aws

개발자 가이드

Amazon Pinpoint

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Pinpoint: 개발자 가이드

Copyright © 2026 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

..... x
Amazon Pinpoint가 무엇인가요? 1
Amazon Pinpoint를 사용하여 대상 세그먼트에 메시지 전송 및 데이터 분석 1
대상 세그먼트 정의 1
메시징 캠페인 예약 2
트랜잭션 메시지 전송 2
분석 및 지표 보고 사용 2
엔드포인트를 사용하여 대상 정의 3
엔드포인트 추가 4
예시 5
관련 정보 10
엔드포인트와 사용자 연결 10
예시 11
관련 정보 15
엔드포인트 배치 추가 16
예제 16
관련 정보 24
엔드포인트 가져오기 24
시작하기 전 준비 사항 24
예제 24
관련 정보 37
Amazon Pinpoint에서 Amazon S3 버킷으로 엔드포인트 내보내기 37
시작하기 전 준비 사항 37
예시 38
관련 정보 49
Amazon Pinpoint 프로젝트에서 엔드포인트 조회 49
예시 49
관련 정보 55
엔드포인트 ID 나열 55
엔드포인트 최대 관리 57
엔드포인트 삭제 58
예시 58
세그먼트 생성 또는 가져오기 62
세그먼트 빌드 62

를 사용하여 세그먼트 빌드 AWS SDK for Java 62
세그먼트 가져오기 66
AWS SDK for Java를 사용하여 세그먼트 가져오기 66
세그먼트 사용자 지정 69
이벤트 데이터 70
Lambda 함수 생성 71
Lambda 함수 정책 할당 73
캠페인에 Lambda 함수 할당 75
Amazon Pinpoint 캠페인을 프로그래밍 방식으로 생성 77
SDK for Java로 캠페인 생성 77
A/B 테스트 캠페인 생성 80
태그 관리 83
IAM 정책에서 태그 사용 84
리소스에 태그 추가 85
API를 사용하여 태그 추가 85
를 사용하여 태그 추가 AWS CLI 86
리소스에 대한 태그 표시 87
API를 사용하여 태그 표시 87
를 사용하여 태그 표시 AWS CLI 88
태그 업데이트 또는 덮어쓰기 89
리소스에서 태그 제거 90
API를 사용하여 태그 제거 90
를 사용하여 태그 제거 AWS CLI 90
애플리케이션과 통합 92
AWS SDK 작업 92
Amplify를 사용하여 프런트엔드 애플리케이션 연결 94
다음 단계 94
앱에 엔드포인트 등록 94
시작하기 전 준비 사항 95
AWS 모바일 SDKs 95
AWS Amplify 95
다음 단계 95
앱에서 이벤트 보고 95
시작하기 전 준비 사항 96
AWS 모바일 SDKs 97
웹 및 React Native 95

Amazon Pinpoint 이벤트 API 97
다음 단계 97
사용자의 앱에서 트랜잭션 메시지 전송 98
트랜잭션 이메일 보내기 98
이메일을 보낼 메서드 선택 99
Amazon Pinpoint와 Amazon SES 중에서 선택 99
API를 사용하여 이메일 전송 99
이메일 구독 취소 헤더 추가 113
SMS 메시지 보내기 115
음성 메시지 보내기 128
SMS and Voice API 사용 137
일회용 암호 생성 138
SendOtpMessage 응답 141
OTP 메시지 검증 141
VerifyOtpMessage 응답 142
Amazon Pinpoint의 OTP 코드 예제 142
참조 ID 생성 142
OTP 코드 전송 143
OTP 코드 확인 144
인앱 메시지 사용자 지정 146
엔드포인트에 대한 인앱 메시지 검색 146
GetInAppMessages API 응답 JSON 예제 148
InAppMessageCampaigns 객체 150
InAppMessage 객체 151
HeaderConfig 객체 152
BodyConfig 객체 153
InAppMessageContent 객체 153
Schedule 객체 154
InAppMessageButton 객체 155
DefaultButtonConfig 객체 156
OverrideButtonConfig 객체 157
전화 번호 확인 159
Amazon Pinpoint 전화번호 검증 사용 사례 159
를 사용하여 전화번호 확인 AWS CLI 160
전화 번호 검증 응답 160
사용자 지정 채널 생성 165

웹후크 사용 165
Lambda 함수 사용 165
개별 캠페인에 Lambda 함수 또는 웹후크 할당 167
Amazon Pinpoint 캠페인에 대한 Lambda 함수 생성 및 구성 168
Lambda 함수 예제 168
Amazon Pinpoint용 Lambda 함수 응답 형식 172
함수를 간접적으로 호출할 수 있는 권한 부여 173
앱 이벤트 데이터 스트리밍 176
이벤트 데이터 스트리밍 설정 177
사전 조건 177
AWS CLI 177
AWS SDK for Java 178
Amazon Pinpoint의 앱 이벤트 데이터 스트림 179
앱 이벤트 예 179
앱 이벤트 속성 180
Amazon Pinpoint의 캠페인 이벤트 데이터 스트림 185
캠페인 이벤트 예제 185
캠페인 이벤트 속성 186
Amazon Pinpoint의 여정 이벤트 데이터 192
여정 이벤트 예제 192
여정 이벤트 속성 193
Amazon Pinpoint에서 이메일 이벤트 데이터 스트리밍 198
이메일 이벤트 예제 198
이메일 이벤트 속성 205
Amazon Pinpoint의 SMS 이벤트 데이터 스트림 211
SMS 이벤트 예 212
SMS 이벤트 속성 213
이벤트 스트림 삭제 221
AWS CLI 221
AWS SDK for Java 221
분석 데이터 쿼리 222
Amazon Pinpoint의 지표 쿼리 222
IAM 정책 223
프로젝트, 캠페인 및 여정에 대한 표준 지표 228
캠페인의 애플리케이션 지표 229
이메일의 애플리케이션 지표 232

SMS의 애플리케이션 지표 240
캠페인 지표 246
여정 참여 지표 254
여정 실행 지표 259
여정 활동 실행 지표 260
여정 및 캠페인 실행 지표 265
캠페인 데이터 쿼리 268
사전 조건 268
하나의 캠페인에 대한 데이터 쿼리 269
여러 캠페인에 대한 데이터 쿼리 274
트랜잭션 메시징 데이터 쿼리 280
사전 조건 280
트랜잭션 이메일에 대한 데이터 쿼리 281
트랜잭션 SMS에 대한 데이터 쿼리 285
JSON 쿼리 결과 사용 291
JSON 구조 292
JSON 객체 및 필드 297
CloudTrail로 API 직접 호출 로깅 299
CloudTrail의 Amazon Pinpoint 정보 299
CloudTrail 로그 파일의 API 작업 300
CloudTrail 로그 파일의 이메일 API 작업 304
CloudTrail 로그 파일에서 지원되는 SMS and Voice API v1 작업 305
CloudTrail 로그 항목 예제 306
추천자 모델 사용 311
메시지에 추천 추가 311
추천자 모델에 대한 Lambda 함수 간접적으로 호출 313
입력 이벤트 데이터 313
응답 데이터 및 요구 사항 316
추천 데이터를 처리할 정책 할당 320
Amazon Pinpoint가 Lambda 함수를 간접적으로 호출하도록 권한 부여 322
추천자 모델 구성 323
Amazon Pinpoint 프로젝트 데이터 삭제 324
Amazon Pinpoint 프로젝트 데이터 삭제 324
코드 예제 326
Amazon Pinpoint 327
기본 사항 328

시나리오 419
Amazon Pinpoint SMS 및 음성 API 431
기본 사항 432
보안 453
데이터 보호 454
데이터 암호화 456
인터네트워크 트래픽 개인 정보 456
Amazon Pinpoint에 대한 인터페이스 VPC 엔드포인트 생성 457
ID 및 액세스 관리 459
대상 459
ID를 통한 인증 459
정책을 사용하여 액세스 관리 460
Amazon Pinpoint에서 IAM을 사용하는 방법 462
Amazon Pinpoint 정책 작업 468
ID 기반 정책 예시 498
일반적인 태스크에 대한 IAM 역할 512
문제 해결 529
로깅 및 모니터링 531
규정 준수 확인 532
복원력 533
인프라 보안 533
구성 및 취약성 분석 534
보안 모범 사례 534
할당량 536
프로젝트 할당량 536
API 요청 할당량 537
캠페인 할당량 539
이메일 할당량 540
이메일 메시지 할당량 541
이메일 발신자 및 수신자 할당량 541
이메일 전송 할당량 542
엔드포인트 할당량 543
엔드포인트 가져오기 할당량 544
이벤트 수집 할당량 545
여정 할당량 546
Lambda 할당량 547

기계 학습 할당량 547
메시지 템플릿 할당량 549
푸시 알림 할당량 550
인앱 메시지 할당량 550
세그먼트 할당량 551
SMS 할당량 551
10DLC 할당량 551
음성 할당량 552
할당량 증가 요청 552
문서 이력 554
이전 업데이트 562

지원 종료 공지: 2026년 10월 30일에 Amazon Pinpoint에 대한 지원이 AWS 종료됩니다. 2026년 10월 30일 이후에는 Amazon Pinpoint 콘솔 또는 Amazon Pinpoint 리소스(엔드포인트, 세그먼트, 캠페인, 여정, 분석)에 더 이상 액세스할 수 없습니다. 자세한 내용은 Amazon Pinpoint 지원 종료를 참조하세요. 참고: SMS, 음성, 모바일 푸시, OTP 및 전화번호 검증과 관련된 APIs는이 변경의 영향을 받지 않으며 AWS 최종 사용자 메시징에서 지원됩니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.

Amazon Pinpoint가 무엇인가요?

Amazon Pinpoint는 다양한 메시징 채널을 통해 고객과 소통하는 데 사용할 수 있는 AWS 서비스입니다. Amazon Pinpoint를 사용해 푸시 알림, 이메일, SMS 문자 메시지 또는 음성 메시지를 전송할 수 있습니다.

이 개발자 안내서는 애플리케이션 개발자를 위한 내용을 담고 있습니다. 이 안내서에는 Amazon Pinpoint 기능을 프로그래밍 방식으로 사용하는 방법이 수록되어 있습니다. 또한 분석 및 메시징 기능을 애플리케이션과 통합하는 절차 등 모바일 앱 개발자에게 특히 흥미로운 정보를 다룹니다.

Amazon Pinpoint는 북미, 유럽, 아시아 및 오세아니아의 여러 AWS 리전에서 사용할 수 있습니다. AWS 리전에 대한 자세한 내용은 Amazon Web Services 일반 참조의 AWS 리전 관리를 참조하세요. 현재 Amazon Pinpoint를 사용할 수 있는 모든 리전의 목록은 Amazon Web Services 일반 참조의 Amazon Pinpoint 엔드포인트 및 할당량과 AWS 서비스 엔드포인트를 참조하세요. 각 리전에서 사용할 수 있는 가용 영역 수에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

Amazon Pinpoint에 대한 자세한 내용은 다음 설명서를 참조하세요.

- Amazon Pinpoint API 참조
- Amazon Pinpoint SMS 및 음성 API
- Amazon Pinpoint 사용 설명서

Amazon Pinpoint를 사용하여 대상 세그먼트에 메시지 전송 및 데이터 분석

Amazon Pinpoint를 사용하여 대상 세그먼트를 정의하고, 메시징 캠페인 및 트랜잭션 메시지를 보내고, 지표를 사용하여 사용자 동작을 분석할 수 있습니다.

대상 세그먼트 정의

대상 세그먼트를 정의하여 메시지에 적절한 대상에게 도달하세요. 세그먼트는 캠페인에서 보내는 메시지를 수신하는 사용자를 지정합니다. 운영 체제 또는 모바일 디바이스 유형과 같은 애플리케이션에서 보고하는 데이터를 기반으로 동적 세그먼트를 정의할 수 있습니다. 다른 서비스나 애플리케이션을 사용하여 정의한 정적 세그먼트를 가져올 수도 있습니다.

메시징 캠페인 예약

메시징 캠페인을 만들어 대상에 접촉합니다. 캠페인은 정의된 일정에 따라 맞춤형 메시지를 보냅니다. 그런 다음 모바일 푸시, 이메일 또는 SMS 메시지를 전송하는 캠페인을 생성할 수 있습니다.

대체 캠페인 전략을 실험하려면 캠페인을 A/B 테스트로 설정하고 Amazon Pinpoint 분석을 사용하여 결과를 분석합니다.

트랜잭션 메시지 전송

특정 사용자에게 새 계정 활성화 메시지, 주문 확인, 암호 재설정 알림 등과 같은 트랜잭션 모바일 푸시 및 SMS 메시지를 직접 보내 고객에게 정보를 제공합니다. Amazon Pinpoint REST API를 사용하여 트랜잭션 메시지를 전송할 수 있습니다.

분석 및 지표 보고 사용

Amazon Pinpoint가 제공하는 분석을 사용하여 대상에 대한 통찰력과 캠페인의 효과에 대한 통찰력을 얻을 수 있습니다. 사용자의 참여 수준, 구매 활동, 인구 통계 등에 대한 추세를 볼 수 있습니다. 또한 캠페인 또는 애플리케이션에 대해 전송하거나 열어본 총 메시지 수와 같은 지표를 보고 메시지 트래픽을 모니터링할 수 있습니다. 애플리케이션은 Amazon Pinpoint API를 통해 Amazon Pinpoint에서 분석할 수 있는 사용자 지정 데이터를 보고할 수 있으며 특정 표준 지표에 대한 분석 데이터를 쿼리할 수 있습니다.

분석 데이터를 Amazon Pinpoint 외부에서 분석 또는 저장하기 위해 Amazon Kinesis로 데이터를 스트리밍하도록 Amazon Pinpoint를 구성할 수 있습니다.

엔드포인트를 사용하여 Amazon Pinpoint에서 대상을 나타냄

Amazon Pinpoint에서 대상의 각 멤버는 하나 이상의 엔드포인트로 표시됩니다. Amazon Pinpoint를 사용하여 메시지를 전송할 때 목표 대상의 멤버를 나타내는 엔드포인트로 메시지를 보냅니다. 각 엔드포인트 정의에는 디바이스 토큰, 이메일 주소 또는 전화번호 같은 메시지 대상이 포함됩니다. 또한 사용자 및 사용자의 디바이스에 대한 데이터도 포함됩니다. 대상을 분석, 세그먼트화 또는 참여시키기 전에 엔드포인트를 Amazon Pinpoint 프로젝트에 추가해야 합니다.

대상이 증가하고 변경되면 엔드포인트 데이터도 증가하고 변경됩니다. 대상에 대해 Amazon Pinpoint에 있는 최신 정보를 보려면 엔드포인트를 개별적으로 조회하거나 Amazon Pinpoint 프로젝트에 대한 모든 엔드포인트를 내보낼 수 있습니다. 엔드포인트 데이터를 보면 사용자에 대한 다음 정보를 볼 수 있습니다.

- 사용자의 디바이스 및 플랫폼.
- 사용자의 시간대.
- 사용자의 디바이스에 설치된 앱의 버전.
- 사용자의 도시 및 국가 위치.
- 기록하는 기타 사용자 지정 속성 및 지표.

Amazon Pinpoint 콘솔은 엔드포인트에 캡처되는 인구 통계 및 사용자 지정 속성에 대한 분석도 제공합니다.

다음 주제에서는 Amazon Pinpoint에서 엔드포인트를 사용하는 방법을 설명합니다. Android, iOS 또는 JavaScript 클라이언트를 사용하여 엔드포인트를 자동으로 추가하는 방법에 대한 자세한 내용은 애플리케이션에 Amazon Pinpoint 엔드포인트 등록 섹션을 참조하세요.

주제

- Amazon Pinpoint에 엔드포인트 추가
- Amazon Pinpoint 엔드포인트와 사용자 연결
- Amazon Pinpoint에 엔드포인트 배치 추가
- Amazon Pinpoint로 엔드포인트 가져오기
- Amazon Pinpoint에서 Amazon S3 버킷으로 엔드포인트 내보내기
- Amazon Pinpoint 프로젝트에서 엔드포인트 조회
- Amazon Pinpoint를 사용하여 엔드포인트 ID 나열

- Amazon Pinpoint의 최대 엔드포인트 수 관리
- Amazon Pinpoint에서 프로그래밍 방식으로 엔드포인트 삭제

Amazon Pinpoint에 엔드포인트 추가

엔드포인트는 메시지를 전송할 수 있는 대상을 나타냅니다(예: 모바일 디바이스, 전화번호 또는 이메일 주소). 대상의 구성원에게 메시지를 보내려면 먼저 해당 개인에 대해 하나 이상의 엔드포인트를 정의해야 합니다.

엔드포인트를 Amazon Pinpoint에 추가하면 대상 데이터의 리포지토리로 증가합니다. 이 데이터는 다음으로 구성됩니다.

- Amazon Pinpoint API를 사용하여 추가하거나 업데이트하는 엔드포인트
- 사용자가 애플리케이션에 연결할 때 클라이언트 코드가 추가하거나 업데이트하는 엔드포인트

엔드포인트를 정의할 때 채널과 주소를 지정합니다. 채널은 엔드포인트에 메시지를 보내는 데 사용하는 플랫폼 유형입니다. 채널의 예에는 푸시 알림 서비스, SMS 또는 이메일이 포함됩니다. 주소는 엔드포인트에 메시지를 보내는 위치입니다(예: 디바이스 토큰, 전화번호 또는 이메일 주소).

대상에 대한 세부 정보를 추가하려면 엔드포인트를 사용자 지정 및 표준 속성으로 보강할 수 있습니다. 이러한 속성에는 사용자, 사용자의 기본 설정, 사용자의 디바이스, 사용자가 사용하는 클라이언트의 버전 또는 사용자의 위치가 포함될 수 있습니다. 이 유형의 데이터를 엔드포인트에 추가할 때 다음을 수행할 수 있습니다.

- Amazon Pinpoint 콘솔에서 대상에 대한 차트를 봅니다.
- 올바른 목표 대상에 메시지를 전송할 수 있도록 엔드포인트 속성을 기반으로 대상을 세그먼트화합니다.
- 엔드포인트 속성 값으로 대체되는 메시지 변수를 통합하여 메시지를 개인 설정합니다.

