대신하여 이러한 세금을 처리할 때 이 거래에서 환급이 처리된 경우 환급된 총 미국 매출세 및 이용세입니다.

열	설명
도매가	채널 파트너 비공개 제안에만 해당됩니다. 리셀러에게 지급되는 제품의 비용입니다. 리셀러가 제조업체의 제품을 판매할 때 제조업체에 지불하는 금액을 예로 들 수 있습니다. 도매가는 정가에 할인율을 곱한 가격입니다.
도매가 환불	채널 파트너 비공개 제안에만 해당됩니다. 리셀러가 환불한 제품 금액입니다.
도매 판매자 세금 공제	판매자가 세금을 지불해야 하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금입니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 판매자 세금 분담 환급	판매자가 책임을 지는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 환급. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 기타 판매자 세금 분담	판매자가 세금을 지불해야 하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금입니다. 이 필드는 거래에 관여한 다른 판매자가 책임을 지게 되면 채워집니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 기타 판매자 세금 분담 환급	판매자가 책임을 지는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 환급. 이 필드는 거래에 관여한 다른 판매자가 책임을 지게 되면 채워집니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.

열	설명
도매 AWS 세금 분담	가 세금을 부담하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 AWS 입니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 AWS 세금 분담 환급	가 세금을 부담하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 환급 AWS . 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
판매자 순수익	거래에 대해 청구되는 총 금액으로, AWS Marketplace 수수료, 환급금, 미국 매출세 및 이용세를 공제한 금액입니다.
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
거래 참조 ID	AWS Marketplace 보고서 간에 트랜잭션을 상호 연관시키는 데 사용할 수 있는 트랜잭션을 나타내는 고유 식별자입니다.
AWS 레코드 판매자	거래를 이끈 기업체의 식별자입니다. 가능한 값은 다음과 같습니다. • AWS_INC: AWS, Inc.의 식별자(미국 기반) • AWS_EUROPE: AWS EMEA SARL의 식별자(룩셈부르크 기반) • AWS AUSTRALIA: AWS Australia Pty Ltd의 식별자 • AWS_JAPAN: AWS 일본 G.K의 식별자입니다. • AWS_KOREA: AWS 한국의 식별자

열	설명
재판매 권한 부여 ID	등록된 기회에 대한 고유 식별자
재판매 권한 부여 이름	등록된 기회의 고유한 이름입니다.
재판매 권한 부여 설명	등록된 기회에 대해 ISV가 정의한 설명입니다.
리셀러 회사 이름	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 이름입니다.
지급인 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
지급인 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 liu-jie@example.com이면 항목은 example.com입니다.
지급인 구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
지급인 시/도 또는 리전	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
지급인 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다.
지급인 우편번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
ISV 계정 ID	제품 또는 서비스 소유자의 식별자입니다.
ISV 회사 이름	제품 또는 서비스 소유자의 상호입니다.
제품 ID	소프트웨어 제품의 익숙한 고유 식별자입니다.
지급된 순수익	판매자에게 지급된 총 거래 금액. 금액이 "판매자 순수익"과 같지 않으면 부분 결제입니다.

열	설명
지급되지 않은 순수익	판매자에게 지급되지 않은 총 거래 금액입니다. 금액이 0이 아닌 경우 고객이 남은 잔액을 지불해야 합니다.

수금 및 지급 대시보드

수금 및 지급 대시보드는 은행 계좌로 지급된 자금에 대한 데이터를 제공합니다. 또한 부분 결제, 전체 결제에 대한 결제가 전송되었거나 결제 실패가 발생한 모든 인보이스 목록도 제공합니다. 이 목록에는 지급이 발생하지 않은 인보이스가 포함되지 않습니다. 지급되지 않은 인보이스를 검토하려면 청구 수익 대시보드를 사용합니다.

지급에는 고객 결제, 제품 구독에 대한 합의된 환급, 고객에게 징수되거나 환급되는 일부 세금이 포함됩니다. 판매자가 환급을 승인한 후 고객에게 환급되므로 대시보드의 환급금은 음수로 표시됩니다.

컬렉션 및 지급 대시보드를 사용하면 고객 지급에 더 빠르게 액세스할 수 있습니다. 대금이 지급되고 5일 후에 생성되는 레거시 지급 보고서에 비해 약 4일을 절약할 수 있을 것으로 예상됩니다.

> **Note**
>
> 고객은 결제 조건이 다르 AWS므로 수금되지 않은 연령 범주의 일부 자금은 고객의 결제 기한이 아닐 수 있습니다.

수금 및 지급 대시보드는 운영 및 재무 프로세스에 대한 정보를 제공합니다. 대시보드는 매일 새로 고침됩니다. 자세한 내용은 다음 항목을 참조하세요.

대시보드를 열려면에 로그인하고 인사이트AWS Marketplace Management Portal, 재무 작업을 선택한 다음 컬렉션 및 지출 탭을 선택합니다.

주제

- 수금 및 지급 대시보드의 새로 고침 빈도
- 섹션 1: 컨트롤
- 섹션 2: 필터
- 섹션 3: 주요 지표
- 섹션 4: 추세

- 섹션 5: 분석
- 섹션 6: 세분화된 데이터

수금 및 지급 대시보드의 새로 고침 빈도

수금 및 지급 대시보드는 북미 영업일에만 업데이트됩니다. 은행에 입금된 후 1일 이내에 지급된 인보이스를 볼 수 있습니다.

섹션 1: 컨트롤

대시보드의 이 섹션에서는 대시보드 데이터를 세분화하는 필터를 제공합니다. 예를 들어 AWS Marketplace 이벤트 알림에서 필드의 필터를 선택하여 특정 고객 계정 ID, 구독자 회사 이름 또는 제안 ID에 대한 지출을 확인할 수 있습니다. 지급 상태로 필터링하여 판매자에게 대금이 지급된 모든 인보이스 또는 대금이 지급되지 않은 미해결 인보이스를 모두 확인할 수도 있습니다. 시각적 객체에 포함할 날짜 범위와 같은 필터를 분석에 추가할 수 있습니다. 컨트롤 내에서 선택한 필터는 지표, 추세, 분류 및 세분화된 데이터 섹션에 표시되는 데이터를 업데이트합니다.

필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. QuickSight

컨트롤 설명

컨트롤 이름	설명
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 회사 이름	제품을 구독하는 계정 이름입니다.
제품 제목	제품의 제목입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제안 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.

컨트롤 이름	설명
AWS 레코드 판매자	거래를 이끈 기업체의 식별자입니다. 가능한 값은 다음과 같습니다. • AWS_INC: Inc AWS.의 식별자(미국 기반) • AWS_EUROPE: AWS EMEA SARL의 식별자(룩셈부르크 기준) • AWS_AUSTRALIA: AWS Australia Pty Ltd(AWS 호주)의 식별자 • AWS_JAPAN: AWS 일본 G.K의 식별자입니다. • AWS_KOREA: AWS 한국의 식별자
지급 상태	AWS 가 은행 계좌에 자금, 부분 또는 전체 금액을 지급하지 않았음을 확인하기 위해 인보이스와 연결된 상태입니다. 가능한 상태는 지급됨, 부분적으로 지급됨, 실패, 지급되지 않음입니다.
지급인 AWS 계정 ID	요금이 청구되는 계정의 ID입니다.
지급인 회사 이름	요금이 청구되는 계정의 상호입니다.
리셀러 회사	소프트웨어 제조업체의 제품 판매 허가를 받은 리셀러 계정의 상호입니다.
리셀러 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
재판매 권한 부여 ID	등록된 기회에 대한 고유 식별자
재판매 권한 부여 이름	등록된 기회의 고유한 이름입니다.
구독자 계정	제품을 구독한 계정과 연결된 2자 국가 코드입니다.
구독자 시/도 또는 리전	제품을 구독한 계정과 연결된 청구지 주소 시/도 또는 리전입니다.

컨트롤 이름	설명
거래 참조 ID	AWS Marketplace 레거시 보고서 간에 트랜잭션을 상호 연관시키는 데 도움이 되는 트랜잭션의 고유 식별자입니다.
지급 은행 추적 ID	지급의 경우 은행이 추적 ID를 할당합니다. 은행 추적 ID를 사용하여 판매자 은행이 제공하는 입금 알림 및 보고서를 AWS Marketplace 보고서의 인보이스와 연결할 수 있습니다.

섹션 2: 필터

대시보드의 이 섹션에서는 제안 통화 및 두 가지 날짜 차원을 기준으로 그리고 날짜 필드 값이 지정된 날짜 이전인지 이후인지 또는 날짜 범위 내에 있는지에 따라 레코드를 세분화하는 필터를 제공합니다. 날짜 차원은 결제 만기일 또는 마지막 지급일입니다. 지급 날짜는 지정된 날짜 범위 내에서 지급된 인보이스로 결과를 제한합니다. 지급 기한에는 지급 날짜와 관계없이 기한이 지정된 범위 내에 있는 인보이스가 포함됩니다. 날짜 범주 필터는 지표, 추세, 분류 및 세분화된 데이터 섹션에 표시되는 데이터를 업데이트합니다. 기본 날짜 범주는 마지막 지급일이며 지난 6개월의 데이터를 가져옵니다.

섹션 3: 주요 지표

대시보드의 이 섹션에는 대금이 지급된 수익 수치와 지급되지 않은 수익 수치를 비교하여 시각화하는 핵심 성과 지표(KPI)가 표시됩니다. 지정된 필터 기준의 총 수익, 순수익, 도매 비용(해당하는 경우), 지급된 금액, 지급되지 않은 금액에 대한 KPI가 표시됩니다.

섹션 4: 추세

대시보드의 이 섹션에서는 지정된 날짜 범위의 대금 지급 및 연체 추세를 볼 수 있습니다. 일, 월, 분기 또는 연도와 같이 지정된 날짜 집계별로 추세를 보고 AWS Marketplace 컬렉션 상태를 파악할 수 있습니다. 추세 보기에는 다음과 같은 항목이 있습니다.

- 지급 추세 - 평균 지급 일수 및 관련 순수익의 개요를 제공합니다. 이 추세는 인보이스 날짜와 지급일 사이의 일수를 측정하여 수금 효율성을 보고합니다. 날짜 집계 필터에서 날짜 범위를 선택할 수 있습니다.
- 지급 기간 - 순수익과 대금이 지급된 인보이스 수를 표준 기한 초과 미수금 버킷(예: 기한 전, 1~30일, 31~60일)별로 분류하여 간략하게 보여줍니다. 이 추세는 결제 만기일과 지급일 사이의 일수를 측정하여 고객의 지급 조건 안에 대금이 지급되었는지 보고합니다.

- 미지급 기간 - 순수익과 미결제 인보이스 수를 기한 초과 버킷(예: 기한 전, 1~30일, 31~60일)별로 정리하여 간략하게 보여줍니다. 아직 기한이 남아 있는 대금이 미수금에 포함될 수 있습니다. 이 추세는 오늘 날짜와 지급 만기일 사이의 일수를 측정하여 들어오는 미수금을 표시합니다.

섹션 5: 분석

대시보드의 이 섹션에서는 제안 ID, 제품명, 지급인 회사 이름, 구독자 회사 이름, 리셀러 이름(채널 파트너 비공개 제안에 참여하는 경우), 지급인 지역 및 구독자 지역별로 미수금을 볼 수 있습니다. 분석 섹션을 사용하여 각 범주의 지급된 미수금을 지급되지 않은 미수금과 비교하여 측정할 수 있습니다.

섹션 6: 세분화된 데이터

대시보드의 이 섹션에는 모든 지급금 및 미수금이 제품, 고객 및 제안 세부 정보별로 표시됩니다.

> **Note**
>
> 2021년 4월 1일 이전에 생성된 인보이스에는 연결된 계약 ID, 제안 ID, 구독자 AWS 계정 ID 또는 구독자 회사 이름이 없을 수 있습니다.

QuickSight 테이블에서 데이터를 내보내고 다운로드하는 방법에 대한 자세한 내용은 QuickSight 사용 설명서의 시각적 객체에서 데이터 내보내기를 참조하세요.

세분화된 데이터 설명

열	설명
인보이스 날짜	고객에게 제품 구독 요금이 청구된 날짜.
결제 기한	YYYY-MM-DD 날짜 형식으로 된 지불 기한.
결제 조건	고객의 AWS 인보이스 결제 조건입니다.
인보이스 ID	요금이 청구된 인 AWS 보이스에 할당된 AWS ID입니다.
리스팅 요금 인보이스 ID	AWS Marketplace 구독이 AWS EMEA SARL, 일본 또는 호주 법인(등록 판매자)을 통해 거래되는 경우 판매를 위한 마켓플레이스 운영자(예: AWS EMEA SARL)는 판매자에게 판매자 등록

열	설명
	수수료에 대한 VAT를 청구해야 합니다. 해당하는 거래의 경우 리스팅 수수료에 부과된 VAT의 인보이스 ID는 소프트웨어 또는 제품 구독 인보이스 ID와 다릅니다.
도매 인보이스 ID	채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매를 나타내는 미지급 인보이스에 할당된 AWS ID입니다. 공개 제안 및 Marketplace 비공개 제안(MPOs 경우이 필드는 '해당 없음'이 됩니다.
구독자 회사 이름	제품을 구독하는 계정 이름입니다.
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 이메일 도메인	제품을 구독한 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 liu-jie@example.com이면 항목은 example.com입니다.
구독자 구/군/시	제품을 구독한 계정과 연결된 청구지 주소 구/군/시입니다.
구독자 시/도 또는 리전	제품을 구독한 계정과 연결된 청구지 주소 시/도입니다.
구독자 계정	제품을 구독한 계정과 연결된 청구지 주소 국가입니다.
구독자 우편 번호	제품을 구독한 계정과 연결된 청구지 우편 번호입니다.
제품 제목	제품의 제목입니다.
제안 이름	판매자가 정의한 제안 이름입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.

열	설명
제안 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.
계약 시작 날짜	고객의 제품 구독이 시작되는 날짜이며 MM-DD-YYYY 형식입니다. 미래 날짜의 계약인 경우 이 날짜가 수락 날짜와 다를 수 있습니다.
계약 종료 날짜	계약이 만료되는 날짜이며 MM-DD-YYYY 형식입니다. 종량제/사용한 만큼만 지불 구독의 경우 이 날짜는 9999년 1월 1일로 설정됩니다.
계약 수락 날짜	고객이 제품을 구독한 날짜이며 MM-DD-YYYY 형식입니다.
사용 기간 종료 날짜	제품 사용 기간의 종료 날짜입니다.
사용 기간 시작 날짜	제품 사용 기간의 시작 날짜입니다.
지급 상태	AWS 가 이전 지급 이후를 수금하여 은행 계좌로 지급했는지 확인하는 인보이스와 관련된 상태입니다. 지급됨 - 관련 인보이스의 금액이 징수되어 지급되었습니다. 지급되지 않음 - 관련 인보이스의 금액이 징수되어 지급되지 않았습니다.
지급 날짜	판매자의 은행에 대한 지급이 AWS 시작된 날짜입니다.

열	설명
지급 은행 추적 ID	지급의 경우 은행이 추적 ID를 할당합니다. 은행 추적 ID를 사용하여 판매자 은행에서 제공한 입금 알림 및 보고서를 AWS Marketplace 보고서의 인보이스와 연관시킬 수 있습니다.
총 수익	사용량 또는 제품의 월간 요금으로 고객에게 청구되는 금액입니다.
총 환급금	데이터 적용 기간 중에 환급이 처리된 경우 고객에게 환급되는 구독 비용의 총 금액입니다.
등록 수수료	청구 금액에서 공제되는 AWS Marketplace 수수료 금액입니다.
등록 수수료 환불	데이터 적용 기간 동안 환급이 처리된 경우 환급되는 AWS Marketplace 수수료 부분입니다.
리스팅 수수료 비율	청구 금액에서 공제할 AWS Marketplace 수수료 비율입니다.
판매자 세금	이 거래에 청구되는 미국 매출세 및 이용세의 총액입니다.
판매자 세금 환급	이 거래에서 환급이 처리된 경우 환급된 미국 매출세 및 이용세의 총액입니다.
AWS 세금 분담 리스팅 요금	여기서 AWS 마켓플레이스 운영자는 판매자의 판매에 대해 자신의 이름으로 VAT를 청구하고 징수해야 합니다. 이 금액은 AWS Marketplace 판매자 계정에 청구되는 VAT 금액에 해당합니다.
AWS 세금 공유 환급 등록 수수료	마켓플레이스 운영자 AWS 는 판매자의 매출에 대해 자신의 이름으로 VAT를 환급해야 합니다. 이 금액은 AWS Marketplace 판매자 계정에 환급된 VAT 금액에 해당합니다.

열	설명
도매가	채널 파트너 비공개 제안에만 해당됩니다. 리셀러에게 지급되는 제품의 비용입니다. 리셀러가 제조업체의 제품을 판매할 때 제조업체에 지불하는 금액을 예로 들 수 있습니다. 도매가는 정가에 할인율을 곱한 가격입니다.
도매가 환불	채널 파트너 비공개 제안에만 해당됩니다. 리셀러가 환불한 제품 금액입니다.
도매 판매자 세금 분담	판매자가 세금을 지불해야 하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금입니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 판매자 세금 분담 환급	판매자가 책임을 지는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 환급. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 기타 판매자 세금 분담	판매자가 세금을 지불해야 하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금입니다. 이 필드는 거래에 관여한 다른 판매자가 책임을 지게 되면 채워집니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 기타 판매자 세금 분담 환급	판매자가 책임을 지는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 환급. 이 필드는 거래에 관여한 다른 판매자가 책임을 지게 되면 채워집니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.

열	설명
도매 AWS 세금 분담	가 세금을 부담하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 AWS 입니다. 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
도매 AWS 세금 분담 환급	가 세금을 부담하는 채널 파트너 비공개 제안(CPPO)에서 ISV와 채널 파트너 간의 판매에 대한 세금 환급 AWS . 공개 제안 및 Marketplace 비공개 제안(MPPOs 경우이 필드는 '해당 없음'이 됩니다.
판매자 순수익	거래에 대해 청구되는 총 금액으로, AWS Marketplace 수수료, 환급금, 미국 매출세 및 이용세를 공제한 금액입니다.
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
거래 참조 ID	AWS Marketplace 보고서 간에 트랜잭션을 상호 연관시키는 데 사용할 수 있는 트랜잭션을 나타내는 고유 식별자입니다.
AWS 레코드 판매자	거래를 이끈 기업체의 식별자입니다. 가능한 값은 다음과 같습니다. • AWS_INC: Inc AWS.의 식별자(미국 기반) • AWS_EUROPE: AWS EMEA SARL의 식별자(룩셈부르크 기준) • AWS Australia Pty Ltd(AWS 호주) • AWS 일본 G.K. • AWS_한국
재판매 권한 부여 ID	등록된 기회에 대한 고유 식별자
재판매 권한 부여 이름	등록된 기회의 고유한 이름입니다.

열	설명
재판매 권한 부여 설명	등록된 기회에 대해 ISV가 정의한 설명입니다.
리셀러 회사 이름	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 이름입니다.
리셀러 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
지급인 회사 이름	요금이 청구되는 계정의 이름입니다.
지급인 AWS 계정 ID	요금이 청구되는 계정의 ID입니다.
지급인 이메일 도메인	요금이 청구되는 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 liu-jie@example.com이면 항목은 example.com입니다.
지급인 구/군/시	요금이 청구되는 계정과 연결된 청구지 주소 구/군/시입니다.
지급인 시/도 또는 리전	요금이 청구되는 계정과 연결된 청구지 주소 시/도입니다.
지급인 국가	요금이 청구되는 계정과 연결된 2자 국가 코드입니다.
지급인 우편번호	요금이 청구되는 계정과 연결된 청구지 주소 우편 번호입니다.
ISV 계정 ID	제품 또는 서비스 소유자의 식별자입니다.
ISV 회사 이름	제품 또는 서비스 소유자의 상호입니다.
제품 ID	소프트웨어 제품의 익숙한 고유 식별자입니다.
지급 상태	AWS 가 은행 계좌에 자금, 부분 또는 전체 금액을 지급하지 않았음을 확인하기 위해 인보이스와 연결된 상태입니다. 가능한 상태는 지급됨, 부분적으로 지급됨, 실패, 지급되지 않음입니다.

열	설명
지급된 순수익	판매자에게 지급된 총 거래 금액. 금액이 "판매자 순수익"과 같지 않으면 부분 결제입니다.
지급되지 않은 순수익	판매자에게 지급되지 않은 거래의 총 금액입니다. 금액이 0이 아닌 경우 고객이 남은 잔액을 지불해야 합니다.
지급 기간	수금이 이루어진 미수금 기간(예: 기한 전, 1~30일, 31~60일)을 설명하는 범주입니다.

과세 대시보드

과세 대시보드는 AWS Marketplace의 트랜잭션에 대한 미국 및 국제 세금에 대한 시각화 및 세분화된 데이터를 제공합니다.

이 대시보드를 열려면에 로그인하고 인사이트AWS Marketplace Management Portal, 마케팅을 선택한 다음 성능 나열 탭을 선택합니다.

주제

- 과세 대시보드의 새로고침 빈도
- 섹션 1: 컨트롤
- 섹션 2: 날짜 필터링
- 섹션 3: 지표
- 섹션 4: 세금 추세
- 섹션 5: 분석
- 섹션 6: 세분화된 데이터
- 미국 공제 코드
- EU 공제 코드

과세 대시보드의 새로고침 빈도

대시보드는 매일 업데이트됩니다. 업스트림 시스템에서 수신한 최신 트랜잭션 데이터가 지연되는 경우 최신 데이터가 대시보드에 반영되는 데 1~2일의 지연이 발생할 수 있습니다.

섹션 1: 컨트롤

대시보드의 이 섹션에서는 과세 데이터를 세분화하는 필터를 제공합니다. 시각적 객체에 포함할 날짜 범위와 같은 필터를 분석에 추가할 수도 있습니다. 컨트롤 내에서 선택한 필터는 지표, 추세, 분류 및 세분화된 데이터 섹션에 표시되는 데이터를 업데이트합니다.

필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. QuickSight

컨트롤 설명

컨트롤 이름	설명
인보이스 ID	고객에게 청구되는 요금에 할당된 AWS ID입니다.
과세된 고객 AWS 계정 ID	제품 구독에 대해 과세된 계정의 AWS ID입니다.
AWS 레코드 판매자	거래를 이끈 기업체의 식별자입니다.
과세된 고객 국가	과세된 고객과 연결된 두 자리 국가 코드
과세된 고객 상태 또는 리전	과세된 고객과 연결된 청구 주소 상태 또는 리전입니다.
과세된 고객 도시	과세된 고객과 연결된 청구 주소 도시
과세 대상	과세 대상, 비과세 대상 및 세금 환급 대상의 소스 트랜잭션입니다.
세금 책임 당사자	`AWS` 또는 `Seller`입니다. 판매자가 세금 책임 당사자인 경우 세금이 징수됩니다. AWS 가 세금 납부 당사자인 경우에서 판매세를 징수하고 송금합니다 AWS. 자세한 내용은 AWS Marketplace Tax Help for Sellers를 참조하세요. 세금이 징수되지 않으면 여기에 값이 표시되지 않습니다. 판매자는 세금 징수에 대한 책임이 있

컨트롤 이름	설명
	으므로 각 인보이스에 대해 일부 세금이 징수되었는지를 확인해야 합니다.
세금 유형	거래에 적용되는 세금 유형입니다. 가능한 값은 `None`, `Sales`, `SellerUse` , `Tax registration type` , `VAT`(부가가치세), `CNPJ`, `IGST`, `CT`입니다.

섹션 2: 날짜 필터링

대시보드의 이 섹션에서는 지정된 날짜 범위에 대한 인보이스 날짜를 기준으로 레코드를 구체화하는 필터를 제공합니다. 인보이스 날짜 내에서 선택한 필터는 지표, 추세, 분류 및 세분화된 데이터 섹션에 표시되는 데이터를 업데이트합니다. 예를 들어 날짜 필터 `Past 30 days`를 선택하면 모든 섹션에 지난 30일 이내에 과세된 인보이스와 관련된 데이터가 반영됩니다.

섹션 3: 지표

대시보드의 이 섹션에는 과세 금액, 비과세 금액, 세금 금액, 과세된 송장 수, 총 거래 수를 포함하여 세금과 관련된 지표를 시각화하는 핵심 성과 지표(KPI)가 표시됩니다. 필터 섹션에서 사용 날짜 기준을 업데이트하여 날짜 범위를 업데이트할 수 있습니다.

섹션 4: 세금 추세

대시보드의 이 섹션에서는 지정된 날짜 범위의 과세 추세를 제공합니다. 일별, 전월 대비, 전분기 대비, 전년 대비 등과 같이 지정된 날짜 집계별로 추세를 확인하여 과세에 대한 인사이트를 얻을 수 있습니다. 다음 과세 추세 정보를 사용할 수 있습니다.

- 세금 추세 - 금액 - 선택한 기간의 과세 금액과 세금 금액의 스냅샷을 인보이스 날짜별로 제공합니다.
- 세금 추세 - 인보이스 수 - 선택한 기간 동안 인보이스 날짜별로 인보이스 수의 스냅샷을 제공합니다.

섹션 5: 분석

대시보드의 이 섹션에서는 인보이스 ID, 세금 유형, 제품 제목 및 고객 지리적 위치별로 비즈니스에 대한 세금 지표를 제공합니다.

섹션 6: 세분화된 데이터

대시보드의 이 섹션에는 AWS Marketplace에서 조직의 트랜잭션에 대한 세분화된 세금 데이터가 표시됩니다.

세분화된 데이터 설명

열	설명
인보이스 ID	요금이 청구된 인 AWS 보이스에 할당된 AWS ID입니다. 구매자 상업용 인보이스, 판매자 도매 인보이스 또는 판매자 등록 수수료 인보이스일 수 있습니다.
리스팅 요금 인보이스 ID	판매자 등록 수수료 및 VAT가 판매자에게 청구된 인보이스에 할당된 AWS ID입니다. ISVs에만 적용됩니다.
항목 ID	각 항목의 고유 식별자입니다. 환급 거래의 항목 ID는 앞으로의 전달 세금과 동일합니다.
고객 청구서 ID	고객 청구서 ID
세금 책임 당사자	`AWS` 또는 `Seller`입니다. 판매자가 세금 책임 당사자인 경우 세금이 징수됩니다. AWS 가 세금 납부 당사자인 경우에서 판매세를 징수하고 송금합니다 AWS. 자세한 내용은 AWS Marketplace Tax Help for Sellers를 참조하세요. 세금이 징수되지 않으면 여기에 값이 표시되지 않습니다. 판매자는 세금 징수에 대한 책임이 있으므로 각 인보이스에 대해 일부 세금이 징수되었는지를 확인해야 합니다.
트랜잭션 유형 코드	거래의 유형 코드입니다. 값은 다음과 같습니다. • `AWS` - 세금 트랜잭션 전달. • `REFUND` - 전체 또는 일부 환급. • `TAXONLYREFUND` - 세금만 환급.

열	설명
	원래의 선도 거래를 포함한 환불 거래 지분 항목 ID입니다.
제품 ID	소프트웨어 제품의 익숙한 고유 식별자입니다.
제품 제목	구매한 제품의 이름입니다.
제품 세금 코드	제품의 과세 속성을 식별하는 표준 코드입니다. 제품을 생성하거나 수정할 때 이 속성을 선택합니다.
인보이스 날짜	고객에게 제품 구독 요금과 세금이 청구된 날짜.
과세된 고객 AWS 계정 ID	과세된 고객의 AWS 계정 ID입니다.
과세된 고객 국가	과세된 고객과 연결된 두 자리 국가 코드입니다.
과세된 고객 상태 또는 리전	과세된 고객과 연결된 청구 주소 상태 또는 리전입니다.
과세된 고객 도시	과세된 고객과 연결된 청구 주소 도시입니다.
과세된 고객 우편번호	세금 계산에 사용되는 우편 번호입니다.
세금 유형	거래에 적용되는 세금 유형입니다. 가능한 값은 `None`, `Sales`, `SellerUse` , `Tax registration type` , `VAT`, `CNPJ`, `IGST`, `CT`입니다.
관할 구역 수준	세금 납부 지역에 사용되는 주소의 관할 구역 수준입니다. 가능한 값은 `State`, `County`, `City` 및 `District`입니다.
조세 관할권	조세 관할 구역 이름입니다.
표시 가격 과세 유형	고객에게 표시되는 가격의 과세 유형입니다. 모든 AWS Marketplace 상품은 배타적입니다.
세금 관할권 요금	관할 구역 수준에서 적용되는 세율입니다.

열	설명
세금 금액	관할 구역 수준에서 부과되는 세금입니다.
통화	과세된 세금의 통화입니다. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
법인세 금액	관할권 수준에서 청구되는 법적 통화로 표시되는 세금 금액입니다.
법인세 통화	부과된 세금의 법적 통화로 표시되는 통화입니다.
세금 계산 이유 코드	관할 구역 수준에서 거래가 과세 대상인지, 비과세 대상인지, 공제인지, 부가가치세가 없는지를 나타냅니다.
세금 계산에 사용되는 날짜	거래의 세금 계산에 사용되는 날짜입니다.
고객 공제 인증서 ID	공제 인증서의 인증서 ID입니다.
고객 공제 인증서 ID 도메인	Amazon 시스템에 인증서가 저장되는 위치입니다.
고객 공제 인증서 수준	공제를 제공한 관할 구역 수준입니다.
고객 공제 코드	공제를 지정하는 코드입니다. 예를 들어 RESALE입니다.
거래 참조 ID	AWS Marketplace 보고서 간에 트랜잭션을 상호 연관시키는 데 도움이 되는 트랜잭션의 고유 식별자입니다.
AWS 레코드 판매자	거래를 이끈 기업체의 식별자입니다.
구매자 인보이스 ID	청구된 인 AWS 보이스에 할당된 AWS ID가 구매자에게 청구되었습니다. 판매자는 도매 인보이스 또는 등록 수수료 인보이스를 해당 구매자 요금에 연결할 수 있습니다.

> **Note**
>
> 총 세금 금액과 총 구매자 비용이 청구된 수익 대시보드의 AWS 세금 공유 및 총 수익 필드로 이동되었습니다.

미국 공제 코드

공제 코드	설명
501C	501C 조직
AP	농업 생산
CO	자선 조직
DIRECT	직접 지불 허가
DISTSBT	Distributor
DP	UN/외교관
DV	장애가 있는 재향 군인
EDI	교육 기관
FG	연방 정부
HCP	의료 서비스 제공자
HO	병원(비영리 또는 주립)
INSUR	보험
IPM	IPM
LB	라이브러리
MPU	다중 사용 지점 면제, 디지털 제품/소프트웨어 면제

공제 코드	설명
NA	북미 원주민
NAI	북미 원주민 개인
NP	비영리 조직
NPR	비영리 종교 단체
OT	기타
재판매	리셀러
SBE	소기업 공제
SLG	주/지방 정부

EU 공제 코드

공제 코드	설명
SpecialOrg	VAT 부과가 면제되는 조직입니다.
SpecialZone	면세 지역입니다.
SplitPayment	구매자는 인보이스에 명시된 VAT를 세무 당국에 직접 납부합니다.

영업 운영을 위한 대시보드

AWS Marketplace에서는 판매 데이터를 추적하는 데 도움이 되는 여러 대시보드를 제공합니다.

주제

- 계약 및 갱신 대시보드
- 사용 대시보드

계약 및 갱신 대시보드

이 대시보드는 계약 서명 후 24시간 이내에 계약 및 갱신에 대한 정보를 제공합니다 AWS Marketplace. 이 대시보드는 만료되는 구독을 추적하는 데 도움이 될 수 있습니다.

대시보드를 열려면를 시작하고 인사이트 탭을 AWS Marketplace Management Portal선택한 다음 영업 작업을 선택합니다.

주제

- 섹션 1: 컨트롤
- 섹션 2: 날짜 범주 선택
- 섹션 3: 지표
- 섹션 4: 구독 추세
- 섹션 5: 분석
- 섹션 6: 세분화된 데이터

섹션 1: 컨트롤

이 섹션에서는 계약 및 갱신 데이터를 구체화하는 필터를 제공합니다. 다음 필터 중에 선택할 수 있습니다.

> **Note**
>
> 필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. QuickSight

컨트롤 설명

컨트롤 이름	설명
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 회사 이름	제품을 구독하는 계정 이름입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.

컨트롤 이름	설명
제안 표시 여부	공개 제안인지, 비공개 제안인지 또는 기업 계약 제안인지 나타냅니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.
CPPO 플래그	채널 파트너 비공개 제안을 사용하여 계약을 체결했는지 여부를 나타내는 예/아니요 플래그입니다. 예이면 등록 판매자가 채널 파트너입니다. 아니오이면 등록 판매자가 소프트웨어 제조업체(독립 소프트웨어 공급업체)입니다.
ISV AWS 계정 ID	제품 또는 서비스 소유자의 계정입니다.
ISV 회사 이름	제품 또는 서비스 소유자의 상호입니다.
레거시 제품 ID	소프트웨어 제품의 레거시 고유 식별자입니다.
제품 제목	제품의 제목입니다.
재판매 권한 부여 ID	ISV가 채널 파트너에게 제공하는 재판매 권한 부여 ID입니다.
재판매 권한 부여 이름	ISV가 채널 파트너에게 제공하는 재판매 권한 부여 이름입니다.
리셀러 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
리셀러 회사 이름	소프트웨어 제조업체의 제품 판매 허가를 받은 리셀러 계정의 상호입니다.
구독자 계정	제품을 구독한 계정과 연결된 2자 국가 코드입니다.
구독자 시/도 또는 리전	제품을 구독한 계정과 연결된 청구지 주소 시/도 또는 리전입니다.

컨트롤 이름	설명
구독자 구/군/시	제품을 구독한 계정과 연결된 청구지 주소 구/군/시입니다.

섹션 2: 날짜 범주 선택

대시보드의 이 섹션에서는 지정된 날짜 범위의 계약 종료 날짜, 계약 종료 기간 또는 계약 시작 날짜를 기준으로 레코드를 세분화하는 필터를 제공합니다. 날짜 범주 선택에서 선택한 필터는 지표, 추세, 분류 및 세부 데이터 섹션에 표시되는 데이터를 업데이트합니다.

최근에 서명된 계약에 대한 정보를 보려면 계약 시작 날짜로 필터링하면 됩니다. 계약 갱신에 대해 알아보려면 계약 종료 날짜를 사용하여 필터링하면 됩니다.

섹션 3: 지표

대시보드의 이 섹션에는 주요 계약 데이터를 시각화하여 비교하는 핵심 성과 지표(KPI)가 표시됩니다. 활성 계약 수 및 종료된 계약 수에 대한 KPI가 표시됩니다. 날짜 범주 선택 필드에서 날짜 기준을 업데이트하여 날짜 범위를 업데이트할 수 있습니다.

섹션 4: 구독 추세

대시보드의 이 섹션에서는 지정된 날짜 범위의 구독 추세를 제공합니다. 일별, 전월 대비, 전분기 대비, 전년 대비 등과 같이 지정된 날짜 집계별로 추세를 확인하여 계약에 대한 인사이트를 얻을 수 있습니다. 계약 및 갱신 추세 정보는 집계 수준에서 제공됩니다.

- 구독 추세 - 선택한 날짜 집계 필터 및 선택한 날짜 범주 필터를 기반으로 활성 계약 수와 종료된 계약 수의 스냅샷을 제공합니다. 계약 시작 날짜에 대한 구독 추세를 보려면 날짜 범주 선택 필터에서 계약 시작 날짜를 선택하면 됩니다. 계약 종료 날짜의 구독 추세를 보려면 날짜 범주 선택 필터에서 계약 종료 날짜를 선택하면 됩니다.
- 갱신 내역 - 종료되었거나 며칠 후에 곧 종료되는 갱신의 스냅샷을 제공합니다. 1~30일 전에 종료된 계약, 31~60일 전에 종료된 계약 또는 종료된 지 60일이 넘은 계약 등 여러 기간의 갱신 내역을 볼 수 있습니다. 0~30일 후에 종료되는 계약, 31~60일 후에 종료되는 계약 등으로 필터링할 수도 있습니다.

섹션 5: 분석

대시보드의 이 섹션에서는 활성 구독자, ISV 및 리셀러에 대한 회사 전체의 비즈니스 지표를 제공합니다. 계약 ID, 종료된 계약 ID, 제안 ID, 갱신된 계약 ID 및 구독자 수를 볼 수 있습니다.

섹션 6: 세분화된 데이터

대시보드의 이 섹션에는 계약, 제안, 제품, 계약 제안자, 구독자, 재판매 권한 부여, 리셀러 및 ISV에 대한 세분화된 데이터가 표시됩니다.

> **Note**
>
> QuickSight 테이블에서 데이터를 내보내고 다운로드하는 방법에 대한 자세한 내용은 QuickSight 사용 설명서의 시각적 객체에서 데이터 내보내기를 참조하세요.

세분화된 데이터 설명

열	설명
구독자 회사 이름	구독자의 회사 이름입니다.
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 이메일 도메인	제품을 구독한 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 liu-jie@example.com이면 항목은 example.com입니다.
구독자 계정	제품을 구독한 계정과 연결된 청구지 주소 국가입니다.
구독자 시/도 또는 리전	제품을 구독한 계정과 연결된 청구지 주소 시/도입니다.
구독자 구/군/시	제품을 구독한 계정과 연결된 청구지 주소 구/군/시입니다.
구독자 우편 번호	제품을 구독한 계정과 연결된 청구지 우편 번호입니다.
제품 제목	제품의 제목입니다.
제품 ID	소프트웨어 제품의 익숙한 고유 식별자입니다.
레거시 제품 ID	소프트웨어 제품의 레거시 고유 식별자입니다.

열	설명
제안 이름	판매자가 정의한 제안 이름입니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제안 표시 여부	공개 제안인지, 비공개 제안인지 또는 기업 계약 제안인지 나타냅니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.
계약 시작 날짜	고객의 제품 구독이 시작되는 날짜이며 MM-DD-YYYY 형식입니다. 미래 날짜의 계약인 경우 이 날짜가 수락 날짜와 다를 수 있습니다.
계약 수락 날짜	고객이 제품을 구독한 날짜이며 MM-DD-YYYY 형식입니다.
계약 종료 날짜	계약이 만료되는 날짜이며 MM-DD-YYYY 형식입니다. 종량제/사용한 만큼만 지불 구독의 경우 이 날짜는 9999년 1월 1일로 설정됩니다.

열	설명
상태 표시기	계약의 현재 상태입니다. 지원되는 상태는 다음과 같습니다. • 활성 - 계약 약관의 일부 또는 전부가 유효합니다. • 종료됨 - AWS가 시작한 종료 이벤트로 인해 사전 합의된 종료 날짜 이전에 계약이 종료되었습니다. 가장 일반적인 경우는 결제 실패입니다. • 취소됨 - 수락자가 종료 날짜 이전에 계약을 종료하기로 선택합니다. • 만료됨 - 계약이 사전 합의된 종료일에 종료되었습니다. • 대체됨 - 대체 제안을 사용하여 계약이 대체되었습니다. • 갱신됨 - 자동 갱신과 같은 기능을 사용하여 계약이 새 계약으로 갱신되었습니다. • 아카이브됨 - 계약이 종료되었습니다. 종결, 종료, 취소 또는 만료됨의 정확한 이유가 지정되지 않았습니다. • 롤백 - 오류로 인해 계약 개정이 롤백되었습니다. 이제 이전 개정이 활성화됩니다. 비활성 계약 개정에만 적용됩니다. • 대체됨 - 계약 개정이 더 이상 활성화되지 않고 다른 계약 개정이 이제 활성화되었습니다. 비활성 계약 개정에만 적용됩니다. • 롤백 - 오류로 인해 계약 개정이 롤백되었습니다. 이제 이전 개정이 활성화됩니다. 비활성 계약 개정에만 적용됩니다.

열	설명
예상 계약 값	계약의 예상 비용으로, '총 계약 금액' 또는 '구매 금액'이라고도 합니다. 이는 SaaS, 전문 서비스 및 서버, 계약 또는 연간 요금 유형의 제품 유형에 적용됩니다.
통화 코드	계약의 예상 비용과 관련된 제안 요금 통화입니다.
리셀러 회사 이름	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 이름입니다.
리셀러 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
재판매 권한 부여 ID	등록된 기회에 대한 고유 식별자
재판매 권한 부여 이름	등록된 기회의 고유한 이름입니다.
재판매 권한 부여 설명	등록된 기회에 대해 ISV가 정의한 설명입니다.
CPPO 플래그	QuickSight의 계산된 필드입니다.

열	설명
계약 의도	현재 계약이 이전 agreement_id 열에 제공된 이전 계약의 갱신인지 아니면 업그레이드인지에 대한 정보를 채웁니다. 가능한 값은 다음과 같습니다. • 업그레이드: 비공개 제안인 경우 업그레이드는 모든 수정(업그레이드, 다운그레이드, 수정, 연장, 갱신)을 총칭하는 용어입니다. 공개 제안인 경우 업그레이드는 조건을 변경(다른 제안 사용)하여 새 계약을 체결하는 특정 계약이며, 특정 계약은 보관됩니다. • 갱신: 공개 제안에만 적용됩니다. 갱신 계약의 시작 날짜가 특정 계약의 종료 날짜이고 제안 ID는 변경되지 않았지만 계약 ID는 변경된 경우입니다. 계약 A가 계약 B로 변경된 경우를 예로 들 수 있습니다. • 자동 갱신: 공개 제안에만 적용됩니다. 현재 계약이 종료될 때 새로운 계약이 생성되는 경우입니다. 구매자가 자동 갱신을 활성화했습니다. • 신규: 구매자가 조건을 수락하고(제안을 사용하여) 새 계약을 구독했습니다.
이전 계약 ID	이 계약이 갱신 또는 업그레이드된 경우 계약 참조입니다.
다음 계약 ID	현재 계약이 갱신 또는 업그레이드된 경우 다음 계약의 계약 ID입니다.
이전 제안 ID	현재 계약이 갱신 또는 업그레이드된 경우 이전 계약과 연결된 제안 ID입니다.
다음 제안 ID	현재 계약이 갱신 또는 업그레이드된 경우 다음 계약과 연결된 제안 ID입니다.

열	설명
다음 계약 ID 수락 날짜	구매자가 다음 계약을 수락한 날짜입니다.
제안자 AWS 계정 ID	제품 또는 서비스 소유자의 식별자입니다.
제안자 회사 이름	제품 또는 서비스 소유자의 상호입니다.
ISV AWS 계정 ID	판매자의 AWS 식별자입니다.
ISV 회사 이름	판매자의 상호입니다.
계약 종료(일)	QuickSight의 계산된 필드입니다. 오늘부터 계약이 종료되는 날까지 남은 일수입니다.
계약 종료 기간	QuickSight의 계산된 필드입니다. 이 기간 내에 계약이 종료됩니다.

사용 대시보드

사용량 대시보드는 SaaS 및 서버 사용량 기반 제품을 사용하는 고객에게 시각화와 세분화된 데이터를 제공합니다. AWS Marketplace 판매자는이 대시보드를 사용하여 사용량 기반 제품의 고객 소비를 추적하여 제품 지원, 요금, 공개 제안에서 비공개 제안으로의 전환, 제품 중단에 대한 결정을 내릴 수 있습니다. 대시보드는 순차적인 기간인 지난 6개월 동안의 데이터를 제공합니다.

대시보드를 열려면에 로그인하고 인사이트AWS Marketplace Management Portal를 선택한 다음 판매 작업을 선택한 다음 사용 탭을 선택합니다.

> **Note**
>
> 이 대시보드에는 오퍼 생성 시 지정한 모든 차원 키의 사용량이 표시됩니다. 예를 들어 Amazon Machine Image(AMI) 제품의 경우 차원 키는 인스턴스 유형이며, 가격이 0 USD인 경우에도 제안에 지정된 모든 인스턴스 유형이이 대시보드에 표시됩니다. 오퍼 생성 시 지정되지 않은 차원의 제품 사용량을 확인하려면 필요한 차원을 포함하도록 제품을 다시 게시해 보세요.

주제

- 사용량 대시보드의 새로고침 빈도
- 섹션 1: 컨트롤

- 섹션 2: 필터
- 섹션 3: 지표
- 섹션 4: 추세
- 섹션 5: 분석
- 섹션 6: 세분화된 데이터

사용량 대시보드의 새로고침 빈도

대시보드는 매일 오후 4시 PST(자정 UTC)에 업데이트됩니다. 사용 데이터는 업스트림 데이터 소스에서 수신되므로 지연이 발생할 수 있습니다. 사용 날짜 및 사용 보고 날짜를 참조하여 사용 발생 시기와 대시보드에 보고된 시점을 비교하여 명확하게 확인할 수 있습니다.

섹션 1: 컨트롤

대시보드의 이 섹션에서는 사용량 데이터를 세분화하는 필터를 제공합니다. 예를 들어 다음 필터 중에서 선택할 수 있습니다.

> **Note**
> 필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. QuickSight

컨트롤 설명

컨트롤 이름	설명
최종 사용자 회사 이름	제품을 사용한 계정 이름입니다.
최종 사용자 AWS 계정 ID	제품을 사용한 계정 ID입니다.
최종 사용자 국가	제품을 사용한 계정과 연결된 2자 국가 코드입니다.
제품 제목	제품의 제목입니다.
제품 코드	제품의 측정에 사용되는 기존 권한 부여 제품 코드입니다. 이 값은 데이터를 보고서와 조인하거

컨트롤 이름	설명
	나 측정 서비스에 제공된 내용을 참조하는 AWS Marketplace 데에도 사용됩니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.
제안 표시 여부	공공, 민간 또는 기업 계약 제안인지 여부입니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.
차원 키	제품 사용과 관련된 리소스 유형입니다. 차원 키는 SaaS 및 서버 사용량 기반 제품에 적용됩니다.
구독자 회사 이름	제품을 구독하는 계정 이름입니다.
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 계정	제품을 구독한 계정과 연결된 2자 국가 코드입니다.
리셀러 회사 이름	제품 제조업체의 제품 판매 허가를 받은 리셀러 계정의 이름입니다.
리셀러 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
재판매 권한 부여 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
CPPO 플래그	채널 파트너 비공개 제안을 사용하여 계약을 체결했는지 여부를 나타내는 예/아니요 플래그입니다. 예이면 등록 판매자가 채널 파트너입니다. 그렇지 않다면 등록 판매자가 제품 제조업체(독립 소프트웨어 개발 판매 회사)입니다.

섹션 2: 필터

대시보드의 이 섹션에서는 사용 날짜를 기준으로 레코드를 세분화하는 필터를 제공합니다. 이러한 필터에서 선택한 값은 지표, 추세, 분류 및 세분화된 데이터 섹션에 표시되는 데이터를 업데이트합니다. 기본 선택은 지난 6개월 사용 데이터를 가져오는 것입니다.

섹션 3: 지표

대시보드의 이 섹션에는 예상 사용량 단위, 사용량이 있는 고객, 사용량이 있는 제품 등 소비와 관련된 지표를 시각화하는 핵심 성과 지표(KPI)가 표시됩니다. 필터 섹션에서 사용 날짜 기준을 업데이트하여 날짜 범위를 업데이트할 수 있습니다. 주요 지표에는 모든 단위 유형에 대한 데이터가 표시된다는 점에 유의하세요.

섹션 4: 추세

대시보드의 이 섹션에서는 지정된 날짜 범위의 사용 추세를 제공합니다. 일별, 전월 대비, 전분기 대비, 전년 대비 등과 같이 지정된 날짜 집계별로 추세를 확인하여 사용량에 대한 인사이트를 얻을 수 있습니다. 사용 단위 유형을 선택하여 사용 추세를 그래픽으로 볼 수도 있습니다.

섹션 5: 분석

대시보드의 이 섹션에서는 선택한 단위 유형에 대한 회사 이름, 제품 제목, 차원 키 및 오퍼 ID 전반의 비즈니스에 대한 예상 사용량 지표를 제공합니다. 보려는 항목 수를 선택할 수도 있습니다.

섹션 6: 세분화된 데이터

대시보드의 이 섹션에는 사용량, 제안, 제품, 구독자, 지불자, 최종 사용자, 재판매 승인, 리셀러, 독립 소프트웨어 개발 판매 회사(ISV)에 대한 세분화된 데이터가 표시됩니다. 세분화된 데이터 표에는 모든 장치 유형에 대한 데이터가 표시된다는 점에 유의하세요.

월말에 청구가 확정되기 전까지는 예상 수익으로 간주해야 합니다. 사용량 기반 청구서는 전월 사용량에 대해 다음 달 2일이나 3일에 구매자에게 제공됩니다. 예를 들어, 11/1에서 11/30 사이의 사용량으로 측정된 고객에게는 12/2 또는 12/3 사용량에 대한 청구서가 제공됩니다. 측정된 사용량은 실제 사용일로부터 며칠 후에 이 대시보드에 도착할 수 있으므로 사용 날짜와 사용량 보고 날짜가 다를 수 있습니다. 즉, 이번 달의 최신 추적을 위해 한 달 내내 매일 대시보드를 방문해야 할 수도 있습니다. 신뢰할 수 있는 고객 청구 정보는 재무 운영 탭의 청구 수익 대시보드를 참조하세요.

> **Note**
>
> 필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 시각적 객체에서 데이터 내보내기를 참조하세요.

세분화된 데이터 설명

열	설명
사용 날짜	고객의 제품 소비 날짜입니다.
사용 보고 날짜	고객의 제품 소비가 인사이트 대시보드에 표시된 날짜입니다.
최종 사용자 회사 이름	제품을 사용한 계정 이름입니다.
최종 사용자 AWS 계정 ID	제품을 사용한 계정 ID입니다.
최종 사용자 이메일 도메인	제품을 사용한 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 abc@example.com이면 항목은 example.com입니다.
최종 사용자 도시	제품을 사용한 계정과 연결된 도시입니다.
최종 사용자 시/도 또는 리전	제품을 사용한 계정과 관련된 시/도 또는 리전입니다.
최종 사용자 국가	제품을 사용한 계정과 연결된 2자 국가 코드입니다.
최종 사용자 우편번호	제품을 사용한 계정과 연결된 청구지 우편 번호입니다.
제품 제목	제품의 제목입니다.
레거시 제품 ID	제품의 레거시 고유 식별자입니다.
제품 ID	제품의 익숙한 고유 식별자입니다.
제품 코드	제품의 측정에 사용되는 기존 권한 부여 제품 코드입니다. 이 값은 데이터를 보고서와 조인하거나 AWS Marketplace Metering Service에 제공된 내용을 참조하는 데에도 사용됩니다.
제안 ID	구매자가 서명한 제안의 식별자입니다.

열	설명
제안 이름	판매자가 정의한 제안 이름입니다.
제안 표시 여부	공개 제안인지, 비공개 제안인지 또는 기업 계약 제안인지 나타냅니다.
계약 ID	제품 사용을 시작하기 위해 제안자와 수락자 간에 체결한 계약의 고유한 계약 데이터 피드 참조입니다.
계약 수락 날짜	고객이 제품을 구독한 시점의 날짜 타임스탬프(UTC)입니다.
계약 시작 날짜	고객의 제품 구독이 시작되는 날짜 타임스탬프(UTC)입니다. 미래 날짜의 계약인 경우 이 날짜가 수락 날짜와 다를 수 있습니다.
계약 종료 날짜	계약이 만료되는 날짜(UTC)입니다. 종량제/사용한 만큼만 지불 구독의 경우 이 날짜는 9999년 1월 1일 오전 12시로 설정됩니다.
차원 키	제품 사용과 관련된 리소스 유형입니다. 차원 키는 SaaS 및 서버 사용량 기반 제품에 적용됩니다.
리전	구매자가 Amazon EC2 인스턴스를 배포한 지역입니다.
예상 사용량	제품에 대해 기록된 사용량입니다.
사용 단위 유형	사용량이 기록되는 단위 유형입니다.
예상 수익	제품 사용으로 인한 수익입니다. 월말에 청구가 확정되기 전까지는 예상 수익으로 간주해야 합니다. 사용량 기반 송장은 전월 사용량에 대해 다음 달 2일 또는 3일에 구매자에게 제공됩니다.

열	설명
통화	거래 통화. 예를 들어 거래가 US 달러로 이루어지는 경우, 이 항목은 USD입니다.
구독자 회사 이름	제품을 구독하는 계정 이름입니다.
구독자 AWS 계정 ID	제품을 구독하는 계정 ID입니다.
구독자 이메일 도메인	제품을 구독한 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 abc@example.com이면 항목은 example.com입니다.
구독자 구/군/시	제품을 구독한 계정과 연결된 청구지 주소 구/군/시입니다.
구독자 시/도 또는 리전	제품을 구독한 계정과 연결된 청구지 주소 시/도입니다.
구독자 계정	제품을 구독한 계정과 연결된 청구지 주소 국가입니다.
구독자 우편 번호	제품을 구독한 계정과 연결된 청구지 우편 번호입니다.
지급인 회사 이름	제품을 결제한 계정의 이름입니다.
지급인 AWS 계정 ID	제품을 결제한 계정의 ID입니다.
지급인 이메일 도메인	제품을 결제한 계정과 연결된 이메일 도메인입니다. 예를 들어 이메일 주소가 abc@example.com이면 항목은 example.com입니다.
지급인 구/군/시	제품을 결제한 계정과 연결된 청구지 주소 구/군/시입니다.
지급인 시/도 또는 리전	제품을 결제한 계정과 연결된 청구지 주소 시/도입니다.

열	설명
지급인 국가	제품을 결제한 계정과 연결된 청구지 주소 국가입니다.
지급인 우편번호	제품을 결제한 계정과 연결된 청구지 우편 번호입니다.
리셀러 회사 이름	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 이름입니다.
리셀러 AWS 계정 ID	고객에게 재판매할 목적으로 ISV에게 도매로 제품 또는 서비스를 구매한 계정의 ID입니다.
재판매 권한 부여 ID	등록된 재판매 기회에 대한 고유 식별자입니다.
재판매 권한 부여 이름	등록된 재판매 기회의 고유한 이름입니다.
재판매 권한 부여 설명	등록된 재판매 기회에 대한 설명입니다.
CPPO 플래그	채널 파트너 비공개 제안을 사용하여 계약을 체결했는지 여부를 나타내는 예/아니요 플래그입니다. 예이면 등록 판매자가 채널 파트너입니다. 그렇지 않다면 등록된 판매자는 상품 제조업체(ISV)입니다.
ISV 회사 이름	제품 또는 서비스 소유자의 이름입니다.
ISV AWS 계정 ID	제품 또는 서비스 소유자의 식별자입니다.

마케팅용 대시보드

AWS Marketplace 는 마케팅 데이터를 추적하는 데 도움이 되는 여러 대시보드를 제공합니다.

주제

- 고객 계약 대시보드
- 리스팅 성과 대시보드
- 검색 성능 대시보드

- AWS 대시보드를 사용한 구매 이해

고객 계약 대시보드

고객 계약 대시보드는에서 제품을 구독하는 계약 및 고객에 대한 개요를 제공합니다 AWS Marketplace. 대시보드는 신규 및 활성 계약에 대한 데이터와 고객 추세 및 프로필을 제공합니다.

> **Note**
>
> - 이 대시보드를 잠금 해제하려면 AWS Marketplace Seller Prime 프로그램에 등록해야 합니다.
> - 이 대시보드를 열려면에 로그인하고 인사이트AWS Marketplace Management Portal, 마케팅을 선택한 다음 고객 계약 탭을 선택합니다.

AWS Marketplace 대시보드 사용에 대한 자세한 내용은이 섹션 판매자 대시보드앞부분의 단원을 참조하십시오.

주제

- 섹션 1: 필터
- 섹션 2: 날짜 필터 심층 분석
- 섹션 3: 공개 및 비공개 제안 계약
- 섹션 4: 고객 지표

섹션 1: 필터

다음 필터를 사용하여 계약 데이터를 구체화할 수 있습니다.

컨트롤 이름	설명
제품 제목	제품의 제목입니다.
날짜 필터	지난 30일, 60일, 90일, 이후 12개월(TTM), 올해 오늘까지(YTD)의 데이터가 포함됩니다. 사용자 지정을 선택하여 특정 시작 및 종료 날짜를 정의합니다.

컨트롤 이름	설명
고객 산업	판매자 제품을 구독한 고객을 위해 AWS 정의하는 산업입니다.
고객 세그먼트	판매자 제품을 구독한 고객에 대해 AWS 정의하는 세그먼트입니다.

필터링에 대한 자세한 내용은 Amazon QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. Amazon QuickSight

섹션 2: 날짜 필터 심층 분석

이 날짜 필터는 고객 계약 대시보드의 모든 지표에 적용됩니다.

예를 들어 연초부터 현재까지의 기본 필터 값을 사용하면 follpwing 지표가 나타납니다.