AWS Mobile SDK 또는 AWS Amplify JavaScript 라이브러리를 사용하여 Amazon Pinpoint를 통합할 경우, 모바일 또는 JavaScript 클라이언트 애플리케이션은 엔드포인트를 자동으로 등록합니다. 클라이언트는 각 새 사용자에 대한 엔드포인트를 등록하고 재방문 사용자에 대한 엔드포인트를 업데이트합니다. 모바일 또는 JavaScript 클라이언트에서 엔드포인트를 등록하려면 애플리케이션에 Amazon Pinpoint 엔드포인트 등록 단원을 참조하십시오.

예시

다음 예제에서는 Amazon Pinpoint 프로젝트에 엔드포인트를 추가하는 방법을 보여 줍니다. 엔드포인트는 시애틀에 거주하고 iPhone을 사용하는 대상 구성원을 나타냅니다. Apple 푸시 알림 서비스(APN)를 통해 이 사람에게 메시지를 보낼 수 있습니다. 엔드포인트의 주소는 APN에서 제공된 디바이스 토큰입니다.

AWS CLI

AWS CLI에서 명령을 실행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 업데이트 명령

엔드포인트를 추가하거나 업데이트하려면 update-endpoint 명령을 사용합니다.

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
> --endpoint-request file://endpoint-request-file.json
```

위치:

- application-id는 엔드포인트를 추가하거나 업데이트할 Amazon Pinpoint 프로젝트의 ID입니다.
- example-endpoint는 새 엔드포인트에 할당할 ID 또는 업데이트할 기존 엔드포인트의 ID입니다.
- endpoint-request-file.json은 `--endpoint-request` 파라미터에 대한 입력이 포함된 로컬 JSON 파일의 파일 경로입니다.

Example엔드포인트 요청 파일

예제 `update-endpoint` 명령은 JSON 파일을 `--endpoint-request` 파라미터의 인수로 사용합니다. 이 파일에는 다음과 같은 엔드포인트 정의가 포함되어 있습니다.

```
{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
```

```
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

엔드포인트를 정의하는 데 사용할 수 있는 속성은 Amazon Pinpoint API 참조의 EndpointRequest 스키마를 참조하세요.

AWS SDK for Java

AWS SDK for Java에서 제공하는 클라이언트를 사용하여 Java 애플리케이션에서 Amazon Pinpoint API를 사용할 수 있습니다.

Example코드

엔드포인트를 추가하려면 `EndpointRequest` 객체를 초기화하고 이를 `AmazonPinpoint` 클라이언트의 `updateEndpoint` 메서드에 전달합니다.

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.*;
import java.util.Arrays;
```

```
public class AddExampleEndpoint {

 public static void main(String[] args) {

  final String USAGE = "\n" +
    "AddExampleEndpoint - Adds an example endpoint to an Amazon Pinpoint
 application." +
    "Usage: AddExampleEndpoint <applicationId>" +
    "Where:\n" +
    "  applicationId - The ID of the Amazon Pinpoint application to add the example
 " +
    "endpoint to.";

  if (args.length < 1) {
   System.out.println(USAGE);
   System.exit(1);
  }

  String applicationId = args[0];

  // The device token assigned to the user's device by Apple Push Notification
  // service (APNs).
  String deviceToken =
 "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f";

  // Initializes an endpoint definition with channel type and address.
  EndpointRequest wangXiulansIphoneEndpoint = new EndpointRequest()
    .withChannelType(ChannelType.APNS)
    .withAddress(deviceToken);

  // Adds custom attributes to the endpoint.
  wangXiulansIphoneEndpoint.addAttributesEntry("interests", Arrays.asList(
    "technology",
    "music",
    "travel"));

  // Adds custom metrics to the endpoint.
  wangXiulansIphoneEndpoint.addMetricsEntry("technology_interest_level", 9.0);
  wangXiulansIphoneEndpoint.addMetricsEntry("music_interest_level", 6.0);
  wangXiulansIphoneEndpoint.addMetricsEntry("travel_interest_level", 4.0);

  // Adds standard demographic attributes.
  wangXiulansIphoneEndpoint.setDemographic(new EndpointDemographic()
    .withAppVersion("1.0")
```

```
    .withMake("apple")
    .withModel("iPhone")
    .withModelVersion("8")
    .withPlatform("ios")
    .withPlatformVersion("11.3.1")
    .withTimezone("America/Los_Angeles"));

  // Adds standard location attributes.
  wangXiulansIphoneEndpoint.setLocation(new EndpointLocation()
    .withCountry("US")
    .withCity("Seattle")
    .withPostalCode("98121")
    .withLatitude(47.61)
    .withLongitude(-122.33));

  // Initializes the Amazon Pinpoint client.
  AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
    .withRegion(Regions.US_EAST_1).build();

  // Updates or creates the endpoint with Amazon Pinpoint.
  UpdateEndpointResult result = pinpointClient.updateEndpoint(new
 UpdateEndpointRequest()
    .withApplicationId(applicationId)
    .withEndpointId("example_endpoint")
    .withEndpointRequest(wangXiulansIphoneEndpoint));

  System.out.format("Update endpoint result: %s\n",
 result.getMessageBody().getMessage());

 }
}
```

HTTP

REST API에 HTTP 요청을 직접 수행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 PUT 요청

엔드포인트를 추가하려면 다음 URI에서 엔드포인트 리소스에 PUT 요청을 발행합니다.

/v1/apps/*application-id*/endpoints/*endpoint-id*

위치:

- application-id는 엔드포인트를 추가하거나 업데이트할 Amazon Pinpoint 프로젝트의 ID입니다.
- endpoint-id는 새 엔드포인트에 할당할 ID 또는 업데이트할 기존 엔드포인트의 ID입니다.

요청에 필수 헤더를 포함시키고 EndpointRequest JSON을 본문으로 제공합니다.

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180428T004705Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180428/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
```

```
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

요청이 성공하면 다음과 비슷한 응답이 수신됩니다.

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

관련 정보

지원되는 HTTP 메서드 및 요청 파라미터를 비롯하여, Amazon Pinpoint API의 엔드포인트 리소스에 대한 자세한 내용은 Amazon Pinpoint API 참조의 엔드포인트 섹션을 참조하세요.

변수를 사용하여 메시지를 개인화하는 방법에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 메시지 변수 섹션을 참조하세요.

할당할 수 있는 속성의 수와 같이 엔드포인트에 적용되는 할당량에 대한 자세한 내용은 the section called “엔드포인트 할당량” 단원을 참조하십시오.

Amazon Pinpoint 엔드포인트와 사용자 연결

엔드포인트에는 사용자를 정의하는 속성이 포함될 수 있습니다. 이 속성은 대상의 개인을 나타냅니다. 예를 들어, 사용자는 모바일 앱을 설치한 사람 또는 웹 사이트에 계정을 가지고 있는 사람을 나타낼 수 있습니다.

고유의 사용자 ID와 선택 사항으로 사용자 지정 사용자 속성을 지정하여 사용자를 정의합니다. 어떤 사람이 여러 디바이스에서 앱을 사용하는 경우 또는 한 사람에게 여러 주소로 메시지를 보낼 수 있는 경우 동일한 사용자 ID를 여러 엔드포인트에 할당할 수 있습니다. 이 경우 Amazon Pinpoint는 전체 엔드포인트에서 사용자 속성을 동기화합니다. 따라서 사용자 속성을 한 엔드포인트에 추가하면 Amazon Pinpoint는 동일한 사용자 ID가 포함된 각 엔드포인트에 해당 속성을 추가합니다.

사용자 속성을 추가하여 개인에게 적용되지만 이 사람이 사용하는 디바이스에 따라 달라지지 않는 데이터를 추적할 수 있습니다. 예를 들어, 한 사람의 이름, 연령 또는 계정 상태에 대한 속성을 추가할 수 있습니다.

> **Tip**
>
> 애플리케이션이 Amazon Cognito 사용자 풀을 사용하여 사용자 인증을 처리하는 경우 Amazon Cognito는 사용자 ID와 속성을 엔드포인트에 자동으로 추가합니다. 엔드포인트 사용자 ID 값의 경우 Amazon Cognito는 사용자 풀에 있는 사용자에게 할당된 sub 값을 할당합니다. Amazon Cognito를 사용하여 사용자를 추가하는 방법을 알아보려면 Amazon Cognito 개발자 안내서의 Amazon Cognito 사용자 풀에서 Amazon Pinpoint 분석 사용 섹션을 참조하세요.

사용자 정의를 엔드포인트에 추가한 후에는 대상을 세그먼트화하는 방법에 대한 추가 옵션이 있습니다. 사용자 속성을 기반으로 세그먼트를 정의하거나, 사용자 ID 목록을 가져와서 세그먼트를 정의할 수 있습니다. 사용자를 기반으로 하는 세그먼트에 메시지를 전송하는 경우 잠재적인 대상에는 세그먼트의 각 사용자와 연결된 각 엔드포인트가 포함됩니다.

또한 대상에게 메시지를 보내는 방법에 대한 추가 옵션도 있습니다. 캠페인을 사용하여 한 세그먼트의 사용자에게 메시지를 보내거나, 사용자 ID 목록에 메시지를 직접 전송할 수 있습니다. 메시지를 개인 설정하려면 사용자 속성 값으로 대체되는 메시지 변수를 포함시킬 수 있습니다.

예시

다음 예제에서는 사용자 정의를 엔드포인트에 추가하는 방법을 보여 줍니다.

AWS CLI

AWS CLI에서 명령을 실행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 업데이트 명령

사용자를 엔드포인트에 추가하려면 update-endpoint 명령을 사용합니다. `--endpoint-request` 파라미터의 경우 사용자가 포함될 수 있는 새 엔드포인트를 정의할 수 있습니다. 또는 기존 엔드포인트를 업데이트하려면 변경할 속성만 제공할 수 있습니다. 다음 예제에서는 사용자 속성만 제공하여 사용자를 기존 엔드포인트에 추가합니다.

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
```

```
> --endpoint-request file://endpoint-request-file.json
```

위치:

- *application-id*는 엔드포인트를 추가하거나 업데이트할 Amazon Pinpoint 프로젝트의 ID입니다.
- *endpoint-id*는 새 엔드포인트에 할당할 ID 또는 업데이트할 기존 엔드포인트의 ID입니다.
- *endpoint-request-file.json*은 `--endpoint-request` 파라미터에 대한 입력이 포함된 로컬 JSON 파일의 파일 경로입니다.

Example엔드포인트 요청 파일

예제 `update-endpoint` 명령은 JSON 파일을 `--endpoint-request` 파라미터의 인수로 사용합니다. 이 파일에는 다음과 같은 사용자 정의가 포함되어 있습니다.

```
{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":["Wang"],
            "LastName":["Xiulan"],
            "Gender":["Female"],
            "Age":["39"]
        }
    }
}
```

사용자를 정의하는 데 사용할 수 있는 속성은 Amazon Pinpoint API 참조에서 EndpointRequest 스키마의 `User` 객체를 참조하세요.

AWS SDK for Java

AWS SDK for Java에서 제공하는 클라이언트를 사용하여 Java 애플리케이션에서 Amazon Pinpoint API를 사용할 수 있습니다.

Example코드

엔드포인트에 사용자를 추가하려면 EndpointRequest 객체를 초기화하고 이를 `AmazonPinpoint` 클라이언트의 `updateEndpoint` 메서드에 전달합니다. 이 객체를 사용하여 사용자가 포함될 수 있는 새 엔드포인트를 정의할 수 있습니다. 또는 기존 엔드포인트를 업데이트하려면 변경할 속성만

업데이트할 수 있습니다. 다음 예제에서는 EndpointUser 객체를 EndpointRequest 객체에 추가하여 사용자를 기존 엔드포인트에 추가합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
   public static void updatePinpointEndpoint(PinpointClient pinpoint, String
 applicationId, String endPointId) {
       try {
           List<String> wangXiList = new ArrayList<>();
           wangXiList.add("cooking");
           wangXiList.add("running");
           wangXiList.add("swimming");

           Map myMapWang = new HashMap<>();
           myMapWang.put("interests", wangXiList);

           List<String> myNameWang = new ArrayList<>();
           myNameWang.add("Wang ");
           myNameWang.add("Xiulan");

           Map wangName = new HashMap<>();
           wangName.put("name", myNameWang);

           EndpointUser wangMajor = EndpointUser.builder()
                   .userId("example_user_10")
                   .userAttributes(wangName)
                   .build();

           // Create an EndpointBatchItem object for Mary Major.
           EndpointRequest wangXiulanEndpoint = EndpointRequest.builder()
                   .channelType(ChannelType.EMAIL)
                   .address("wang_xiulan@example.com")
```

```
                    .attributes(myMapWang)
                    .user(wangMajor)
                    .build();

            // Adds multiple endpoint definitions to a single request object.
            UpdateEndpointRequest endpointList = UpdateEndpointRequest.builder()
                    .applicationId(applicationId)
                    .endpointRequest(wangXiulanEndpoint)
                    .endpointId(endPointId)
                    .build();

            UpdateEndpointResponse result = pinpoint.updateEndpoint(endpointList);
            System.out.format("Update endpoint result: %s\n",
result.messageBody().message());

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

전체 SDK 예제를 보려면 GitHub의 AddExampleUser.java를 참조하세요.

HTTP

REST API에 HTTP 요청을 직접 수행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example사용자 정의가 포함된 엔드포인트 Put 요청

엔드포인트에 사용자를 추가하려면 다음 URI에서 엔드포인트 리소스에 PUT 요청을 발행합니다.

`/v1/apps/application-id/endpoints/endpoint-id`

위치:

- *application-id*는 엔드포인트를 추가하거나 업데이트할 Amazon Pinpoint 프로젝트의 ID입니다.
- *endpoint-id*는 새 엔드포인트에 할당할 ID 또는 업데이트할 기존 엔드포인트의 ID입니다.

요청에 필수 헤더를 포함시키고 EndpointRequest JSON을 본문으로 제공합니다. 요청 본문은 사용자가 포함될 수 있는 새 엔드포인트를 정의할 수 있습니다. 또는 기존 엔드포인트를 업데이트하려면 변경할 속성만 제공할 수 있습니다. 다음 예제에서는 사용자 속성만 제공하여 사용자를 기존 엔드포인트에 추가합니다.

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":"Wang",
            "LastName":"Xiulan",
            "Gender":"Female",
            "Age":"39"
        }
    }
}
```

요청이 성공하면 다음과 비슷한 응답이 수신됩니다.

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

관련 정보

지원되는 HTTP 메서드 및 요청 파라미터를 비롯하여, Amazon Pinpoint API의 엔드포인트 리소스에 대한 자세한 내용은 Amazon Pinpoint API 참조의 엔드포인트 섹션을 참조하세요.

변수를 사용하여 메시지를 개인화하는 방법에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 메시지 변수 섹션을 참조하세요.

사용자 ID 목록을 가져와서 세그먼트를 정의하는 방법을 알아보려면 Amazon Pinpoint 사용 설명서의 세그먼트 가져오기 섹션을 참조하세요.

최대 100개의 사용자 ID에 DM(Direct Message)을 보내는 방법에 대한 내용은 Amazon Pinpoint API 참조의 사용자 메시지 섹션을 참조하세요.

할당할 수 있는 사용자 속성의 수를 포함해 엔드포인트에 적용되는 할당량에 대한 자세한 내용은 the section called “엔드포인트 할당량” 단원을 참조하십시오.

Amazon Pinpoint에 엔드포인트 배치 추가

엔드포인트를 배치로 제공하여 단일 작업에서 여러 엔드포인트를 추가하거나 업데이트할 수 있습니다. 각 배치 요청에는 최대 100개의 엔드포인트 정의가 포함될 수 있습니다.

단일 작업에서 100개보다 많은 엔드포인트를 추가하거나 업데이트하려는 경우 그 대신 Amazon Pinpoint로 엔드포인트 가져오기 단원을 참조하십시오.

예제

다음 예제에서는 엔드포인트를 배치 요청에 포함시켜 두 개의 엔드포인트를 한 번에 추가하는 방법을 보여줍니다.

AWS CLI

AWS CLI에서 명령을 실행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 배치 업데이트 명령

엔드포인트 배치 요청을 제출하려면 `update-endpoints-batch` 명령을 사용합니다.

```
$ aws pinpoint update-endpoints-batch \
> --application-id application-id \
> --endpoint-batch-request file://endpoint_batch_request_file.json
```

위치:

- *`application-id`*는 엔드포인트를 추가하거나 업데이트할 Amazon Pinpoint 프로젝트의 ID입니다.
- *`endpoint_batch_request_file.json`*은 `--endpoint-batch-request` 파라미터에 대한 입력이 포함된 로컬 JSON 파일의 파일 경로입니다.

Example엔드포인트 배치 요청 파일

예제 `update-endpoints-batch` 명령은 JSON 파일을 `--endpoint-request` 파라미터의 인수로 사용합니다. 이 파일에는 다음과 같은 엔드포인트 배치 정의가 포함되어 있습니다.