지표	설명
공개 제안 계약 수	올해 오늘까지 최소 1일 이상 활성 상태로 유지된 공개 제안이 있는 계약의 수입니다.
비공개 제안 계약 수	올해 오늘까지 최소 1일 이상 활성 상태로 유지된 비공개 제안이 있는 계약의 수입니다.
새 공개 제안 계약 수	올해 오늘까지 승인 날짜가 있는 공개 제안이 있는 계약의 수입니다.
새 비공개 제안 계약 수	올해 오늘까지 승인 날짜가 있는 비공개 제안이 있는 계약의 수입니다.
활성 고객	구독자 AWS 계정 ID로 식별되고 YTD 내에 최소 1개의 활성 계약이 있는 고유한 고객 수입니다. 활성 계약은 해당 기간 동안 최소 1일의 활성 계약이 있는 계약으로 정의됩니다.
신규 유료 고객	YTD 내에 첫 번째 결제 월이 있는 구독 AWS 자 계정 ID로 식별되는 고유 고객 수입니다.

섹션 3: 공개 및 비공개 제안 계약

대시보드의 이 섹션에는 계약의 개요가 표시됩니다. 핵심 성과 지표(KPI)에는 공개 제안 계약 수, 비공개 제안 계약 수, 신규 공개 제안 계약 수, 신규 비공개 제안 계약 수가 포함됩니다. 전년 대비 또는 전 기간 대비 볼륨 및 백분율의 변경 사항을 확인할 수 있습니다. 필터 섹션에서 사용 날짜 기준을 업데이트하여 날짜 범위를 업데이트할 수 있습니다.

계약은 제안자(제품 또는 서비스 소유자)와 수락자(고객) 간에 제품 사용을 시작하기 위해 서명된 계약입니다.

> **Note**
>
> 새 계약 지표에는 활성, 만료, 취소 및 종료된 계약이 포함됩니다.

다음 표에서는 계약 상태를 나열하고 설명합니다.

상태 표시기	설명
활성	계약 조건의 일부 또는 전부가 유효합니다.
만료됨	계약이 사전 합의된 종료일에 종료되었습니다.
취소됨	수락자가 종료 날짜 이전에 계약을 종료하기로 선택합니다.
종료됨	결제 실패와 같은 AWS 시작 종료 이벤트로 인해 사전 합의된 종료일 이전에 계약이 종료되었습니다.
갱신됨	자동 갱신과 같은 기능을 사용하여 계약이 새 계약으로 갱신되었습니다.
대체됨	대체 제안을 사용하여 계약이 대체되었습니다.

섹션 4: 고객 지표

대시보드의 이 섹션에서는 지정된 날짜 범위의 사용 추세를 제공합니다. KPI에는 활성 고객 수와 신규 결제 고객 수가 포함됩니다.

- 월별 추세 - 차트 내 필터를 기반으로 활성 고객 수와 신규 결제 고객 수에 대한 월별 보기를 제공합니다.
- 고객 지표 - 선택한 지표 필터를 기반으로 활성 고객 수와 신규 결제 고객 수에 대한 데이터를 제공합니다. 고객 세그먼트 또는 산업을 선택하여 각 세그먼트가 총 고객 지표에 어떻게 기여하는지 이해할 수 있습니다.

> **Note**
> 계약은 지정된 상태입니다. 상태에 대한 자세한 내용은 이전 섹션의 표를 참조하세요.

다음 표에서는 지표를 나열하고 설명합니다.

Metrics	설명
공개 제안 계약	공개 제안 가시성이 있는 계약 ID로 식별된 계약 수입니다. 계약 길이는 선택한 날짜 범위와 적어도 하루 이상 겹칩니다. 길이는 계약 시작일과 종료일 사이의 기간이 식별됩니다.
비공개 제안 계약	비공개 제안 가시성이 있는 계약 ID로 식별된 계약 수입니다. 계약 길이는 선택한 날짜 범위와 적어도 하루 이상 겹칩니다. 길이는 계약 시작일과 종료일 사이의 기간이 식별됩니다.
새로운 공개 제안 계약	공개 제안 가시성이 있는 계약 ID로 식별된 계약 수입니다. 계약 수락 날짜가 선택한 날짜 범위 내에 있습니다.
새로운 비공개 제안 계약	비공개 제안 가시성이 있는 계약 ID로 식별된 계약 수입니다. 계약 수락 날짜가 선택한 날짜 범위 내에 있습니다.
활성 고객	하나 이상의 활성 계약이 있는 구독자 AWS 계정 IDs로 식별된 고객 수입니다. 계약 기간은 선택한 날짜 범위와 1일 이상 겹칩니다.

Metrics	설명
신규 유료 고객	선택한 날짜 범위 내에 첫 번째 결제 월이 있는 구독자 AWS 계정 IDs로 식별된 고객 수입니다.

계약 및 수익에 대한 자세한 내용은 이 섹션의 계약 및 갱신 대시보드 및 청구 수익 대시보드 섹션을 참조하세요.

리스팅 성과 대시보드

리스팅 성과 대시보드는 AWS Marketplace 리스팅에 대한 개요와 세부 데이터를 제공합니다. 대시보드는 트래픽 볼륨과 고객이 제품을 구독하기 위해 취하는 단계에 대한 데이터를 제공합니다. 또한 대시보드는 마케팅 채널별 트래픽에 대한 세부 분석을 제공합니다.

> **Note**
>
> - 이 대시보드를 잠금 해제하려면 AWS Marketplace Seller Prime 프로그램에 등록해야 합니다.
> - 이 대시보드를 열려면에 로그인하고 인사이트AWS Marketplace Management Portal, 마케팅을 선택한 다음 성능 나열 탭을 선택합니다.

AWS Marketplace 대시보드 사용에 대한 자세한 내용은이 섹션 판매자 대시보드앞부분의 단원을 참조하십시오.

주제

- 섹션 1: 필터
- 섹션 2: 날짜 필터 심층 분석
- 섹션 3: 요약 지표
- 섹션 4: 월별 추세 및 퍼널 변환
- 섹션 5: 채널별 트래픽 추세
- 섹션 6: 월별 트래픽 및 계약 지표
- 섹션 7: 고유한 방문자를 위한 웹 트래픽 소스
- 섹션 8: 웹 트래픽 소스 분석

섹션 1: 필터

다음 필터를 사용하여 데이터를 세분화할 수 있습니다.

컨트롤 이름	설명
제품 제목	제품의 제목입니다.
날짜	지난 30일, 60일, 90일, 이후 12개월(TTM), 올해 오늘까지(YTD)가 포함됩니다. 사용자 지정을 선택하여 특정 시작 및 종료 날짜를 정의할 수 있습니다.

필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. QuickSight

섹션 2: 날짜 필터 심층 분석

이 날짜 필터는 리스팅 성과 대시보드의 모든 지표에 적용됩니다.

예를 들어, 올해 오늘까지의 기본 필터 값을 사용하면 다음 지표가 나타납니다.

지표	설명
고유 방문자	YTD의 AWS Marketplace 목록 페이지를 방문한 고유한 개인 수입니다.
새로운 공개 제안 계약	올해 오늘까지 최소 1일 이상 활성 상태로 유지된 공개 제안이 있는 계약의 수입니다.
새로운 고유 방문자	YTD에서 처음으로 AWS Marketplace 목록 페이지를 방문한 고유한 개인 수입니다.
다시 방문한 고유한 방문자	선택한 연도가 시작되기 전에 이전에를 방문하여 YTD에 다시 방문한 AWS Marketplace 목록 페이지를 방문한 고유한 개인 수입니다.

섹션 3: 요약 지표

대시보드의 이 섹션에는 트래픽 및 공개 제안 계약에 대한 요약 시각화가 표시됩니다. 핵심 성과 지표(KPI)에는 고유 방문자 수, 새 공개 제안 계약 수, 새 고유 방문자 수, 다시 방문한 고유 방문자 수가 포함됩니다. 전년 대비 또는 전 기간 대비 볼륨 및 백분율의 변경 사항을 확인할 수 있습니다. 필터 섹션의 날짜 필드에서 날짜 기준을 업데이트하여 날짜 범위를 업데이트할 수 있습니다.

Metrics	설명
고유 방문자	AWS Marketplace 목록 페이지를 방문한 고유한 개인 수입니다.
계약	제안자(제품 또는 서비스 소유자)와 수락자(고객) 간에 제품 사용을 시작하기 위해 서명된 계약입니다.
새로운 공개 제안 계약	선택한 날짜 범위 내에 승인 날짜가 지정된 공개 제안이 있는 계약의 수입니다. 상태가 활성, 만료, 취소 및 종료인 제안을 포함합니다.
상태	활성 - 계약 조건의 일부 또는 전부가 유효합니다. 만료됨 - 계약이 사전 합의된 종료일에 종료되었습니다. 취소됨 - 수락자가 종료 날짜 이전에 계약을 종료하기로 선택합니다. 종료됨 - 결제 실패와 같은 AWS 시작 종료 이벤트로 인해 사전 합의된 종료일 이전에 계약이 종료되었습니다. 갱신됨 - 자동 갱신과 같은 기능을 사용하여 계약이 새 계약으로 갱신되었습니다. 대체됨 - 대체 제안을 사용하여 계약이 대체되었습니다.

섹션 4: 월별 추세 및 퍼널 변환

대시보드의 이 섹션에서는 지정된 날짜 범위의 트래픽 및 계약 추세를 제공합니다. 핵심 성과 지표에는 신규 공개 제안 계약을 의미하는 신규 계약 수와 순 방문자 수가 포함됩니다. 월별 순 방문자 - 월별 순 방문자 수를 보여줍니다. 월별 순 방문자 수는 한 명의 방문자가 여러 달에 걸쳐 방문할 수 있기 때문에 요약 지표 섹션의 총 순 방문자 수보다 많을 수 있습니다. 월별 신규 계약 - 신규 공개 제안 계약 수에 대한 월별 보기를 제공합니다. 퍼널 전환 - 단계별 전환 보기를 제공합니다. 여기에는 리스팅 페이지, 조달 페이지, 구성 페이지 및 이행 페이지가 포함됩니다. 막대로 표시되는 고유한 방문자, 페이지 뷰 및 고유한 계정을 선택할 수 있습니다. 각 막대 위에 마우스를 올려 놓으면 현재 막대의 값으로 정의된 변환 백분율을 이전 막대의 백분율로 볼 수 있습니다.

Metrics	설명
고유 방문자	AWS Marketplace 목록 페이지를 방문한 고유한 개인 수입니다.
페이지 뷰	AWS Marketplace 목록 페이지 방문 횟수입니다.
고유 계정	방문자와 연결된 고유 AWS 계정 ID의 수입니다. 이 지표는 방문자가 인증한 후에만 적용됩니다.

섹션 5: 채널별 트래픽 추세

대시보드의 이 섹션에서는 월별 트래픽 추세에 대해 자세히 알아봅니다. 고유한 방문자 및 페이지 뷰를 막대로 표시할 수 있습니다. 채널 필터에서 단일 채널 또는 여러 채널을 선택하여 선택한 채널의 월별 추세를 볼 수 있습니다.

섹션 6: 월별 트래픽 및 계약 지표

대시보드의 이 섹션에서는 고유한 방문자, 새 계약, 전환율 및 총 계약을 월별 보기로 제공합니다. 전환율은 매월 고유한 방문자로 나눈 새 계약으로 정의됩니다.

일부 월의 날짜 범위를 선택하더라도 표의 데이터는 한 달 전체를 나타냅니다. 예를 들어 2024년 1월 15일부터 2024년 3월 15일까지의 기간을 표시하도록 데이터를 필터링하는 경우, 테이블에는 2024년 1월 1일부터 2024년 3월 31일까지의 전체 월별 데이터가 계속 포함됩니다.

테이블의 데이터는 월별로 정렬되며 가장 최근 월이 먼저 표시됩니다. 데이터를 오름차순 또는 내림차순으로 정렬하는 옵션이 있는 해당 열 헤더를 클릭하여 사용 가능한 열을 기준으로 테이블을 정렬할 수 있습니다.

섹션 7: 고유한 방문자를 위한 웹 트래픽 소스

대시보드의 이 섹션에서는 마케팅 소스별로 고유한 방문자에 대한 심층 분석을 제공합니다. 범주에는 판매자 유기적 검색, 판매자 주도 마케팅 캠페인, AWS 유기적 검색 및 AWS 주도 캠페인이 포함됩니다. 이러한 범주는 방문자 또는 방문과 연결된 추적 코드로 식별됩니다.

트래픽 소스	설명
판매자 오가닉 검색	웹 사이트 또는 앱 내 알림에서 작업 및 검색 지점에 대한 호출을 통해 목록에 도달하는 고유한 방문자입니다.
판매자 주도 마케팅 캠페인	유료 광고 또는 홍보 채널을 통해 리스팅에 도달하는 고유한 방문자입니다.
AWS 유기 검색	유기적 AWS 웹 사이트 트래픽, 콘솔(AWS 콘솔) 검색 및 AWS Marketplace 검색과 같은 AWS 채널을 통해 목록에 도착하는 고유한 방문자입니다.
AWS 마케팅 캠페인	AWS 주도 캠페인 및 유료 프로모션을 통해 리스팅에 도달하는 고유한 방문자입니다.

섹션 8: 웹 트래픽 소스 분석

웹 트래픽 소스 분석 표는 트래픽을 구동하는 마케팅 활동에 대한 세분화된 분석을 제공합니다.

- 트래픽 소스 AWS- 두 가지 주요 범주로 포함 및 판매자 주도.
- 트래픽 유형 - 판매자 유기적 검색, 판매자 주도 마케팅 캠페인, AWS 유기적 검색 및 주도 마케팅 캠페인 AWS이 포함됩니다.
- 채널 - 유료 디스플레이, 유료 검색, 유료 소셜 미디어, 이메일, 오가닉, 내부 사이트 트래픽 외부 링크 및 기타 다양한 마케팅 전술이 포함됩니다.
- 게시자 - Facebook, Google, LinkedIn, AWS Seller-led 등을 포함하여 캠페인이 게시되는 엔터티를 나타냅니다.
- 프로모션 - 전용 추적 코드가 생성된 특정 캠페인을 나타냅니다. 예시는 `psm_linked_post_free-trial-global-ver-a`가 될 수 있습니다.

Metrics	설명
기여한 계약	각 프로모션에 대한 방문으로 인한 총 계약입니다.
전환율	기여한 계약은 각 프로모션에 대해 고유한 방문자로 나뉩니다.

검색 성능 대시보드

검색 성능 대시보드는 AWS Marketplace 검색 경험에서 목록이 수행되는 방식에 AWS Marketplace 대한 지표를 제공합니다. 대시보드를 사용하여 목록의 검색 가능성과 성능을 개선할 수 있습니다. 대시보드는 노출, 클릭 및 클릭률 수를 추적합니다. 또한 사용자가 제공하는 상위 키워드와 고객이 사용하는 키워드를 추적하고 일반적인 고객 검색을 기반으로 키워드 제안을 제공합니다.

대시보드의 인사이트를 통해 제품 목록을 구체화하여 가시성을 개선하고, 키워드를 고객 기본 설정과 일치시키고, 비즈니스에 더 적합한 트래픽을 유도할 수 있습니다 AWS Marketplace.

대시보드를 열려면 AWS Marketplace 관리 포털을 시작하고 인사이트 탭으로 이동합니다.

대시보드에 대한 액세스 제어에 대한 자세한 내용은이 섹션 판매자 대시보드앞부분의 섹션을 참조하세요.

섹션 1: 대시보드 액세스

이 대시보드를 잠금 해제하려면 AWS Marketplace 판매자 프라임 프로그램의 멤버여야 합니다.

판매자 프라임 프로그램에 등록한 후 판매자 계정의 시스템 관리자는의 인사이트 탭을 사용하여 검색 성능 대시보드를 AWS Marketplace Management Portal 시작할 수 있습니다.

시스템 관리자는 AWS Identity and Access Management (IAM) 정책을 생성하여 판매자 회사의 다른 사용자에게 특정 대시보드에 대한 액세스를 제공할 수도 있습니다.

> **Note**
>
> 2023년 9월부터 레거시 IAM 권한으로 활성화된 판매자 대시보드에 대한 액세스를 더 이상 지원하지 않습니다. 다음 예제와 같이 새 Amazon 리소스 이름(ARN) 형식을 사용하여 IAM 권한을 업데이트합니다.

정책 생성에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 정책 생성을 참조하세요.

정책을 사용하여 액세스 제어

다음 정책 중 하나를 사용하여 나열 성능 대시보드에 대한 액세스를 제공합니다.

다음 예제에서는 현재 또는 미래 데이터 피드에 관계없이 대시보드 및 보고서를 포함한 모든 현재 및 미래 AWS Marketplace 리소스에 대한 액세스를 제공합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/*",
    ]
  }]
}
```

다음 예제에서는 ARN을 포함하여 검색 성능 대시보드에 대한 액세스를 제공합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:GetSellerDashboard"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::awsAccountID:AWSMarketplace/ReportingData/Marketing_V1/
Dashboard/SearchPerformance_V1"
  }]
}
```

섹션 2: 필터

대시보드의이 섹션에서는 다음 필터를 제공합니다. 이를 사용하여 데이터를 구체화합니다.

컨트롤 이름	설명
제품 제목	제품의 제목입니다. 이 필터는 노출, 클릭, 클릭률 및 검색 키워드 권장 사항에 영향을 줍니다. 제품 범주별 상위 검색 키워드 테이블에는 영향을 주지 않습니다.

컨트롤 이름	설명
날짜 필터	지난 30일, 60일, 90일, 후행 12개월(TTM) 및 현재까지의 연도(YTD)를 기준으로 필터링할 수 있습니다. 사용자 지정을 선택하여 특정 시작 및 종료 날짜를 사용할 수 있습니다. 이 필터는 노출, 클릭 및 클릭률에 영향을 줍니다. 검색 AWS Marketplace 키워드 권장 사항 또는 제품 범주 테이블별로 검색된 상위 키워드에는 영향을 미치지 않습니다.

필터링에 대한 자세한 내용은 Amazon QuickSight 사용 설명서의 Amazon QuickSight에서 데이터 필터링을 참조하세요. Amazon QuickSight

섹션 3:성능 AWS Marketplace 검색

대시보드의이 섹션에는 AWS Marketplace 목록의 검색 가능성 및 성능에 대한 핵심 성과 지표(KPIs)가 표시됩니다. 대시보드는 노출 수, 클릭 수 및 클릭률에 대한 KPIs를 제공합니다.

year-over-year 또는 period-over-period 볼륨 및 백분율의 변경 사항을 확인할 수 있습니다. 날짜 범위를 변경할 수도 있습니다.

지표	설명
노출	상위 20개 검색 결과에 목록이 표시되는 횟수입니다 AWS Marketplace.
클릭	사용자가 AWS Marketplace 검색 결과에서 목록을 클릭한 횟수입니다.
클릭률(CTR)	클릭 수와 노출 수 간의 비율입니다. 목록이 검색 결과에서 고객을 참여시키는 데 얼마나 효과적인지 측정한 값입니다.

섹션 4: 노출, 클릭 및 클릭률에 대한 월별 추세

대시보드의이 섹션에서는 지정된 날짜 범위에 대한 월별 추세를 제공합니다. 노출 수, 클릭 수 및 클릭률에 대한 KPIs를 제공합니다.

섹션 5:AWS Marketplace 키워드 권장 사항 검색

대시보드의이 섹션에서는 현재 키워드와 각 목록에 권장되는 키워드를 나열하는 테이블을 제공합니다.

열 이름	설명
제품 제목	제품의 제목입니다. 링크를 클릭하면 AWS Marketplace Management Portal의 제품 개요 페이지로 이동하며, 여기에서 목록을 관리하고 편집할 수 있습니다.
현재 키워드	AWS Marketplace Management Portal의 제품 정보 업데이트 섹션에 제공한 키워드입니다.
최고 성능 키워드	제품에 대한 상위 5개 고객 검색에 표시되는 키워드입니다.
AWS-권장 키워드	유사한 제품을 검색할 때 고객이 가장 많이 사용하는 키워드입니다.
예상 트래픽 이득 %	권장 키워드를 채택하면 트래픽이 증가할 수 있습니다. 날짜 필터의 영향을 받지 않습니다.

> **Note**
>
> 키워드는 중요하지만 다른 요인이 검색 결과에 영향을 미칠 수 있습니다.
>
> - 목록 최적화
> - 짧은 설명과 긴 설명 등 목록의 다른 부분을 최적화합니다.
> - 무료 평가판과 같은 기능을 제공하여 목록의 어필을 개선하는 것이 좋습니다.
> - 키워드를 목록 및 대상 고객과 일치
> - 가능한 한 자주 사용하는 키워드와 목록을 일치시킵니다.
> - 제품 및 기능과 관련된 키워드만 사용합니다. 그렇지 않으면 더 많은 수의 검색에 목록이 표시될 수 있지만 더 적은 수의 사용자가 선택할 수 있습니다.
> - 키워드 변형 이해
> - "git" 및 "gitops"와 같은 유사한 용어는 동일한 검색어로 취급될 수 있습니다.
> - 변형의 관련성을 평가하고 그에 따라 키워드를 조정합니다.
> - 키워드 업데이트

1. 에서 AWS Marketplace Management Portal제품 메뉴를 열고 제품 범주를 선택합니다.
2. 제품 링크를 선택하고 제품 정보 탭을 선택한 다음 제품 정보 제공 페이지로 이동합니다.
3. AWS Marketplace 검색 가능성 섹션으로 스크롤하여 AWS Marketplace 검색 결과에 대한 키워드를 찾은 다음 키워드를 조정합니다.

섹션 6: 제품 범주별로 검색된 상위 키워드

사용자와 유사한 제품의 각 제품 범주에서 상위 5개 고객 검색 키워드입니다. 이 대시보드의 다른 제품 및 날짜 필터는이 데이터에 영향을 주지 않습니다.

AWS 대시보드를 사용한 구매 이해

를 사용하여 구매 AWS 대시보드는 직접 call-to-action 버튼을 사용하여 구매 AWS하고 조달을 통해 구매 AWS 페이지를 방문하는 고객이 생성한 웹 트래픽, 참여 및 계약에 대한 개요를 제공합니다 AWS Marketplace.

대시보드 액세스에 대한 자세한 내용은이 가이드 판매자 대시보드앞부분의 섹션을 참조하세요.

주제

- 섹션 1: 필터
- 섹션 2: 요약 지표
- 섹션 3: 페이지 보기 및 전체 계약의 총 클릭 및 추세 데이터
- 섹션 4:AWS 로그인 페이지 드롭오프 비율
- 섹션 5: 조달을 통한 AWS 구매 페이지의 참여
- 섹션 6: AWS 조달을 통한 구매 페이지의 참여 지표에 대한 추세 데이터

섹션 1: 필터

이 섹션에서는 다음과 같은 데이터 필터를 제공합니다.

필터 이름	설명
제품 제목	제품의 제목입니다.
제품 ID	소프트웨어 제품의 친숙한 고유 식별자입니다.

필터 이름	설명
날짜 필터	지난 30일, 60일 또는 90일, 이후 12개월 또는 현재까지의 연도를 기준으로 필터링합니다. 다른 시작 및 종료 날짜를 입력하려면 사용자 지정을 선택합니다.

필터링에 대한 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 데이터 필터링을 참조하세요. QuickSight

섹션 2: 요약 지표

이 섹션에는 call-to-action을 통한 구매 AWS 버튼과 조달을 통한 구매 AWS 페이지에 대한 정보가 표시됩니다.

다음 표에서는 지표를 나열하고 설명합니다. 날짜 범위를 업데이트하려면 필터 섹션에서 날짜 기준을 변경합니다.

지표	설명
총 버튼 클릭 수	판매자 웹 사이트에서 AWS로 구매와 같은 call-to-action 버튼을 클릭한 총 횟수입니다.
AWS 페이지 뷰를 사용한 구매 수	조달을 통한 구매 AWS 페이지의 총 방문 횟수입니다. 여기에는 반복 방문도 포함됩니다.
를 사용한 구매 AWS 페이지의 총 계약 수	조달을 통한 구매 AWS 페이지에서 생성된 총 계약 수입니다.

섹션 3: 페이지 보기 및 전체 계약의 총 클릭 및 추세 데이터

이 섹션에서는 지정된 날짜 범위에 대한 월별 또는 주간 추세를 제공합니다. 다음 표에서는 KPIs를 나열하고 설명합니다.

지표	설명
총 버튼 클릭 수	판매자 웹 사이트의 call-to-action 버튼을 클릭한 총 횟수입니다.

지표	설명
페이지 뷰	AWS 구매로 구매 페이지를 방문하는 총 페이지 수입니다. 데이터 표시 기준 목록을 사용하여 월별 보기와 주간 보기 중에서 선택합니다.
총 계약 수	AWS 구매 구매 페이지에서 생성된 총 계약 수입니다. 데이터 표시 기준 목록을 사용하여 월별 보기와 주간 보기 중에서 선택합니다.

섹션 4:AWS 로그인 페이지 드롭오프 비율

이 섹션에는 AWS 로그인 페이지의 인계율이 표시됩니다.이 비율은 로그인하고 계약을 생성한 사용자 대 로그인만 한 사용자 수의 비율입니다. 데이터 표시 기준 목록을 사용하여 주 또는 월을 기준으로 필터링합니다.

섹션 5: 조달을 통한 AWS 구매 페이지의 참여

이 섹션에서는 지정된 날짜 범위에 대한 요약 지표와 추세 데이터를 제공합니다. 다음 표에서는 지표를 나열하고 설명합니다. 필터 섹션의 날짜 필드를 사용하여 날짜 범위를 업데이트합니다.

지표	설명
총 고유 방문자 수	총 고유 조달 페이지 사용자 수입니다.
반송률	한 사이트 페이지를 본 후 로그인하고 나가는 사용자와 두 개 이상의 페이지를 방문하는 사용자의 비율입니다.
분 단위의 평균 사용자 체류 시간	사용자가 로그인한 시점과 해당 사용자가 페이지를 떠난 시점 사이의 시간입니다.

섹션 6: AWS 조달을 통한 구매 페이지의 참여 지표에 대한 추세 데이터

차트에는 섹션 5의 지표에 대한 월별 또는 주간 추세가 표시됩니다. 데이터 표시 기준 목록을 사용하여 월별 보기와 주간 보기 중에서 선택합니다.

AWS Marketplace 공급업체 인사이트

AWS Marketplace Vendor Insights는 조직이 신뢰하고 표준을 충족하는 조달 소프트웨어를 보호하기 위해 조직에서 수행하는 소프트웨어 위험 평가를 간소화하는 기능입니다. AWS Marketplace Vendor Insights를 사용하면 구매자는 단일 콘솔에서 거의 실시간으로 제품의 보안 프로파일을 모니터링할 수 있습니다. AWS Marketplace Vendor Insights는 구매자의 조달 프로세스를 간소화하고 잠재적으로 판매자의 판매를 늘릴 수 있습니다. 또한 소프트웨어 제품의 보안 및 규정 준수 정보에 대한 대시보드를 제공하여 구매자의 평가 부담을 줄여줍니다.

AWS Marketplace Vendor Insights 대시보드의 모든 보안 및 규정 준수 정보는 다음 소스에서 수집한 증거를 기반으로 합니다.

- AWS Marketplace Vendor Insights 보안 자체 평가 및 합의 평가 이니셔티브 설문(CAIQ)을 포함한 판매자의 자체 증명
- 산업 표준 감사 보고서(예: 국제표준화기구 ISO 27001)
- 판매자의 프로덕션 환경에서 증거를 수집하는 작업을 자동화하는AWS Audit Manager

AWS Marketplace Vendor Insights는 제품에 대한 규정 준수 아티팩트 및 보안 제어 정보를 수집하여 대시보드에 제공합니다. 이 대시보드는 판매자의 자체 평가 데이터, 감사 보고서의 증거, AWS 계정의 실시간 증거를 가져옵니다. 이 데이터는 보안 컨트롤에 전달된 후, 구매자가 검토할 수 있도록 대시보드에 전달됩니다. 실시간 증거는 여러 소스의 데이터를 일관적으로 업데이트하여 최신 정보를 제공하는 방법입니다. AWS Config 는 판매자의 환경에서 활성화됩니다. 구성, 활성화된 백업 및 기타 정보에 대한 데이터는 자동으로 업데이트됩니다. 예를 들어 제품의 액세스 제어가 규정을 준수하고 Amazon Simple Storage Service(S3) 버킷이 퍼블릭으로 된다고 가정하겠습니다. 대시보드에는 제어 상태가 규정 준수에서 미확인으로 변경되었다고 표시됩니다.

AWS Marketplace Vendor Insights를 사용하기 AWS 계정 전에에서 기준 리소스 및 인프라를 설정해야 합니다. 설정이 완료되면 AWS Marketplace , Vendor Insights는의 서비스형 소프트웨어(SaaS) 제품에 대한 정보를 수집하고 보안 프로필을 생성할 수 있습니다 AWS Marketplace.