```
{
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
```

```
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
        }
    ]
}
```

엔드포인트를 정의하는 데 사용할 수 있는 속성은 Amazon Pinpoint API 참조의 EndpointBatchRequest 스키마를 참조하세요.

AWS SDK for Java

AWS SDK for Java에서 제공하는 클라이언트를 사용하여 Java 애플리케이션에서 Amazon Pinpoint API를 사용할 수 있습니다.

Example코드

엔드포인트 배치 요청을 제출하려면 `EndpointBatchRequest` 객체를 초기화하고 이를 `AmazonPinpoint` 클라이언트의 `updateEndpointsBatch` 메서드에 전달합니다. 다음 예제에서는 `EndpointBatchRequest` 객체를 두 개의 `EndpointBatchItem` 객체로 채웁니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class AddExampleEndpoints {

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <appId>

                                Where:
                                   appId - The ID of the application.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String applicationId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                updateEndpointsViaBatch(pinpoint, applicationId);
                pinpoint.close();
        }
```

```
       public static void updateEndpointsViaBatch(PinpointClient pinpoint, String
applicationId) {
               try {
                       List<String> myList = new ArrayList<>();
                       myList.add("music");
                       myList.add("books");

                       Map myMap = new HashMap<String, List>();
                       myMap.put("attributes", myList);

                       List<String> myNames = new ArrayList<String>();
                       myList.add("Richard");
                       myList.add("Roe");

                       Map myMap2 = new HashMap<String, List>();
                       myMap2.put("name", myNames);

                       EndpointUser richardRoe = EndpointUser.builder()
                                       .userId("example_user_1")
                                       .userAttributes(myMap2)
                                       .build();

                       // Create an EndpointBatchItem object for Richard Roe.
                       EndpointBatchItem richardRoesEmailEndpoint =
EndpointBatchItem.builder()
                                       .channelType(ChannelType.EMAIL)
                                       .address("richard_roe@example.com")
                                       .id("example_endpoint_1")
                                       .attributes(myMap)
                                       .user(richardRoe)
                                       .build();

                       List<String> myListMary = new ArrayList<String>();
                       myListMary.add("cooking");
                       myListMary.add("politics");
                       myListMary.add("finance");

                       Map myMapMary = new HashMap<String, List>();
                       myMapMary.put("interests", myListMary);

                       List<String> myNameMary = new ArrayList<String>();
                       myNameMary.add("Mary ");
                       myNameMary.add("Major");
```

```
                        Map maryName = new HashMap<String, List>();
                        myMapMary.put("name", myNameMary);

                        EndpointUser maryMajor = EndpointUser.builder()
                                        .userId("example_user_2")
                                        .userAttributes(maryName)
                                        .build();

                        // Create an EndpointBatchItem object for Mary Major.
                        EndpointBatchItem maryMajorsSmsEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.SMS)
                                        .address("+16145550100")
                                        .id("example_endpoint_2")
                                        .attributes(myMapMary)
                                        .user(maryMajor)
                                        .build();

                        // Adds multiple endpoint definitions to a single request
object.
                        EndpointBatchRequest endpointList =
EndpointBatchRequest.builder()
                                        .item(richardRoesEmailEndpoint)
                                        .item(maryMajorsSmsEndpoint)
                                        .build();

                        // Create the UpdateEndpointsBatchRequest.
                        UpdateEndpointsBatchRequest batchRequest =
UpdateEndpointsBatchRequest.builder()
                                        .applicationId(applicationId)
                                        .endpointBatchRequest(endpointList)
                                        .build();

                        // Updates the endpoints with Amazon Pinpoint.
                        UpdateEndpointsBatchResponse result =
pinpoint.updateEndpointsBatch(batchRequest);
                        System.out.format("Update endpoints batch result: %s\n",
result.messageBody().message());

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
```

```
        }
}
```

전체 SDK 예제를 보려면 GitHub의 AddExampleEndpoints.java를 참조하세요.

HTTP

REST API에 HTTP 요청을 직접 수행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 Put 요청

엔드포인트 배치 요청을 제출하려면 다음 URI에서 엔드포인트 리소스에 PUT 요청을 발행합니다.

`/v1/apps/application-id/endpoints`

*application-id*는 엔드포인트를 추가하거나 업데이트할 Amazon Pinpoint 프로젝트의 ID입니다.

요청에 필수 헤더를 포함시키고 EndpointBatchRequest JSON을 본문으로 제공합니다.

```
PUT /v1/apps/application_id/endpoints HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180501T184948Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
```

```
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
        }
    ]
}
```

요청이 성공하면 다음과 비슷한 응답이 수신됩니다.

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

관련 정보

지원되는 HTTP 메서드 및 요청 파라미터를 비롯하여, Amazon Pinpoint API의 엔드포인트 리소스에 대한 자세한 내용은 Amazon Pinpoint API 참조의 엔드포인트 섹션을 참조하세요.

Amazon Pinpoint로 엔드포인트 가져오기

Amazon S3 버킷에서 엔드포인트를 가져와서 많은 수의 엔드포인트를 추가하거나 업데이트할 수 있습니다. 엔드포인트 가져오기는 Amazon Pinpoint 외부의 대상에 대한 레코드가 있고 이 정보를 Amazon Pinpoint 프로젝트에 추가하려는 경우에 유용합니다. 이 경우 다음을 수행합니다.

1. 자체 대상 데이터를 기반으로 하는 엔드포인트 정의를 생성합니다.
2. 이 엔드포인트 정의를 하나 이상의 파일에 저장하고 파일을 Amazon S3 버킷에 업로드합니다.
3. 버킷에서 엔드포인트를 가져와서 엔드포인트를 Amazon Pinpoint 프로젝트에 추가합니다.

각 가져오기 작업을 통해 최대 1GB의 데이터를 전송할 수 있습니다. 각 엔드포인트가 4KB 이하인 일반적인 작업에서는 약 250,000개의 엔드포인트를 가져올 수 있습니다. AWS 계정당 최대 2개의 동시 가져오기 작업을 실행할 수 있습니다. 가져오기 작업에 더 많은 대역폭이 필요한 경우에 서비스 할당량 증가 요청을 제출할 수 있습니다 지원. 자세한 내용은 할당량 증가 요청 단원을 참조하십시오.

시작하기 전 준비 사항

엔드포인트를 가져오려면 먼저 AWS 계정에서 다음 리소스가 필요합니다.

- Amazon S3 버킷. 버킷을 생성하는 방법은 Amazon Simple Storage Service 사용 설명서의 버킷 생성을 참조하세요.
- Amazon S3 버킷에 대한 Amazon Pinpoint 읽기 권한을 부여하는 AWS Identity and Access Management (IAM) 역할입니다. Amazon S3 역할을 생성하려면 엔드포인트 또는 세그먼트를 가져오기 위한 IAM 역할 단원을 참조하십시오.

예제

다음 예제에서는 엔드포인트 정의를 Amazon S3 버킷에 추가한 다음 해당 엔드포인트를 Amazon Pinpoint 프로젝트로 가져오는 방법을 보여 줍니다.

엔드포인트 정의가 있는 파일

Amazon S3 버킷에 추가하는 파일에는 CSV 또는 줄 바꿈으로 구분된 JSON 형식의 엔드포인트 정의가 포함될 수 있습니다. 엔드포인트를 정의하는 데 사용할 수 있는 속성은 Amazon Pinpoint API 참조의 EndpointRequest JSON 스키마를 참조하세요.

CSV

다음 예와 같이 CSV 파일로 정의된 엔드포인트를 가져올 수 있습니다.

```
ChannelType,Address,Location.Country,Demographic.Platform,Demographic.Make,User.UserId
SMS,12065550182,CN,Android,LG,example-user-id-1
APNS,1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f,US,iOS,Apple,example-user-id-2
EMAIL,john.stiles@example.com,US,iOS,Apple,example-user-id-2
```

첫 번째 줄은 엔드포인트 속성을 포함하는 헤더입니다. `Location.Country`와 같이 점 표기법을 사용하여 중첩 속성을 지정합니다.

후속 줄은 헤더에 있는 각 속성의 값을 제공하여 엔드포인트를 정의합니다.

값에 쉼표 또는 큰따옴표를 포함하려면 `"aaa,bbb"`와 같이 값을 큰따옴표로 묶습니다.

CSV의 값 내에서는 줄 바꿈이 지원되지 않습니다.

JSON

다음 예와 같이 줄 바꿈으로 구분된 JSON 파일로 정의된 엔드포인트를 가져올 수 있습니다.

```
{"ChannelType":"SMS","Address":"12065550182","Location":
{"Country":"CN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

이 형식에서 각 줄은 개별 엔드포인트 정의가 포함된 완전한 JSON 객체입니다.

가져오기 작업 요청

다음 예제에서는 로컬 파일을 버킷에 업로드하여 엔드포인트 정의를 Amazon S3에 추가하는 방법을 보여줍니다. 그런 다음, 이 예제에서는 엔드포인트 정의를 Amazon Pinpoint 프로젝트로 가져옵니다.

AWS CLI

AWS CLI에서 명령을 실행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example S3 CP 명령

Amazon S3 버킷에 로컬 파일을 업로드하려면 Amazon S3 cp 명령을 사용합니다.

```
$ aws s3 cp ./endpoints-file s3://bucket-name/prefix/
```

위치:

- ./endpoints-file은 엔드포인트 정의가 포함된 로컬 파일의 파일 경로입니다.
- bucket-name/prefix/는 Amazon S3 버킷의 이름 및 선택 사항으로 버킷의 객체를 계층 구조로 정리하는 데 도움이 되는 접두사입니다. 예를 들어, 유용한 접두사는 `pinpoint/imports/endpoints/`일 수 있습니다.

Example가져오기 작업 생성 명령

Amazon S3 버킷에서 엔드포인트 정의를 가져오려면 `create-import-job` 명령을 사용합니다.

```
$ aws pinpoint create-import-job \
> --application-id application-id \
> --import-job-request \
> S3Url=s3://bucket-name/prefix/key,\
> RoleArn=iam-import-role-arn,\
> Format=format,\
> RegisterEndpoints=true
```

위치:

- application-id는 엔드포인트를 가져올 Amazon Pinpoint 프로젝트의 ID입니다.
- bucket-name/prefix/key는 가져올 객체가 하나 이상 포함된 Amazon S3 내의 위치입니다. 이 위치는 개별 객체의 키로 끝나거나, 여러 객체를 한정하는 접두사로 끝날 수 있습니다.

- iam-import-role-arn은 버킷에 Amazon Pinpoint 읽기 액세스를 부여하는 IAM 역할의 Amazon 리소스 이름(ARN)입니다.
- format은 엔드포인트를 정의하는 데 사용한 형식에 따라 `JSON` 또는 `CSV`일 수 있습니다. Amazon S3 위치에 혼합된 형식의 여러 객체가 포함된 경우 Amazon Pinpoint는 지정된 형식과 일치하는 객체만 가져옵니다.
- RegisterEndpoints는 `true` 또는 `false` 중 하나일 수 있습니다. true로 설정하면 엔드포인트 정의를 가져올 경우 가져오기 작업이 Amazon Pinpoint에 엔드포인트를 등록합니다.

  RegisterEndpoints와 DefineSegments 조합

RegisterEndpoints	DefineSegments	설명
true	true	Amazon Pinpoint가 엔드포인트를 가져오고 엔드포인트가 포함된 세그먼트를 생성합니다.
true	false	Amazon Pinpoint가 엔드포인트를 가져오지만 세그먼트를 생성하지는 않습니다.
false	true	Amazon Pinpoint가 엔드포인트를 가져오고 엔드포인트가 포함된 세그먼트를 생성합니다. 엔드포인트가 저장되지 않으며 기존 엔드포인트를 덮어쓰지 않습니다.
false	false	Amazon Pinpoint가 이 요청을 거부합니다.

응답에는 가져오기 작업에 대한 세부 정보가 포함되어 있습니다.

```
{
    "ImportJobResponse": {
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
```

```
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://bucket-name/prefix/key"
        },
        "Id": "d5ecad8e417d498389e1d5b9454d4e0c",
        "JobStatus": "CREATED",
        "Type": "IMPORT"
    }
}
```

응답은 `Id` 속성이 있는 작업 ID를 제공합니다. 이 ID를 사용하여 가져오기 작업의 현재 상태를 확인할 수 있습니다.

Example가져오기 작업 Get 명령

가져오기 작업의 현재 상태를 확인하려면 `get-import-job` 명령을 사용합니다.

```
$ aws pinpoint get-import-job \
> --application-id application-id \
> --job-id job-id
```

위치:

- application-id는 가져오기 작업이 초기화된 Amazon Pinpoint 프로젝트의 ID입니다.
- job-id는 확인할 가져오기 작업의 ID입니다.

이 명령에 대한 응답은 가져오기 작업의 현재 상태를 제공합니다.

```
{
    "ImportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-24T21:26:45.308Z",
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
```

```
            "S3Url": "s3://s3-bucket-name/prefix/endpoint-definitions.json"
        },
        "FailedPieces": 0,
        "Id": "job-id",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 3,
        "Type": "IMPORT"
    }
}
```

응답은 `JobStatus` 속성이 있는 작업 상태를 제공합니다.

AWS SDK for Java

AWS SDK for Java에서 제공하는 클라이언트를 사용하여 Java 애플리케이션에서 Amazon Pinpoint API를 사용할 수 있습니다.

Example코드

엔드포인트 정의가 포함된 파일을 Amazon S3에 업로드하려면 `AmazonS3` 클라이언트의 `putObject` 메서드를 사용합니다.

엔드포인트를 Amazon Pinpoint 프로젝트로 가져오려면 `CreateImportJobRequest` 객체를 초기화합니다. 그런 다음 이 객체를 `AmazonPinpoint` 클라이언트의 `createImportJob` 메서드에 전달합니다.

```
package com.amazonaws.examples.pinpoint;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.CreateImportJobRequest;
import com.amazonaws.services.pinpoint.model.CreateImportJobResult;
import com.amazonaws.services.pinpoint.model.Format;
import com.amazonaws.services.pinpoint.model.GetImportJobRequest;
import com.amazonaws.services.pinpoint.model.GetImportJobResult;
import com.amazonaws.services.pinpoint.model.ImportJobRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AmazonS3Exception;
```

```
import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.List;
import java.util.concurrent.TimeUnit;

public class ImportEndpoints {

    public static void main(String[] args) {

        final String USAGE = "\n" +
                "ImportEndpoints - Adds endpoints to an Amazon Pinpoint application
 by: \n" +
                "1.) Uploading the endpoint definitions to an Amazon S3 bucket. \n"
 +
                "2.) Importing the endpoint definitions from the bucket to an Amazon
 Pinpoint " +
                "application.\n\n" +
                "Usage: ImportEndpoints <endpointsFileLocation> <s3BucketName>
 <iamImportRoleArn> " +
                "<applicationId>\n\n" +
                "Where:\n" +
                "  endpointsFileLocation - The relative location of the JSON file
 that contains the " +
                "endpoint definitions.\n" +
                "  s3BucketName - The name of the Amazon S3 bucket to upload the
 JSON file to. If the " +
                "bucket doesn't exist, a new bucket is created.\n" +
                "  iamImportRoleArn - The ARN of an IAM role that grants Amazon
 Pinpoint read " +
                "permissions to the S3 bucket.\n" +
                "  applicationId - The ID of the Amazon Pinpoint application to add
 the endpoints to.";

        if (args.length < 1) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String endpointsFileLocation = args[0];
        String s3BucketName = args[1];
        String iamImportRoleArn = args[2];
        String applicationId = args[3];
```

```
        Path endpointsFilePath = Paths.get(endpointsFileLocation);
        File endpointsFile = new
File(endpointsFilePath.toAbsolutePath().toString());
        uploadToS3(endpointsFile, s3BucketName);

        importToPinpoint(endpointsFile.getName(), s3BucketName, iamImportRoleArn,
applicationId);

    }

    private static void uploadToS3(File endpointsFile, String s3BucketName) {

        // Initializes Amazon S3 client.
        final AmazonS3 s3 = AmazonS3ClientBuilder.defaultClient();

        // Checks whether the specified bucket exists. If not, attempts to create
one.
        if (!s3.doesBucketExistV2(s3BucketName)) {
            try {
                s3.createBucket(s3BucketName);
                System.out.format("Created S3 bucket %s.\n", s3BucketName);
            } catch (AmazonS3Exception e) {
                System.err.println(e.getErrorMessage());
                System.exit(1);
            }
        }

        // Uploads the endpoints file to the bucket.
        String endpointsFileName = endpointsFile.getName();
        System.out.format("Uploading %s to S3 bucket %s . . .\n", endpointsFileName,
s3BucketName);
        try {
            s3.putObject(s3BucketName, "imports/" + endpointsFileName,
endpointsFile);
            System.out.println("Finished uploading to S3.");
        } catch (AmazonServiceException e) {
            System.err.println(e.getErrorMessage());
            System.exit(1);
        }
    }

    private static void importToPinpoint(String endpointsFileName, String
s3BucketName,
            String iamImportRoleArn, String applicationId) {
```

```
        // The S3 URL that Amazon Pinpoint requires to find the endpoints file.
        String s3Url = "s3://" + s3BucketName + "/imports/" + endpointsFileName;

        // Defines the import job that Amazon Pinpoint runs.
        ImportJobRequest importJobRequest = new ImportJobRequest()
                .withS3Url(s3Url)
                .withRegisterEndpoints(true)
                .withRoleArn(iamImportRoleArn)
                .withFormat(Format.JSON);
        CreateImportJobRequest createImportJobRequest = new CreateImportJobRequest()
                .withApplicationId(applicationId)
                .withImportJobRequest(importJobRequest);

        // Initializes the Amazon Pinpoint client.
        AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
                .withRegion(Regions.US_EAST_1).build();

        System.out.format("Importing endpoints in %s to Amazon Pinpoint application
%s . . .\n",
                endpointsFileName, applicationId);

        try {

            // Runs the import job with Amazon Pinpoint.
            CreateImportJobResult importResult =
pinpointClient.createImportJob(createImportJobRequest);

            String jobId = importResult.getImportJobResponse().getId();
            GetImportJobResult getImportJobResult = null;
            String jobStatus = null;

            // Checks the job status until the job completes or fails.
            do {
                getImportJobResult = pinpointClient.getImportJob(new
GetImportJobRequest()
                        .withJobId(jobId)
                        .withApplicationId(applicationId));
                jobStatus =
getImportJobResult.getImportJobResponse().getJobStatus();
                System.out.format("Import job %s . . .\n", jobStatus.toLowerCase());
                TimeUnit.SECONDS.sleep(3);
            } while (!jobStatus.equals("COMPLETED") && !jobStatus.equals("FAILED"));
```

```
            if (jobStatus.equals("COMPLETED")) {
                System.out.println("Finished importing endpoints.");
            } else {
                System.err.println("Failed to import endpoints.");
                System.exit(1);
            }

            // Checks for entries that failed to import.
            // getFailures provides up to 100 of the first failed entries for the
 job, if
            // any exist.
            List<String> failedEndpoints =
 getImportJobResult.getImportJobResponse().getFailures();
            if (failedEndpoints != null) {
                System.out.println("Failed to import the following entries:");
                for (String failedEndpoint : failedEndpoints) {
                    System.out.println(failedEndpoint);
                }
            }

        } catch (AmazonServiceException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }

    }

}
```

HTTP

REST API에 HTTP 요청을 직접 수행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example S3 객체 PUT 요청

엔드포인트 정의를 버킷에 추가하려면 Amazon S3 PUT 객체 작업을 사용하고 엔드포인트 정의를 본문으로 제공합니다.

```
PUT /prefix/key HTTP/1.1
Content-Type: text/plain
Accept: application/json
Host: bucket-name.s3.amazonaws.com
X-Amz-Content-Sha256:
 c430dc094b0cec2905bc88d96314914d058534b14e2bc6107faa9daa12fdff2d
```

```
X-Amz-Date: 20180605T184132Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/s3/aws4_request, SignedHeaders=accept;cache-control;content-
length;content-type;host;postman-token;x-amz-content-sha256;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{"ChannelType":"SMS","Address":"2065550182","Location":
{"Country":"CAN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

위치:

- /prefix/key는 업로드 후 엔드포인트 정의가 포함될 객체의 접두사 및 키 이름입니다. 접두사를 사용하여 객체를 계층 구조로 정리할 수 있습니다. 예를 들어, 유용한 접두사는 `pinpoint/imports/endpoints/`일 수 있습니다.
- bucket-name은 엔드포인트 정의를 추가할 Amazon S3 버킷의 이름입니다.

Example가져오기 작업 POST 요청

Amazon S3 버킷에서 엔드포인트 정의를 가져오려면 가져오기 작업 리소스에 POST 요청을 실행합니다. 요청에 필수 헤더를 포함시키고 ImportJobRequest JSON을 본문으로 제공합니다.

```
POST /v1/apps/application_id/jobs/import HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T214912Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3Url": "s3://bucket-name/prefix/key",
```

```
  "RoleArn": "iam-import-role-arn",
  "Format": "format",
  "RegisterEndpoints": true
}
```

위치:

- application-id는 엔드포인트를 가져올 Amazon Pinpoint 프로젝트의 ID입니다.
- bucket-name/prefix/key는 가져올 객체가 하나 이상 포함된 Amazon S3 내의 위치입니다. 이 위치는 개별 객체의 키로 끝나거나, 여러 객체를 한정하는 접두사로 끝날 수 있습니다.
- iam-import-role-arn은 버킷에 Amazon Pinpoint 읽기 액세스를 부여하는 IAM 역할의 Amazon 리소스 이름(ARN)입니다.
- format은 엔드포인트를 정의하는 데 사용한 형식에 따라 `JSON` 또는 `CSV`일 수 있습니다. Amazon S3 위치에 혼합된 형식의 여러 파일이 포함된 경우 Amazon Pinpoint는 지정된 형식과 일치하는 파일만 가져옵니다.

요청이 성공하면 다음과 비슷한 응답이 수신됩니다.

```
{
    "Id": "a995ce5d70fa44adb563b7d0e3f6c6f5",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-05T21:49:15.288Z",
    "Type": "IMPORT",
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key",
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

응답은 `Id` 속성이 있는 작업 ID를 제공합니다. 이 ID를 사용하여 가져오기 작업의 현재 상태를 확인할 수 있습니다.

Example가져오기 작업 GET 요청

가져오기 작업의 현재 상태를 확인하려면 가져오기 작업 리소스에 `GET` 요청을 발행합니다.

```
GET /v1/apps/application_id/jobs/import/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T220744Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

위치:

- application_id는 가져오기 작업이 초기화된 Amazon Pinpoint 프로젝트의 ID입니다.
- job_id는 확인할 가져오기 작업의 ID입니다.

요청이 성공하면 다음과 비슷한 응답이 수신됩니다.

```
{
    "ApplicationId": "application_id",
    "Id": "70a51b2cf442447492d2c8e50336a9e8",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-05T22:04:49.213Z",
    "CompletionDate": "2018-06-05T22:04:58.034Z",
    "Type": "IMPORT",
    "TotalFailures": 0,
    "TotalProcessed": 3,
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key.json",
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

응답은 `JobStatus` 속성이 있는 작업 상태를 제공합니다.

관련 정보

지원되는 HTTP 메서드 및 요청 파라미터를 비롯하여, Amazon Pinpoint API의 가져오기 작업 리소스에 대한 자세한 내용은 Amazon Pinpoint API 참조의 가져오기 작업 섹션을 참조하세요.

Amazon Pinpoint에서 Amazon S3 버킷으로 엔드포인트 내보내기

Amazon Pinpoint에 있는 대상에 대한 정보를 모두 가져오려면 프로젝트에 속한 엔드포인트 정의를 내보낼 수 있습니다. 내보내기를 수행하면 Amazon Pinpoint는 사용자가 지정하는 Amazon S3 버킷에 엔드포인트 정의를 배치합니다. 엔드포인트 내보내기는 다음을 수행하려는 경우에 유용합니다.

- 클라이언트 애플리케이션이 Amazon Pinpoint에 등록한 신규 및 기존 엔드포인트에 대한 최신 데이터를 봅니다.
- Amazon Pinpoint에 있는 엔드포인트 데이터를 자체 고객 관계 관리(CRM) 시스템과 동기화합니다.
- 고객 데이터에 대한 보고서를 생성하거나 고객 데이터를 분석합니다.

> **Note**
>
> Amazon S3 버킷에 전달한 콘텐츠에는 고객 콘텐츠가 포함될 수 있습니다. Amazon S3 버킷으로 내보낸 엔드포인트 데이터를 삭제해야 하는 경우 Amazon S3에서 삭제해야 합니다. 민감한 데이터 제거에 대한 자세한 내용은 S3 버킷을 비우는 방법 또는 S3 버킷을 삭제하는 방법을 참조하세요.

시작하기 전 준비 사항

엔드포인트를 내보내려면 먼저 AWS 계정에서 다음 리소스가 필요합니다.

- Amazon S3 버킷. 버킷을 생성하는 방법은 Amazon Simple Storage Service 사용 설명서의 버킷 생성을 참조하세요.
- Amazon S3 버킷에 대한 Amazon Pinpoint 쓰기 권한을 부여하는 AWS Identity and Access Management(IAM) 역할입니다. 역할을 생성하려면 엔드포인트 또는 세그먼트를 내보내기 위한 IAM 역할 단원을 참조하십시오.

예시

다음 예제에서는 Amazon Pinpoint 프로젝트에서 엔드포인트를 내보낸 다음 Amazon S3 버킷에서 해당 엔드포인트를 다운로드하는 방법을 보여줍니다.

AWS CLI

AWS CLI에서 명령을 실행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example내보내기 작업 생성 명령

Amazon Pinpoint 프로젝트에서 엔드포인트를 내보내려면 `create-export-job` 명령을 사용합니다.

```
$ aws pinpoint create-export-job \
> --application-id application-id \
> --export-job-request \
> S3UrlPrefix=s3://bucket-name/prefix/,\
> RoleArn=iam-export-role-arn
```

위치:

- *application-id*는 엔드포인트가 포함된 Amazon Pinpoint 프로젝트의 ID입니다.
- *bucket-name/prefix/*는 Amazon S3 버킷의 이름이며, 선택 사항으로 버킷의 객체를 계층 구조로 정리하는 데 도움이 되는 접두사입니다. 예를 들어, 유용한 접두사는 `pinpoint/exports/endpoints/`일 수 있습니다.
- *iam-export-role-arn*은 Amazon Pinpoint 쓰기 액세스 권한을 버킷에 부여하는 IAM 역할의 Amazon 리소스 이름(ARN)입니다.

이 명령에 대한 응답은 내보내기 작업에 대한 세부 정보를 제공합니다.

```
{
    "ExportJobResponse": {
        "CreationDate": "2018-06-04T22:04:20.585Z",
        "Definition": {
            "RoleArn": "iam-export-role-arn",
            "S3UrlPrefix": "s3://s3-bucket-name/prefix/"
        },
        "Id": "7390e0de8e0b462380603c5a4df90bc4",
        "JobStatus": "CREATED",
```

```
        "Type": "EXPORT"
    }
}
```

응답은 `Id` 속성이 있는 작업 ID를 제공합니다. 이 ID를 사용하여 내보내기 작업의 현재 상태를 확인할 수 있습니다.

Example내보내기 작업 Get 명령

내보내기 작업의 현재 상태를 확인하려면 `get-export-job` 명령을 사용합니다.

```
$ aws pinpoint get-export-job \
> --application-id application-id \
> --job-id job-id
```

위치:

- *application-id*는 엔드포인트를 내보낸 Amazon Pinpoint 프로젝트의 ID입니다.
- *job-id*는 확인할 작업의 ID입니다.

이 명령에 대한 응답은 내보내기 작업의 현재 상태를 제공합니다.

```
{
    "ExportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-08T22:16:48.228Z",
        "CreationDate": "2018-05-08T22:16:44.812Z",
        "Definition": {},
        "FailedPieces": 0,
        "Id": "6c99c463f14f49caa87fa27a5798bef9",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 215,
        "Type": "EXPORT"
    }
}
```

응답은 `JobStatus` 속성이 있는 작업 상태를 제공합니다. 작업 상태 값이 `COMPLETED`이면 내보낸 엔드포인트를 Amazon S3 버킷에서 가져올 수 있습니다.

Example S3 CP 명령

내보낸 엔드포인트를 다운로드하려면 Amazon S3 cp 명령을 사용합니다.

```
$ aws s3 cp s3://bucket-name/prefix/key.gz /local/directory/
```

위치:

- *bucket-name/prefix/key*는 엔드포인트를 내보낼 때 Amazon Pinpoint가 버킷에 추가한 .gz 파일의 위치입니다. 이 파일에는 내보낸 엔드포인트 정의가 포함되어 있습니다. 예를 들어, `https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv` URL에서 `PINPOINT-EXAMPLE-BUCKET`은 버킷의 이름이고 `Exports/example.csv`는 키입니다. 키에 대한 자세한 내용은 Amazon S3 사용 설명서의 키 섹션을 참조하십시오.
- */local/directory/*는 엔드포인트를 다운로드할 로컬 디렉터리의 파일 경로입니다.

AWS SDK for Java

AWS SDK for Java에서 제공하는 클라이언트를 사용하여 Java 애플리케이션에서 Amazon Pinpoint API를 사용할 수 있습니다.

Example코드

Amazon Pinpoint 프로젝트에서 엔드포인트를 내보내려면 `CreateExportJobRequest` 객체를 초기화합니다. 그런 다음 이 객체를 `AmazonPinpoint` 클라이언트의 `createExportJob` 메서드에 전달합니다.

Amazon Pinpoint에서 내보낸 엔드포인트를 다운로드하려면 `AmazonS3` 클라이언트의 `getObject` 메서드를 사용합니다.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
```

```
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;
```

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;
```

```
/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and prints
 them.
                Usage: ExportEndpoints <applicationId> <s3BucketName>
 <iamExportRoleArn> <path>

                Where:
                  applicationId - The ID of the Amazon Pinpoint application that has
 the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the JSON
 file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
 Pinpoint write permissions to the S3 bucket.  path - The path where the files
 downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
                """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
```

```
        String s3BucketName = args[1];
        String iamExportRoleArn = args[2];
        String path = args[3];
        System.out.println("Deleting an application with ID: " + applicationId);

        Region region = Region.US_EAST_1;
        PinpointClient pinpoint = PinpointClient.builder()
                .region(region)
                .build();

        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
        pinpoint.close();
        s3Client.close();
    }

    public static void exportAllEndpoints(PinpointClient pinpoint,
            S3Client s3Client,
            String applicationId,
            String s3BucketName,
            String path,
            String iamExportRoleArn) {

        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint, S3Client
s3Client, String s3BucketName,
```

```
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
                    .build();

            System.out.format("Exporting endpoints from Amazon Pinpoint application
 %s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);

            CreateExportJobResponse exportResult =
 pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
```

```
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                    .jobId(jobId)
                    .applicationId(applicationId)
                    .build();

            do {
                getExportJobResult = pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
```

```
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path location.
    public static void downloadFromS3(S3Client s3Client, String path, String
 s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
 s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();

                // Write the data to a local file.
                String fileSuffix = new
 SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

전체 SDK 예제를 보려면 GitHub의 ExportEndpoints.java를 참조하세요.

HTTP

REST API에 HTTP 요청을 직접 수행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example내보내기 작업 POST 요청

Amazon Pinpoint 프로젝트의 엔드포인트를 내보내려면 작업 내보내기 리소스에 POST 요청을 발행합니다.

```
POST /v1/apps/application_id/jobs/export HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T001238Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3UrlPrefix": "s3://bucket-name/prefix",
  "RoleArn": "iam-export-role-arn"
}
```

위치:

- *application-id*는 엔드포인트가 포함된 Amazon Pinpoint 프로젝트의 ID입니다.
- *bucket-name/prefix*는 Amazon S3 버킷의 이름이며, 선택 사항으로 버킷의 객체를 계층 구조로 정리하는 데 도움이 되는 접두사입니다. 예를 들어, 유용한 접두사는 `pinpoint/exports/endpoints/`일 수 있습니다.
- *iam-export-role-arn*은 Amazon Pinpoint 쓰기 액세스 권한을 버킷에 부여하는 IAM 역할의 Amazon 리소스 이름(ARN)입니다.

이 요청에 대한 응답은 내보내기 작업에 대한 세부 정보를 제공합니다.

```
{
    "Id": "611bdc54c75244bfa51fe7001ddb2e36",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "Type": "EXPORT",
    "Definition": {
        "S3UrlPrefix": "s3://bucket-name/prefix",
        "RoleArn": "iam-export-role-arn"
```

```
    }
}
```

응답은 Id 속성이 있는 작업 ID를 제공합니다. 이 ID를 사용하여 내보내기 작업의 현재 상태를 확인할 수 있습니다.

Example내보내기 작업 GET 요청

내보내기 작업의 현재 상태를 확인하려면 내보내기 작업 리소스에 GET 요청을 발행합니다.

```
GET /v1/apps/application_id/jobs/export/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T002443Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

위치:

- *application-id*는 엔드포인트를 내보낸 Amazon Pinpoint 프로젝트의 ID입니다.
- *job-id*는 확인할 작업의 ID입니다.

이 요청에 대한 응답은 내보내기 작업의 현재 상태를 제공합니다.

```
{
    "ApplicationId": "application_id",
    "Id": "job_id",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "CompletionDate": "2018-06-06T00:13:01.141Z",
    "Type": "EXPORT",
    "TotalFailures": 0,
    "TotalProcessed": 217,
    "Definition": {}
```

```
}
```

응답은 `JobStatus` 속성이 있는 작업 상태를 제공합니다. 작업 상태 값이 `COMPLETED`이면 내보낸 엔드포인트를 Amazon S3 버킷에서 가져올 수 있습니다.

관련 정보

특정 엔드포인트의 엔드포인트 ID를 찾으려면 엔드포인트가 속한 세그먼트를 확인한 다음 Amazon Pinpoint에서 해당 세그먼트를 내보냅니다. 내보낸 데이터는 각 엔드포인트의 엔드포인트 ID를 포함합니다. Amazon Pinpoint 콘솔을 사용하여 세그먼트를 파일로 내보낼 수 있습니다. 세그먼트 내보내기에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 세그먼트 내보내기를 참조하세요.

지원되는 HTTP 메서드 및 요청 파라미터를 비롯하여, Amazon Pinpoint API의 내보내기 작업 리소스에 대한 자세한 내용은 Amazon Pinpoint API 참조의 내보내기 작업 섹션을 참조하세요.

Amazon Pinpoint 프로젝트에서 엔드포인트 조회

Amazon Pinpoint 프로젝트에 추가된 개별 엔드포인트에 대한 세부 정보를 조회할 수 있습니다. 이러한 세부 정보에는 메시지의 대상 주소, 메시징 채널, 사용자의 디바이스에 대한 데이터, 사용자의 위치에 대한 데이터, 엔드포인트에서 기록하는 사용자 지정 속성이 포함됩니다.

엔드포인트를 조회하려면 엔드포인트 ID가 필요합니다. ID를 모르는 경우 그 대신 내보내기를 통해 엔드포인트 데이터를 가져올 수 있습니다. 엔드포인트를 내보내려면 the section called “Amazon Pinpoint에서 Amazon S3 버킷으로 엔드포인트 내보내기” 단원을 참조하십시오.

예시

다음 예제에서는 ID를 지정하여 개별 엔드포인트를 조회하는 방법을 보여 줍니다.

AWS CLI

AWS CLI에서 명령을 실행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 Get 명령

엔드포인트를 조회하려면 `get-endpoint` 명령을 사용합니다.

```
$ aws pinpoint get-endpoint \
> --application-id application-id \
```

```
> --endpoint-id endpoint-id
```

위치:

- *application-id*는 엔드포인트가 포함된 Amazon Pinpoint 프로젝트의 ID입니다.
- *endpoint-id*는 조회할 엔드포인트의 ID입니다.

이 명령에 대한 응답은 다음 예제와 같은 엔드포인트의 JSON 정의입니다.

```
{
    "EndpointResponse": {
        "Address":
 "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
        "ApplicationId": "application-id",
        "Attributes": {
            "Interests": [
                "Technology",
                "Music",
                "Travel"
            ]
        },
        "ChannelType": "APNS",
        "CohortId": "63",
        "CreationDate": "2018-05-01T17:31:01.046Z",
        "Demographic": {
            "AppVersion": "1.0",
            "Make": "apple",
            "Model": "iPhone",
            "ModelVersion": "8",
            "Platform": "ios",
            "PlatformVersion": "11.3.1",
            "Timezone": "America/Los_Angeles"
        },
        "EffectiveDate": "2018-05-07T19:03:29.963Z",
        "EndpointStatus": "ACTIVE",
        "Id": "example_endpoint",
        "Location": {
            "City": "Seattle",
            "Country": "US",
            "Latitude": 47.6,
            "Longitude": -122.3,
            "PostalCode": "98121"
```

```
        },
        "Metrics": {
            "music_interest_level": 6.0,
            "travel_interest_level": 4.0,
            "technology_interest_level": 9.0
        },
        "OptOut": "ALL",
        "RequestId": "7f546cac-6858-11e8-adcd-2b5a07aab338",
        "User": {
            "UserAttributes": {
                "Gender": "Female",
                "FirstName": "Wang",
                "LastName": "Xiulan",
                "Age": "39"
            },
            "UserId": "example_user"
        }
    }
}
```

AWS SDK for Java

AWS SDK for Java에서 제공하는 클라이언트를 사용하여 Java 애플리케이션에서 Amazon Pinpoint API를 사용할 수 있습니다.

Example코드

엔드포인트를 조회하려면 `GetEndpointRequest` 객체를 초기화합니다. 그런 다음 이 객체를 `AmazonPinpoint` 클라이언트의 `getEndpoint` 메서드에 전달합니다.

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
```

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <endpoint>

                Where:
                  appId - The ID of the application to delete.
                  endpoint - The ID of the endpoint.\s
                  """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String appId,
 String endpoint) {
```

```
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
            System.out.println(endpointJson);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

엔드포인트 데이터를 읽을 수 있는 형식으로 인쇄하기 위해 이 예제에서는 Google GSON 라이브러리를 사용하여 EndpointResponse 객체를 JSON 문자열로 변환합니다.

HTTP

REST API에 HTTP 요청을 직접 수행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 GET 요청

엔드포인트를 조회하려면 엔드포인트 리소스에 GET 요청을 발행합니다.

```
GET /v1/apps/application_id/endpoints/endpoint_id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

위치:

- *application-id*는 엔드포인트가 포함된 Amazon Pinpoint 프로젝트의 ID입니다.
- *endpoint-id*는 조회할 엔드포인트의 ID입니다.

이 요청에 대한 응답은 다음 예제와 같은 엔드포인트의 JSON 정의입니다.