내용

- AWS Marketplace Vendor Insights 이해
- AWS Marketplace Vendor Insights 설정
- AWS Marketplace Vendor Insights 프로필 보기
- AWS Marketplace Vendor Insights에서 스냅샷 관리

- AWS Marketplace Vendor Insights의 액세스 권한 제어

AWS Marketplace Vendor Insights 이해

AWS Marketplace Vendor Insights는 제품에 대한 규정 준수 아티팩트 및 보안 제어 정보를 수집하여 대시보드에 제공합니다. 이 대시보드는 제품 소유자의 자체 평가 데이터, 감사 보고서의 증거, AWS 계정의 실시간 증거를 가져옵니다. 이 데이터는 보안 컨트롤에 전달된 후, 구매자가 검토할 수 있도록 대시보드에 전달됩니다.

대시보드는 여러 보안 제어 범주에서 AWS Marketplace Vendor Insights가 수집한 증거 기반 정보를 제공합니다. 따라서 보안 프로필을 거의 실시간으로 볼 수 있으므로 인사이트를 얻을 수 있으며 구매자와 판매자 간의 논쟁을 줄일 수 있습니다. 구매자는 몇 시간 내에 평가를 완료하는 판매자의 정보를 검증할 수 있습니다. AWS Marketplace Vendor Insights는 판매자가 보안 및 규정 준수 태세 정보를 자동으로 up-to-date 상태로 유지할 수 있는 메커니즘을 제공합니다. 판매자는 이러한 상태 정보를 구매자와 온디맨드로 공유할 수 있으므로 무작위로 설문지에 응답할 필요가 없습니다.

AWS Marketplace Vendor Insights는 다음 세 가지 소스에서 증거 기반 정보를 수집합니다.

- 공급업체 자체 평가 - 지원되는 자체 평가에는 AWS Marketplace Vendor Insights 보안 자체 평가 및 합의 평가 이니셔티브 설문지(CAIQ)가 포함됩니다.
- 프로덕션 계정 - 여러 컨트롤 중에서 25개 컨트롤은 판매자의 프로덕션 계정에서 실시간으로 증거를 수집할 수 있도록 지원합니다. 각 컨트롤에 대한 실시간 증거는 하나 이상의 AWS Config 규칙을 사용하여 AWS 리소스의 구성 설정을 평가하여 생성됩니다.는 증거를 AWS Audit Manager 캡처하고 AWS Marketplace Vendor Insights가 사용할 수 있도록 준비합니다. 온보딩 AWS CloudFormation 템플릿은 실시간 증거 수집을 활성화하는 데 필요한 사전 조건 단계를 자동화합니다. AWS Config는 판매자의 환경에서 활성화됩니다. 구성, 활성화된 백업 및 기타 정보에 대한 데이터는 자동으로 업데이트됩니다. 예를 들어 제품의 액세스 제어가 규정을 준수하고 Amazon S3 버킷이 퍼블릭으로 된다고 가정하겠습니다. 대시보드에는 제어 상태가 규정 준수에서 미확인으로 변경되었다고 표시됩니다.
  - AWS Config 및 AWS Audit Manager 서비스를 켭니다.
  - AWS Config 규칙 및 AWS Audit Manager 자동 평가 생성.
  - AWS Marketplace Vendor Insights가 평가 결과를 가져올 수 있도록 AWS Identity and Access Management (IAM) 역할을 프로비저닝합니다.
- ISO 27001 및 SOC2 Type II 보고서 - 컨트롤 범주는 국제 표준화 기구(ISO) 보고서와 시스템 및 조직 제어(SOC2) 보고서의 컨트롤에 매핑됩니다. 이러한 보고서를 AWS Marketplace Vendor Insights와 공유하면 Vendor Insights는 보고서에서 관련 증거를 추출하여 대시보드에 표시할 수 있습니다.

AWS Marketplace Vendor Insights 설정

다음 절차에서는 서비스형 소프트웨어(SaaS) 목록에 AWS Marketplace Vendor Insights AWS Marketplace 를 설정하는 상위 단계를 설명합니다.

SaaS 목록에 AWS Marketplace Vendor Insights를 설정하려면

1. the section called “보안 프로필 생성”.
2. (선택 사항) the section called “인증서 업로드”.
3. the section called “자체 평가 업로드”.
4. the section called “AWS Audit Manager 자동 평가 활성화”(선택 사항).

보안 프로필 생성

보안 프로필은 구매자에게 소프트웨어 제품의 보안 태세에 대한 구체적인 정보를 제공합니다. 보안 프로필은 자체 평가, 인증 및 AWS Audit Manager 자동 평가를 비롯한 관련 데이터 소스를 사용합니다.

> **Note**
> 제한된 수의 보안 프로필을 생성할 수 있습니다. 보안 프로필을 더 생성하려면 할당량을 늘려 달라고 요청합니다. 자세한 내용은AWS 일반 참조의 AWS Service Quotas를 참조하세요.

보안 프로필을 생성하는 방법

1. AWS Marketplace 판매자 계정에 액세스할 수 있는 IAM 사용자 또는 역할을 사용하여 로그인합니다.
2. 제품을 선택하고 SaaS를 선택하여 SaaS 제품 페이지로 이동합니다.
3. 제품을 선택합니다.
4. Vendor Insights 탭을 선택하고 보안 프로필 추가에 대한 지원 문의를 선택합니다.
5. 양식을 작성하고 제출을 선택합니다.

   AWS Marketplace 판매자 운영 팀이 보안 프로필을 생성합니다. 보안 프로필이 준비되면 양식에 기재된 수취인에게 알림 이메일 메시지가 전송됩니다.

인증서 업로드

인증은 여러 차원에서 제품의 보안 태세에 대한 증거를 제공하는 데이터 소스입니다. AWS Marketplace 공급업체 인사이트는 다음 인증을 지원합니다.

- FedRAMP 인증 - 미국 정부 클라우드 보안 표준을 준수한다는 것을 인증합니다.
- GDPR 준수 보고서 - 일반 데이터 보호 규정(GDPR) 요구 사항을 준수한다는 것을 증명하며 개인 데이터 및 개인의 프라이버시를 보호합니다.
- HIPAA 준수 보고서 - 미국 건강 보험 양도 및 책임에 관한 법(HIPAA) 규정을 준수한다는 것을 증명하며 보호 대상 건강 정보를 보호합니다.
- ISO/IEC 27001 감사 보고서 - 국제표준화기구(ISO) 및 국제전기기술위원회(IEC) 27001을 준수한다는 것을 증명하며 정보 보안 표준을 강조합니다.
- PCI DSS 감사 보고서 - PCI 보안 표준 위원회에서 정한 보안 표준을 준수한다는 것을 증명합니다.
- SOC 2 유형 2 감사 보고서 - 서비스 조직 통제(SOC) 데이터 프라이버시 및 보안 컨트롤을 준수한다는 것을 증명합니다.

인증서를 업로드하는 방법

1. Vendor Insights 탭에서 데이터 소스 섹션으로 이동합니다.
2. 인증에서 인증서 업로드를 선택합니다.
3. 인증 세부 정보에서 요청된 정보를 제공하고 인증서를 업로드합니다.
4. (선택 사항) 태그에서 새 태그를 추가합니다.

> **Note**
>
> 태그에 대한 자세한 내용은 AWS 리소스 태그 지정 사용 설명서의 리소스 태그 지정을 참조하세요. AWS

5. 인증서 업로드를 선택합니다.

> **Note**
>
> 인증서는 현재 보안 프로필과 자동으로 연결됩니다. 이미 업로드한 인증서를 연결할 수도 있습니다. 제품 세부 정보 페이지의 인증에서 인증서 연결을 선택하고 목록에서 인증서를 선택한 다음, 인증서 연결을 선택합니다.

인증을 업로드한 후 제품 상세 페이지의 인증 다운로드 버튼을 사용하여 다운로드할 수 있습니다. 인증 업데이트 버튼을 사용하여 인증 세부 정보를 업데이트할 수도 있습니다.

인증 세부 정보가 검증될 때까지 인증 상태는 ValidationPending으로 바뀝니다. 데이터 소스가 처리되는 동안 그리고 처리된 후에 대체 상태가 나타납니다.

- Available - 데이터 소스가 업로드되었고 시스템 검증이 성공적으로 완료되었습니다.
- AccessDenied - AWS Marketplace Vendor Insights가 더 이상 데이터 소스의 외부 소스 참조에 액세스할 수 없습니다.
- ResourceNotFound - Vendor Insights가 더 이상 데이터 소스의 외부 소스 참조에 액세스하여 읽을 수 없습니다.
- ResourceNotSupported - 데이터 소스가 업로드되었지만 제공된 소스가 아직 지원되지 않습니다. 검증 오류에 대한 자세한 내용은 상태 메시지를 참조하세요.
- ValidationPending - 데이터 소스가 업로드되었지만 시스템 검증이 아직 실행 중입니다. 이 단계에서는 실행할 수 있는 작업 항목이 없습니다. 상태가 Available, ResourceNotSupported 또는 ValidationFailed로 업데이트됩니다.
- ValidationFailed - 데이터 소스가 업로드되었지만 한 가지 이상의 이유로 시스템 검증이 실패했습니다. 검증 오류에 대한 자세한 내용은 상태 메시지를 참조하세요.

자체 평가 업로드

자체 평가는 제품의 보안 태세에 대한 증거를 제공하는 데이터 소스 유형입니다. AWS Marketplace 공급업체 인사이트는 다음과 같은 자체 평가를 지원합니다.

- AWS Marketplace Vendor Insights 자체 평가
- Consensus Assessment Initiative Questionnaire(CAIQ). 자세한 내용은 Cloud Security Alliance 웹사이트의 What is CAIQ를 참조하세요.

자체 평가를 업로드하는 방법

1. https://console.aws.amazon.com/marketplace AWS Marketplace 콘솔을 엽니다.
2. Vendor Insights 탭에서 데이터 소스 섹션으로 이동합니다.
3. 자체 평가에서 자체 평가 업로드를 선택합니다.
4. 자체 평가 세부 정보에서 다음 정보를 작성합니다.

a. 이름 - 자체 평가의 이름을 입력합니다.

b. Type - 드롭다운 목록에서 평가 유형을 선택합니다.

> **Note**
>
> Vendor Insights 보안 자체 평가를 선택한 경우 템플릿 다운로드를 선택하여 자체 평가를 다운로드합니다. 스프레드시트에서 각 질문에 대한 답변으로 예, 아니요 또는 N/A를 선택합니다.

5. 완료된 평가를 업로드하려면 자체 평가 업로드를 선택합니다.
6. (선택 사항) 태그에서 새 태그를 추가합니다.

> **Note**
>
> 태그에 대한 자세한 내용은 AWS 리소스 태그 지정 사용 설명서의 리소스 태그 지정을 참조하세요. AWS

7. 자체 평가 업로드를 선택합니다.

> **Note**
>
> 자체 평가는 현재 보안 프로필과 자동으로 연결됩니다. 이미 업로드한 자체 평가를 연결할 수도 있습니다. 제품 세부 정보 페이지의 자체 평가에서 자체 평가 연결을 선택하고, 목록에서 자체 평가를 선택하고, 자체 평가 연결을 선택합니다.
> 자체 평가를 업로드한 후 제품 상세 페이지의 자체 평가 다운로드 버튼을 사용하여 다운로드할 수 있습니다. 자체 평가 업데이트 버튼을 사용하여 자체 평가 세부 정보를 업데이트할 수도 있습니다.

상태가 다음 중 하나로 업데이트됩니다.

- Available - 데이터 소스가 업로드되었고 시스템 검증이 성공적으로 완료되었습니다.
- AccessDenied - Vendor Insights가 더 이상 데이터 소스의 외부 소스 참조에 액세스하여 읽을 수 없습니다.
- ResourceNotFound - Vendor Insights가 더 이상 데이터 소스의 외부 소스 참조에 액세스하여 읽을 수 없습니다.

- ResourceNotSupported - 데이터 소스가 업로드되었지만 제공된 소스가 아직 지원되지 않습니다. 검증 오류에 대한 자세한 내용은 상태 메시지를 참조하세요.
- ValidationPending - 데이터 소스가 업로드되었지만 시스템 검증이 아직 실행 중입니다. 이 단계에서는 실행할 수 있는 작업 항목이 없습니다. 상태가 Available, ResourceNotSupported 또는 ValidationFailed로 업데이트됩니다.
- ValidationFailed - 데이터 소스가 업로드되었지만 한 가지 이상의 이유로 시스템 검증이 실패했습니다. 검증 오류에 대한 자세한 내용은 상태 메시지를 참조하세요.

AWS Audit Manager 자동 평가 활성화

AWS Marketplace Vendor Insights는 여러 AWS 서비스 를 사용하여 보안 프로필에 대한 증거를 자동으로 수집합니다.

자동화된 평가를 위해서는 다음과 같은 AWS 서비스 및 리소스가 필요합니다.

- AWS Audit Manager - AWS Marketplace Vendor Insights 설정을 간소화하기 위해 필요한 리소스를 프로비저닝하고 구성하는 AWS CloudFormation Stacks 및 StackSets를 사용합니다. 이 스택 세트는 AWS Config에 의해 자동으로 채워지는 컨트롤이 포함된 자동 평가를 생성합니다.

  에 대한 자세한 내용은 AWS Audit Manager 사용 설명서를 AWS Audit Manager참조하세요.
- AWS Config - 스택 세트는 적합성 팩을 AWS Config 배포하여 필요한 AWS Config 규칙을 설정합니다. 이러한 규칙을 통해 Audit Manager 자동 평가는 해당에 AWS 서비스 배포된 다른에 대한 실시간 증거를 수집할 수 있습니다 AWS 계정. AWS Config 기능에 대한 자세한 내용은 AWS Config 개발자 안내서를 참조하세요.

> **Note**
>
> 를 기록한 첫 달에는 다음 달에 AWS Config 비해 계정에서 활동이 증가하는 것을 확인할 수 있습니다. 초기 부트스트래핑 프로세스 중에는에서 기록하도록 선택한 AWS Config 계정의 모든 리소스를 AWS Config 검토합니다.
> 임시 워크로드를 실행하는 경우 이러한 임시 리소스 생성 및 삭제와 관련된 구성 변경을 기록 AWS Config 하므로의 활동이 증가할 수 있습니다. 임시 워크로드는 컴퓨팅 리소스를 필요할 때 로드 및 실행하는 일시적 사용입니다.
> 임시 워크로드의 예로는 Amazon Elastic Compute Cloud(Amazon EC2) 스팟 인스턴스, Amazon EMR 작업 AWS Auto Scaling및 등이 있습니다 AWS Lambda. 활동 증가로 인해 임시 워크로드가 실행되지 않도록 하려면 AWS Config 가 꺼져 있는 별도의 계정에서 이러한

유형의 워크로드를 실행할 수 있습니다. 이 방법을 사용하면 구성 기록 및 규칙 평가가 늘어나는 것을 방지할 수 있습니다.

- Amazon S3 - 스택 세트는 다음과 같은 두 개의 Amazon Simple Storage Service 버킷을 생성합니다.
  - vendor-insights-stack-set-output-bucket-{account number} - 이 버킷에는 스택 세트 실행의 출력이 포함되어 있습니다. AWS Marketplace 판매자 운영 팀은 출력을 사용하여 자동 데이터 소스 생성 프로세스를 완료합니다.
  - vendor-insights-assessment-reports-bucket-{account number} -는이 Amazon S3 버킷에 평가 보고서를 AWS Audit Manager 게시합니다. 평가 보고서 게시에 대한 자세한 내용은 AWS Audit Manager 사용 설명서의 평가 보고서를 참조하세요.

    Amazon S3 기능에 대한 자세한 내용은 Amazon S3 사용 설명서를 참조하세요.
- IAM - 온보딩 스택 세트는 계정에서 다음 AWS Identity and Access Management (IAM) 역할을 프로비저닝합니다.
  - `VendorInsightsPrerequisiteCFT.yml` 템플릿이 배포되면 해당 템플릿은 관리자 역할 `AWSVendorInsightsOnboardingStackSetsAdmin` 및 실행 역할 `AWSVendorInsightsOnboardingStackSetsExecution`을 생성합니다. 이 스택 세트는 관리자 역할을 사용하여 필요한 스택을 여러 AWS 리전 에 동시에 배포합니다. 관리자 역할은 AWS Marketplace Vendor Insights 설정 프로세스의 일부로 필요한 상위 스택과 중첩 스택을 배포하는 실행 역할을 수임합니다. 자체 관리형 권한에 대한 자세한 내용은 AWS CloudFormation 사용 설명서의 자체 관리형 권한 부여를 참조하세요.
  - `AWSVendorInsightsRole` 역할은 AWS Marketplace Vendor Insights에 AWS Audit Manager 리소스의 평가를 읽을 수 있는 액세스 권한을 제공합니다. AWS Marketplace Vendor Insights는 AWS Marketplace Vendor Insights 프로파일의 평가에 있는 증거를 표시합니다.
  - 는 `vendor-insights-stack-set-output-bucket` 버킷의 객체를 나열하고 읽을 수 있는 액세스 권한을 AWS Marketplace Vendor Insights에 `AWSVendorInsightsOnboardingDelegationRole` 제공합니다. 이 기능을 사용하면 AWS Marketplace 카탈로그 운영 팀이 AWS Marketplace Vendor Insights 프로파일 설정을 지원할 수 있습니다.
  - 이 `AWSAuditManagerAdministratorAccess` 역할은 평가 AWS Audit Manager, 제어 및 프레임워크를 활성화 또는 비활성화하고, 설정을 업데이트하고, 관리할 수 있는 관리 액세스 권한을 제공합니다. 판매자 또는 판매자의 팀이 이 역할을 맡아 AWS Audit Manager에서 자동 평가에 대한 조치를 취할 수 있습니다.

AWS Audit Manager 자동 평가를 활성화하려면 온보딩 스택을 배포해야 합니다.

온보딩 스택 배포

AWS Marketplace Vendor Insights 설정을 간소화하기 위해 필요한 리소스를 프로비저닝하고 구성하는 AWS CloudFormation Stacks 및 StackSets를 사용합니다. 여러 계정 또는 여러 AWS 리전 SaaS 솔루션이 있는 경우 StackSets를 사용하면 중앙 관리 계정에서 온보딩 스택을 배포할 수 있습니다.

CloudFormation StackSets에 대한 자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation StackSets 작업을 참조하세요.

AWS Marketplace Vendor Insights를 설정하려면 다음 CloudFormation 템플릿을 사용해야 합니다.

- `VendorInsightsPrerequisiteCFT` - 계정에서 CloudFormation StackSets를 실행하는 데 필요한 관리자 역할 및 권한을 설정합니다. 판매자 계정에서 이 스택을 생성하세요.
- `VendorInsightsOnboardingCFT` - 필수 AWS 서비스 를 설정하고 적절한 IAM 권한을 구성합니다. 이러한 권한을 통해 AWS Marketplace Vendor Insights는에서 실행되는 SaaS 제품에 대한 데이터를 수집하고 AWS Marketplace Vendor Insights 프로파일에 데이터를 AWS 계정 표시할 수 있습니다. StackSets를 통해 SaaS 솔루션을 호스팅하는 판매자 계정과 프로덕션 계정 모두에서 이 스택을 생성하세요.

VendorInsightsPrerequisiteCFT 스택을 생성합니다.

`VendorInsightsPrerequisiteCFT` CloudFormation 스택을 실행하여 스택 세트를 온보딩하기 위한 IAM 권한을 설정합니다.

VendorInsightsPrerequisiteCFT 스택을 생성하는 방법

1. GitHub 웹 사이트의 Vendor Insights용AWS 샘플 리포지토리 템플릿 폴더에서 최신 `VendorInsightsPrerequisiteCFT.yml` 파일을 검토하고 다운로드합니다.
2. AWS Marketplace 판매자 계정을 AWS Management Console 사용하여에 로그인한 다음 https://console.aws.amazon.com/cloudformation 콘솔을 AWS CloudFormation 엽니다.
3. CloudFormation 콘솔 탐색 창에서 스택을 선택한 다음, 드롭다운에서 스택 생성을 선택하고 새 리소스 사용(표준)을 선택합니다. (탐색 창이 보이지 않으면 왼쪽 상단 모서리에서 탐색 창을 선택하고 확장합니다.)
4. 템플릿 지정에서 템플릿 파일 업로드를 선택합니다. 다운로드한 `VendorInsightsPrerequisiteCFT.yml` 파일을 업로드하려면 파일 선택을 사용합니다. 그런 다음 다음을 선택합니다.
5. 스택의 이름을 입력하고 다음을 선택합니다.

6. (선택 사항) 스택 옵션을 원하는 대로 구성합니다.

   다음을 선택합니다.
7. [검토] 페이지에서 선택 사항을 검토합니다. 변경 작업을 수행하려면 변경하려는 영역에서 편집을 선택합니다. 스택을 생성하려면 먼저 기능 영역에서 승인 확인란을 선택해야 합니다.

   제출을 선택합니다.
8. 스택이 생성된 후에는 리소스 탭을 선택하고, 생성된 다음 역할을 기록해 둡니다.

   - `AWSVendorInsightsOnboardingStackSetsAdmin`
   - `AWSVendorInsightsOnboardingStackSetsExecution`

VendorInsightsOnboardingCFT 스택 세트 생성

`VendorInsightsOnboardingCFT` CloudFormation 스택 세트를 실행하여 필요한를 설정하고 적절한 IAM 권한을 AWS 서비스 구성합니다. 이를 통해 AWS Marketplace Vendor Insights는에서 실행되는 SaaS 제품에 대한 데이터를 수집하여 AWS Marketplace Vendor Insights 프로파일에 AWS 계정 표시할 수 있습니다.

여러 계정 솔루션이 있거나 판매자 계정과 프로덕션 계정이 분리되어 있는 경우 이 스택을 여러 계정에 배포해야 합니다. StackSets를 사용하면 사전 조건 스택을 생성한 관리 계정에서 이 작업을 수행할 수 있습니다.

스택 세트는 자체 관리형 권한을 사용하여 배포합니다. 자세한 내용은 AWS CloudFormation 사용 설명서의 자체 관리 권한으로 스택 세트 만들기를 참조하세요.

VendorInsightsOnboardingCFT 스택 세트를 생성하는 방법

1. GitHub 웹 사이트의 Vendor Insights용AWS 샘플 리포지토리 템플릿 폴더에서 최신 `VendorInsightsOnboardingCFT.yml` 파일을 검토하고 다운로드합니다.
2. AWS Marketplace 판매자 계정을 AWS Management Console 사용하여에 로그인한 다음 https://console.aws.amazon.com/cloudformation 콘솔을 AWS CloudFormation 엽니다.
3. CloudFormation 콘솔 탐색 창에서 StackSet 생성을 선택합니다. (탐색 창이 보이지 않으면 왼쪽 상단 모서리에서 탐색 창을 선택하고 확장합니다.)
4. 권한에서 관리자 역할로 IAM 역할 이름을 선택한 다음, 드롭다운에서 역할 이름으로 AWSVendorInsightsOnboardingStackSetsAdmin을 선택합니다.
5. IAM 실행 역할 이름으로 **`AWSVendorInsightsOnboardingStackSetsExecution`**을 입력합니다.

6. 템플릿 지정에서 템플릿 파일 업로드를 선택합니다. 다운로드한 `VendorInsightsOnboardingCFT.yml` 파일을 업로드하려면 파일 선택을 사용한 후 다음을 선택합니다.
7. 다음 StackSet 파라미터를 입력하고 다음을 선택합니다.
   - `CreateVendorInsightsAutomatedAssessment` -이 파라미터는에서 AWS Audit Manager 자동 평가를 설정합니다 AWS 계정. 관리 계정과 프로덕션 계정이 분리되어 있는 경우 이 옵션은 프로덕션 계정에만 선택하고 관리 계정에는 선택하면 안 됩니다.
   - `CreateVendorInsightsIAMRoles` -이 파라미터는 AWS Marketplace Vendor Insights가의 평가 데이터를 읽을 수 있도록 IAM 역할을 프로비저닝합니다 AWS 계정.
   - `PrimaryRegion` - 이 파라미터는 SaaS 배포에 사용할 기본 AWS 리전 을 설정합니다. 에서 Amazon S3 버킷이 생성되는 리전입니다 AWS 계정. SaaS 제품이 한 리전에만 배포되는 경우 해당 리전이 기본 리전입니다.
8. StackSet 옵션을 원하는 대로 구성합니다. 실행 구성을 비활성으로 유지하고 다음을 선택합니다.
9. 배포를 구성합니다. 계정 솔루션이 여러 개 있는 경우 한 번의 작업으로 여러 계정과 여러 리전에 배포하도록 스택 세트를 구성할 수 있습니다. 다음을 선택합니다.

> **Note**
>
> 계정 솔루션이 여러 개 있는 경우 모든 계정을 단일 스택 세트로 배포하는 것은 좋지 않습니다. 7단계에서 정의한 파라미터에 주의를 기울여야 합니다. 배포하려는 계정 유형에 따라 일부 파라미터를 활성화하거나 비활성화해야 할 수 있습니다. StackSets는 단일 배포에서 지정된 모든 계정에 동일한 파라미터를 적용합니다. 계정을 스택 세트로 그룹화하여 배포 시간을 줄일 수 있지만, 다중 계정 솔루션의 경우 여전히 여러 번 배포해야 합니다.

> **Important**
>
> 여러 리전에 배포하는 경우 목록에 있는 첫 번째 리전이 `PrimaryRegion`이어야 합니다. 리전 동시성 옵션은 기본 설정인 순차적적으로 둡니다.

10. [검토] 페이지에서 선택 사항을 검토합니다. 변경 작업을 수행하려면 변경하려는 영역에서 편집을 선택합니다. 스택 세트를 생성하려면 먼저 기능 영역에서 승인 확인란을 선택해야 합니다.

    제출을 선택합니다.

    스택 세트를 생성하는 데 리전당 5분 가량 소요됩니다.

AWS Marketplace Vendor Insights 프로필 보기

AWS Marketplace Vendor Insights의 프로필은 구매자가 제품을 평가할 때 사용할 수 있는 중요한 정보를 제공합니다. 데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS Identity and Access Management (IAM)을 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이러한 접근 방식에서는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. IAM를 사용하여 사용자를 생성하는 방법에 대한 자세한 내용은 the section called “그룹 생성 또는 사용” 섹션을 참조하세요.

구매자는 AWS Marketplace Vendor Insights 대시보드를 사용하여 제품을 평가할 수 있습니다. 구매자는 이 대시보드에서 판매자가 프로필에 추가한 데이터 소스를 통해 정의된 제품 개요를 볼 수 있습니다. 보안 프로필은 10개 범주의 여러 보안 컨트롤을 통해 정의됩니다.

데이터를 정의하는 데 사용되는 10개 보안 범주는 다음과 같습니다.

- 액세스 관리
- 애플리케이션 보안
- 감사, 규정 준수 및 보안 정책
- 비즈니스 복원력
- 데이터 보안 및 개인정보 보호
- 최종 사용자 디바이스 및 모바일 보안
- 인적 자원
- 인프라 보안
- 위험 관리 및 인시던트 대응
- 보안 및 구성 정책

자세한 내용은 AWS Marketplace사용 설명서의 컨트롤 범주 이해를 참조하세요.

AWS Marketplace Vendor Insights를 설정하고 사용하면 사용자 정보를 비공개로 안전하게 유지하기 위해 AWS 서비스 약관 및 데이터 개인 정보 보호 규칙을 준수하는 데 동의하는 것입니다. AWS 데이터 프라이버시 조건에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 서비스 약관에 대한 자세한 내용은 AWS 서비스 약관을 참조하세요.

판매자로 보안 프로필 보기

자체 평가를 완료하고 다른 실시간 증거를 추가한 후에는 판매자로 프로필을 보는 것이 중요합니다. 프로필에 추가된 정보를 검토하는 것이 좋습니다.

> **Note**
>
> 이 프로필은 AWS Marketplace Vendor Insights 지원 팀에 가시성 업데이트를 요청할 때까지 구매자가 볼 수 없습니다. 지원 팀이 업데이트를 완료하면 비밀 유지 계약(NDA)에 서명한 구매자는 보안 프로필에 액세스할 수 있습니다.
> AWS Marketplace Vendor Insights 프로필에서 구독자의 개인 식별 정보(PII) 데이터를 삭제하려면에 문의하여 지원 사례를 시작합니다지원.

구매자가 보는 것처럼 보안 프로필을 보는 방법

1. AWS Management Console에 로그인합니다.
2. 포털의 SaaS 제품 페이지로 이동합니다.
3. 관련 보안 프로필이 있는 제품을 선택합니다.
4. Vendor Insights 탭을 선택하고 최근에 공개된 스냅샷 보기를 선택합니다.
5. 개요 탭에는 판매자가 업로드한 모든 인증서 배지가 표시됩니다.
6. 여러 컨트롤에서 수집한 데이터를 볼 수 있는 보안 및 규정 준수 탭을 선택합니다. 더 자세한 내용을 보려면 각 컨트롤 세트를 선택합니다.

AWS Marketplace Vendor Insights에서 스냅샷 관리

스냅샷은 보안 프로필의 특정 시점 태세입니다. AWS Marketplace Vendor Insights에서 스냅샷을 사용하여 언제든지 판매자의 제품을 평가할 수 있습니다. 판매자는 여러 시점의 프로필 보안 태세를 비교하거나 여러 보안 프로필의 최신 스냅샷을 비교하여 의사 결정에 활용할 수 있습니다. 스냅샷은 데이터의 최신성과 출처에 대한 투명성뿐 아니라 필요한 보안 정보를 제공합니다.

AWS Marketplace 콘솔의 AWS Marketplace Vendor Insights 스냅샷 요약 섹션에서 생성 및 릴리스 일정에 대한 다음과 같은 스냅샷 세부 정보를 볼 수 있습니다.

- 마지막으로 생성된 스냅샷 - 이 프로필에 대해 마지막으로 생성된 스냅샷입니다.
- 다음 예약 생성 - 생성하기로 예약된 다음 스냅샷입니다.
- 생성 빈도 - 스냅샷 생성 간격 또는 스냅샷 생성 빈도입니다.
- 다음 예약 릴리스 - 릴리스하기로 예약된 다음 스냅샷입니다.
- 스테이징 시간 - 스냅샷은 적어도 이 시간 동안 스테이징되며, 이후 스냅샷 릴리스 이벤트 중에 릴리스할 수 있습니다.

- 릴리스 빈도 - 릴리스 이벤트 간격입니다.

스냅샷 목록 섹션의 스냅샷 상태는 다음과 같습니다.

- 릴리스됨 - 스냅샷이 공개되었으며 이 제품에 대한 권한이 있는 사용자는 스냅샷을 볼 수 있습니다.
- 릴리스 보류 중 - 스냅샷이 완료되었거나 필수 최소 스테이징 기간 중에 있으며 다음 릴리스로 예정되어 있습니다.
- 프라이빗 - 보안 프로필이 활성화되기 전에 생성되었거나 검증 오류가 발생하여 일반에 공개되지 않는 스냅샷입니다. 프라이빗 스냅샷은 판매자만 볼 수 있는 상태로 유지됩니다.

주제

- 스냅샷 생성
- 스냅샷 보기
- 스냅샷 내보내기
- 최근에 릴리스된 스냅샷 보기
- 스냅샷 릴리스 연기하기
- 스냅샷 목록의 기본 설정 변경

스냅샷 생성

프로필 스냅샷을 생성하려면 다음 단계를 수행합니다. 하루에 최대 20개의 스냅샷을 생성할 수 있습니다.

1. AWS Management Console에 로그인하고 AWS Marketplace 콘솔을 엽니다.
2. Vendor Insights를 선택합니다.
3. Vendor Insights에서 제품을 선택합니다.
4. 제품 프로필 페이지에서 스냅샷 목록으로 이동하여 새 스냅샷 생성을 선택합니다.
5. 스냅샷 일정이 변경된다고 알려주는 메시지가 나타납니다. 생성(Create)을 선택합니다.

> **Note**
>
> 새 스냅샷이 생성되면 스냅샷 일정이 변경됩니다. 새 스냅샷은 수동으로 생성한 스냅샷과 같은 시간에 예약됩니다. 이 메시지에는 새 일정이 포함되어 있습니다.

새 스냅샷이 30분 내에 생성되어 스냅샷 목록에 추가됩니다. 새 스냅샷은 릴리스 보류 중 상태로 생성됩니다. 상태가 릴리스됨으로 변경될 때까지 아무도 새 스냅샷을 볼 수 없습니다.

스냅샷 보기

프로필 스냅샷을 보려면 다음 단계를 수행합니다.

1. AWS Management Console에 로그인하고 AWS Marketplace 콘솔을 엽니다.
2. Vendor Insights를 선택합니다.
3. Vendor Insights에서 제품을 선택합니다.
4. 제품 프로필 페이지에서 스냅샷 목록으로 이동한 후, 보고 싶은 스냅샷의 스냅샷 ID를 선택합니다.
5. 모두 마쳤으면 뒤로를 선택하여 스냅샷 보기를 종료합니다.

스냅샷 내보내기

JSON 또는 CSV 형식으로 내보낼 수 있습니다. 스냅샷을 내보내려면 다음 단계를 수행합니다.

1. AWS Management Console에 로그인하고 AWS Marketplace 콘솔을 엽니다.
2. Vendor Insights를 선택합니다.
3. Vendor Insights에서 제품을 선택합니다.
4. 제품 프로필 페이지에서 스냅샷 목록으로 이동한 후, 내보내려는 스냅샷의 스냅샷 ID를 선택합니다.
5. 내보내기를 선택합니다.
6. 드롭다운 목록에서 다운로드(JSON) 또는 다운로드(CSV)를 선택합니다.

최근에 릴리스된 스냅샷 보기

가장 최근에 릴리스된 스냅샷은 사용자가 제품 상태를 보고 평가하는 데 사용하는 스냅샷입니다. 제품을 정확하게 묘사할 수 있도록 최근에 릴리스된 스냅샷의 내용을 아는 것이 중요합니다. 프로필의 최근 스냅샷을 보려면 다음 단계를 수행합니다.

1. AWS Management Console에 로그인하고 AWS Marketplace 콘솔을 엽니다.
2. Vendor Insights를 선택합니다.
3. Vendor Insights에서 제품을 선택합니다.

4. 제품 프로필 페이지에서 스냅샷 목록으로 이동한 후, 보고 싶은 스냅샷의 스냅샷 ID를 선택합니다.
5. 최근에 릴리스된 스냅샷 보기를 선택합니다.
6. 모두 마쳤으면 뒤로를 선택하여 스냅샷 보기를 종료합니다.

스냅샷 릴리스 연기하기

프로필 스냅샷의 릴리스를 연기하려면 특정 스냅샷 ID에 대한 스냅샷 릴리스를 연기하면 됩니다.

1. AWS Management Console에 로그인하고 AWS Marketplace 콘솔을 엽니다.
2. Vendor Insights를 선택합니다.
3. Vendor Insights에서 제품을 선택합니다.
4. 제품 프로필 페이지에서 스냅샷 목록으로 이동한 후, 릴리스를 연기하려는 스냅샷의 스냅샷 ID를 선택합니다.
5. 스냅샷 요약에서 스냅샷 릴리스 연기를 선택합니다.
6. 스냅샷 일정이 변경된다고 알려주는 메시지가 나타납니다. 연기를 선택합니다.

이 제품의 스냅샷 릴리스를 연기했다고 알려주는 성공 메시지가 나타납니다.

스냅샷 목록의 기본 설정 변경

스냅샷을 생성한 후, 스냅샷 목록에서 스냅샷이 표시되는 방식의 기본 설정을 변경할 수 있습니다.

1. AWS Management Console에 로그인하고 AWS Marketplace 콘솔을 엽니다.
2. Vendor Insights를 선택합니다.
3. Vendor Insights에서 제품을 선택합니다.
4. 제품 프로필 페이지에서 스냅샷 목록으로 이동한 후, 변경하려는 스냅샷의 스냅샷 ID를 선택합니다.
5. 기본 설정 아이콘을 선택합니다. 스냅샷의 다음 기본 설정을 사용자 지정할 수 있습니다.
   - 페이지 크기 - 각 페이지에 나열할 스냅샷 수를 페이지당 리소스 10개, 리소스 20개 또는 리소스 50개 중에 선택합니다.
   - 줄 바꿈 - 전체 레코드를 표시할 줄 바꿈 옵션을 선택합니다.
   - 시간 형식 - 절대, 상대 또는 ISO 중에서 원하는 형식을 선택합니다.
   - 표시되는 열 - 스냅샷 세부 정보에 표시할 옵션(스냅샷 ID, 상태, 생성된 날짜)을 선택합니다.

AWS Marketplace Vendor Insights의 액세스 권한 제어

AWS Identity and Access Management (IAM)는 AWS 리소스에 대한 액세스를 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다. 관리자인 경우 리소스를 사용할 AWS Marketplace 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는 사용자를 제어합니다. AWS Marketplace Vendor Insights는 IAM을 사용하여 판매자 데이터, 평가, 판매자 자체 증명 및 업계 표준 감사 보고서에 대한 액세스를 제어합니다.

에서 누가 무엇을 할 수 있는지 제어하는 권장 방법은 IAM을 사용하여 사용자와 그룹을 생성하는 AWS Marketplace Management Portal 것입니다. 그런 다음 사용자를 그룹에 추가하고 그룹을 관리합니다. 읽기 전용 권한을 제공하는 정책 또는 권한을 해당 그룹에 할당할 수 있습니다. 읽기 전용 액세스가 필요한 다른 사용자들이 있는 경우에는 이들에게 권한을 추가하지 말고 생성한 그룹에 이들을 추가하면 됩니다.

정책은 사용자, 그룹 또는 역할에 적용되는 권한을 정의하는 문서입니다. 권한은 사용자가 AWS에서 수행할 수 있는 작업을 결정합니다. 정책은 일반적으로 특정 작업에 대한 액세스를 허용하며 Amazon EC2 인스턴스, Amazon S3 buckets 버킷과 같은 특정 리소스에 대한 작업을 허용하도록 선택적으로 권한을 부여할 수 있습니다. 정책은 액세스를 명시적으로 거부할 수도 있습니다. 권한은 특정 리소스에 대한 액세스를 허용하거나 거부하는 정책 내 문입니다.

> **Important**
>
> 생성하는 모든 사용자는 자신의 자격 증명을 사용하여 인증합니다. 그러나 이들은 동일한 AWS 계정을 사용합니다. 사용자에 의한 모든 변경은 전체 계정에 영향을 줄 수 있습니다.

AWS Marketplace 에는 해당 권한이 있는 사용자가에서 수행할 수 있는 작업을 제어하도록 정의된 권한이 있습니다 AWS Marketplace Management Portal. 또한 여러 권한을 결합하는를 AWS Marketplace 생성하고 관리하는 정책도 있습니다. 이 `AWSMarketplaceSellerProductsFullAccess` 정책은 사용자에게 AWS Marketplace Management Portal의 제품에 대한 전체 액세스 권한을 부여합니다.

사용 가능한 작업, 리소스 및 조건 키에 대한 자세한 내용은 서비스 권한 부여 참조의 AWS Marketplace Vendor Insights에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

AWS Marketplace Vendor Insights 판매자에 대한 권한

AWS Marketplace Vendor Insights에 대한 IAM 정책에서 다음 권한을 사용할 수 있습니다. 이러한 권한을 단일 IAM 정책에 결합하여 원하는 권한을 부여할 수 있습니다.

CreateDataSource

CreateDataSource는 사용자가 새 데이터 소스 리소스를 생성할 수 있습니다. 지원되는 데이터 소스는 다음과 같습니다.

- SOC2Type2AuditReport
- ISO27001AuditReport
- AWSAuditManagerSecurityAutomatedAssessment
- FedRAMPCertification
- GDPRComplianceReport
- HIPAAComplianceReport
- PCIDSSAuditReport
- SecuritySelfAssessment

작업 그룹: 읽기-쓰기

필수 리소스: 없음

리소스 생성: DataSource

DeleteDataSource

DeleteDataSource는 사용자가 본인 소유의 데이터 소스를 삭제할 수 있습니다. 삭제하려면 모든 프로필에서 데이터 소스를 분리해야 합니다. 자세한 내용은 the section called "AssociateDataSource" 단원을 참조하세요.

작업 그룹: 읽기-쓰기

필수 리소스: DataSource

GetDataSource

GetDataSource는 사용자가 데이터 소스의 세부 정보를 검색할 수 있습니다. 데이터 소스의 세부 정보는 관련 타임스탬프, 원래 생성 파라미터, 처리 정보(있는 경우)와 같은 메타데이터 정보입니다.

작업 그룹: 읽기 전용, 읽기-쓰기

필수 리소스: DataSource

UpdateDataSource

UpdateDataSource는 사용자가 데이터 소스의 세부 정보를 업데이트할 수 있습니다. 세부 정보는 이름 및 소스 정보(예: 역할, 소스 Amazon 리소스 이름(ARN), 원본 콘텐츠)와 같은 메타데이터 정보입니다.

작업 그룹: 읽기 전용, 읽기-쓰기

필수 리소스: DataSource

ListDataSources

ListDataSources는 사용자가 본인 소유의 데이터 소스를 나열할 수 있습니다.

작업 그룹: 읽기 전용, 읽기-쓰기, 나열 전용

필수 리소스: 없음

CreateSecurityProfile

CreateSecurityProfile은 사용자가 새 보안 프로필을 생성할 수 있습니다. 보안 프로필은 스냅샷 생성 방법과 시기를 관리하는 리소스입니다. 또한 사용자는 프로필의 상태 및 해당 조건을 제어하여 구매자가 스냅샷에 액세스하는 방법을 제어할 수 있습니다.

작업 그룹: 읽기 전용, 읽기-쓰기, 나열 전용

필수 리소스: 없음

리소스 생성: SecurityProfile

ListSecurityProfiles

ListSecurityProfiles는 사용자가 본인 소유의 보안 프로필을 나열할 수 있습니다.

작업 그룹: 읽기 전용, 읽기-쓰기, 나열 전용

필수 리소스: 없음

GetSecurityProfile

CreateSecurityProfile은 사용자가 보안 프로필의 세부 정보를 얻을 수 있습니다.

작업 그룹: 읽기 전용 및 읽기-쓰기

필수 리소스: `SecurityProfile`

AssociateDataSource

`AssociateDataSource`를 사용하면 사용자가 기존를 AWS Marketplace Vendor Insights 프로필`DataSource`과 연결할 수 있습니다. 사용자는 데이터 소스를 프로필에 연결하거나 연결을 해제하여 스냅샷의 콘텐츠를 제어할 수 있습니다.

작업 그룹: 읽기-쓰기

필수 리소스: `SecurityProfile` 및 `DataSource`

DisassociateDataSource

`DisassociateDataSource`를 사용하면 사용자가 기존를 AWS Marketplace Vendor Insights 프로필`DataSource`과 연결 해제할 수 있습니다. 사용자는 데이터 소스를 프로필에 연결하거나 연결을 해제하여 스냅샷의 콘텐츠를 제어할 수 있습니다.

작업 그룹: 읽기-쓰기

필수 리소스: `SecurityProfile` 및 `DataSource`

UpdateSecurityProfile

`UpdateSecurityProfile`은 사용자가 이름이나 설명과 같은 보안 프로필 속성을 수정할 수 있습니다.

작업 그룹: 읽기-쓰기

필수 리소스: `SecurityProfile`

ActivateSecurityProfile

`ActivateSecurityProfile`은 사용자가 보안 프로필의 `Active` 상태를 설정할 수 있습니다. 보안 프로필이 활성화되면 다른 조건이 충족될 때 해제할 수 있는 `Staged` 상태로 새 스냅샷을 생성할 수 있습니다. 자세한 내용은 the section called "`UpdateSecurityProfileSnapshotReleaseConfiguration`" 단원을 참조하십시오.

`Released` 스냅샷이 하나 이상 있는 `Active` 보안 프로필은 최종 사용자를 위한 AWS Marketplace Vendor Insights 검색에 사용할 수 있습니다.

작업 그룹: 읽기-쓰기

필수 리소스: SecurityProfile

DeactivateSecurityProfile

DeactivateSecurityProfile은 사용자가 보안 프로필의 Inactive 상태를 설정할 수 있습니다. 보안 프로필이 이 터미널 상태이면 공유 상태에서 프로필을 삭제하는 것과 같습니다. 사용자는 프로필의 활성 구독자가 없는 경우에만 보안 프로필을 비활성화할 수 있습니다.

작업 그룹: 읽기-쓰기

필수 리소스: SecurityProfile

UpdateSecurityProfileSnapshotCreationConfiguration

UpdateSecurityProfileSnapshotCreationConfiguration은 사용자가 스냅샷 생성 구성을 위한 사용자 지정 일정을 정의할 수 있습니다. 주간 생성의 기본 생성 구성을 이 작업으로 재정의할 수 있습니다.

사용자는 이 작업을 사용하여 일정을 취소하거나, 일정을 미래 날짜로 연기하거나, 새 스냅샷 생성을 더 일찍 시작하는 등 일정을 변경할 수 있습니다.

작업 그룹: 읽기-쓰기

필수 리소스: SecurityProfile

UpdateSecurityProfileSnapshotReleaseConfiguration

UpdateSecurityProfileSnapshotReleaseConfiguration은 사용자가 스냅샷 릴리스 구성을 위한 사용자 지정 일정을 정의할 수 있습니다. 검토할 준비 기간이 이틀인 주간 릴리스의 기본 생성 구성을 이 작업으로 재정의할 수 있습니다.

사용자는 이 작업을 사용하여 일정을 취소하거나 일정을 미래 날짜로 연기하는 등 일정을 변경할 수 있습니다.

작업 그룹: 읽기-쓰기

필수 리소스: SecurityProfile

ListSecurityProfileSnapshots

ListSecurityProfileSnapshots는 사용자가 본인 소유의 보안 프로필에 대한 스냅샷을 나열할 수 있습니다.

작업 그룹: 읽기 전용, 나열 전용. 읽기-쓰기

필수 리소스: SecurityProfile

GetSecurityProfileSnapshot

GetSecurityProfileSnapshot은 사용자가 본인 소유의 보안 프로필에 대한 스냅샷을 가져올 수 있습니다.

작업 그룹: 읽기 전용 및 읽기-쓰기

필수 리소스: SecurityProfile

TagResource

TagResource는 사용자가 리소스에 새 태그를 추가할 수 있습니다. 지원되는 리소스는 SecurityProfile 및 DataSource입니다.

작업 그룹: 태그 지정

선택적 리소스: SecurityProfile 및 DataSource

UntagResource

UntagResource는 사용자가 리소스의 태그를 제거할 수 있습니다. 지원되는 리소스는 SecurityProfile 및 DataSource입니다.

작업 그룹: 태그 지정

선택적 리소스: SecurityProfile 및 DataSource

ListTagsForResource

ListTagsForResource는 사용자가 리소스의 태그를 나열할 수 있습니다. 지원되는 리소스는 SecurityProfile 및 DataSource입니다.

작업 그룹: 읽기 전용

선택적 리소스: SecurityProfile 및 DataSource

추가 리소스

IAM 사용 설명서의 다음 리소스에는 IAM을 시작하고 사용하는 방법이 자세히 나옵니다.

- IAM의 보안 모범 사례
- IAM 정책 관리
- IAM 사용자 그룹에 정책 연결
- IAM ID(사용자, 사용자 그룹, 역할)
- AWS 계정에 IAM 사용자 생성
- IAM 사용자 그룹 만들기
- 정책을 사용하여 AWS 리소스에 대한 액세스 제어

AWS Marketplace 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 AWS 규정 준수 프로그램 제공 범위 내 서비스를 AWS Marketplace 참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 데이터의 민감도, 조직의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다 AWS Marketplace. 다음 주제에서는 보안 및 규정 준수 목표를 충족하기 위해 AWS Marketplace 에 대한 액세스를 관리 AWS Identity and Access Management 하도록를 구성하는 방법을 보여줍니다. 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배울 수도 있습니다 AWS Marketplace .

에서 제공하는 제품과 관련된 보안 및 기타 정책에 대해 자세히 알아보려면 다음 주제를 AWS Marketplace참조하세요.

- 에 대한 AMI 기반 제품 요구 사항 AWS Marketplace
- 에 대한 컨테이너 기반 제품 요구 사항 AWS Marketplace
- 기계 학습 제품 생성을 위한 요구 사항 및 모범 사례
- 에 대한 SaaS 제품 지침 AWS Marketplace
- 의 전문 서비스 제품에 대한 요구 사항 AWS Marketplace

> **Note**
>
> AWS Data Exchange의 데이터 제품 보안에 대한 자세한 내용은 AWS Data Exchange 사용 설명서의 보안을 참조하세요.

의 구매자 보안에 대한 자세한 내용은 AWS Marketplace 구매자 안내서의 의 보안을 AWS Marketplace AWS Marketplace참조하세요.

주제

- AWS Marketplace Management Portal에 대한 액세스 제어
- AWS Marketplace 판매자를 위한 정책 및 권한
- AWSAWS Marketplace 판매자를 위한 관리형 정책
- AWS Marketplace 상거래 분석 서비스 계정 권한
- Amazon SQS 권한
- AWS Marketplace 측정 및 권한 부여 API 권한
- 에서 재판매 권한 부여에 서비스 연결 역할 사용 AWS Marketplace
- 를 사용하여 AWS Marketplace 측정 API 호출 로깅 AWS CloudTrail

AWS Marketplace Management Portal에 대한 액세스 제어

AWS Identity and Access Management (IAM)는 AWS 리소스에 대한 액세스를 제어하는 데 도움이 AWS 서비스 되는 입니다. 관리자인 경우 AWS Marketplace 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

에서 누가 무엇을 할 수 있는지 제어하는 권장 방법은 IAM을 사용하여 사용자와 그룹을 생성하는 AWS Marketplace Management Portal 것입니다. 그런 다음 사용자를 그룹에 추가하고 그룹을 관리합니다. 예를 들어 John이 자신의 제품을 볼 수 있으려면 사용자를 생성하고 읽기 전용 액세스용으로 생성한 그룹에 해당 사용자를 추가해야 합니다. 읽기 전용 권한을 제공하는 정책 또는 권한을 해당 그룹에 할당할 수 있습니다. 읽기 전용 액세스가 필요한 다른 사용자들이 있는 경우에는 이들에게 권한을 추가하지 말고 생성한 그룹에 이들을 추가하면 됩니다. John의 역할이 달라져 더 이상 읽기 전용 액세스가 필요 없는 경우에는 해당 그룹에서 John을 제거하면 됩니다.

정책은 사용자, 그룹 또는 역할에 적용되는 권한을 정의하는 문서입니다. 그리고 권한은 사용자가 AWS에서 수행할 수 있는 작업을 결정합니다. 정책은 일반적으로 특정 작업에 대한 액세스를 허용하며 Amazon EC2 인스턴스, Amazon S3 buckets 버킷과 같은 특정 리소스에 대한 작업을 허용하도록 선택적으로 권한을 부여할 수 있습니다. 정책은 액세스를 명시적으로 거부할 수도 있습니다. 권한은 특정 리소스에 대한 액세스를 허용하거나 거부하는 정책 내 문입니다. 권한을 다음과 같이 설명할 수

있습니다. "A는 C에 대해 B를 수행할 권한이 있습니다." 예를 들어 Jane(A)은 John의 Amazon Simple Queue Service 대기열(C)에서 메시지를 읽을(B) 권한이 있습니다. Jane이 John의 대기열을 사용하도록 Amazon SQS에 요청을 보낼 때마다 서비스에서는 Jane에게 권한이 있는지 확인합니다. 또한 요청이 John이 권한에 지정된 조건을 충족하는지 여부도 확인합니다.

> **Important**
>
> 생성하는 모든 사용자는 자신의 자격 증명을 사용하여 인증합니다. 그러나 이들은 동일한 AWS 계정을 사용합니다. 사용자에 의한 모든 변경은 전체 계정에 영향을 줄 수 있습니다.

AWS Marketplace 에는 해당 권한이 있는 사용자가 수행할 수 있는 작업을 제어하도록 정의된 권한이 있습니다 AWS Marketplace Management Portal. 또한 여러 권한을 결합하는 정책을 AWS Marketplace 생성하고 관리합니다.

다음 리소스에는 IAM을 시작하고 사용하는 방법이 자세히 나옵니다.

- 관리 사용자 생성
- IAM의 보안 모범 사례
- IAM 정책 관리
- 정책을 IAM 사용자 그룹에 연결
- IAM 자격 증명(사용자, 그룹, 및 역할)
- 정책을 사용하여 AWS 리소스에 대한 액세스 제어

다음 주제는 사용자와 그룹을 생성하고 사용자로 로그인하는 방법에 대한 간략한 지침을 제공합니다.

주제

- 사용자 생성
- 그룹 생성 또는 사용
- 사용자로 로그인

사용자 생성

회사 직원이에 로그인하도록 허용하려면 액세스 권한이 필요한 각 사용자에 대해 사용자를 AWS Marketplace Management Portal생성합니다.

사용자를 만들려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. 탐색 창의 액세스 관리에서 사용자를 선택한 다음 역할 생성을 선택합니다.
3. 번호가 매겨진 텍스트 상자에 만들고자 하는 각 사용자의 이름을 입력합니다.
4. Generate an access key for each user(각 사용자에 대한 액세스 키 생성) 확인란의 선택을 취소한 다음, 생성을 선택합니다.

방금 생성한 각 사용자에게 암호를 할당하려면

1. 사용자 목록에서 새로운 사용자의 이름을 선택합니다.
2. Security Credentials(보안 자격 증명) 탭을 선택한 후 Manage Password(암호 관리)를 선택합니다.
3. 자동으로 생성된 암호 또는 사용자 지정 암호 옵션을 선택합니다. 선택적으로, 사용자가 다음에 로그인할 때 새 암호를 선택하도록 요구하려면 Require user to create a new password at next sign-in(사용자가 다음에 로그인할 때 새 암호를 생성하도록 요청) 확인란을 선택합니다. 적용을 선택합니다.
4. 로그인 보안 인증 정보 및 계정별 로그인 URL을 쉼표로 구분된 값(CSV) 파일로 컴퓨터에 저장하려면 자격 증명 다운로드를 선택합니다. 그런 다음 닫기를 선택합니다.

> **Note**
>
> 방금 생성한 로그인 보안 인증 정보로 로그인하려면 사용자가 계정별 로그인 URL로 이동해야 합니다. 이 URL은 방금 다운로드한 보안 인증 정보 파일에 있으며 콘솔에서도 찾을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자가 AWS 계정에 로그인하는 방법을 참조하세요.

> **Tip**
>
> AWS 계정 소유자라도 로그인 보안 인증 정보를 직접 생성할 수 있습니다. 계정 소유자를 비롯한 모든 사용자가 AWS Marketplace 에서 사용자로 작업하는 것이 가장 좋습니다. 관리자 권한이 있는 사용자를 직접 생성하는 방법에 대한 지침은 IAM 사용 설명서의 관리 사용자 생성을 참조하세요.

그룹 생성 또는 사용

사용자를 만든 후에는 그룹을 만들고, AWS Marketplace Management Portal페이지에 액세스할 수 있는 권한을 만들고, 해당 권한을 그룹에 추가한 다음 사용자를 해당 그룹에 추가합니다.

그룹에 권한을 할당하면 해당 그룹의 구성원이 특정 작업을 수행할 수 있습니다. 그룹에 새 사용자를 추가하면 해당 사용자는 자동으로 그룹에 할당된 권한을 얻습니다. 그룹은 둘 이상의 작업에 대한 권한을 가질 수 있습니다. 자체 정책을 생성하지 말고 AWS Marketplace 관리형 정책을 사용하는 것이 좋습니다.

그룹에에 대한 관리형 정책을 할당 AWS Marketplace 하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹을 선택한 후 정책을 연결할 그룹을 선택합니다.
3. 그룹 요약 페이지의 Permission(권한) 탭에서 Attach Policy(정책 연결)을 선택합니다.
4. Attach Policy(정책 연결) 페이지의 Filter:(필터:) 옆에 awsmarketplace을 입력합니다.
5. 연결할 정책을 선택한 다음 정책 연결을 선택합니다.

AWS Marketplace Management Portal 권한을 사용하여 정책을 생성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
3. [Policy Generator] 옆의 [Select]를 선택합니다.
4. 권한 편집 페이지에서 다음을 수행합니다.
   a. [Effect]에서 [Allow]를 선택합니다.
   b. AWS 서비스에 대해 AWS Marketplace Management Portal를 선택합니다.
   c. 작업에서 허용할 권한을 선택합니다.
   d. 설명문 추가를 선택합니다.
   e. 다음 단계를 선택합니다.
5. 정책 검토 페이지에서 다음과 같이 합니다.
   a. 정책 이름에 이 정책의 이름을 입력합니다. 이후 단계에서 정책 이름이 필요하므로 정책 이름을 기록해 둡니다.
   b. (선택 사항) 설명에 이 정책에 대한 설명을 입력합니다.

c. 정책 생성(Create Policy)을 선택합니다.