```
{
    "ChannelType": "APNS",
    "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "RequestId": "b720cfa8-6924-11e8-aeda-0b22e0b0fa59",
    "Location": {
        "Latitude": 47.6,
        "Longitude": -122.3,
        "PostalCode": "98121",
        "City": "Seattle",
        "Country": "US"
    },
    "Demographic": {
        "Make": "apple",
        "Model": "iPhone",
        "ModelVersion": "8",
        "Timezone": "America/Los_Angeles",
        "AppVersion": "1.0",
        "Platform": "ios",
        "PlatformVersion": "11.3.1"
    },
    "EffectiveDate": "2018-06-06T00:58:19.865Z",
    "Attributes": {
        "Interests": [
            "Technology",
            "Music",
            "Travel"
        ]
    },
    "Metrics": {
        "music_interest_level": 6,
        "travel_interest_level": 4,
        "technology_interest_level": 9
    },
```

```
    "User": {},
    "ApplicationId": "application_id",
    "Id": "example_endpoint",
    "CohortId": "39",
    "CreationDate": "2018-06-06T00:58:19.865Z"
}
```

관련 정보

Amazon Pinpoint API의 엔드포인트 리소스에 대한 자세한 내용은 Amazon Pinpoint API 참조의 엔드포인트 섹션을 참조하세요.

Amazon Pinpoint를 사용하여 엔드포인트 ID 나열

엔드포인트를 업데이트하거나 삭제하려면 엔드포인트 ID가 필요합니다. 따라서 Amazon Pinpoint 프로젝트의 모든 엔드포인트에서 이러한 작업을 수행하려는 경우 첫 번째 단계는 해당 프로젝트에 속한 모든 엔드포인트 ID를 나열하는 것입니다. 그런 다음, 이러한 ID를 반복하여 예를 들면 글로벌로 속성을 추가하거나 프로젝트에 있는 모든 엔드포인트를 삭제할 수 있습니다.

다음 예제에서는 AWS SDK for Java를 사용하여 다음을 수행합니다.

1. Amazon Pinpoint에서 엔드포인트 내보내기의 예제 코드에서 예제 `exportEndpointsToS3` 메서드를 호출합니다. 이 메서드는 Amazon Pinpoint 프로젝트에서 엔드포인트 정의를 내보냅니다. 엔드포인트 정의는 gzip 파일로 Amazon S3 버킷에 추가됩니다.
2. 내보낸 gzip 파일을 다운로드합니다.
3. gzip 파일을 읽고 각 엔드포인트의 JSON 정의에서 엔드포인트 ID를 가져옵니다.
4. 엔드포인트 ID를 콘솔로 인쇄합니다.
5. Amazon Pinpoint가 Amazon S3에 추가한 파일을 삭제하여 정리합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <applicationId> <userId>

                Where:
                   applicationId - The ID of the Amazon Pinpoint application that has
 the endpoint.
                   userId - The user id applicable to the endpoints""";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String userId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllEndpoints(pinpoint, applicationId, userId);
        pinpoint.close();
    }

    public static void listAllEndpoints(PinpointClient pinpoint,
```

```
            String applicationId,
            String userId) {

        try {
            GetUserEndpointsRequest endpointsRequest =
 GetUserEndpointsRequest.builder()
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

            GetUserEndpointsResponse response =
 pinpoint.getUserEndpoints(endpointsRequest);
            List<EndpointResponse> endpoints = response.endpointsResponse().item();

            // Display the results.
            for (EndpointResponse endpoint : endpoints) {
                System.out.println("The channel type is: " + endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

전체 SDK 예제를 보려면 GitHub의 ListEndpointIs.java를 참조하세요.

Amazon Pinpoint의 최대 엔드포인트 수 관리

각 대상 멤버는 UserId와 연결된 엔드포인트를 최대 15개까지 보유할 수 있습니다(엔드포인트 할당량 참조). 16번째 엔드포인트를 추가하려고 할 경우 ChannelType에 따라 BadRequestException이 발생하거나, EffectiveDate가 가장 오래된 엔드포인트를 제거하여 추가 작업이 성공합니다.

16번째 엔드포인트 추가

- 엔드포인트의 새 채널 유형이 SMS, PUSH, VOICE, EMAIL, CUSTOM 또는 IN_APP인 경우 대상 멤버가 최대 엔드포인트 수에 도달하므로, BadRequestException이 반환됩니다. 대상 멤버와 연결된 엔드포인트를 제거하고 다시 시도해야 합니다. Amazon Pinpoint에서 프로그래밍 방식으로 엔드포인트 삭제 섹션을 참조하세요.

- 엔드포인트의 새 채널 유형이 ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX 또는 BAIDU인 경우:
  - 대상 멤버와 현재 연결된 하나 이상의 엔드포인트에 ADM, GCM, APNS, APNS_VOICE, APNS_VOIP_SANDBOX 또는 BAIDU라는 ChannelType이 있는지 확인합니다. 없는 경우 BadRequestException이 반환되며, 다시 시도하기 전에 엔드포인트를 제거해야 합니다(Amazon Pinpoint에서 프로그래밍 방식으로 엔드포인트 삭제 참조).
  - 그렇지 않으면 ChannelType이 ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX 또는 BAIDU인 경우 EffectiveDate가 가장 오래된 엔드포인트는 `INACTIVE`로 설정됩니다.
    - 이전 엔드포인트의 UserId가 제거됩니다.
    - 새 엔드포인트는 대상 멤버와 연결되며 대상 멤버의 엔드포인트 수는 여전히 최대입니다.

상태를 `ACTIVE`로 설정하고 UserId를 엔드포인트에 다시 추가하여 엔드포인트를 다시 활성화할 수 있습니다.

Amazon Pinpoint에서 프로그래밍 방식으로 엔드포인트 삭제

엔드포인트는 고객 중 한 명에게 접촉하기 위한 단일 방법입니다. 각 엔드포인트는 고객의 이메일 주소, 모바일 디바이스 식별자, 전화번호 또는 메시지를 보낼 수 있는 기타 유형의 대상을 참조할 수 있습니다. 대부분 지역에서 이 유형의 정보는 개인 정보로 간주될 수 있습니다. 특정 대상에 더 이상 메시지를 보내고 싶지 않은 경우(예: 목적지에 연결할 수 없거나, 고객이 계정을 폐쇄하는 경우), 엔드포인트를 삭제할 수 있습니다.

예시

다음 예제에서는 엔드포인트를 삭제하는 방법을 보여줍니다.

AWS CLI

AWS CLI에서 명령을 실행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 삭제 명령

엔드포인트를 삭제하려면 `delete-endpoint` 명령을 사용합니다.

```
$ aws pinpoint delete-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id
```

위치:

- application-id는 엔드포인트가 포함된 Amazon Pinpoint 프로젝트의 ID입니다.
- endpoint-id는 삭제할 엔드포인트의 ID입니다.

이 요청에 대한 응답은 삭제한 엔드포인트의 JSON 정의입니다.

AWS SDK for Java

AWS SDK for Java에서 제공하는 클라이언트를 사용하여 Java 애플리케이션에서 Amazon Pinpoint API를 사용할 수 있습니다.

Example코드

엔드포인트를 삭제하려면 `AmazonPinpoint` 클라이언트의 `deleteEndpoint` 메서드를 사용합니다. `DeleteEndpointRequest` 객체를 메서드 인수로 제공합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appName> <endpointId >
```

```
                Where:
                  appId - The id of the application to delete.
                  endpointId - The id of the endpoint to delete.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
 String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

전체 SDK 예제를 보려면 GitHub의 DeleteEndpoint.java를 참조하세요.

HTTP

REST API에 HTTP 요청을 직접 수행하여 Amazon Pinpoint를 사용할 수 있습니다.

Example엔드포인트 DELETE 요청

엔드포인트를 삭제하려면 엔드포인트 리소스에 DELETE 요청을 발행합니다.

```
DELETE /v1/apps/application-id/endpoints/endpoint-id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

위치:

- application-id는 엔드포인트가 포함된 Amazon Pinpoint 프로젝트의 ID입니다.
- endpoint-id는 삭제할 엔드포인트의 ID입니다.

이 요청에 대한 응답은 삭제한 엔드포인트의 JSON 정의입니다.

Amazon Pinpoint에서 세그먼트 생성 또는 가져오기

사용자 세그먼트는 얼마나 최근에 앱을 사용했는지 또는 어느 디바이스에서 앱을 사용했는지와 같은 공유된 특성에 따른 사용자 하위 집합을 나타냅니다. 세그먼트는 캠페인이 제공하는 메시지를 수신하는 사용자를 지정합니다. 사용자를 앱으로 다시 유치하거나, 특별 제안을 제시하거나, 이밖에 사용자 참여 및 구매를 높이고자 할 때 올바른 대상에 도달할 수 있도록 세그먼트를 정의합니다.

세그먼트를 생성한 후 하나 이상의 캠페인에서 사용할 수 있습니다. 캠페인은 세그먼트 내 사용자에게 맞춤형 메시지를 전달합니다.

자세한 내용은 세그먼트를 참조하세요.

주제

- Amazon Pinpoint에서 세그먼트 빌드
- Amazon Pinpoint에서 세그먼트 가져오기
- AWS Lambda 함수를 사용하여 Amazon Pinpoint 세그먼트 사용자 지정

Amazon Pinpoint에서 세그먼트 빌드

캠페인이 의도하는 타겟에 도달하려면 앱이 보고하는 데이터를 기반으로 세그먼트를 구축합니다. 예를 들어, 최근에 앱을 사용하지 않은 사용자에 도달하려면 이전 30일간 앱을 사용하지 않은 사용자로 세그먼트를 정의할 수 있습니다.

추가 코드 예제는 코드 예제를 참조하세요.

를 사용하여 세그먼트 빌드 AWS SDK for Java

다음 예제는 AWS SDK for Java를 사용하여 세그먼트를 구축하는 방법을 보여 줍니다. 이 예제에서는 팀이 `Lakers`이고 지난 30일 동안 활성 상태였던 사용자 세그먼트를 생성합니다. 세그먼트가 빌드되면 캠페인 또는 여정의 일부로 사용할 수 있습니다. 캠페인과 함께 세그먼트를 사용하는 예제는 Amazon Pinpoint 캠페인을 프로그래밍 방식으로 생성 섹션을 참조하세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
```

```
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
                final String usage = """

                                Usage:   <appId>
```

```
                                Where:
                                  appId - The application ID to create a segment for.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();
                        segmentAttributes.put("Team", AttributeDimension.builder()
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());

                        RecencyDimension recencyDimension = RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors = SegmentBehaviors.builder()
                                        .recency(recencyDimension)
                                        .build();

                        SegmentDemographics segmentDemographics = SegmentDemographics
                                        .builder()
                                        .build();
```

```
                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
 WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
 CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();

                        CreateSegmentResponse createSegmentResult =
 client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
 createSegmentResult.segmentResponse().id());
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
        }
}
```

이 예제를 실행하면 IDE의 콘솔 창에 다음과 같이 표시됩니다.

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

전체 SDK 예제를 보려면 GitHub에서 CreateSegment.java를 참조하세요.

Amazon Pinpoint에서 세그먼트 가져오기

Amazon Pinpoint에서는 세그먼트에 속하는 엔드포인트에 대한 정보를 가져와 사용자 세그먼트를 정의할 수 있습니다. 엔드포인트는 단일 메시징 대상(예: 모바일 푸시 디바이스 토큰, 휴대폰 번호, 이메일 주소)입니다.

Amazon Pinpoint 외부에서 사용자 세그먼트를 이미 만들었지만 사용자를 Amazon Pinpoint 캠페인에 참여시키기 원할 경우 세그먼트 가져오기가 유용합니다.

세그먼트를 가져오면 Amazon Pinpoint는 Amazon Simple Storage Service(Amazon S3)에서 세그먼트의 엔드포인트를 가져옵니다. 가져오기 전에 Amazon S3에 엔드포인트를 추가하고 Amazon Pinpoint에 Amazon S3 액세스 권한을 부여하는 IAM 역할을 생성합니다. 그런 다음 Amazon Pinpoint에 엔드포인트가 저장된 Amazon S3 위치를 제공합니다. 그러면 Amazon Pinpoint가 세그먼트에 각 엔드포인트를 추가합니다.

IAM 역할을 생성하려면 엔드포인트 또는 세그먼트를 가져오기 위한 IAM 역할 단원을 참조하세요. Amazon Pinpoint 콘솔을 사용하여 세그먼트를 가져오는 방법에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 세그먼트 가져오기를 참조하세요.

추가 코드 예제는 코드 예제를 참조하세요.

AWS SDK for Java를 사용하여 세그먼트 가져오기

다음 예제는 AWS SDK for Java를 사용하여 세그먼트를 가져오는 방법을 보여 줍니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
```

```
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the segment
 definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon Pinpoint to access S3.
 You need to set trust management for this to work. See https://docs.aws.amazon.com/
IAM/latest/UserGuide/reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket, key,
 roleArn);
```

```
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for import
 job status.");
        System.out.println("See application " + response.jobStatus() + " for import job
 status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse = client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

전체 SDK 예제를 보려면 GitHub에서 ImportingSegments.java를 참조하세요.

AWS Lambda 함수를 사용하여 Amazon Pinpoint 세그먼트 사용자 지정

이 시험판 설명서는 공개 베타 버전의 기능에 관한 것입니다. 내용은 변경될 수 있습니다.

AWS Lambda 를 사용하여 Amazon Pinpoint 캠페인이 대상 고객과 소통하는 방식을 조정할 수 있습니다. 를 사용하면 Amazon Pinpoint가 캠페인의 메시지를 보내는 순간 캠페인의 세그먼트를 수정할 AWS Lambda수 있습니다.

AWS Lambda 는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하는 데 사용할 수 있는 컴퓨팅 서비스입니다. 코드를 패키징해서 Lambda에 Lambda 함수로 업로드합니다. Lambda는 수동으로 또는 이벤트에 대한 응답으로 자동으로 함수가 간접 호출되면 그 함수를 실행합니다. 자세한 내용은 개발자 안내서AWS Lambda를 참조하세요.

Lambda 함수를 캠페인에 할당하려면 Amazon Pinpoint API의 캠페인 리소스를 사용하여 캠페인의 `CampaignHook` 설정을 정의합니다. 이러한 설정에는 Lambda 함수 이름이 포함됩니다. 또한 Amazon Pinpoint가 함수로부터 반환 값을 받는지 여부를 지정하는 `CampaignHook` 모드도 포함됩니다.

캠페인에 할당한 Lambda 함수를 Amazon Pinpoint 확장이라고 합니다.

`CampaignHook` 설정이 정의되면 Amazon Pinpoint는 캠페인을 실행할 때 캠페인의 메시지를 전송하기 전에 자동으로 Lambda 함수를 간접 호출합니다. Amazon Pinpoint는 함수를 간접 호출할 때 메시지 전송에 대한 이벤트 데이터를 제공합니다. 이 데이터에는 Amazon Pinpoint가 메시지를 보내는 엔드포인트 목록인 캠페인 세그먼트가 포함됩니다.

`CampaignHook` 모드가 `FILTER`로 설정되면 Amazon Pinpoint는 메시지를 보내기 전에 함수가 세그먼트를 수정하고 반환하도록 허용합니다. 예를 들어 함수는 Amazon Pinpoint 외부의 소스에서 가져온 데이터가 포함된 속성으로 엔드포인트 정의를 업데이트할 수 있습니다. 또는 함수는 함수 코드의 조건을 기반으로 특정 엔드포인트를 제거하여 세그먼트를 필터링할 수 있습니다. Amazon Pinpoint는 함수로부터 수정된 세그먼트를 수신한 후 캠페인의 전송 채널을 사용하여 세그먼트의 각 엔드포인트에 메시지를 보냅니다.

를 사용하여 세그먼트를 처리하면 메시지를 보내는 대상과 해당 메시지에 포함된 내용을 더 잘 제어할 AWS Lambda수 있습니다. 캠페인 메시지가 전송되는 시점에 실시간으로 캠페인을 맞춤 조정할 수 있습니다. 세그먼트를 필터링하면 더 협소하게 정의된 세그먼트의 하위 집합과 접촉할 수 있습니다. 또한 엔드포인트 속성을 추가하거나 업데이트하면 메시지 변수에 새 데이터를 사용할 수 있습니다.

> **Note**
>
> CampaignHook 설정을 사용하여 메시지 전송을 처리하는 Lambda 함수를 할당할 수도 있습니다. 이 유형의 함수는 소셜 미디어 플랫폼과 같이 Amazon Pinpoint에서 지원하지 않는 사용자 지정 채널을 통한 메시지 전송에 유용합니다. 자세한 내용은 웹후크 또는 Lambda 함수를 사용하여 Amazon Pinpoint에서 사용자 지정 채널 생성 단원을 참조하십시오.
> Amazon Pinpoint를 사용하여 Lambda 후크를 간접 호출할 때는 Lambda 함수도 Amazon Pinpoint 프로젝트와 동일한 리전에 있어야 합니다.

를 사용하여 캠페인 세그먼트를 수정하려면 AWS Lambda먼저 Amazon Pinpoint에서 전송한 이벤트 데이터를 처리하고 수정된 세그먼트를 반환하는 함수를 생성합니다. 그런 다음 Lambda 함수 정책을 할당하여 함수를 간접 호출하도록 Amazon Pinpoint를 승인합니다. 마지막으로 CampaignHook 설정을 정의하여 하나 이상의 캠페인에 함수를 할당합니다.

추가 코드 예제는 코드 예제를 참조하세요.

이벤트 데이터

Amazon Pinpoint는 Lambda 함수를 간접 호출할 때 다음 페이로드를 이벤트 데이터로 제공합니다.

```
{
  "MessageConfiguration": {Message configuration}
  "ApplicationId": ApplicationId,
  "CampaignId": CampaignId,
  "TreatmentId": TreatmentId,
  "ActivityId": ActivityId,
  "ScheduledTime": Scheduled Time,
  "Endpoints": {
    EndpointId: {Endpoint definition}
    . . .
  }
}
```

AWS Lambda 는 이벤트 데이터를 함수 코드로 전달합니다. 이벤트 데이터는 다음 속성을 제공합니다.

- MessageConfiguration - Amazon Pinpoint API에 있는 메시지 리소스의 DirectMessageConfiguration 객체와 구조가 동일합니다.
- ApplicationId - 캠페인이 속한 Amazon Pinpoint 프로젝트의 ID입니다.
- CampaignId - 함수가 간접 호출된 Amazon Pinpoint 캠페인의 ID입니다.

- `TreatmentId` - A/B 테스트에 사용되는 캠페인 변형의 ID입니다.
- `ActivityId` - 캠페인에 의해 수행 중인 활동의 ID입니다.
- `ScheduledTime` - 캠페인의 메시지가 전달될 때 ISO 8601 형식의 날짜 및 시간입니다.
- `Endpoints` - 엔드포인트 ID와 엔드포인트 정의를 연결하는 맵입니다. 각 이벤트 데이터 페이로드에는 최대 50개의 엔드포인트가 포함됩니다. 캠페인 세그먼트에 포함된 엔드포인트가 50개를 초과하는 경우 Amazon Pinpoint는 모든 엔드포인트가 처리될 때까지 한 번에 최대 50개의 엔드포인트를 사용하여 함수를 반복적으로 간접 호출합니다.

Lambda 함수 생성

Lambda 함수를 생성하는 방법을 알아보려면 AWS Lambda 개발자 안내서의 시작하기를 참조하세요. 함수를 만들 때 다음 조건에서는 메시지 전달이 실패한다는 점에 유의하세요.

- Lambda 함수에서 수정된 세그먼트를 반환하는 데 15초 이상 소요됩니다.
- Amazon Pinpoint는 함수의 반환 값을 디코딩할 수 없습니다.
- 함수를 성공적으로 간접 호출하기 위해 Amazon Pinpoint에서 3번 이상 시도해야 합니다.

Amazon Pinpoint는 함수의 반환 값에 있는 엔드포인트 정의만 수락합니다. 함수는 이벤트 데이터의 다른 요소를 수정할 수 없습니다.

Lambda 함수 예제

Lambda 함수는 Amazon Pinpoint가 전송한 이벤트 데이터를 처리하며, 수정된 엔드포인트를 반환합니다(Node.js로 작성된 다음 예제 핸들러 참조).

```
'use strict';

exports.handler = (event, context, callback) => {
    for (var key in event.Endpoints) {
        if (event.Endpoints.hasOwnProperty(key)) {
            var endpoint = event.Endpoints[key];
            var attr = endpoint.Attributes;
            if (!attr) {
                attr = {};
                endpoint.Attributes = attr;
            }
            attr["CreditScore"] = [ Math.floor(Math.random() * 200) + 650];
        }
```

```
    }
    console.log("Received event:", JSON.stringify(event, null, 2));
    callback(null, event.Endpoints);
};
```

Lambda는 이벤트 데이터를 핸들러에 `event` 파라미터로 전달합니다.

이 예제에서 핸들러는 `event.Endpoints` 객체의 각 엔드포인트를 거쳐 반복되며 새 속성 `CreditScore`를 엔드포인트에 추가합니다. `CreditScore` 속성의 값은 단순한 난수입니다.

`console.log()` 명령문은 CloudWatch Logs에 이벤트를 로그합니다.

`callback()` 명령문은 수정된 엔드포인트를 Amazon Pinpoint에 반환합니다. 일반적으로 `callback` 파라미터는 Node.js Lambda 함수에서 선택 사항이지만 이 컨텍스트에서는 함수가 업데이트된 엔드포인트를 Amazon Pinpoint에 반환해야 하므로 필수입니다.

함수는 이벤트 데이터가 제공한 것과 동일한 형식으로 엔드포인트를 반환해야 하며, 이는 엔드포인트 ID와 엔드포인트 정의를 연결하는 맵입니다(다음 예제 참조).

```
{
    "eqmj8wpxszeqy/b3vch04sn41yw": {
        "ChannelType": "GCM",
        "Address": "4d5e6f1a2b3c4d5e6f7g8h9i0j1a2b3c",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "android"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    },
    "idrexqqtn8sbwfex0ouscod0yto": {
        "ChannelType": "APNS",
        "Address": "1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "apple"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    }
}
```

예제 함수는 이벤트 데이터에서 받은 `event.Endpoints` 객체를 수정하고 반환합니다.

선택에 따라 반환하는 엔드포인트 정의에 `TitleOverride` 및 `BodyOverride` 속성을 포함시킬 수 있습니다.

> **Note**
>
> 이 솔루션을 사용해 메시지를 전송할 때 Amazon Pinpoint는 엔드포인트에 대해 `BodyOverride` 및 `ChannelType` 속성만 인식합니다. 여기에서 `ADM` 속성의 값은 `TitleOverride`, `APNS`, `APNS_SANDBOX`, `APNS_VOIP`, `APNS_VOIP_SANDBOX`, `BAIDU`, `GCM` 또는 `SMS` 중 하나입니다.
> Amazon Pinpoint는 `ChannelType` 속성 값이 `EMAIL`인 엔드포인트에 대해서는 이러한 속성을 인식하지 않습니다.

Lambda 함수 정책 할당

Lambda 함수를 사용하여 엔드포인트를 처리하기 전에 Lambda 함수를 간접 호출하도록 Amazon Pinpoint를 승인해야 합니다. 간접 호출 권한을 부여하려면 함수에 Lambda 함수 정책을 할당합니다. Lambda 함수 정책은 리소스 기반 권한 정책으로, 함수를 사용할 수 있는 개체와 이러한 개체가 실행할 수 있는 작업을 지정합니다.

자세한 내용은 AWS Lambda 개발자 안내서에서 AWS Lambda에 리소스 기반 정책 사용을 참조하세요.

예제 함수 정책

다음 정책은 Amazon Pinpoint 서비스 보안 주체에게 특정 캠페인(*campaign-id*)에 대한 `lambda:InvokeFunction` 작업을 사용할 수 있는 권한을 부여합니다.

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:account-id:function:function-name}",
  "Condition": {
    "StringEquals": {
```

```
      "AWS:SourceAccount": "111122223333"
    },
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-
id/campaigns/campaign-id"
    }
  }
}
```

함수 정책에는 AWS:SourceArn 키를 포함하는 Condition 블록이 필요합니다. 이 코드는 함수를 간접 호출하도록 허용되는 Amazon Pinpoint 캠페인을 지정합니다. 이 예제에서 정책은 하나의 캠페인에만 권한을 부여합니다. Condition 블록에는 작업을 호출할 수 있는 AWS 계정을 제어하는 AWS:SourceAccount 키도 포함되어야 합니다.

더 일반적인 정책을 작성하려면 다중 문자 매칭 와일드카드(*)를 사용하세요. 예를 들어 다음 Condition 블록을 사용하여 특정 Amazon Pinpoint 프로젝트(*application-id*)의 모든 캠페인이 함수를 간접 호출하도록 허용할 수 있습니다.

```
...
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "111122223333"
  },
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/*"
  }
}
...
```

Lambda 함수가 프로젝트의 모든 캠페인에서 사용하는 기본 함수가 되게 하려면 앞의 방법으로 정책에 대한 Condition 블록을 구성하는 것이 좋습니다. 프로젝트의 모든 캠페인에 대해 Lambda 함수를 기본값으로 설정하는 방법에 대한 자세한 내용은 캠페인에 Lambda 함수 할당 단원을 참조하세요.

Amazon Pinpoint 간접 호출 권한 부여

AWS Command Line Interface (AWS CLI)를 사용하여 Lambda 함수에 할당된 Lambda 함수 정책에 권한을 추가할 수 있습니다. Amazon Pinpoint가 특정 캠페인에 대한 함수를 간접 호출하도록 허용하려면 다음 예제와 같이 Lambda add-permission 명령을 사용합니다.

```
$ aws lambda add-permission \
```

```
> --function-name function-name \
> --statement-id sid \
> --action lambda:InvokeFunction \
> --principal pinpoint.us-east-1.amazonaws.com \
> --source-account 111122223333
> --source-arn arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/campaign-id
```

AWS CLI에서 get-campaigns 명령을 사용하여 캠페인 ID를 조회할 수 있습니다. get-apps 명령을 사용하여 애플리케이션 ID를 조회할 수도 있습니다.

Lambda `add-permission` 명령을 실행하면 Lambda에서 다음 출력이 반환됩니다.

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/application-id/
campaigns/campaign-id\"}}
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
}
```

`Statement` 값은 Lambda 함수 정책에 추가된 문의 JSON 문자열 버전입니다.

캠페인에 Lambda 함수 할당

개별 Amazon Pinpoint 캠페인에 Lambda 함수를 할당할 수 있습니다. 또는 Lambda 함수를 개별적으로 함수를 할당한 캠페인을 제외한 모든 캠페인에서 프로젝트에 대해 사용되는 기본값으로 설정할 수 있습니다.

개별 캠페인에 Lambda 함수를 할당하려면 Amazon Pinpoint API를 사용하여 `Campaign` 객체를 생성하거나 업데이트하고 `CampaignHook` 속성을 정의합니다. Lambda 함수를 프로젝트의 모든 캠페인에 대한 기본값으로 설정하려면 프로젝트에 대한 `Settings` 리소스를 생성하거나 업데이트하고 `CampaignHook` 객체를 정의합니다.

두 경우 모두 다음 CampaignHook 속성을 설정합니다.

- LambdaFunctionName - Amazon Pinpoint가 캠페인에 대한 메시지를 보내기 전에 간접 호출하는 Lambda 함수의 이름 또는 ARN입니다.
- Mode – FILTER로 설정합니다. 이 모드에서 Amazon Pinpoint는 함수를 간접 호출하고 수정된 엔드포인트가 반환될 때까지 기다립니다. Amazon Pinpoint는 이를 수신한 후 메시지를 전송합니다. Amazon Pinpoint는 15초까지 기다린 후 메시지 전송을 실패로 처리합니다.

CampaignHook 설정이 캠페인에 대해 정의되면 Amazon Pinpoint는 캠페인 메시지를 보내기 전에 지정된 Lambda 함수를 간접 호출합니다. Amazon Pinpoint는 함수의 수정된 엔드포인트를 수신하기 위해 대기합니다. Amazon Pinpoint는 업데이트된 엔드포인트를 수신하면 업데이트된 엔드포인트 데이터를 사용하여 메시지 전송 작업을 계속 진행합니다.

Amazon Pinpoint 캠페인을 프로그래밍 방식으로 생성

앱에 대한 사용자 참여를 높이기 위해 Amazon Pinpoint를 사용하여 특정 사용자 세그먼트에 도달하는 푸시 알림 캠페인을 생성하고 관리합니다.

예를 들어, 캠페인은 최근에 앱을 실행한 적이 없는 사용자를 다시 앱으로 유치하거나 최근에 구매한 적이 없는 사용자에게 특별 제안을 제시할 수 있습니다.

캠페인은 지정된 사용자 세그먼트에 맞춤형 메시지를 전송합니다. 캠페인은 세그먼트의 모든 사용자에게 메시지를 전송할 수 있습니다. 또는 홀드아웃, 즉 메시지를 수신하지 않는 사용자의 백분율을 할당할 수도 있습니다.

캠페인 일정을 설정하여 메시지를 한 번만 또는 반복적으로(예: 매주 한 번) 전송할 수 있습니다. 사용자가 불편한 시간대에 메시지를 수신하지 않도록 일정에 메시지가 전혀 전송되지 않는 침묵 시간이 포함될 수 있습니다.

대체 캠페인 전략을 실험하려면 캠페인을 A/B 테스트로 설정합니다. A/B 테스트에는 메시지 또는 일정에 대한 처리가 2개 이상 포함됩니다. 처리는 메시지 또는 일정의 변형입니다. 사용자가 캠페인에 응답하면 캠페인 분석을 통해 각 처리의 효과를 비교할 수 있습니다.

자세한 내용은 Amazon Pinpoint REST API 가이드의 캠페인 또는 Amazon Pinpoint 사용 설명서의 캠페인을 참조하세요.

표준 Amazon Pinpoint 캠페인 생성

표준 캠페인은 사용자 지정 푸시 알림을 정의된 일정에 따라 지정된 세그먼트로 전송합니다. 다음 예제는 AWS SDK for Java를 사용하여 캠페인을 생성하는 방법을 보여 줍니다. 전달할 세그먼트를 생성하는 예제는 Amazon Pinpoint에서 세그먼트 빌드 섹션을 참조하세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
```

```
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>

                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
```

```
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId, String
segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
    }

    public static CampaignResponse createCampaign(PinpointClient client, String appID,
String segmentID) {

        try {
            Schedule schedule = Schedule.builder()
                    .startTime("IMMEDIATE")
                    .build();

            Message defaultMessage = Message.builder()
                    .action(Action.OPEN_APP)
                    .body("My message body.")
                    .title("My message title.")
                    .build();

            MessageConfiguration messageConfiguration = MessageConfiguration.builder()
                    .defaultMessage(defaultMessage)
                    .build();

            WriteCampaignRequest request = WriteCampaignRequest.builder()
                    .description("My description")
                    .schedule(schedule)
                    .name("MyCampaign")
                    .segmentId(segmentID)
                    .messageConfiguration(messageConfiguration)
                    .build();

            CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());
```

```
            System.out.println("Campaign ID: " + result.campaignResponse().id());
            return result.campaignResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }
}
```

이 예제를 실행하면 IDE의 콘솔 창에 다음과 같이 표시됩니다.

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

전체 SDK 예제를 보려면 GitHub에서 CreateCampaign.java를 참조하세요.

를 사용하여 A/B 테스트 Amazon Pinpoint 캠페인 생성 AWS SDK for Java

A/B 테스트 캠페인은 표준 캠페인과 비슷하게 동작하지만 캠페인의 메시지 또는 일정에 대해 다양한 처리를 정의할 수 있습니다. A/B 테스트에는 메시지 또는 일정에 대한 처리가 2개 이상 포함됩니다. 처리는 메시지 또는 일정의 변형입니다. 사용자가 캠페인에 응답하면 캠페인 분석을 통해 각 처리의 효과를 비교할 수 있습니다.

다음 예제는 AWS SDK for Java를 사용하여 A/B 테스트 캠페인을 생성하는 방법을 보여 줍니다.

```
import com.amazonaws.services.pinpoint.AmazonPinpointClient;
import com.amazonaws.services.pinpoint.model.Action;
import com.amazonaws.services.pinpoint.model.CampaignResponse;
import com.amazonaws.services.pinpoint.model.CreateCampaignRequest;
import com.amazonaws.services.pinpoint.model.CreateCampaignResult;
import com.amazonaws.services.pinpoint.model.Message;
import com.amazonaws.services.pinpoint.model.MessageConfiguration;
import com.amazonaws.services.pinpoint.model.Schedule;
import com.amazonaws.services.pinpoint.model.WriteCampaignRequest;
import com.amazonaws.services.pinpoint.model.WriteTreatmentResource;

import java.util.ArrayList;
import java.util.List;
```

```
public class PinpointCampaignSample {

    public CampaignResponse createAbCampaign(AmazonPinpointClient client, String appId,
 String segmentId) {
        Schedule schedule = new Schedule()
                .withStartTime("IMMEDIATE");

        // Default treatment.
        Message defaultMessage = new Message()
                .withAction(Action.OPEN_APP)
                .withBody("My message body.")
                .withTitle("My message title.");

        MessageConfiguration messageConfiguration = new MessageConfiguration()
                .withDefaultMessage(defaultMessage);

        // Additional treatments
        WriteTreatmentResource treatmentResource = new WriteTreatmentResource()
                .withMessageConfiguration(messageConfiguration)
                .withSchedule(schedule)
                .withSizePercent(40)
                .withTreatmentDescription("My treatment description.")
                .withTreatmentName("MyTreatment");

        List<WriteTreatmentResource> additionalTreatments = new
 ArrayList<WriteTreatmentResource>();
        additionalTreatments.add(treatmentResource);

        WriteCampaignRequest request = new WriteCampaignRequest()
                .withDescription("My description.")
                .withSchedule(schedule)
                .withSegmentId(segmentId)
                .withName("MyCampaign")
                .withMessageConfiguration(messageConfiguration)
                .withAdditionalTreatments(additionalTreatments)
                .withHoldoutPercent(10); // Hold out of A/B test

        CreateCampaignRequest createCampaignRequest = new CreateCampaignRequest()
                .withApplicationId(appId).withWriteCampaignRequest(request);

        CreateCampaignResult result = client.createCampaign(createCampaignRequest);

        System.out.println("Campaign ID: " + result.getCampaignResponse().getId());
```

```
        return result.getCampaignResponse();
    }

}
```

이 예제를 실행하면 IDE의 콘솔 창에 다음과 같이 표시됩니다.

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Amazon Pinpoint 리소스 태그 관리

태그는 선택적으로 정의하여 특정 유형의 Amazon Pinpoint AWS 리소스를 포함한 리소스와 연결하는 레이블입니다. 태그를 사용하면 용도, 소유자, 환경 또는 기타 기준과 같은 다양한 방법으로 리소스를 분류하고 관리하는 데 도움이 됩니다. 예를 들어 태그를 사용하여 정책 또는 자동화를 적용하거나, 특정 규정 준수 요구 사항이 적용되는 리소스를 식별할 수 있습니다. 태그를 추가할 수 있는 Amazon Pinpoint 리소스 유형은 다음과 같습니다.

- Campaigns
- 메시지 템플릿
- 프로젝트(애플리케이션)
- Segments

리소스는 최대 50개의 태그를 가질 수 있습니다. 각 태그는 사용자가 정의하는 필수 태그 키와 선택적 태그 값으로 구성됩니다. 태그 키는 더 구체적인 태그 값에 대해 카테고리와 같은 역할을 하는 일반적인 레이블입니다. 태그 값은 태그 키에 대한 설명자 역할을 합니다.

태그 키에는 최대 128자를 사용할 수 있습니다. 태그 값에는 최대 256자를 사용할 수 있습니다. 문자는 유니코드 문자, 숫자, 공백 또는 다음 기호 중 하나일 수 있습니다. _ . : / = + -. 태그에 적용되는 추가 제한 사항은 다음과 같습니다.