적절한 권한이 있는 IAM 그룹을 생성하고 해당 그룹에 사용자를 추가하는 방법

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 그룹을 선택한 다음, 새 그룹 생성을 선택합니다.
3. 그룹 이름:에 그룹의 이름을 입력합니다. 그런 다음 [Next Step]을 선택합니다.
4. 정책 연결 페이지에서 다음 작업을 수행합니다.
   a. 필터:에서 Customer Managed Policies(고객 관리형 정책)를 선택합니다.
   b. 이 그룹에 연결하려는 정책의 이름 옆에 있는 확인란을 선택합니다. 이는 일반적으로 방금 만든 정책입니다.
   c. 다음 단계를 선택합니다.
5. 그룹 생성을 선택합니다.
6. 그룹 목록에서 새 그룹을 찾은 다음 옆에 있는 확인란을 선택합니다. Group Actions(그룹 작업), Add Users to Group(그룹에 사용자 추가)을 선택합니다.
7. 그룹에 추가할 각 사용자 옆에 있는 확인란을 선택한 다음 사용자 추가를 선택합니다.

사용자로 로그인

IAM에서 사용자를 생성한 후, 해당 사용자는 자신의 로그인 보안 인증 정보로 로그인할 수 있습니다. 로그인하려면 AWS 계정과 연결된 고유의 URL을 사용해야 합니다. 로그인 URL을 가져와서 사용자에게 배포할 수 있습니다.

계정의 고유한 로그인 URL을 가져오려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. 콘텐츠 창의 맨 위 근처에서 IAM 사용자 로그인 링크:를 찾아 다음과 같은 형식의 로그인 링크를 메모합니다.

```
https://AWS_account_ID.signin.aws.amazon.com/console/
```

> **Note**
>
> 로그인 페이지의 URL에 AWS 계정 ID 대신 회사 이름(또는 기타 친숙한 식별자)을 포함시키려면 사용자 지정을 선택하여 계정의 별칭을 생성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 계정 ID 및 별칭을 참조하세요.

4. 이 URL을 함께 작업할 수 있는 회사 직원에게 각각에 대해 생성한 로그인 자격 증명과 AWS Marketplace함께 배포합니다. AWS Marketplace에 액세스하기 전에 계정의 고유한 로그인 URL을 사용하여 로그인하라고 안내합니다.

AWS Marketplace 판매자를 위한 정책 및 권한

AWS Marketplace 는와 함께 사용할 수 있는 관리형 정책 세트를 제공합니다 AWS Marketplace Management Portal. 또한 개별 권한을 사용하여 자체 AWS Identity and Access Management (IAM) 정책을 생성할 수 있습니다.

설정, 문의처, 파일 업로드 및 인사이트 탭에 AWS Marketplace Management Portal 대해에 대한 세분화된 액세스를 제공할 수도 있습니다. 세분화된 액세스를 통해 다음을 수행할 수 있습니다.

- 암호나 액세스 키를 공유하지 않고도 AWS 계정 의 리소스를 관리하고 사용할 수 있는 권한을 다른 사람에게 부여합니다.
- 여러 사람에게 다양한 리소스에 대한 세분화된 권한을 부여합니다. 예를 들어 어떤 사용자에게는 AWS Marketplace Management Portal에서 설정 탭을 볼 수 있는 액세스 권한을 부여합니다. 또 어떤 사용자에게는 설정 및 AWS에 문의 탭에서 편집할 수 있는 액세스 권한을 부여합니다.

> **Note**
>
> 데이터 제품에 대한 AWS Data Exchange의 정책 및 권한에 대한 자세한 내용은 AWS Data Exchange 사용 설명서의 AWS Data Exchange의 자격 증명 및 액세스 관리를 참조하세요. AWS Marketplace 구매자의 정책 및 권한에 대한 자세한 내용은 AWS Marketplace 구매자 안내서의 AWS Marketplace 구독에 대한 액세스 제어를 참조하세요.

AWS Marketplace 판매자를 위한 정책

다음 관리형 정책을 사용하여 사용자에게 AWS Marketplace Management Portal에 대한 통제된 액세스 권한을 제공할 수 있습니다.

`AWSMarketplaceSellerFullAccess`

Amazon Machine Image(AMI) 관리와 같은 AWS Marketplace Management Portal 및 기타 AWS 서비스의 모든 페이지에 대한 전체 액세스를 허용합니다.

`AWSMarketplaceSellerProductsFullAccess`

AWS Marketplace Management Portal의 제품 페이지에 대한 전체 액세스 권한을 허용합니다.

`AWSMarketplaceSellerProductsReadOnly`

AWS Marketplace Management Portal의 제품 페이지에 대한 읽기 전용 액세스 권한을 허용합니다.

> **Important**
>
> AWS Marketplace 구매자는 관리형 정책을 사용하여 구매한 구독을 관리할 수 있습니다. 에서 사용하는 관리형 정책의 이름은 로 AWS Marketplace Management Portal 시작합니다`AWSMarketplaceSeller`. IAM에서 정책을 검색할 때, `AWSMarketplaceSeller`로 시작하는 정책을 검색해야 합니다. 이러한 정책에 대한 자세한 내용은 AWS 관리형 정책 참조를 참조하세요.

AWS Marketplace 는 특정 시나리오에 대한 특수 관리형 정책도 제공합니다. AWS Marketplace 판매자를 위한 AWS 관리형 정책의 전체 목록과 판매자가 제공하는 권한에 대한 설명은 섹션을 참조하세요AWSAWS Marketplace 판매자를 위한 관리형 정책.

AWS Marketplace 판매자에 대한 권한

AWS Marketplace Management Portal의 정책에서 다음 권한을 사용할 수 있습니다.

`aws-marketplace-management:PutSellerVerificationDetails`

고객확인제도(KYC) 프로세스를 시작할 수 있는 액세스 권한을 허용합니다.

`aws-marketplace-management:GetSellerVerificationDetails`

AWS Marketplace Management Portal에서 KYC 상태를 볼 수 있는 액세스 권한을 허용합니다.

aws-marketplace-management:PutBankAccountVerificationDetails

은행 계좌 확인 프로세스를 시작할 수 있는 액세스 권한을 허용합니다.

aws-marketplace-management:GetBankAccountVerificationDetails

AWS Marketplace Management Portal에서 은행 계좌 확인 상태를 볼 수 있는 액세스 권한을 허용합니다.

aws-marketplace-management:PutSecondaryUserVerificationDetails

에서 보조 사용자를 추가할 수 있는 액세스를 허용합니다 AWS Marketplace Management Portal.

aws-marketplace-management:GetSecondaryUserVerificationDetails

AWS Marketplace Management Portal에서 보조 사용자 상태를 볼 수 있는 액세스 권한을 허용합니다.

aws-marketplace-management:GetAdditionalSellerNotificationRecipients

AWS Marketplace 알림에 대한 이메일 연락처를 볼 수 있는 액세스를 허용합니다.

aws-marketplace-management:PutAdditionalSellerNotificationRecipients

AWS Marketplace 알림에 대한 이메일 연락처를 업데이트할 수 있는 액세스를 허용합니다.

tax:PutTaxInterview

AWS Marketplace Management Portal에서 세금 인터뷰에 참여할 수 있는 액세스 권한을 허용합니다.

tax:GetTaxInterview

AWS Marketplace Management Portal에서 세금 인터뷰 상태를 볼 수 있는 액세스 권한을 허용합니다.

tax:GetTaxInfoReportingDocument

AWS Marketplace 판매자가 세금 대시보드에서 세금 문서(예: 1099-K 양식)를 보고 다운로드할 수 있도록 허용

payments:CreatePaymentInstrument

에 은행 계좌를 추가할 수 있는 액세스를 허용합니다 AWS Marketplace Management Portal.

payments:GetPaymentInstrument

에서 기존 은행 계좌를 볼 수 있는 액세스를 허용합니다 AWS Marketplace Management Portal.

`support:CreateCase`

내에서 AWS Marketplace 사례를 생성할 수 있는 액세스를 허용합니다 AWS Marketplace Management Portal.

`aws-marketplace-management:viewSupport`

AWS Marketplace Management Portal의 고객 지원 자격 요건 페이지에 액세스할 수 있는 권한을 허용합니다.

`aws-marketplace-management:viewReports`

AWS Marketplace Management Portal의 보고서 페이지에 액세스할 수 있는 권한을 허용합니다.

`aws-marketplace:ListEntities`

에 객체를 나열할 수 있는 액세스를 허용합니다 AWS Marketplace Management Portal. AWS Marketplace Management Portal의 파일 업로드, 제안 및 파트너 페이지에 액세스하려면 이 권한이 필요합니다.

> **Note**
>
> 설정 탭을 볼 수 있는 액세스 권한을 허용하려면 이 권한, `ListEntity` 권한 및 Amazon 리소스 이름(ARN) `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`를 사용하면 됩니다.

`aws-marketplace:DescribeEntity`

에서 객체의 세부 정보를 볼 수 있는 액세스를 허용합니다 AWS Marketplace Management Portal. AWS Marketplace Management Portal의 파일 업로드, 제안, 파트너 및 계약서 페이지에 액세스하려면 이 권한이 필요합니다.

> **Note**
>
> 설정 탭을 볼 수 있는 액세스 권한을 허용하려면 이 권한, `DescribeEntity` 권한 및 ARN `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/*`를 사용하면 됩니다.

aws-marketplace:StartChangeSet

에서 제품 변경 사항을 생성할 수 있는 액세스를 허용합니다 AWS Marketplace Management Portal. AWS Marketplace Management Portal의 파일 업로드, 제안, 파트너 및 계약서 페이지에서 변경 작업을 수행하려면 이 권한이 필요합니다.

> **Note**
>
> 에 판매자로 등록할 수 있는 액세스를 허용하려면이 권한 AWS Marketplace, `catalog:ChangeType: "CreateSeller"` 조건 키 및 ARN을 사용할 수 있습니다`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`.
> 에서 판매자 프로필을 업데이트할 수 있는 액세스를 허용하려면이 권한 AWS Marketplace, `catalog:ChangeType: "UpdateInformation"` 조건 키 및 ARN을 사용할 수 있습니다`arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`.
> Amazon Web Services의 결제 기본 설정을 업데이트할 수 있는 액세스 권한을 허용하려면 이 권한, `catalog:ChangeType: "UpdateDisbursementPreferences"` 조건 키 및 ARN `arn:{partition}:{aws-marketplace}:{region}:{account-id}:AWSMarketplace/Seller/{entity-id}`를 사용하면 됩니다.

aws-marketplace:SearchAgreements

계약 페이지에서 상위 수준의 계약서 목록을, 파트너 페이지에서 ISV와 채널 파트너 간의 영업 기회를 볼 수 있는 액세스 권한을 허용합니다.

aws-marketplace:DescribeAgreement

계약 페이지에서 상위 수준 계약서 세부 정보를, 파트너 페이지에서 ISV와 채널 파트너 간의 영업 기회를 볼 수 있는 액세스 권한을 허용합니다.

aws-marketplace:GetAgreementTerms

계약서 페이지에서 모든 계약 조건 세부 정보를, 파트너 페이지에서 ISV와 채널 파트너 간의 영업 기회를 볼 수 있는 액세스 권한을 허용합니다.

aws-marketplace:GetSellerDashboard

AWS Marketplace Management Portal의 인사이트 페이지에서 대시보드에 액세스할 수 있는 권한을 허용합니다.

aws-marketplace:ListAssessments

판매자 작업이 보류 중인 평가 목록을 볼 수 있는 액세스를 허용합니다.

aws-marketplace:DescribeAssessment

판매자 작업 보류 중인 평가의 세부 정보를 볼 수 있는 액세스를 허용합니다.

> **Note**
>
> 사용자가 제품 관리 페이지에 액세스할 수 있도록 하려면 `AWSMarketplaceSellerProductsFullAccess` 또는 `AWSMarketplaceSellerProductsReadOnly` 관리형 권한을 사용해야 합니다.

위의 권한을 단일 IAM 정책에 결합하여 원하는 권한을 부여할 수 있습니다. 다음 예시를 참조하세요.

예제 1: KYC 상태를 볼 수 있는 권한

에서 KYC 상태를 볼 수 있는 권한을 부여하려면 다음 예제와 유사한 정책을 AWS Marketplace Management Portal사용합니다.

에서 KYC 상태를 볼 수 있는 권한을 부여하려면 다음 예제와 유사한 정책을 AWS Marketplace Management Portal사용합니다.

JSON

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

예제 2: 비공개 제안에 대한 업그레이드 및 갱신을 생성할 수 있는 권한

계약서 페이지를 살펴보고 사용하여 비공개 제안을 업그레이드 및 갱신할 수 있는 권한을 부여하려면 다음 예와 비슷한 정책을 사용합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aws-marketplace:SearchAgreements",
                "aws-marketplace:DescribeAgreement",
                "aws-marketplace:GetAgreementTerms",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:StartChangeSet"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws-marketplace:PartyType": "Proposer"
                },
                "ForAllValues:StringEquals": {
                    "aws-marketplace:AgreementType": [
                        "PurchaseAgreement"
                    ]
                }
            }
        }
    ]
}
```

예제 3: 제안 페이지에 액세스하고 새 비공개 제안을 생성할 수 있는 권한

제안 페이지를 살펴보고 사용하여 기존의 비공개 제안을 살펴볼 수 있는 권한을 부여하려면 다음 예제와 비슷한 정책을 사용합니다.

JSON

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

예제 4: 설정 페이지에 액세스할 수 있는 권한

설정 페이지를 살펴보고 사용할 수 있는 권한을 부여하려면 다음 예제와 비슷한 정책을 사용합니다.

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

예제 5: 파일 업로드 페이지에 액세스할 수 있는 권한

파일 업로드 페이지를 살펴보고 사용할 수 있는 권한을 부여하려면 다음 예제와 비슷한 정책을 사용합니다.

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

IAM 그룹 사용

또는 별도의 IAM 그룹을 생성하여 AWS Marketplace Management Portal의 개별 페이지에 대한 액세스 권한을 부여할 수도 있습니다. 사용자는 둘 이상의 그룹에 속할 수 있습니다. 따라서 사용자가 둘 이상의 페이지에 액세스해야 할 경우에는 해당 사용자를 해당 그룹 전체에 추가할 수 있습니다. 예를 들어 IAM 그룹을 하나 만들고 해당 그룹에 인사이트 페이지에 대한 액세스 권한을 부여하고, 다른 그룹을 생성하고 해당 그룹에 파일 업로드 페이지에 대한 액세스 권한을 부여하는 방식으로 권한을 부여할 수 있습니다. 사용자가 인사이트 페이지와 파일 업로드 페이지에 모두 액세스할 수 있는 권한이 필요한 경우 사용자를 두 그룹에 모두 추가합니다.

사용자 및 그룹에 대한 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명(사용자, 그룹, 및 역할)을 참조하세요.

AWSAWS Marketplace 판매자를 위한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 미칩니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하세요.

이 섹션에는 AWS Marketplace에 대한 판매자 액세스를 관리하는 데 사용되는 각 정책이 나열됩니다. 구매자 정책에 대한 자세한 내용은 AWS Marketplace 구매자 안내서의 AWSAWS Marketplace 구매자를 위한 관리형 정책을 참조하세요.

주제

- AWS 관리형 정책: AWSMarketplaceAmiIngestion
- AWS 관리형 정책: AWSMarketplaceFullAccess
- AWS 관리형 정책: AWSMarketplaceGetEntitlements
- AWS 관리형 정책: AWSMarketplaceMeteringFullAccess
- AWS 관리형 정책: AWSMarketplaceMeteringRegisterUsage
- AWS 관리형 정책: AWSMarketplaceSellerFullAccess
- AWS 관리형 정책: AWSMarketplaceSellerProductsFullAccess
- AWS 관리형 정책: AWSMarketplaceSellerProductsReadOnly
- AWS 관리형 정책: AWSMarketplaceSellerOfferManagement
- AWS 관리형 정책: AWSVendorInsightsVendorFullAccess
- AWS 관리형 정책: AWSVendorInsightsVendorReadOnly
- AWS MarketplaceAWS 관리형 정책에 대한 업데이트

AWS 관리형 정책: AWSMarketplaceAmiIngestion

에서 사용자를 대신하여 작업을 수행하는 AWS Marketplace 데 사용할 수 있는이 정책을 사용하여 서비스 역할을 생성할 수 있습니다. `AWSMarketplaceAmiIngestion` 사용에 관한 자세한 내용은 AMI에 대한 AWS Marketplace 액세스 권한 부여 부분을 참조하세요.

이 정책은가 Amazon Machine Image(AMIs) AWS Marketplace 를 복사하여 나열할 수 있는 권한을 기고자에게 부여합니다 AWS Marketplace.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceAmiIngestion를 참조하세요.

AWS 관리형 정책: AWSMarketplaceFullAccess

`AWSMarketplaceFullAccess` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 판매자와 구매자 모두 AWS Marketplace 및 관련 서비스에 대한 전체 액세스를 허용하는 관리 권한을 부여합니다. 이러한 권한으로 다음과 같은 작업을 수행할 수 있습니다.

- AWS Marketplace 소프트웨어를 구독 및 구독 취소합니다.
- 에서 AWS Marketplace 소프트웨어 인스턴스를 관리합니다 AWS Marketplace.
- 계정에서 프라이빗 마켓플레이스를 생성하고 관리합니다.
- Amazon EC2 AWS CloudFormation및 Amazon EC2 Systems Manager에 대한 액세스 권한을 제공합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceFullAccess를 참조하세요.

AWS 관리형 정책: AWSMarketplaceGetEntitlements

`AWSMarketplaceGetEntitlements` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 서비스형 소프트웨어(SaaS) 제품 판매자가 고객이 AWS Marketplace SaaS 제품을 구독했는지 확인할 수 있는 읽기 전용 권한을 부여합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceGetEntitlements를 참조하세요.

AWS 관리형 정책: AWSMarketplaceMeteringFullAccess

`AWSMarketplaceMeteringFullAccess` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 AWS Marketplace의 유연한 소비 요금이 적용되는 AMI 및 컨테이너 제품에 해당하는 측정 사용량을 보고할 수 있는 기여자 권한을 부여합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceMeteringFullAccess를 참조하세요.

AWS 관리형 정책: AWSMarketplaceMeteringRegisterUsage

`AWSMarketplaceMeteringRegisterUsage` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 시간당 요금이 적용되는 컨테이너 제품에 해당하는 측정된 사용량을 보고할 수 있는 권한을 기여자에게 부여합니다 AWS Marketplace.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceMeteringRegisterUsage를 참조하세요.

AWS 관리형 정책: AWSMarketplaceSellerFullAccess

`AWSMarketplaceSellerFullAccess` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 AMI 기반 제품에 사용되는 Amazon EC2 AMI를 AWS Marketplace포함하여 AWS Marketplace Management Portal의 모든 판매자 작업에 대한 전체 액세스를 허용하는 관리 권한을 부여합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MarketplaceManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace-management:uploadFiles",
                "aws-marketplace-management:viewReports",
                "aws-marketplace-management:viewSupport",
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:StartChangeSet",
                "aws-marketplace:CancelChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:GetSellerDashboard",
                "aws-marketplace:ListAssessments",
                "aws-marketplace:DescribeAssessment",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots",
                "ec2:ModifyImageAttribute",
                "ec2:ModifySnapshotAttribute"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "AgreementAccess",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:SearchAgreements",
                "aws-marketplace:DescribeAgreement",
                "aws-marketplace:GetAgreementTerms"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws-marketplace:PartyType": "Proposer"
                },
                "ForAllValues:StringEquals": {
                    "aws-marketplace:AgreementType": [
                        "PurchaseAgreement"
                    ]
                }
            }
        },
        {
            "Sid": "IAMGetRole",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Sid": "AssetScanning",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "assets.marketplace.amazonaws.com"
                }
            }
        },
        {
            "Sid": "VendorInsights",
            "Effect": "Allow",
```

```
            "Action": [
                "vendor-insights:GetDataSource",
                "vendor-insights:ListDataSources",
                "vendor-insights:ListSecurityProfiles",
                "vendor-insights:GetSecurityProfile",
                "vendor-insights:GetSecurityProfileSnapshot",
                "vendor-insights:ListSecurityProfileSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TagManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:TagResource",
                "aws-marketplace:UntagResource",
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Sid": "SellerSettings",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace-management:GetSellerVerificationDetails",
                "aws-marketplace-management:PutSellerVerificationDetails",
                "aws-marketplace-management:GetBankAccountVerificationDetails",
                "aws-marketplace-management:PutBankAccountVerificationDetails",
                "aws-marketplace-management:GetSecondaryUserVerificationDetails",
                "aws-marketplace-management:PutSecondaryUserVerificationDetails",
                "aws-marketplace-
management:GetAdditionalSellerNotificationRecipients",
                "aws-marketplace-
management:PutAdditionalSellerNotificationRecipients",
                "payments:GetPaymentInstrument",
                "payments:CreatePaymentInstrument",
                "tax:GetTaxInterview",
                "tax:PutTaxInterview",
                "tax:GetTaxInfoReportingDocument",
                "tax:ListSupplementalTaxRegistrations",
                "tax:PutSupplementalTaxRegistration",
                "tax:DeleteSupplementalTaxRegistration",
                "tax:GetTaxRegistration"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "Support",
            "Effect": "Allow",
            "Action": [
                "support:CreateCase"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ResourcePolicyManagement",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:GetResourcePolicy",
                "aws-marketplace:PutResourcePolicy",
                "aws-marketplace:DeleteResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "resale-
authorization.marketplace.amazonaws.com"
                }
            }
        }
    ]
}
```

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceSellerFullAccess를 참조하세요.

AWS 관리형 정책: AWSMarketplaceSellerProductsFullAccess

`AWSMarketplaceSellerProductsFullAccess` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 기고자에게 제품 및에 대한 전체 액세스 권한을 허용 AWS Marketplace Management Portal 하고 AMI 기반 제품에 사용되는 Amazon EC2 AMI를 관리할 수 있는 권한을 부여합니다.

권한 세부 정보

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:StartChangeSet",
                "aws-marketplace:CancelChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:ListAssessments",
                "aws-marketplace:DescribeAssessment",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots",
                "ec2:ModifyImageAttribute",
                "ec2:ModifySnapshotAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
```

```
                    "iam:PassedToService": "assets.marketplace.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "vendor-insights:GetDataSource",
                "vendor-insights:ListDataSources",
                "vendor-insights:ListSecurityProfiles",
                "vendor-insights:GetSecurityProfile",
                "vendor-insights:GetSecurityProfileSnapshot",
                "vendor-insights:ListSecurityProfileSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:TagResource",
                "aws-marketplace:UntagResource",
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:GetResourcePolicy",
                "aws-marketplace:PutResourcePolicy",
                "aws-marketplace:DeleteResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        }
    ]
}
```

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceSellerProductsFullAccess를 참조하세요.

AWS 관리형 정책: AWSMarketplaceSellerProductsReadOnly

`AWSMarketplaceSellerProductsReadOnly` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 AWS Marketplace Management Portal의 제품을 보고 AMI 기반 제품에 사용되는 Amazon EC2 AMI를 볼 수 있는 읽기 전용 권한을 부여합니다.

권한 세부 정보

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListChangeSets",
                "aws-marketplace:DescribeChangeSet",
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity",
                "aws-marketplace:ListAssessments",
                "aws-marketplace:DescribeAssessment",
                "ec2:DescribeImages",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListTagsForResource"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:GetResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/*"
```

```
        }
    ]
}
```

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceSellerProductsReadOnly를 참조하세요.

AWS 관리형 정책: AWSMarketplaceSellerOfferManagement

`AWSMarketplaceSellerOfferManagement` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 판매자에게 제안 및 계약 관리 활동에 대한 액세스 권한을 부여합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSMarketplaceSellerOfferManagement를 참조하세요.

AWS 관리형 정책: AWSVendorInsightsVendorFullAccess

`AWSVendorInsightsVendorFullAccess` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 AWS Marketplace Vendor Insights에서 모든 리소스를 생성하고 관리할 수 있는 전체 액세스 권한을 부여합니다. AWS Marketplace Vendor Insights에서 평가자는 구매자와 동일하고, 공급업체는 이 가이드의 목적상 판매자와 동일합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSVendorInsightsVendorFullAccess를 참조하세요.

AWS 관리형 정책: AWSVendorInsightsVendorReadOnly

`AWSVendorInsightsVendorReadOnly` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 AWS Marketplace Vendor Insights 프로필 및 관련 리소스를 볼 수 있는 읽기 전용 액세스 권한을 부여합니다. AWS Marketplace Vendor Insights에서 평가자는 구매자와 동일하고, 공급업체는이 가이드의 목적상 판매자와 동일합니다.

이 정책의 권한을 보려면AWS 관리형 정책 참조의 AWSVendorInsightsVendorReadOnly를 참조하세요.

AWS MarketplaceAWS 관리형 정책에 대한 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 AWS Marketplace 이후부터의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 AWS Marketplace 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경 사항	설명	날짜
AWSMarketplaceSellerFullAccess - 업데이트된 정책	AWS Marketplace 는 보충 세금 프로필 기능에 대해 `ListSupplementalTaxRegistrations` , , `PutSupplementalTaxRegistration` , `DeleteSupplementalTaxRegistration` 라는 네 가지 새로운 `SellerSettings` 권한을 추가했습니다`GetTaxRegistration` .	2024년 12월 20일
AWSMarketplaceSellerFullAccess - 업데이트된 정책 AWSMarketplaceSellerProductsFullAccess - 업데이트된 정책 AWSMarketplaceSellerFullAccess - 업데이트된 정책 AWSMarketplaceSellerProductsReadOnly - 업데이트된 정책	AWS Marketplace 는 `ListTasks` , `DescribeTask` `UpdateTasks` , 및 `CompleteTasks` 권한을 제거했습니다.	2024년 12월 10일
AWSMarketplaceSellerOfferManagement - 새 정책 추가	AWS Marketplace 에 새 정책이 추가되었습니다. `AWSMarketplaceSellerOfferManagement`	2024년 11월 18일

변경 사항	설명	날짜
AWSMarketplaceSellerFullAccess - 업데이트된 정책	AWS Marketplace 에서 `UploadFiles` 권한을 추가했습니다. 변경 사항을 통해 판매자는에서 더 이상 사용되지 않는 페이지를 사용할 수 있습니다 AWS Marketplace Management Portal.	2024년 11월 6일
AWSMarketplaceSellerFullAccess - 업데이트된 정책	AWS Marketplace 에 `ListAssessments` 및 `DescribeAssessments` 권한이 추가되었습니다. 변경 사항을 통해 SSLv2 사용자는 평가 데이터에 액세스할 수 있습니다.	2024년 10월 22일
AWSMarketplaceSellerProductsFullAccess - 업데이트된 정책	AWS Marketplace 에 `ListAssessments` 및 `DescribeAssessments` 권한이 추가되었습니다. 변경 사항을 통해 SSLv2 사용자는 평가 데이터에 액세스할 수 있습니다.	2024년 10월 22일
AWSMarketplaceSellerProductsReadOnly - 업데이트된 정책	AWS Marketplace 에 `ListAssessments` 및 `DescribeAssessments` 권한이 추가되었습니다. 변경 사항을 통해 SSLv2 사용자는 평가 데이터에 액세스할 수 있습니다.	2024년 10월 22일

변경 사항	설명	날짜
AWSMarketplaceSellerFullAccess - 정책 업데이트	`aws-marketplace-management:viewMarketing` , `aws-marketplace-management:viewSettings` 및 `aws-marketplace-management:uploadFiles` 작업의 제거를 반영하도록 `AWSMarketplaceSellerFullAccess` 설명서를 업데이트했습니다. 이 업데이트에는 세분화된 권한 사용 섹션 제거도 포함됩니다.	2024년 6월 4일
AWSMarketplaceGetEntitlements - 정책 업데이트	AWS Marketplace 정책 설명`sid`에를 추가`AWSMarketplaceGetEntitlements` 하도록 업데이트되었습니다.	2024년 3월 22일
AWSMarketplaceSellerFullAccess - 정책 업데이트	AWS Marketplace 서비스 연결 역할을 생성할 수 있는 권한을 추가`AWSMarketplaceSellerFullAccess` 하도록 업데이트되었습니다.	2024년 3월 15일
AWSMarketplaceSellerFullAccess - 정책 업데이트	AWS Marketplace 세금 정보에 액세스할 수 있는 권한을 추가`AWSMarketplaceSellerFullAccess` 하도록 업데이트되었습니다.	2024년 2월 8일

변경 사항	설명	날짜
AWSVendorInsightsVendorFullAccess - 정책 업데이트	AWS Marketplace 데이터 소스를 업데이트할 수 있는 권한을 추가`AWSVendorInsightsVendorFullAccess` 하도록 가 업데이트되었습니다.	2023년 10월 18일
AWSMarketplaceSellerFullAccess - 정책 업데이트	AWS Marketplace 가 엔터티 공유 권한을 추가`AWSMarketplaceSellerFullAccess` 하도록 업데이트되었습니다.	2023년 6월 1일
AWSMarketplaceSellerFullAccess - 정책 업데이트	AWS Marketplace 계정 확인, 은행 계좌 확인, 사례 관리 및 판매자 알림 세부 정보와 관련된 권한을 추가`AWSMarketplaceSellerFullAccess` 하도록 업데이트되었습니다.	2023년 6월 1일
AWSMarketplaceSellerFullAccess - 정책 업데이트	AWS Marketplace 판매자 대시보드에 액세스할 수 있는 권한을 추가`AWSMarketplaceSellerFullAccess` 하도록 업데이트되었습니다.	2022년 12월 23일
AWSMarketplaceSellerFullAccess, AWSMarketplaceSellerProductsFullAccess, AWSMarketplaceSellerProductsReadOnly – 기존 정책의 업데이트	AWS Marketplace 는 새로운 태그 기반 권한 부여 기능에 대한 정책을 업데이트했습니다.	2022년 12월 9일

변경 사항	설명	날짜
AWS Marketplace 업데이트됨 AWSVendorInsightsVendorFullAccess	AWS Marketplace `AWSMarketplaceSellerProductsFullAccess` 는 계약 검색, 프로필 스냅샷 업데이트, 공급업체 태그 지정을 추가하고 AWS Artifact 타사 보고서에 대한 읽기 전용 액세스(미리 보기)를 허용하도록 업데이트되었습니다.	2022년 11월 30일
AWS Marketplace 업데이트됨 AWSVendorInsightsVendorReadOnly	AWS Marketplace 태그를 나열할 수 있는 권한을 추가하고 타사 보고서(미리 보기)에 대한 AWS Artifact 읽기 전용 액세스를 허용`AWSVendorInsightsVendorReadOnly` 하도록 업데이트되었습니다.	2022년 11월 30일
AWSVendorInsightsVendorFullAccess 및 AWSVendorInsightsVendorReadOnly – 새 정책 추가	AWS Marketplace 에 새 기능 AWS Marketplace Vendor Insights: `AWSMarketplaceSellerProductsFullAccess` 및에 대한 정책이 추가되었습니다`AWSVendorInsightsVendorReadOnly` .	2022년 7월 26일

변경 사항	설명	날짜
AWSMarketplaceSellerProductsFullAccess 및 AWSMarketplaceSellerFullAccess- 정책 업데이트	AWS Marketplace AWS Marketplace Vendor Insights 의 새로운 기능인 `AWSMarketplaceSellerProductsFullAccess` 및에 대한 정책이 업데이트되었습니다 `AWSMarketplaceSellerFullAccess` .	2022년 7월 26일
AWSMarketplaceSellerFullAccess 및 AWSMarketplaceSellerProductsFullAccess - 기존 정책 업데이트	AWS Marketplace 는 `iam:PassedToService` 조건이 에만 적용되도록 정책을 업데이트했습니다`iam:PassRole` .	2021년 11월 22일
AWSMarketplaceFullAccess - 기존 정책 업데이트	AWS Marketplace 가 `AWSMarketplaceFullAccess` 정책에서 중복 `ec2:DescribeAccountAttributes` 권한을 제거했습니다.	2021년 7월 20일
AWS Marketplace 에서 변경 내용 추적 시작	AWS Marketplace 가 AWS 관리형 정책에 대한 변경 내용 추적을 시작했습니다.	2021년 4월 20일

AWS Marketplace 상거래 분석 서비스 계정 권한

다음 IAM 권한 정책을 사용하여 AWS Marketplace 상거래 분석 서비스에 등록합니다.

등록 방법에 대한 지침은 온보딩 가이드를 따릅니다.

JSON

```
{
```

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

다음 IAM 권한 정책을 사용하여 사용자가 AWS Marketplace 상거래 분석 서비스에 대한 요청을 생성할 수 있도록 허용합니다.

JSON

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

이 기능에 대한 자세한 내용은 AWS Marketplace Commerce Analytics Service를 사용하면 프로그래밍 방식으로 제품 및 고객 데이터에 액세스할 수 있습니다. 섹션을 참조하세요.

Amazon SQS 권한

SaaS 제품 게시 프로세스의 일환으로는 고객의 구독 또는 권한 상태가 변경될 경우 알림을 수신하는 데 사용할 수 있는 Amazon SNS 주제를 AWS Marketplace 제공합니다. 대기열이 알림에 대한 조치를

취할 수 있도록 주제에 대한 하나 이상의 Amazon SQS 대기열을 구성할 수 있습니다. 예를 들어 고객이 SaaS 제품에 대해 보유한 구독에 더 많은 스토리지를 추가하는 경우, Amazon SNS 주제는 Amazon SQS 대기열에 해당 고객이 사용할 수 있는 스토리지 용량을 자동으로 증가시키는 프로세스를 시작하는 메시지를 보낼 수 있습니다.

Amazon Simple Queue Service(Amazon SQS) 대기열을 제공된 Amazon SNS 주제에 구독하면 주제가 대기열에 메시지를 게시할 수 있는 권한이 자동으로 추가됩니다. 그러나 AWS Marketplace 측정 및 권한 부여 서비스 API 사용자에게 대기열에 대한 액세스 권한을 부여하려면 여전히 IAM 정책이 필요합니다. 동일한 자격 증명으로 서비스가 실행될 경우 이러한 사항이 동일한 사용자에 적용될 수 있습니다. 다음 콘텐츠가 있는 정책을 생성하고 사용자 또는 역할에 연결합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:ReceiveMessage", "sqs:DeleteMessage",
 "sqs:GetQueueAttributes", "sqs:GetQueueUrl"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:sqs:REGION_HERE:XXXXXXXXXXXX:NAME_HERE"
        }
    ]
}
```

> **Note**
>
> `Resource` 필드는 Amazon SQS 대기열의 Amazon 리소스 이름(ARN)입니다.

SaaS 제품의 메시지 알림 및 대기열에 대한 자세한 내용은 the section called “SNS 주제에 대한 SQS 대기열 구독” 및 the section called “AWS Marketplace 측정 및 권한 부여 서비스 API 액세스” 단원을 참조하십시오.

AWS Marketplace 측정 및 권한 부여 API 권한

서비스형 소프트웨어(SaaS) 제품, Amazon Machine Image(AMI) 제품 및 컨테이너 제품은 AWS Marketplace Metering Service 및 AWS Marketplace Entitlement Service APIs. 각 유형은 서로 다른 AWS Identity and Access Management (IAM) 권한이 필요합니다. 제품 또는 제품의 경우 모든 사용량을 측정하며, 제공한 측정 레코드에 AWS 따라에서 고객에게 요금이 청구됩니다. 측정 레코드를 제공하는 데 필요한 통합 AWS Marketplace 을 활성화하려면 통합이 사용 중인 서비스 계정에 액세스를 활성화하는 제한된 IAM 정책이 필요합니다. 측정 정보를 전송하고 있는 제품 유형에 대한 정책을, 사용 중인 사용자 또는 역할에 연결하여 통합할 수 있습니다.

주제

- SaaS 제품에 적용되는 IAM 정책
- AMI 제품에 적용되는 IAM 정책
- 컨테이너 제품에 적용되는 IAM 정책

SaaS 제품에 적용되는 IAM 정책

다음 정책에서는 모든 SaaS 통합에 첫 번째 권한 `aws-marketplace:ResolveCustomer`가 필요합니다. 두 번째 권한인 `aws-marketplace:BatchMeterUsage`는 AWS Marketplace Metering Service API에 필요합니다. 세 번째 권한인 `aws-marketplace:GetEntitlements`는 AWS Marketplace Entitlement Service API에 필요합니다.

JSON

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

SaaS 제품에 대한 자세한 내용은 의 SaaS 기반 제품 AWS Marketplace 섹션을 참조하세요.

AMI 제품에 적용되는 IAM 정책

AMI 제품에는 다음과 같은 IAM 정책을 사용합니다.

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

AMI 제품에 대한 자세한 내용은 의 AMI 기반 제품 AWS Marketplace 섹션을 참조하세요.

컨테이너 제품에 적용되는 IAM 정책

컨테이너 제품에는 다음과 같은 IAM 정책을 사용합니다.

JSON

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

컨테이너 제품에 대한 자세한 내용은 의 컨테이너 기반 제품 AWS Marketplace 섹션을 참조하세요.

사용자 생성에 대한 자세한 내용은 IAM 사용 설명서의 AWS 계정에서 사용자 생성 섹션을 참조하세요. 정책 생성 및 할당에 대한 자세한 내용은 IAM 사용자의 권한 변경을 참조하세요.

이 정책은 해당 정책을 연결하는 IAM 역할 또는 사용자에 대한 API에 액세스할 수 있는 권한을 부여합니다. 이러한 API 호출에 대해 다른 계정에서 역할 가정을 활성화하는 방법에 대한 자세한 내용은 AWS Partner Network (APN) 블로그의 How to Best Architect Your AWS Marketplace SaaS Subscription Across Multiple AWS 계정을 참조하세요.

에서 재판매 권한 부여에 서비스 연결 역할 사용 AWS Marketplace

AWS Marketplace 는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 직접 연결된 고유한 유형의 IAM 역할입니다 AWS Marketplace. 서비스 연결 역할은에서 사전 정의 AWS Marketplace 하며 서비스가 사용자를 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할을 더 AWS Marketplace 쉽게 설정할 수 있습니다.는 서비스 연결 역할의 권한을 AWS Marketplace 정의하며, 달리 정의되지 않은 한 만 해당 역할을 수임 AWS Marketplace 할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔티티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 AWS Marketplace 리소스에 대한 액세스 권한을 실수로 제거할 수 없기 때문에 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 AWS IAM으로 작업하는 서비스를 참조하고 서비스 연결 역할 열에서 예인 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

주제

- 에 대한 서비스 연결 역할 권한 AWS Marketplace
- 에 대한 서비스 연결 역할 생성 AWS Marketplace
- 에 대한 서비스 연결 역할 편집 AWS Marketplace
- 에 대한 서비스 연결 역할 삭제 AWS Marketplace
- AWS Marketplace 서비스 연결 역할에 지원되는 리전

에 대한 서비스 연결 역할 권한 AWS Marketplace

AWS Marketplace 는 AWSServiceRoleForMarketplaceResaleAuthorization이라는 서비스 연결 역할을 사용합니다.이 역할을 사용하면에서 재판매 권한 부여에 사용하거나 관리하는 AWS 서비스 및 리소스 AWS Marketplace 에 액세스할 수 있습니다.

AWSServiceRoleForMarketplaceResaleAuthorization 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `resale-authorization.marketplace.amazonaws.com`

AWSMarketplaceResaleAuthorizationServiceRolePolicy라는 역할 권한 정책은 AWS Marketplace 가 지정된 리소스에 대해 다음 작업을 수행하도록 허용합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "AllowResaleAuthorizationShareActionsRAMCreate",
            "Effect": "Allow",
            "Action": [
                "ram:CreateResourceShare"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ],
            "Condition": {
                "StringEquals": {
                    "ram:RequestedResourceType": "aws-marketplace:Entity"
                },
                "ArnLike": {
                    "ram:ResourceArn": "arn:aws:aws-
marketplace:*:*:AWSMarketplace/ResaleAuthorization/*"
                },
                "Null": {
                    "ram:Principal": "true"
                }
            }
        },
        {
```

```
            "Sid": "AllowResaleAuthorizationShareActionsRAMAssociate",
            "Effect": "Allow",
            "Action": [
                "ram:AssociateResourceShare"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ],
            "Condition": {
                "Null": {
                    "ram:Principal": "false"
                },
                "StringEquals": {
                    "ram:ResourceShareName": "AWSMarketplaceResaleAuthorization"
                }
            }
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsRAMAccept",
            "Effect": "Allow",
            "Action": [
                "ram:AcceptResourceShareInvitation"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ],
            "Condition": {
                "StringEquals": {
                    "ram:ResourceShareName": "AWSMarketplaceResaleAuthorization"
                }
            }
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsRAMGet",
            "Effect": "Allow",
            "Action": [
                "ram:GetResourceShareInvitations",
                "ram:GetResourceShareAssociations"
            ],
            "Resource": [
                "arn:aws:ram:*:*:*"
            ]
        },
        {
```

```
            "Sid": "AllowResaleAuthorizationShareActionsMarketplace",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:PutResourcePolicy",
                "aws-marketplace:GetResourcePolicy"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/
ResaleAuthorization/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": ["ram.amazonaws.com"]
                }
            }
        },
        {
            "Sid": "AllowResaleAuthorizationShareActionsMarketplaceDescribe",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:DescribeEntity"
            ],
            "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/
ResaleAuthorization/*"
        }
    ]
}
```

사용자, 그룹 또는 역할이 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 사용 권한을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

에 대한 서비스 연결 역할 생성 AWS Marketplace

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. 에서 서비스 연결 역할을 선택하면가 서비스 연결 역할을 AWS Marketplace Management Portal AWS Marketplace 생성합니다.

서비스 연결 역할 생성

1. AWS Marketplace Management Portal에서 관리 계정에 로그인하고 설정을 선택합니다.
2. 설정 섹션에서 서비스 연결 역할 탭을 선택합니다.
3. 서비스 연결 역할 페이지에서 재판매 권한 부여에 대한 서비스 연결 역할 또는 재판매 권한 부여 통합을 선택한 다음 서비스 연결 역할 생성 또는 통합 구성을 선택합니다.

4. 재판매 권한 부여에 대한 서비스 연결 역할 또는 재판매 권한 부여 통합 만들기 페이지에서 정보를 검토하고 서비스 연결 역할 생성 또는 통합 생성을 선택하여 확인합니다.

   서비스 연결 역할 페이지에 재판매 권한 부여 서비스 연결 역할이 성공적으로 생성되었음을 나타내는 메시지가 나타납니다.

서비스 연결 역할을 삭제하는 경우 다음 단계에 따라 역할을 다시 생성할 수 있습니다.

에 대한 서비스 연결 역할 편집 AWS Marketplace

AWS Marketplace 에서는 AWSServiceRoleForMarketplaceResaleAuthorization 서비스 연결 역할을 편집할 수 없습니다. 서비스 링크 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

에 대한 서비스 연결 역할 삭제 AWS Marketplace

서비스 연결 역할이 있어야 하는 기능 또는 서비스가 더 이상 필요 없다면 그 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔터티가 없도록 합니다.

> **Note**
>
> 독립 소프트웨어 공급업체(ISVs)에 역할이 없는 경우 AWS Resource Access Manager 는 대상 채널 파트너와 새 재판매 승인을 자동으로 공유하지 않습니다. 채널 파트너에게 역할이 없는 경우 AWS Resource Access Manager 는 해당 파트너를 대상으로 하는 재판매 승인을 자동으로 수락하지 않습니다.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 AWSServiceRoleForMarketplaceResaleAuthorization 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오.

AWS Marketplace 서비스 연결 역할에 지원되는 리전

AWS Marketplace 는 서비스를 사용할 수 있는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 AWS 리전 및 엔드포인트 단원을 참조하세요.

를 사용하여 AWS Marketplace 측정 API 호출 로깅 AWS CloudTrail

AWS Marketplace 는 사용자 AWS CloudTrail, 역할 또는 AWS 서비스 의가 수행한 작업에 대한 레코드를 제공하는 서비스와 통합됩니다 AWS Marketplace. CloudTrail은에 대한 API 호출을 이벤트 AWS Marketplace 로 캡처합니다. 캡처되는 호출에는 AWS Marketplace 콘솔의 호출과 AWS Marketplace API 작업에 대한 코드 호출이 포함됩니다.

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. 지원되는 이벤트 활동이에서 발생하면 AWS Marketplace해당 활동이 이벤트 기록의 다른 이벤트와 함께 CloudTrail AWS 서비스 이벤트에 기록됩니다. 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청했는지 여부
- 다른 AWS 서비스에서 요청했는지 여부.

AWS Marketplace 는 `BatchMeterUsage` 작업을 CloudTrail 로그 파일에 이벤트로 로깅할 수 있도록 지원합니다.

AWS Marketplace 측정 API 로그 파일 항목 예제

예시: **`BatchMeterUsage`**

다음 예제에서는 AWS Marketplace Metering Service의 `BatchMeterUsage` 작업을 시연하는 CloudTrail 로그 항목을 보여줍니다. 판매자가에 나열된 서비스형 소프트웨어(SaaS) 제품에 대한 고객의 사용을 보고하기 위해 측정 레코드를 보내면 AWS Marketplace이 CloudTrail 로그 항목이 판매자의에 기록됩니다 AWS 계정.

```
{
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/*****",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
```

```
                "userName": "*****"
            },
            "eventTime": "2018-04-19T16:32:51Z",
            "eventSource": "metering-marketplace.amazonaws.com",
            "eventName": "BatchMeterUsage",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "************",
            "userAgent": "Coral/Netty14",
            "requestParameters": {
                "usageRecords": [
                    {
                        "dimension": "Dimension1",
                        "timestamp": "Apr 19, 2018 4:32:50 PM",
                        "customerIdentifier": "customer1",
                        "customerAWSAccountID": "987654321098",
                        "quantity": 1
                    }
                ],
                "productCode": "EXAMPLE_proCode"
            },
            "responseElements": {
                "results": [
                    {
                        "usageRecord": {
                            "dimension": "Dimension1",
                            "timestamp": "Apr 19, 2018 4:32:50 PM",
                            "customerIdentifier": "customer1",
                            "customerAWSAccountID": "987654321098",
                            "quantity": 1
                        },
                        "meteringRecordId": "bEXAMPLE-98f0-4e90-8bd2-bf0EXAMPLE1e",
                        "status": "Success"
                    }
                ],
                "unprocessedRecords": [ ]
            },
            "requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
            "eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
            "readOnly": false,
            "eventType": "AwsApiCall",
            "recipientAccountId": "123456789012"
        }
    ]
```

```
}
```

예: 컨테이너의 **RegisterUsage**

다음 예제에서는 AWS Marketplace Metering Service의 `RegisterUsage` 작업을 시연하는 CloudTrail 로그 항목을 보여줍니다. 의 시간당 가격 컨테이너 제품이 구매자의에 AWS Marketplace 배포되면 컨테이너의 AWS 계정소프트웨어는 구매자의 `RegisterUsage` 내에서를 호출 AWS 계정 하여 해당 Amazon Elastic Container Service(Amazon ECS) 작업 또는 Amazon Elastic Kubernetes Service(Amazon EKS) 포드에 대한 시간당 측정을 시작합니다. 이 CloudTrail 로그 항목은 구매자의 AWS 계정에 로깅됩니다.

```
{
    "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"
}
```

예: Amazon EKS의 컨테이너에 대한 **MeterUsage**

다음 예제는 Amazon EKS의 컨테이너에 AWS Marketplace Metering Service 대한의 `MeterUsage` 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다. 에서 사용자 지정 측정 AWS Marketplace 이 포함된 컨테이너 제품이 구매자의에 배포되면 컨테이너의 AWS 계정소프트웨어는 구매자의 `MeterUsage` 내에서를 호출 AWS 계정 하여 매시간 보고합니다. 이 CloudTrail 로그 항목은 구매자의 AWS 계정에 로깅됩니다.

```
{
    "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"
}
```

예: AMI의 **`MeterUsage`**

다음 예제는 Amazon Machine Image(AMIs AWS Marketplace Metering Service 용의 `MeterUsage` 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다. 에서 사용자 지정 측정 AWS Marketplace 이 포함된 AMI 제품이 구매자의에 배포되면 AMI의 AWS 계정소프트웨어는 구매자의 `MeterUsage` 내에서를 호출 AWS 계정 하여 매시간 사용량을 보고합니다. 이 CloudTrail 로그 항목은 구매자의 AWS 계정에 로깅됩니다.

```
{
    "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"
}
```

문제 해결

일반 판매자 등록 오류 메시지

다음 표에서는 일반적인 판매자 등록 오류 메시지와 해결 방법을 설명합니다.

오류 메시지	Solution
규정 준수 요구 사항이 완료되지 않음	모든 일본 판매자의 경우 유료 제품을 게시하기 전에 규제 목적으로 추가 정보를 수집해야 합니다. 일본에 기반을 둔 판매자이고 등록에 관심이 있는 경우 `<aws-jp-marketplace@amazon.co.jp>` 으로 일본 판매자 온보딩 팀에 AWS Marketplace문의하여 자세한 내용을 확인하세요. 자세한 내용은 JP SOR FAQ를 참조하세요.
AWS Marketplace 는 내부 문제로 인해 현재 요청을 처리할 수 없습니다. 나중에 다시 시도해 주세요. 문제가 지속되면 문의하세요.	브라우저의 캐시를 지우고 다시 시도하세요. 문제가 지속되면 AWS Marketplace 판매자 운영 팀에 문의하세요.
제품을 퍼블릭으로 업데이트하려면 먼저 판매자 계정에 퍼블릭 프로필을 추가해야 합니다.	제품을 퍼블릭으로 업데이트하기 전에 판매자 계정에 퍼블릭 프로필을 추가합니다. AWS Marketplace 관리 포털에서 설정으로 이동한 다음 퍼블릭 프로필로 이동합니다.
CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.	통화 코드와 호환되도록 결제 설정을 업데이트합니다. 현지 통화는 소비 요금에 지원되지 않으므로 USD를 추가해야 합니다. 자세한 내용은 사용 가능한 통화 단원을 참조하십시오.
유료 판매자로 유지하려면 VAT/GST 정보가 필요합니다.	여기에는 두 가지 옵션이 있습니다. 1. 기본 계정의 세금 설정을 업데이트합니다. 세금 상속 설명서를 검토하여 연결된 모든 계정에 미치는 영향을 이해합니다. 2. 조직에서이 계정을 제거하고 독립 실행형 계정으로 만듭니다. 즉, 결제의 세금 설정 페이지에서 VAT/GST 설정을 수동으로 추가해야 합니다.

오류 메시지	Solution
세금 인터뷰 위치가 비즈니스 위치와 일치하지 않음	현재 위치 설정을 확인하려면: 1. 세금 인터뷰 위치: • AWS Marketplace 관리 포털에서 설정으로 이동한 다음 세금 정보로 이동합니다. • 세금 정보 업데이트를 선택한 다음 재시도를 선택합니다. • 세금 프로필 업데이트를 선택한 후 아래로 스크롤하여 제공한 국가를 확인합니다. 2. 비즈니스 위치: • AWS Marketplace 관리 포털의 설정 탭에서 계정 요약에 제공된 비즈니스 위치를 확인합니다. 위치가 일치하지 않는 경우: • 결제의 세금 정보 업데이트: 자세한 내용은 계정 위치를 참조하세요. • AMMP 설정에서 세금 인터뷰 업데이트: 세금 정보에서 세금 인터뷰 재실행 결제 처리 문제를 방지하려면 두 위치가 동일한지 확인합니다.
사용을 시작하려면 AWS 계정 등록을 AWS Marketplace완료합니다.	AWS 계정 등록을 완료합니다. 판매자 등록에 대한 자세한 내용은 섹션을 참조하세요AWS Marketplace 판매자로 등록.
퍼블릭 프로필을 검토 중입니다. 퍼블릭 프로필이 확인될 때까지 지급 방법을 추가할 수 없습니다. 영업일 기준 2일 후에 다시 확인하세요. 현재 무료 제품만 게시할 수 있습니다.	퍼블릭 프로필을 검토 중입니다. 공개 프로필이 확인될 때까지 지급 방법을 추가하고 KYC를 완료할 수 없습니다. 승인하는 데 영업일 기준 2~3일이 걸립니다. 현재 무료 제품만 게시할 수 있습니다. 프로필이 승인되면 루트 이메일 주소로 이메일을 받게 됩니다. 영업일 기준 3일 후에도 결제 정보를 추가할 수 없는 경우 AWS Marketplace 판매자 운영 팀에 문의하세요.

오류 메시지	Solution
비즈니스 위치가 지원되지 않는 관할권에 있습니다. 유료 제품을 게시할 수 없습니다.	비즈니스 위치가 지원되지 않는 관할권에 있습니다. 유료 제품을 게시할 수 없습니다. 적격 관할 지역에 대한 자세한 내용은 섹션을 참조하세요유료 상품에 대한 적격 관할 구역.
AWS 계정이 판매자로 등록되어 있지 않습니다 AWS Marketplace.	판매자로 등록되지 않았습니다 AWS Marketplace. AWS Marketplace Management Portal에서 등록을 완료합니다.
판매자가 유료 판매자가 되려면 공개 프로필이 있어야 합니다.	판매자 공개 프로필을 작성하고 승인해야 합니다. 이를 완료하고 AWS Marketplace Management Portal에서 계정 상태를 확인합니다.

문서 기록

다음 표에서는 AWS Marketplace 판매자 설명서의 이번 릴리스를 소개합니다.