- 태그 키와 값은 대소문자를 구분합니다.
- 연결된 각 리소스에 대해 각 태그 키는 고유해야 하고 하나의 값만 가질 수 있습니다.
- `aws:` 접두사는에서 사용하도록 예약되어 있습니다. 정의한 태그 키 또는 값에는 사용할 AWS수 없습니다. 또는 이 접두사를 사용하는 태그 키 또는 값을 편집하거나 제거할 수 없습니다. 이 접두사를 사용하는 태그는 리소스당 50개의 할당량에 포함되지 않습니다.
- 태그만을 기반으로 하는 리소스는 업데이트하거나 삭제할 수 없습니다. 사용하는 작업에 따라 Amazon 리소스 이름(ARN) 또는 리소스 ID도 지정해야 합니다.
- 태그를 퍼블릭 또는 공유 리소스와 연결할 수 있습니다. 그러나 태그는 사용자 AWS 계정에만 사용할 수 있으며 리소스를 공유하는 다른 계정에는 사용할 수 없습니다. 또한 태그는 AWS 계정의 지정된 AWS 리전에 있는 리소스에만 사용할 수 있습니다.

Amazon Pinpoint 리소스에서 태그 키와 값을 추가, 표시, 업데이트 및 제거하려면 AWS Command Line Interface (AWS CLI), Amazon Pinpoint API, AWS Resource Groups 태그 지정 API 또는 SDK를

AWS 사용할 수 있습니다. AWS 계정의 특정 AWS 리전(Amazon Pinpoint AWS 리소스 포함)에 있는 모든 리소스에서 태그 키와 값을 관리하려면 AWS Resource Groups 태그 지정 API를 사용합니다.

Amazon Pinpoint 리소스를 관리할 때 사용할 수 있는 CLI 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 Amazon Pinpoint 단원을 참조하세요.

지원되는 HTTP(S) 메서드, 파라미터 및 스키마를 포함하여 Amazon Pinpoint API의 리소스에 대한 자세한 내용은 Amazon Pinpoint API 참조를 참조하세요.

IAM 정책 및 API 작업에 Amazon Pinpoint 태그 사용

태그 구현을 시작한 후 AWS Identity and Access Management (IAM) 정책 및 API 작업에 태그 기반 리소스 수준 권한을 적용할 수 있습니다. 여기에는 리소스가 생성될 때 리소스에 태그 추가를 지원하는 작업이 포함됩니다. 이러한 방식으로 태그를 사용하면 리소스 생성 및 태그 지정 권한이 있는 AWS 계정의 그룹 및 사용자와 보다 일반적으로 태그를 생성, 업데이트 및 제거할 권한이 있는 그룹 및 사용자를 세부적으로 제어할 수 있습니다.

예를 들어, 사용자가 이름이 리소스의 `Owner` 태그 값인 모든 Amazon Pinpoint 리소스에 대한 전체 액세스 권한을 갖도록 허용하는 정책을 생성할 수 있습니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "ModifyResourceIfOwner",
         "Effect": "Allow",
         "Action": "mobiletargeting:*",
         "Resource": "*",
         "Condition": {
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

태그 기반의 리소스 수준 권한을 정의하면 권한이 즉시 적용됩니다. 즉, 리소스를 생성하자마자 더 안전하게 보호할 수 있으며 새로운 리소스에 태그 사용 적용을 빠르게 시작할 수 있습니다. 리소스 수준 권한을 사용하여 새로운 및 기존 리소스와 연결할 수 있는 태그 키와 값을 제어할 수도 있습니다. 자세한 내용은AWS IAM 사용 설명서의 태그를 사용한 액세스 제어를 참조하세요.

프로그래밍 방식으로 Amazon Pinpoint 리소스에 태그 추가

다음 예제는 AWS CLI 및 Amazon Pinpoint REST API를 사용하여 Amazon Pinpoint 리소스에 태그를 추가하는 방법을 보여줍니다. 지원되는 AWS SDK를 사용하여 리소스에 태그를 추가할 수도 있습니다.

단일 작업으로 여러 Amazon Pinpoint 리소스에 태그를 추가하려면 AWS CLI 또는 태그 지정 AWS Resource Groups API의 리소스 그룹 태그 지정 작업을 사용합니다.

API를 사용하여 태그 추가

새 리소스를 만들고 Amazon Pinpoint REST API를 사용하여 이 리소스에 태그를 추가하려면 적절한 리소스 URI에 POST 요청을 전송합니다. 요청 본문에 `tags` 파라미터 및 값을 포함합니다. 다음 예제는 새 프로젝트를 만들 때 태그를 지정하는 방법을 보여줍니다.

```
POST /v1/apps HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Cache-Control: no-cache

{
   "Name":"MyProject",
   "tags":{
      "key1":"value1"
   }
}
```

기존 리소스에 태그를 추가하려면 태그 URI에 POST 요청을 보내세요. 리소스의 Amazon 리소스 이름(ARN)을 URI에 포함합니다. ARN은 URL로 인코딩되어야 합니다. 요청 본문에 다음 예제와 같이 `tags` 파라미터와 값을 포함합니다.

```
POST /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
```

```
Cache-Control: no-cache

{
   "tags":{
      "key1":"value1"
   }
}
```

를 사용하여 태그 추가 AWS CLI

를 사용하여 새 리소스를 생성하고 태그를 추가하려면 리소스에 적절한 `create` 명령을 AWS CLI사용합니다. `tags` 파라미터와 값을 포함합니다. 다음 예제는 새 프로젝트를 만들 때 태그를 지정하는 방법을 보여줍니다.

Linux, macOS, or Unix

```
$ aws pinpoint create-app \
  --create-application-request '{
    "Name":"MyProject",
    "tags": {
      "key1":"value1",
      "key2":"value2"
    }
  }'
```

Windows Command prompt

```
C:\> aws pinpoint create-app ^
     --create-application-request Name=MyProject,tags={key1=value1,key2=value2}
```

이전 예제에서 다음을 수행합니다.

- *MyProject*를 프로젝트에 지정하려는 이름으로 바꿉니다.
- *key1* 및 *key2*를 리소스에 추가하려는 태그의 키로 바꿉니다.
- *value1* 및 *value2*를 각 키에 추가하려는 태그의 값으로 바꿉니다.

Amazon Pinpoint 리소스를 만들 때 사용할 수 있는 명령에 대한 자세한 내용은 AWS CLI 명령 참조를 참조하세요.

기존 리소스에 태그를 추가하려면 `tag-resource` 명령을 사용하여 필요한 파라미터에 적합한 값을 지정합니다.

Linux, macOS, or Unix

```
$ aws pinpoint tag-resource \
  --resource-arn resource-arn \
  --tags-model '{
    "tags": {
      "key1":"value1",
      "key2":"value2"
    }
  }'
```

Windows Command Prompt

```
C:\> aws pinpoint tag-resource ^
     --resource-arn resource-arn ^
     --tags-model tags={key1=value1,key2=value2}
```

이전 예제에서 다음을 수행합니다.

- *resource-arn*을 태그를 추가하려는 리소스의 Amazon 리소스 이름(ARN)으로 바꿉니다.
- *key1* 및 *key2*를 리소스에 추가하려는 태그의 키로 바꿉니다.
- *value1* 및 *value2*를 각 키에 추가하려는 태그의 값으로 바꿉니다.

프로그래밍 방식으로 Amazon Pinpoint 리소스에 대한 태그 표시

다음 예제에서는 AWS CLI 및 Amazon Pinpoint REST API를 사용하여 Amazon Pinpoint 리소스와 연결된 모든 태그(키 및 값)를 나열하는 방법을 보여줍니다. 지원되는 AWS SDK를 사용하여 리소스와 연결된 태그를 표시할 수도 있습니다.

API를 사용하여 태그 표시

Amazon Pinpoint REST API를 사용하여 특정 리소스와 연결된 모든 태그를 표시하려면 URI에 리소스의 Amazon 리소스 이름(ARN)을 포함하여 태그 URI에 GET 요청을 보냅니다. ARN은 URL로 인코딩되어야 합니다. 예를 들어 다음 요청은 지정된 캠페인(*resource-arn*)과 연결된 모든 태그를 검색합니다.

```
GET /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

요청에 대한 JSON 응답에는 tags 객체가 포함됩니다. tags 객체에는 캠페인과 연결된 모든 태그 키와 값이 나열됩니다.

둘 이상의 동일한 유형의 리소스와 연결된 모든 태그를 표시하려면, 해당하는 적절한 리소스 유형의 URI에 GET 요청을 보냅니다. 예를 들어 다음 요청은 지정된 프로젝트(*application-id*)의 모든 캠페인에 대한 정보를 가져옵니다.

```
GET /v1/apps/application-id/campaigns HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

이 요청에 대한 JSON 응답은 해당 프로젝트의 모든 캠페인을 나열합니다. 각 캠페인의 tags 객체는 해당 캠페인과 연결된 모든 태그 키와 값을 나열합니다.

를 사용하여 태그 표시 AWS CLI

AWS CLI 를 사용하여 특정 리소스와 연결된 태그 목록을 표시하려면 다음 예제와 같이 `list-tags-for-resource` 명령을 실행하고 `resource-arn` 파라미터에 대한 리소스의 Amazon 리소스 이름(ARN)을 지정합니다.

Linux, macOS, or Unix

```
$ aws pinpoint list-tags-for-resource \
  --resource-arn resource-arn
```

Windows Command Prompt

```
C:\> aws pinpoint list-tags-for-resource ^
     --resource-arn resource-arn
```

태그가 있는 모든 Amazon Pinpoint 리소스 목록과 각 리소스와 연결된 모든 태그를 표시하려면 AWS Resource Groups Tagging API의 get-resources 명령을 사용합니다. 다음 예제에 표시된 대로 `resource-type-filters` 파라미터를 `mobiletargeting`로 설정합니다.

Linux, macOS, or Unix

```
$ aws resourcegroupstaggingapi get-resources \
     --resource-type-filters "mobiletargeting"
```

Windows Command Prompt

```
C:\> aws resourcegroupstaggingapi get-resources ^
     --resource-type-filters "mobiletargeting"
```

명령 출력은 태그가 있는 모든 Amazon Pinpoint 리소스의 ARN 목록입니다. 목록에는 각 리소스와 연결된 모든 태그 키와 값이 포함되어 있습니다.

프로그래밍 방식으로 Amazon Pinpoint 리소스에 대한 태그 업데이트 또는 덮어쓰기

Amazon Pinpoint 리소스에 대한 태그를 업데이트하는(덮어쓰는) 방법에는 여러 가지가 있습니다. 태그를 업데이트하는 가장 좋은 방법은 다음에 따라 다릅니다.

- 태그를 업데이트하려는 리소스의 유형
- 한 리소스의 태그를 업데이트할지, 동시에 여러 리소스의 태그를 업데이트할지
- 태그 키를 업데이트할지, 태그 값을 업데이트할지, 둘 다 업데이트할지

Amazon Pinpoint 프로젝트 또는 여러 리소스의 태그를 동시에 업데이트하려면 AWS CLI 또는 태그 지정 AWS Resource Groups API의 리소스 그룹 태그 지정 작업을 사용합니다. 현재 Amazon Pinpoint API는 이러한 작업 중 하나에 대한 직접적인 지원을 제공하지 않습니다.

한 리소스의 태그를 업데이트하려면 Amazon Pinpoint API를 사용하여 현재 태그를 제거하고 새 태그를 추가할 수 있습니다.

프로그래밍 방식으로 Amazon Pinpoint 리소스에서 태그 제거

다음 예제는 AWS CLI 및 Amazon Pinpoint REST API를 사용하여 Amazon Pinpoint 리소스에서 태그(키와 값 모두)를 제거하는 방법을 보여줍니다. 지원되는 AWS SDK를 사용하여 리소스에서 태그를 제거할 수도 있습니다.

단일 작업으로 여러 Amazon Pinpoint 리소스에서 태그를 제거하려면 AWS CLI 또는 Tagging AWS Resource Groups API의 리소스 그룹 태깅 작업을 사용합니다. 리소스에서 태그 키가 아니라 특정 태그 값만 제거하려면 리소스의 태그를 업데이트합니다.

API를 사용하여 태그 제거

Amazon Pinpoint REST API를 사용하여 리소스에서 태그를 제거하려면 태그 URI에 DELETE 요청을 보냅니다. URI의 경우 태그를 제거할 리소스의 Amazon 리소스 이름(ARN)을 포함한 후, `tagKeys` 파라미터와 제거할 태그를 추가합니다. 예제:

```
https://endpoint/v1/tags/resource-arn?tagKeys=key
```

위치:

- *`endpoint`*는 리소스를 호스트하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- *`resource-arn`*은 태그를 제거할 리소스의 ARN입니다.
- *`key`*는 리소스에서 제거할 태그입니다.

모든 파라미터는 URL로 인코딩해야 합니다.

리소스에서 여러 태그 키와 연결된 값을 제거하려면 제거할 각 추가 태그에 대해 `tagKeys` 파라미터와 인수를 추가하고 앰퍼샌드(&)로 구분합니다. 예제:

```
https://endpoint/v1/tags/resource-arn?tagKeys=key1&tagKeys=key2
```

모든 파라미터는 URL로 인코딩해야 합니다.

를 사용하여 태그 제거 AWS CLI

를 사용하여 리소스에서 태그를 제거하려면 `untag-resource` 명령을 AWS CLI실행합니다. 다음 예제에 표시된 대로 `tag-keys` 파라미터 및 인수를 포함합니다.

Linux, macOS, or Unix

```
$ aws pinpoint untag-resource \
  --resource-arn resource-arn \
  --tag-keys key1 key2
```

Windows Command Prompt

```
C:\> aws pinpoint untag-resource ^
     --resource-arn resource-arn ^
     --tag-keys key1 key2
```

이전 예제에서 다음과 같이 변경합니다.

- *resource-arn*을 태그를 제거하려는 리소스의 ARN으로 바꿉니다.
- *key1* 및 *key2*를 리소스에서 제거하려는 태그의 키로 바꿉니다.

Amazon Pinpoint를 애플리케이션과 통합

Amazon Pinpoint를 클라이언트 코드와 통합하면 사용자를 파악하고 사용자의 관심을 끌 수 있습니다.

통합한 후 사용자가 애플리케이션을 실행하면 애플리케이션이 Amazon Pinpoint 서비스에 연결하여 엔드포인트를 추가하거나 업데이트합니다. 엔드포인트는 메시지를 전송할 수 있는 대상을 나타냅니다(예: 사용자 디바이스, 이메일 주소 또는 전화번호).

또한 애플리케이션은 사용 데이터 또는 이벤트를 제공할 수 있습니다. Amazon Pinpoint 콘솔에서 이벤트 데이터를 보고 사용자 수, 사용자의 애플리케이션 사용 빈도, 애플리케이션 사용 시점 등을 파악할 수 있습니다.

애플리케이션에서 제공한 엔드포인트 및 이벤트 정보로 특정 대상 또는 세그먼트에 맞춰 메시징 캠페인을 맞춤화할 수 있습니다. 또한 캠페인을 생성하지 않고 간단한 수신자 목록에 직접 메시지를 전송할 수도 있습니다.

이 섹션의 주제를 참조하여 Amazon Pinpoint를 모바일 또는 웹 애플리케이션과 통합합니다. 이러한 주제에는 JavaScript, Android, Swift, 또는 Flutter 애플리케이션과의 통합을 위한 코드 예제 및 절차가 나와 있습니다. 앱 통합을 시작하려면 the section called “Amplify를 사용하여 프런트엔드 애플리케이션 연결” 단원을 참조하십시오.

클라이언트 외부에서는 지원되는 AWS SDK 또는 Amazon Pinpoint API를 사용하여 엔드포인트를 가져오고, 이벤트 데이터를 내보내고, 고객 세그먼트를 정의하고, 캠페인을 실행하는 등의 작업을 수행할 수 있습니다.

주제

- AWS SDK에서 Amazon Pinpoint 사용
- AWS Amplify를 사용하여 프런트엔드 애플리케이션을 Amazon Pinpoint에 연결
- 애플리케이션에 Amazon Pinpoint 엔드포인트 등록
- 애플리케이션에서 Amazon Pinpoint 이벤트 보고

AWS SDK에서 Amazon Pinpoint 사용

다양한 프로그래밍 언어에 대해 AWS 소프트웨어 개발 키트(SDK)을 사용할 수 있습니다. 각 SDK는 개발자가 선호하는 언어로 애플리케이션을 쉽게 구축할 수 있도록 하는 API, 코드 예제 및 설명서를 제공합니다.

SDK 설명서	코드 예시
AWS SDK for C++	AWS SDK for C++ 코드 예시
AWS CLI	AWS CLI 코드 예시
AWS SDK for Go	AWS SDK for Go 코드 예시
AWS SDK for Java	AWS SDK for Java 코드 예시
AWS SDK for JavaScript	AWS SDK for JavaScript 코드 예시
AWS SDK for Kotlin	AWS SDK for Kotlin 코드 예시
AWS SDK for .NET	AWS SDK for .NET 코드 예시
AWS SDK for PHP	AWS SDK for PHP 코드 예시
AWS Tools for PowerShell	AWS Tools for PowerShell 코드 예시
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 코드 예시
AWS SDK for Ruby	AWS SDK for Ruby 코드 예시
AWS SDK for Rust	AWS SDK for Rust 코드 예시
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP 코드 예시
AWS SDK for Swift	AWS SDK for Swift 코드 예시

Amazon Pinpoint 관련 예는 AWS SDK를 사용한 Amazon Pinpoint용 코드 예제 섹션을 참조하세요.

> **가용성 예제**
>
> 필요한 예제를 찾을 수 없습니까? 이 페이지 하단의 피드백 제공 링크를 사용하여 코드 예시를 요청하세요.

AWS Amplify를 사용하여 프런트엔드 애플리케이션을 Amazon Pinpoint에 연결

AWS Amplify를 사용하여 앱을와 통합합니다 AWS. Swift 앱의 경우 Swift용 Amplify 설명서에서 시작하기 섹션을 참조하세요. Android 앱의 경우 Android SDK 설명서의 시작하기 섹션을 참조하세요. React 네이티브 앱의 경우 Amplify JavaScript 설명서에서 시작하기 섹션을 참조하세요. Flutter 앱의 경우 Flutter SDK 설명서의 시작하기 섹션을 참조하세요. 이러한 주제는 다음 작업을 수행하는 데 도움이 됩니다.

- 백엔드 리소스를 설정합니다.
- Amplify 라이브러리를 사용하여 앱을 백엔드 리소스에 연결합니다.

분석, 인앱 메시징, 푸시 알림을 위해 프런트엔드 앱을 Amazon Pinpoint에 연결하는 방법을 자세히 알아보려면 AWS Amplify 섹션을 참조하세요.

다음 단계

AWS Amplify를 애플리케이션과 통합한 후 코드를 업데이트하여 사용자의 디바이스를 엔드포인트로 등록합니다. 자세한 내용은 애플리케이션에 Amazon Pinpoint 엔드포인트 등록 단원을 참조하십시오.