이 설명서가 업데이트될 때 알림을 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
판매자 등록에 지급 기본 설정을 지정하는 단계가 추가되었습니다.	판매자 등록 단계는 섹션을 참조https://docs.aws.amazon.com/marketplace/latest/userguide/seller-registration-process.html하세요.	2025년 5월 21일
Marketplace 표준 계약에 추가된 연방 부록	부록은 미국 정부 조달을 간소화하는 데 도움이 될 수 있습니다. 자세한 내용은 에서 표준화된 계약 사용을 AWS Marketplace 참조하세요.	2025년 4월 29일
SaaS 지침 업데이트	"배포 날짜" AWS지정에 대한 정보가 추가되었습니다. 자세한 내용은 https://docs.aws.amazon.com/marketplace/latest/userguide/saas-guidelines.html 섹션을 참조하세요.	2025년 4월 8일
권한 부여 서한	누군가에게 회사를 대신하여 AWS 계정을 개설할 수 있는 권한을 부여하는 데 사용되는 letter-of-authorization 템플릿을 추가했습니다. 자세한 내용은 AWS Marketplace 판매자로 등록을 참조하세요.	2025년 4월 8일
ML에 셀프 서비스 목록 정보 추가	셀프 서비스 목록 정보는 섹션을 참조https://docs.aws.amazon.com/marketplace/	2025년 3월 26일

	latest/userguide/machine-learning-products하세요.	
업데이트된 관리형 정책	AWSMarketplaceSellerFullAccess 정책에 `ListSupplementalTaxRegistrations` , `PutSupplementalTaxRegistration` , `DeleteSupplementalTaxRegistratio` 및 권한이 추가되었습니다`GetTaxRegistration` .	2024년 12월 20일
프로모션 미디어로 AWS Marketplace 제품 개선	프로모션 미디어 사용 방법에 대한 가이드가 추가되었습니다.	2024년 12월 17일
업데이트된 관리형 정책	다음 AWS 관리형 정책에서 `ListTasks` , `DescribeTask` `UpdateTasks` , 및 `CompleteTasks` 권한이 제거되었습니다. • AWSMarketplaceSellerFullAccess • AWSMarketplaceSellerProductsFullAccess • AWSMarketplaceSellerProductsReadOnly	2024년 12월 10일

AWS 기능을 사용한 새 구매에 대한 설명서 추가	를 사용하여 구매 AWS하면 AWS Marketplace 판매자가 자체 웹 사이트에서 AWS Marketplace 제품을 표시하고 판매할 수 있습니다. AWS Marketplace 판매자는 CTA(AWS Marketplace call-to-action) 버튼 및 APIs와 통합하여 판매자 소유 웹 사이트에서 AWS기반 소프트웨어 검색 및 조달 경험을 구축할 수 있습니다.	2024년 12월 4일
EC2 Image Builder 구성 요소 기반 제품	EC2 Image Builder 구성 요소 기반 제품이 제품 제공 방법으로 추가되었습니다.	2024년 12월 1일
새 AWS 파트너 어시스턴트	AWS Partner Assistant는 AWS 파트너를 위한 생성형 AI 기반 챗봇입니다. 파트너 센트럴과 AWS Marketplace 포털 (AMMP) 모두에서 액세스할 수 있습니다.	2024년 11월 25일
업데이트된 관리형 정책	이제 다음 정책에 `uploadFiles` 권한이 부여됩니다. • awsmarketplacefullaccess • AWSMarketplaceSellerFullAccess	2024년 11월 18일
새 관리형 정책 추가	`AWSMarketplaceSellerOfferManagement` 설명서가 추가되었습니다.	2024년 11월 18일

업데이트된 관리형 정책	다음 AWS 관리형 정책에는 이제 `DescribeAssessment` 및 `ListAssessments` 권한이 있습니다. • AWSMarketplaceSellerFullAccess • AWSMarketplaceSellerProductsFullAccess • AWSMarketplaceSellerProductsReadOnly	2024년 10월 9일
더 이상 사용되지 않는 권한 제거	세 가지 작업을 제거하도록 `AWSMarketplaceSellerFullAccess` 설명서를 업데이트했습니다.	2024년 6월 4일
Amazon Machine Image(AMI) 연간 계약 수정 옵션 업데이트됨	이제 판매자는 기존 구독에 대한 인스턴스 유형을 추가하거나 전환할 수 있습니다.	2024년 5월 30일
고객확인제도(KYC) 프로세스 업데이트	판매자를 위한 KYC 프로세스에 추가 단계를 추가했습니다.	2024년 5월 21일
AWS Marketplace 판매자를 위한 비공개 제안 경험 업데이트	비공개 제안을 생성하고 관리하도록 향상된 경험을 위한 콘텐츠가 추가되었습니다.	2024년 5월 20일
Amazon EKS 추가 기능 제품에 대한 요구 사항 업데이트	"컨테이너 제품을 AWS Marketplace 추가 기능으로 준비" 단원을 업데이트하고 "추가 기능 구성 요구 사항 및 추가 기능 공급자 모범 사례"를 추가했습니다.	2024년 5월 8일
AWS Marketplace 판매자에 대한 업데이트된 권한	구문 오류를 수정하기 위해 권한 예제를 업데이트했습니다.	2024년 4월 2일

SaaS 계약 요금 업데이트	종량제 요금 모델을 사용한 SaaS 계약의 초과 보고와 관련된 콘텐츠가 업데이트되었습니다.	2024년 4월 2일
의 새로운 데모 및 비공개 제안 옵션 AWS Marketplace	AWS Marketplace 는 이제 일부 판매자를 위한 제품 세부 정보 페이지에서 데모 및 비공개 제안 요청 옵션을 지원합니다.	2024년 4월 1일
과세를 위한 새 대시보드	AWS Marketplace 는 이제의 거래에 대한 미국 및 국제 세금에 대한 시각화 및 세분화된 데이터를 제공하는 과세 대시보드를 지원합니다 AWS Marketplace.	2024년 3월 29일
`AWSMarketplaceGetEntitlements` 업데이트	`AWSMarketplaceGetEntitlements` 관리형 정책에 `sid`를 추가합니다.	2024년 3월 22일
전문 서비스 절차 업데이트	편집 제품 가시성을 업데이트하고 전문 서비스 제품 절차를 삭제했습니다.	2024년 3월 19일
AMI 액세스 정책 업데이트	Linux별 및 Unix 유사 AMI 액세스 정책을 명확히 하기 위해 섹션을 업데이트했습니다.	2024년 3월 19일
에 대한 검색 엔진 최적화 정보 추가 AWS Marketplace	검색 엔진 최적화와 관련된 내용이 추가되었습니다 AWS Marketplace.	2024년 3월 19일
에서 판매자를 위한 관리형 정책 업데이트 AWS Marketplace	서비스 연결 역할 만들기와 관련된 권한을 추가하도록 `AWSMarketplaceSellerFullAccess` 가 업데이트되었습니다.	2024년 3월 15일

의 제품에 대한 새로운 서비스 연결 역할 AWS Marketplace	AWS Marketplace 는 이제 재판매 권한 부여를 위해에서 사용하거나 관리하는 서비스 및 리소스 AWS Marketplace 에 액세스할 수 있는 AWS 서비스 연결 역할을 제공합니다.	2024년 3월 15일
에서 판매자에 대한 관리형 정책 업데이트 AWS Marketplace	세금 정보 액세스와 관련된 권한을 추가하도록 `AWSMarketplaceSellerFullAccess` 가 업데이트되었습니다.	2024년 2월 8일
보안 보고서를 위한 Amazon EventBridge 이벤트	AWS Marketplace 는 이제 판매자의 제품에 대해 보안 취약성 보고서를 사용할 수 있을 때 이전에는 Amazon CloudWatch Events라고 했던 Amazon EventBridge 이벤트를 지원합니다.	2024년 1월 31일
Amazon EKS 추가 기능 지원	AWS Marketplace 컨테이너 기반 제품의 Amazon EKS 추가 기능에 게시와 관련된 콘텐츠 및 절차가 추가되었습니다.	2024년 1월 29일
전문 서비스 판매자의 제품 재판매 지원 추가	판매자는 이제 독립 소프트웨어 개발 판매 회사(ISV)로서 채널 파트너를 위한 재판매 기회를 창출할 수 있습니다.	2024년 1월 18일
에서 미래 날짜의 계약에 대한 일반 가용성 AWS Marketplace	모든 ISVs와 AWS Marketplace 채널 파트너는 선결제 요금으로 비공개 제안을 게시하는 과정에서 향후 시작일을 지정할 수 있습니다.	2024년 1월 16일

사용을 위한 새로운 대시보드	AWS Marketplace 는 이제 SaaS 및 서버 사용량 기반 제품을 사용하는 고객에게 시각화 및 세분화된 데이터를 제공하는 사용 대시보드를 지원합니다.	2024년 1월 10일
판매자를 위한 새로운 빠른 시작 배포 옵션	서비스형 소프트웨어(SaaS) 제품의 새로운 빠른 시작 배포 옵션과 관련된 콘텐츠 및 절차가 추가되었습니다.	2023년 11월 29일
비공개 제안에 대한 유연한 결제 일정이 제공됩니다.	이제 AWS Marketplace의 모든 고객이 비공개 제안에 대한 유연한 결제 일정(FPS)을 이용할 수 있습니다.	2023년 11월 17일
컨테이너 제품에 대한 셀프 서비스 콘텐츠 추가	컨테이너 제품의 셀프 서비스 작업과 관련된 콘텐츠와 절차를 추가했습니다.	2023년 11월 3일
계약 및 갱신에 대한 새 대시보드	AWS Marketplace 는 이제 판매자를 위한 계약 및 갱신 대시보드를 제공합니다.	2023년 10월 31일
변경 세트를 위한 Amazon EventBridge 이벤트	AWS Marketplace 는 이제 변경 세트가 성공, 실패 또는 취소됨 상태로 완료될 때 이전에 Amazon CloudWatch Events라고 했던 Amazon EventBridge 이벤트를 지원합니다.	2023년 10월 31일
AWS Marketplace Vendor Insights에서 판매자를 위한 관리형 정책 업데이트	AWS Marketplace 데이터 소스를 업데이트할 수 있는 권한을 추가`AWSVendorInsightsVendorFullAccess` 하도록 가 업데이트되었습니다.	2023년 10월 18일

데스크톱 애플리케이션(AMDA)에 AWS Marketplace 대한 지원 종료	AWS Marketplace 는 2023년 10월 2일에 AMDA에 대한 지원을 종료했습니다. AMDA와 관련된 모든 콘텐츠 및 절차를 설명서에서 삭제했습니다.	2023년 10월 2일
SaaS 제품에 대한 셀프 서비스 콘텐츠 추가	서비스형 소프트웨어(SaaS) 제품의 셀프 서비스 작업과 관련된 콘텐츠와 절차를 추가했습니다.	2023년 9월 12일
이제 AWS Marketplace 판매자가 일일 지출을 사용할 수 있습니다.	AWS Marketplace Management Portal 이제의 판매자는 매일 또는 매월 지출을 받을 수 있습니다.	2023년 9월 7일
Amazon EventBridge 지원	AWS Marketplace 는 이제 판매자가 새 제안을 받을 때 EventBridge 이벤트를 지원합니다.	2023년 9월 6일
AWS Marketplace Vendor Insights를 위한 새로운 셀프 서비스 온보딩 경험	AWS Marketplace Vendor Insights는 이제 셀프 서비스 온보딩 환경을 지원합니다.	2023년 8월 17일
에서 판매자에 대한 관리형 정책 업데이트 AWS Marketplace	`AWSMarketplaceSellerFullAccess` 를 업데이트하여 엔터티 공유와 관련된 권한을 추가했습니다.	2023년 6월 1일
에서 판매자에 대한 관리형 정책 업데이트 AWS Marketplace	`AWSMarketplaceSellerFullAccess` 를 업데이트하여 계정 확인, 은행 계좌 확인, 사례 관리 및 판매자 알림 세부 정보와 관련된 권한을 추가했습니다.	2023년 6월 1일

에 대한 세분화된 액세스에 대한 콘텐츠 추가 AWS Marketplace	에 세분화된 액세스를 위한 개요 정보, 권한 및 절차가 추가되었습니다 AWS Marketplace Management Portal.	2023년 6월 1일
AMI 제품에 대한 절차 추가	AMI 제품에 대한 셀프 서비스 절차를 추가했습니다.	2023년 5월 12일
에서 Amazon 세금 공제 프로그램 및 미국 세금 계산 서비스에 대한 지원 종료 AWS Marketplace	아마존 면세 프로그램 및 미국 세금 계산 서비스에 대한 지원이 2023년 3월 6일에 종료되었으므로 설명서에서 두 서비스를 삭제했습니다.	2023년 3월 6일
컨테이너 제품에 대한 절차 추가	컨테이너 제품 설정을 변경하는 새 절차를 추가했습니다.	2023년 2월 13일
AWS Marketplace 제품 지원 연결에 대한 지원 종료	AWS Marketplace 제품 지원 연결 및 상거래 분석 서비스를 사용한 고객 연락처 세부 정보 공유는 2022년 11월 30일부터 더 이상 지원되지 않습니다. 가이드에서 AWS Marketplace 제품 지원 연결 콘텐츠를 AWS Marketplace 제거했습니다.	2023년 1월 27일
컨설팅 파트너 이름 변경	AWS Marketplace 는 이제 컨설팅 파트너를 채널 파트너로 지칭합니다. 변경된 이름만 반영하도록 설명서를 업데이트했습니다.	2023년 1월 26일
고객 서비스 팀 이름 변경	관리형 카탈로그 운영(MCO) 팀이 이름을 AWS Marketplace 판매자 운영 팀으로 변경했습니다. 변경된 이름만 반영하도록 설명서를 업데이트했습니다.	2023년 1월 24일

비공개 제안 페이지	이제 인증된 구매자는 AWS Marketplace 비공개 제안 페이지에서에 확장된 AWS 계정 비공개 제안을 볼 수 있습니다.	2023년 1월 19일
SaaS 제품 설정에 대한 주제 추가	컨테이너 제품 설정을 변경하는 절차가 포함된 새 주제를 추가했습니다.	2023년 1월 6일
에서 판매자에 대한 관리형 정책 업데이트 AWS Marketplace	AWS Marketplace 판매자 대시보드에 액세스할 수 있는 권한을 추가`AWSMarketplaceSellerFullAccess` 하도록 업데이트되었습니다.	2022년 12월 23일
판매자를 위한 이메일 알림 업데이트	이제 비공개 제안이 게시되면 판매자에게 알림이 전송됩니다.	2022년 12월 22일
이제 AWS Marketplace의 판매자는 구독의 SaaS 무료 평가판 사용 가능	이제 판매자는 구독 제품의 무료 평가판을 생성할 수 있습니다.	2022년 12월 16일
AWS Marketplace의 AMI 셀프 서비스 목록(버전 2)에 대한 판매자 경험 업데이트	이제의 판매자는 단일 AMI 제품 셀프 서비스 목록을 생성할 AWS Marketplace 수 있습니다. 판매자는 작업이 처리될 때까지 추가로 기다릴 필요 없이 직접 업데이트할 수 있습니다.	2022년 12월 14일

태그 기반 권한 부여에 대한 세 가지 정책 업데이트	AWS Marketplace 태그 기반 권한 부여 기능에 대한 세 가지 정책(`AWSMarketplaceSellerFullAccess` `AWSMarketplaceSellerProductsFullAccess` , 및 `AWSMarketplaceSellerProductsReadOnly`) 이 업데이트되었습니다.	2022년 12월 9일
AWS Marketplace Vendor Insights에서 판매자를 위한 정책 업데이트	AWS Marketplace Vendor Insights 판매자에 `AWSVendorInsightsVendorReadOnly` 대한 관리형 정책 `AWSVendorInsightsAssessorFullAccess` 및를 업데이트했습니다.	2022년 11월 30일
AWS Marketplace Vendor Insights에서 판매자에 대한 액세스 제어	판매자가 사용할 수 있는 작업 및 권한을 설명하기 위해 AWS Marketplace Vendor Insights에 대한 새 주제가 추가되었습니다.	2022년 11월 30일
AWS Marketplace Vendor Insights에 대한 네 가지 관리형 정책 업데이트	AWS Marketplace Vendor Insights에 대한 `AWSVendorInsightsVendorFullAccess` 및 `AWSVendorInsightsVendorReadOnly` 관리형 정책을 업데이트했습니다.	2022년 11월 28일

판매자가 Amazon EKS에 추가 기능 제품 게시 가능	AWS Marketplace 및 Amazon EKS의 통합을 통해 판매자는 Amazon EKS 콘솔에서 구매자를 위해 제품을 제공할 수 있습니다.	2022년 11월 28일
AWS Marketplace Vendor Insights 설정 업데이트	AWS Marketplace Vendor Insights의 설정 절차를 업데이트했습니다.	2022년 11월 18일
AWS Marketplace Vendor Insights에 대한 두 정책 업데이트	AWS Marketplace Vendor Insights`AWSMarketplaceSellerFullAccess` 에 대한 두 가지 정책 `AWSMarketplaceSellerProductsFullAccess` 및를 업데이트했습니다.	2022년 7월 26일
소프트웨어 위험 평가를 제공하는 AWS Marketplace Vendor Insights 기능에 대한 두 가지 정책이 추가되었습니다.	AWS Marketplace Vendor Insights`AWSVendorInsightsVendorFullAccess` `AWSVendorInsightsVendorReadOnly` 에 소프트웨어 위험 평가를 제공하는 두 가지 정책과 기능이 추가되었습니다.	2022년 7월 26일
AWS Marketplace Vendor Insights는에 추가된 새로운 기능입니다. AWS Marketplace	AWS Marketplace Vendor Insights는 소프트웨어 위험 평가를 제공하는 기능입니다. AWS Marketplace Vendor Insights Vendor Insights는 소프트웨어 위험 평가를 제공하는 기능입니다.	2022년 7월 26일

AWS Marketplace 상거래 분석 서비스 권한 업데이트	AWS Marketplace 상거래 분석 서비스에는 추가 IAM 권한이 있습니다.	2022년 7월 21일
판매자 제공 데이터 피드 서비스 단원 추가	설명서만 업데이트하여 판매자 제공 데이터 피드 서비스 단원을 추가하고 데이터 피드와 관련된 단원을 재구성했습니다.	2022년 6월 15일
추가 보고서 단원 추가	에서 최근 기능 시작을 AWS Marketplace 제공하는 보충 보고서에 대한 새 섹션이 추가되었습니다.	2022년 6월 14일
이제에서 판매자가 계약에 대한 SaaS 무료 평가판을 사용할 수 있습니다. AWS Marketplace	이제 판매자는 추가 개발 작업 없이 무료 평가판 기간, 평가 기간 동안 사용할 수 있는 크기, 고객에게 제공되는 무료 사용 용량을 정의하여 무료 평가판을 생성할 수 있습니다.	2022년 5월 31일
구매자와 판매자의 트랜잭션에 이메일 알림 추가	제안 및 계약을 확인하는 구매자와 판매자에게 이메일 알림을 제공하는 새로운 기능입니다 AWS Marketplace.	2022년 5월 23일
기계 학습 제품 목록 생성에 예제 추가	기계 학습 제품을 생성할 때 판매자의 관점과 구매자의 관점을 비교하는 예제를 포함하도록 기계 학습 섹션의 설명서만 업데이트했습니다.	2022년 4월 22일
기계 학습 단원 업데이트	절차를 명확히 설명하기 위해 기계 학습 섹션의 설명서만 업데이트했습니다.	2022년 4월 15일

이스라엘은 이제 적격 관할 구역	이스라엘 거주자는 이제 AWS Marketplace에서 제품을 판매할 수 있습니다.	2022년 4월 13일
제안 데이터 피드의 새 항목 업데이트	이제 제조업체는 채널 파트너가 생성한 제안에 대한 제안 정보를 받게 됩니다. 여기에는 관련 계정이 해당 제안의 공식 판매자인 제안과 제조업체인 제안도 포함됩니다.	2022년 3월 29일
리셀러 영업 기회 알림	이제 판매자는 리셀러 영업 기회에 대한 알림을 받을 수 있습니다.	2022년 3월 28일
전문 서비스 제품에 비디오 추가	서비스 제품 관리 방법에 대한 자세한 내용이 포함된 비디오를 전문 서비스 제품 페이지에 추가했습니다.	2022년 2월 24일
서버리스 SaaS 통합 솔루션 배포에 대한 새 주제	배포 단계에서 참조할 수 있도록 AWS Quick Start 링크를 포함하여 서버리스 SaaS 배포 통합에 대한 새 정보를 추가했습니다.	2022년 2월 15일
컨테이너 기반 요구 사항 및 AMI 단원을 최소한으로 업데이트	컨테이너 기반 요구 사항의 정책을 최소한으로 업데이트하고 AMI 요금 계약에 대한 잘못된 정보를 제거했습니다.	2022년 2월 14일
컨테이너 버전 관리 업데이트	컨테이너 이미지 및 기타 아티팩트를 리포지토리로 푸시하는 방법을 명확히 하기 위해 설명서만 업데이트했습니다.	2022년 2월 10일

SaaS 제품의 `ResolveCustomer` 코드 예제 업데이트	`CustomerAWSAccountID` 를 포함하도록 SaaS 제품의 `ResolveCustomer` 코드 예제를 업데이트했습니다.	2022년 2월 3일
Containers Anywhere 제품용 AWS License Manager 와 통합하기 AWS Marketplace 위한 설명서 추가	License Manager와 통합하여 for Containers Anywhere 제품에 계약 가격을 추가하는 방법에 대한 자세한 지침을 추가하기 AWS Marketplace 위한 설명서 전용 업데이트입니다.	2022년 2월 1일
SaaS SNS 알림 업데이트	SaaS 알림 메시지를 명확히 하기 위해 설명서만 업데이트했습니다.	2022년 1월 25일
판매자가 Amazon Web Services EMEA SARL을 통해 EMEA 거주 구매자와 거래 가능	이제 적격 AWS Marketplace 판매자는 Amazon Web Services EMEA SARL을 통해 유럽, 중동 및 아프리카(EMEA)의 국가 및 지역에 기반을 둔 고객과 AWS 계정 거래할 수 있습니다.	2022년 1월 7일

차트 Helm 제공 방법을 통해 컨테이너 기반 제품의 제공 옵션을 생성하는 방법에 대한 문서 추가	이제 판매자는 차트 Helm 제공 방법을 통한 제공 옵션을 제공할 수 있습니다. 구매자는 판매자가 제공하는 차트 Helm을 실행 환경에 설치하면 이러한 제공 옵션을 사용하여 컨테이너 기반 애플리케이션을 실행할 수 있습니다. 차트 Helm 제공 방법을 제공할 때 판매자는 구매자에게 빠른 시작을 활성화할 수 있습니다. 빠른 시작은 구매자가 새 Amazon EKS 클러스터 AWS CloudFormation 를 빠르게 생성하고 해당 클러스터에서 컨테이너 기반 애플리케이션을 시작하는 데 사용할 수 있는 기능입니다.	2021년 11월 29일
기존 정책 업데이트	AWS Marketplace 판매자를 위한 보안 정책이 업데이트되었습니다.	2021년 11월 22일
AMI 및 컨테이너 기반 제품의 계약 요금	이제 독립 소프트웨어 개발 판매 회사(ISV)는 새 AMI 기반 제품 또는 컨테이너 기반 제품을 나열하고 구매자에게 선불 계약 요금을 제시할 수 있습니다.	2021년 11월 17일
공급업체 미터식 태그 지정	공급업체 미터식 태그 지정(코드 예제 포함)에 대한 설명서만 업데이트했습니다.	2021년 11월 11일

AMI 또는 컨테이너 제품을 위한 Amazon Simple Notification Service	독립 소프트웨어 개발 판매 회사(ISV)는 고객이 Amazon Simple Notification Service를 통해 AMI 또는 컨테이너 제품을 구독하거나 구독 취소할 때 알림을 받을 수 있습니다.	2021년 11월 10일
새 판매자 권한	AWS Marketplace 는의 제안 및 파트너 탭에 액세스할 수 있는 새 권한을 추가했습니다 AWS Marketplace Management Portal.	2021년 11월 9일
기계 학습 제품의 비동기 추론을 위해 구성된 엔드포인트 배포 가능	최대보다 높은 페이로드 추론을 예상하거나 호출당 최대 처리 시간을 초과하는 처리 시간이 필요한 기계 학습 소프트웨어의 경우 구매자는 Amazon SageMaker AI 비동기 추론에 대해 구성된 엔드포인트를 배포할 수 있습니다.	2021년 11월 8일
환불 정책 및 승인	환불 정책을 명확히 하고 모든 환불 정보를 AWS Marketplace 판매자 설명서의 한 중앙 위치로 이동하기 위해 설명서만 업데이트했습니다.	2021년 8월 20일
컨설팅 파트너 비공개 제안에 사용할 EULA를 선택 또는 업로드	이제 독립 소프트웨어 개발 판매 회사는 컨설팅 파트너의 재판매 기회를 생성할 때 최종 사용자 라이선스 계약(EULA)을 선택하거나 업로드할 수 있습니다.	2021년 8월 17일

SaaS 계약 제품의 사용자 지정 제품 크기	이제 독립 소프트웨어 개발 판매 회사(ISV)는 컨설팅 파트너의 재판매 기회를 생성할 때 SaaS 계약 제품 크기를 사용자 지정할 수 있습니다.	2021년 8월 17일
AWS Marketplace 필드 데모 프로그램	AWS Marketplace 필드 데모 프로그램을 위한 AWS Data Exchange 데이터 세트 제품의 요구 사항을 명확히 하기 위한 설명서 전용 업데이트입니다.	2021년 8월 3일
SaaS 제품 가이드라인 업데이트	SaaS 제품 가이드라인을 업데이트했습니다.	2021년 7월 29일
컨테이너 기반 제품 요구 사항 업데이트	컨테이너 기반 제품 요구 사항을 업데이트했습니다.	2021년 7월 29일
AMI 보안 정책 업데이트	AMI 제품의 보안 정책이 업데이트되었습니다.	2021년 7월 29일
적격 관할 구역 추가	이제 AWS Marketplace홍콩 SAR 및 카타르에서 판매자가 될 수 있습니다.	2021년 6월 23일
데이터 피드 개요	판매자가 사용할 수 있는 데이터 피드 구조의 개요를 제공하기 위해 설명서만 업데이트했습니다.	2021년 6월 23일
기계 학습 챕터 업데이트	기계 학습 제품의 생성 및 유지 관리에 대한 설명서만 업데이트했습니다.	2021년 5월 27일

컨테이너 제품의 셀프 서비스 업데이트	이제 판매자는 AWS Marketplace Management Portal을 통해 컨테이너 기반 제품을 더 간단하고 빠르게 업데이트할 수 있습니다.	2020년 12월 17일
전문 서비스	이제 판매자는 AWS Marketplace 구매자에게 전문 서비스를 제공할 수 있습니다. 문서에 전문 서비스 단원을·추가했습니다.	2020년 12월 3일
AMI 제품의 셀프 서비스 업데이트	이제 판매자는 AWS Marketplace Management Portal을 통해 Amazon Machine Image(AMI) 기반 제품을 더 간단하고 빠르게 업데이트할 수 있습니다.	2020년 11월 23일
적격 관할 구역 추가	이제 AWS Marketplace바레인, 노르웨이, 스위스 및 아랍에미리트(UAE)에서 판매자가 될 수 있습니다.	2020년 6월 17일
승인된 비공개 제안에 대해 업그레이드 및 갱신을 제공할 수 있습니다.	SaaS 계약 및 소비 제품이 포함된 SaaS 계약의 경우 판매자는 이전에 수락된 비공개 제안에 대해 비공개 제안을 사용하여 업그레이드 및 갱신을 제공할 수 있습니다.	2020년 5월 28일
데이터 피드에서 자세한 정보 확인 가능	데이터를 더 쉽게 찾고 분석할 수 있도록 보고서의 추가 정보를 더 작은 데이터 피드로 세분화했습니다.	2020년 5월 21일

이제 표준화된 라이선스 조건 사용 가능	사용자 지정 EULA 대신에 표준화된 라이선스 조건을 제공하여 계약 프로세스를 간소화할 수 있습니다.	2020년 4월 28일
호주 및 뉴질랜드는 적격 관할 구역	이제 (i) 호주 AWS Marketplace(AU) 뉴질랜드(NZ)의 영주권자 및 시민 또는 (ii) 해당 영역 중 하나에서 조직되거나 설립된 사업체에 대한 판매자가 될 수 있습니다.	2020년 4월 2일
이제 컨테이너 제품이 향상된 사용자 지정 측정 및 요금 기능 지원	자체 요금 단위를 정의하고 결제를 위해 사용량을 측정하려면 AWS Marketplace 측정 서비스의 `meterUsage` 작업과 통합하세요.	2019년 11월 14일
AWS Marketplace 는 AWS Data Exchange를 통해 데이터 제품을 지원합니다.	이제 AWS Marketplace에서 데이터 제품을 제공할 수 있습니다.	2019년 11월 13일
AWS Marketplace 카탈로그 API 서비스 소개	AWS Marketplace 카탈로그 API 서비스는 승인된 판매자가 제품을 프로그래밍 방식으로 관리할 수 있는 API 인터페이스를 제공합니다.	2019년 11월 12일
AWS Marketplace 는 유료 시간당 컨테이너를 지원합니다.	AWS Marketplace 는 이제 Amazon Elastic Kubernetes Service(Amazon EKS)에서 실행되는 유료 시간당 컨테이너를 지원합니다.	2019년 9월 25일
AMI 제품 기능 업데이트	이제 AMIs와 Lambda 함수를 함께 배포할 수 있습니다 AWS CloudFormation.	2019년 9월 11일

보안 섹션 추가	새 보안 섹션 아래에 보안 콘텐츠가 통합되었습니다.	2019년 5월 7일
AMI 보안 정책 업데이트	AMI 제품의 보안 정책이 업데이트되었습니다.	2022년 4월 11일
기계 학습 제품 섹션에 버전 관리 정보 추가	기계 학습 제품의 제품 버전 기능을 설명하는 내용이 추가됨.	2019년 3월 21일
기계 학습 제품 섹션 추가	기계 학습 제품을 게시하기 위한 콘텐츠가 추가되었습니다.	2018년 11월 28일
컨테이너 기반 제품 섹션 추가	컨테이너 기반 제품을 게시하기 위한 내용이 추가됨.	2018년 11월 27일
판매자 도움말 요청 제출용 링크 업데이트	이메일 주소가 webform 주소로 변경되었습니다.	2018년 10월 22일
소비 콘텐츠가 포함된 SaaS 계약 추가	SaaS 콘텐츠가 재구성되었고 소비 기능을 포함한 SaaS 계약의 릴리스를 지원하는 콘텐츠가 추가되었습니다.	2018년 10월 18일
비공개 제안에 대한 유연한 결제 일정을 다루는 콘텐츠 추가	비공개 제안 콘텐츠에 대한 유연한 결제 스케줄러의 릴리스를 지원하는 콘텐츠가 추가되었습니다.	2018년 10월 15일
IAM 권한 콘텐츠 업데이트	AMMP 읽기 전용 액세스에 대한 새로운 IAM 권한을 지원하는 콘텐츠가 추가되었습니다.	2018년 10월 9일
컨설팅 파트너 비공개 제안에 대한 콘텐츠 추가	컨설팅 파트너 비공개 제안 기능 릴리스 지원에 대한 콘텐츠가 추가되었습니다.	2018년 10월 9일
프라이빗 이미지 빌드에 대한 콘텐츠 추가	AMI 기능에 대한 프라이빗 이미지 지원 릴리스가 추가되었습니다.	2018년 8월 13일

판매자를 위한 검색 엔진 최적화 지침이 추가되었습니다.	제품을 검색에 최적화하려는 판매자를 위한 지침이 추가되었습니다.	2018년 7월 3일
AWS Marketplace 로고에 대한 링크 업데이트	새 AWS Marketplace 로고를 가리키도록 링크가 업데이트되었습니다.	2018년 6월 12일
판매자 가이드 추가	모든 PDF 판매자 가이드가 온라인 콘텐츠로 변환되었습니다.	2018년 5월 9일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.