애플리케이션에 Amazon Pinpoint 엔드포인트 등록

사용자가 세션을 시작하면(예: 모바일 앱 시작) 모바일 또는 웹 애플리케이션이 Amazon Pinpoint에 엔드포인트를 자동으로 등록(또는 업데이트)할 수 있습니다. 엔드포인트는 사용자가 세션을 시작한 디바이스를 나타냅니다. 여기에는 디바이스를 설명하는 속성이 들어 있으며, 사용자가 정의하는 사용자 지정 속성도 포함될 수 있습니다. 엔드포인트가 고객과 통신하는 다른 방법(예: 이메일 주소, 휴대폰 번호)을 나타낼 수도 있습니다.

애플리케이션이 엔드포인트를 등록하면 엔드포인트 속성을 기준으로 사용자를 분류할 수 있습니다. 그런 다음 맞춤형 메시징 캠페인에 이러한 세그먼트를 적용할 수 있습니다. 이와 함께 Amazon Pinpoint의 분석 페이지에서도 새 엔드포인트 수 및 일별 활성 엔드포인트 수 등 엔드포인트 등록 및 활동에 관한 차트를 볼 수 있습니다.

하나의 사용자 ID를 여러 엔드포인트에 할당할 수 있습니다. 사용자 ID는 사용자 한 명을 나타내고, 사용자 ID에 할당된 각 엔드포인트는 사용자의 디바이스 한 대를 나타냅니다. 엔드포인트에 사용자 ID를 할당한 뒤에는 콘솔에서 [일별 활성 사용자] 및 [월별 활성 사용자] 등의 사용자 활동에 관한 차트를 볼 수 있습니다.

시작하기 전 준비 사항

아직 통합하지 않았다면 Android 또는 iOS용 AWS Mobile SDK를 통합하거나 AWS Amplify JavaScript 라이브러리를 애플리케이션과 통합합니다. 자세한 내용은 AWS Amplify를 사용하여 프런트엔드 애플리케이션을 Amazon Pinpoint에 연결 단원을 참조하십시오.

Android 또는 iOS용 AWS 모바일 SDKs에 엔드포인트 등록

Android 또는 iOS용 AWS Mobile SDKs를 사용하여 엔드포인트를 등록하고 사용자 지정할 수 있습니다. 자세한 내용과 코드 샘플은 다음 문서를 참조하십시오.

- Android SDK 설명서의 애플리케이션에서 엔드포인트 등록을 참조하십시오.
- iOS SDK 설명서의 애플리케이션에서 엔드포인트 등록을 참조하십시오.

AWS Amplify JavaScript 라이브러리에 엔드포인트 등록

AWS Amplify JavaScript 라이브러리를 사용하여 앱에서 엔드포인트를 등록하고 업데이트할 수 있습니다. 자세한 내용과 코드 예제를 보려면 AWS Amplify JavaScript 설명서의 엔드포인트 업데이트를 참조하세요.

다음 단계

엔드포인트를 등록하도록 앱을 업데이트하면 사용자가 앱을 시작할 때 디바이스 정보와 사용자 지정 속성이 Amazon Pinpoint에 제공됩니다. 이러한 정보를 사용하여 대상 세그먼트를 정의할 수 있습니다. 콘솔을 사용하여 엔드포인트 지표와 사용자 ID가 할당된 사용자도 확인할 수 있습니다. 애플리케이션에서 Amazon Pinpoint 이벤트 보고의 단계를 완료하여 사용 데이터를 보고하도록 앱을 업데이트할 수도 있습니다.

애플리케이션에서 Amazon Pinpoint 이벤트 보고

모바일 또는 웹 애플리케이션에서 AWS Mobile SDKs 또는 Amazon Pinpoint 이벤트 API를 사용하여 사용 데이터 또는 이벤트를 Amazon Pinpoint에 보고할 수 있습니다. 이벤트를 보고하여 앱 세션 시간, 사용자의 구매 행동, 로그인 시도 또는 필요한 사용자 지정 이벤트 유형과 같은 정보를 캡처할 수 있습니다.

애플리케이션에서 이벤트를 보고하면 Amazon Pinpoint 콘솔에서 분석 결과를 볼 수 있습니다. [Analytics] 페이지의 차트는 다양한 사용자 행동 측면에 대한 측정치를 제공합니다. 자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint 분석용 차트 참조 섹션을 참조하세요.

이벤트 데이터를 Amazon Pinpoint 외부에서 분석 또는 저장하기 위해 Amazon Kinesis로 데이터를 스트리밍하도록 Amazon Pinpoint를 구성할 수 있습니다. 자세한 내용은 Amazon Pinpoint를 사용하여 Kinesis 및 Firehose를 통해 앱 이벤트 데이터 스트리밍 단원을 참조하십시오.

AWS Mobile SDKs 및 AWS Amplify JavaScript 라이브러리를 사용하여 Amazon Pinpoint API를 호출하여 다음 유형의 이벤트를 보고할 수 있습니다.

세션 이벤트

사용자가 앱을 열고 닫는 시점과 빈도를 나타냅니다.

애플리케이션이 세션 이벤트를 보고한 후 Amazon Pinpoint 콘솔의 분석 페이지를 사용하여 세션, 일별 활성 엔드포인트 수, 7일 보존율 등에 대한 차트를 봅니다.

사용자 지정 이벤트

사용자 지정 이벤트 유형을 할당하여 정의한 비표준 이벤트입니다. 사용자 지정 이벤트에는 사용자 지정 속성과 지표를 추가할 수 있습니다.

분석 페이지의 이벤트 탭에는 해당 앱이 보고한 모든 사용자 지정 이벤트의 지표가 표시됩니다.

수익 창출 이벤트

애플리케이션에서 창출된 수익과 사용자가 구입한 항목 수를 보고합니다.

분석 페이지의 매출 탭에는 매출, 유료 사용자, 판매량 등의 차트가 표시됩니다.

인증 이벤트

사용자가 애플리케이션을 인증한 빈도를 나타냅니다.

분석 페이지의 사용자 탭에는 Sign-ins(로그인 횟수), Sign-ups(가입 횟수) 및 Authentication failures(인증 실패 횟수) 등의 차트가 표시됩니다.

시작하기 전 준비 사항

아직 수행하지 않은 경우 다음 작업을 수행하십시오.

- 앱을 AWS Amplify와 통합합니다. AWS Amplify를 사용하여 프런트엔드 애플리케이션을 Amazon Pinpoint에 연결을(를) 참조하세요.
- 애플리케이션을 업데이트해 엔드포인트를 등록합니다. 애플리케이션에 Amazon Pinpoint 엔드포인트 등록을(를) 참조하세요.

Android 또는 iOS용 AWS 모바일 SDKs를 사용하여 이벤트 보고

iOS 및 Android용 Mobile SDKs를 사용하여 AWS 모바일 앱이 Amazon Pinpoint에 이벤트를 보고하도록 할 수 있습니다.

이벤트를 기록하고 Amazon Pinpoint에 제출하도록 앱을 업데이트하는 방법에 대한 자세한 내용은 AWS Amplify 설명서의 다음 페이지를 참조하세요.

- iOS SDK 설명서의 분석
- Android SDK 설명서의 분석

AWS Amplify JavaScript 라이브러리를 사용하여 이벤트 보고

AWS Amplify JavaScript 라이브러리를 사용하여 JavaScript 및 React Native 앱을 활성화하여 애플리케이션 사용 이벤트를 Amazon Pinpoint에 보고할 수 있습니다. 이벤트를 기록하고 Amazon Pinpoint에 제출하기 위해 앱을 업데이트하는 방법은 AWS Amplify JavaScript 설명서의 분석 섹션을 참조하세요.

Amazon Pinpoint API를 사용하여 이벤트 보고

Amazon Pinpoint API 또는 AWS SDK를 사용하여 Amazon Pinpoint에 이벤트를 대량으로 제출할 수 있습니다. 자세한 내용은 Amazon Pinpoint API 참조의 이벤트 섹션을 참조하세요.

다음 단계

앱을 업데이트하여 이벤트를 보고하면 사용 데이터를 Amazon Pinpoint로 전송합니다. 콘솔에서 이러한 데이터를 확인하고 Amazon Kinesis로 스트리밍할 수 있습니다. 그런 다음 앱을 업데이트하여 Amazon Pinpoint를 사용해 보내는 푸시 알림을 처리하도록 할 수 있습니다. 자세한 내용은 AWS End User Messaging Push 사용 설명서의 다음 주제를 참조하세요.

- 푸시 알림 설정
- Swift 푸시 알림 설정
- Android 푸시 알림 설정
- Flutter 푸시 알림 설정
- React Native 푸시 알림 설정
- 프로젝트 만들기
- 푸시 알림 처리

Amazon Pinpoint를 사용하여 앱에서 트랜잭션 메시지 전송

Amazon Pinpoint API 및 AWS SDKs를 사용하여 앱에서 직접 트랜잭션 메시지를 보낼 수 있습니다. 트랜잭션 메시지는 세그먼트에 보내는 메시지와 달리, 특정 수신자에게 보내는 메시지입니다. 캠페인 기반 메시지 대신에 트랜잭션 메시지를 보내려는 데는 여러 가지 이유가 있습니다. 예를 들어 고객이 주문할 때 이메일로 주문 확인을 보낼 수 있습니다. 고객이 서비스에 대한 계정을 만드는 프로세스를 완료하는 데 사용할 수 있는 일회용 암호를 SMS 또는 음성으로 보낼 수도 있습니다.

이 단원에는 트랜잭션 이메일, SMS 메시지, 음성 메시지를 보내기 시작할 때 사용할 수 있는 여러 프로그래밍 언어로 된 코드 예제가 나와 있습니다.

엔드포인트, 세그먼트 및 채널에 대한 추가 코드 예제는 코드 예제를 참조하세요.

이 섹션의 주제:

- Amazon Pinpoint를 사용하여 트랜잭션 이메일 전송
- Amazon Pinpoint를 사용하여 트랜잭션 SMS 메시지 전송
- Amazon Pinpoint를 사용하여 음성 메시지 전송

Amazon Pinpoint를 사용하여 트랜잭션 이메일 전송

이 단원에는 Amazon Pinpoint를 통해 트랜잭션 이메일 메시지를 보낼 때 사용할 수 있는 전체 코드 샘플이 나와 있습니다.

- Amazon Pinpoint API의 SendMessages 작업을 사용하여: Amazon Pinpoint API의 `SendMessages` 작업을 사용하여 푸시 알림, SMS, 음성, 이메일 채널 등 Amazon Pinpoint가 지원하는 모든 채널에서 메시지를 보낼 수 있습니다.

  이 작업을 사용하는 이점은 메시지를 보내기 위한 요청 구문이 모든 채널에서 매우 비슷하다는 것입니다. 따라서 기존 코드를 용도 변경하기 더 쉽습니다. 또한 `SendMessages` 작업을 통해 이메일 메시지의 내용을 대체할 수 있고, 특정 이메일 주소가 아닌 Amazon Pinpoint 엔드포인트 ID에 이메일을 보낼 수 있습니다.

이 단원에는 트랜잭션 이메일을 보내기 시작할 때 사용할 수 있는 여러 프로그래밍 언어로 된 코드 예제가 나와 있습니다.

엔드포인트, 세그먼트 및 채널에 대한 추가 코드 예제는 코드 예제를 참조하세요.

이메일을 보낼 메서드 선택

트랜잭션 이메일을 보내는 데 사용할 수 있는 가장 좋은 방법은 사용 사례에 따라 다릅니다. 예를 들어 타사 애플리케이션을 사용하여 이메일을 보내야 하거나 프로그래밍 언어에 사용할 수 있는 AWS SDK가 없는 경우 SMTP 인터페이스를 사용해야 할 수 있습니다. Amazon Pinpoint가 지원하는 다른 채널에서 메시지를 전송하고 이러한 요청을 하기 위해 일관된 코드를 사용하고자 하는 경우, Amazon Pinpoint API의 `SendMessages` 작업을 사용해야 합니다.

Amazon Pinpoint와 Amazon SES 중에서 선택

구매 확인 또는 암호 재설정 메시지와 같은 많은 수의 트랜잭션 이메일을 보내는 경우 Amazon SES 사용을 고려하세요. Amazon SES에는 API 및 SMTP 인터페이스가 포함되어 있으며, 이 두 인터페이스 모두 애플리케이션 또는 서비스에서 이메일을 보내는 데 적합합니다. 또한 이메일 수신 기능, 구성 집합 및 전송 권한 부여 기능을 비롯한 추가 이메일 기능을 제공합니다.

Amazon SES에는 Salesforce와 같은 고객 관계 관리(CRM) 서비스를 비롯하여 기존 타사 애플리케이션과 통합할 수 있는 SMTP 인터페이스도 포함되어 있습니다. Amazon SES를 사용한 이메일 전송에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서를 참조하세요.

Amazon Pinpoint API를 사용하여 이메일 보내기

이 단원에는 AWS SDK를 사용하여 Amazon Pinpoint API를 통해 이메일을 보낼 때 사용할 수 있는 전체 코드 예제가 나와 있습니다. 이메일을 보내기 전에 이메일 주소 또는 도메인을 확인해야 합니다.

C#

이 예제를 사용하여 AWS SDK for .NET를 사용해 이메일을 보냅니다. 이 예제에서는 SDK for .NET를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 AWS SDK for .NET 개발자 안내서의 AWS SDK for .NET 시작하기를 참조하세요.

이 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 AWS SDK for .NET 개발자 안내서의 AWS 보안 인증 정보 구성을 참조하세요.

이 코드 예제는 AWS SDK for .NET 버전 3.3.29.13 및 .NET Core 런타임 버전 2.1.2를 사용하여 테스트되었습니다.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
```

```
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;

        // The Amazon Pinpoint project/application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }
```

```
    public static async Task<MessageResponse> SendEmailMessage(
        string region, string appId, string toAddress, string senderAddress)
    {
        var client = new
 AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        // The subject line of the email.
        string subject = "Amazon Pinpoint Email test";

        // The body of the email for recipients whose email clients don't
        // support HTML content.
        string textBody = @"Amazon Pinpoint Email Test (.NET)"
                          + "\n---------------------------------"
                          + "\nThis email was sent using the Amazon Pinpoint API
 using the AWS SDK for .NET.";

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK for .NET)</
h1>"
                          + "\n  <p>This email was sent using the "
                          + "\n    <a href='https://aws.amazon.com/pinpoint/'>Amazon
 Pinpoint</a> API "
                          + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                          + "\n  </p>"
                          + "\n</body>"
                          + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
```

```
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
                },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
                            },
                            Subject = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
        return response.MessageResponse;
    }
}
```

Java

이 예제를 사용하여 AWS SDK for Java를 사용해 이메일을 보냅니다. 이 예제에서는 AWS SDK for Java 2.x를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 AWS SDK for Java 2.x 개발자 안내서의 시작하기를 참조하세요.

이 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 AWS SDK for Java 개발자 안내서의 기본 보안 인증 정보 및 리전 설정을 참조하세요.

이 코드 예제는 AWS SDK for Java 버전 2.3.1 및 OpenJDK 버전 11.0.1을 사용하여 테스트되었습니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
```

```
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendEmailMessage {

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        public static String charset = "UTF-8";

    // The body of the email for recipients whose email clients support HTML
 content.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Java 2.x

        """;

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
 <toAddress>
```

```
            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
               toAddress - The to address. This address has to be verified in Amazon
Pinpoint in the region you're using to send email\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        System.out.println("Sending a message");
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        sendEmail(pinpoint, subject, senderAddress, toAddress);
        System.out.println("Email was sent");
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress) {
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
```

```
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

전체 SDK 예제를 보려면 GitHub에서 SendEmailMessage.java를 참조하세요.

JavaScript (Node.js)

Node.js의 JavaScript용 AWS SDK를 사용하여 이메일을 보내려면 이 예제를 사용하세요. 이 예제에서는 Node.js의 JavaScript용 SDK를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 Node.js의 JavaScript용 AWS SDK 개발자 설명서의 시작하기를 참조하세요.

이 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 Node.js의 JavaScript용 AWS SDK 개발자 설명서의 보안 인증 정보 설정을 참조하세요.

이 코드 예제는 Node.js 버전 2.388.0 및 Node.js 버전 11.7.0의 JavaScript용 SDK를 사용하여 테스트되었습니다.

```
"use strict";

const AWS = require("aws-sdk");
```

```
// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
var charset = "UTF-8";
```

```
// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
```

```
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
  }
});
```

Python

이 예제를 사용하여 AWS SDK for Python (Boto3)를 사용해 이메일을 보냅니다. 이 예제에서는 Python용 SDK(Boto3)를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 Python용 AWS SDK(Boto3) API 참조의 빠른 시작을 참조하세요.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
```

```
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
 account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and message
                     body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
 can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data": html_message},
                            "TextPart": {"Charset": char_set, "Data": text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
```

```
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -------------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

또한 다음 예제와 같이 메시지 템플릿을 사용하여 이메일 메시지를 보낼 수도 있습니다.

```
import logging
import boto3
```

```
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
                         account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress": sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
                    }
                },
            },
        )
    except ClientError:
```

```
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

이들 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 Python용 AWS SDK(Boto3) API 참조의 보안 인증 정보를 참조하세요.

Amazon Pinpoint를 사용하여 이메일에 구독 취소 헤더 추가

> **Note**
>
> 이메일 헤더를 사용하려면 먼저 캠페인 또는 여정에서 이메일을 보내는 경우 이메일 오케스트레이션 전송 역할을 설정해야 합니다. 직접 전송 이메일의 경우 `ses:SendEmail` 및

`ses:SendRawEmail`에 대한 권한이 있어야 합니다. 자세한 내용은 Amazon Pinpoint 사용 설명서의 이메일 오케스트레이션 전송 역할 생성을 참조하세요.

이메일에 구독 취소 링크를 포함하는 것이 모범 사례이며, 일부 국가에서는 법적 요구 사항이기도 합니다. 원클릭 구독 취소 링크를 추가하려면 다음 헤더를 추가합니다.

1. 헤더 이름을 `List-Unsubscribe`로 설정하고 값을 구독 취소 링크로 설정합니다. 수신자의 구독 취소 요청을 처리하려면 링크에서 HTTP POST 요청을 지원해야 합니다.
2. 헤더 이름을 `List-Unsubscribe-Post`로 설정하고 값을 `List-Unsubscribe=One-Click`으로 설정합니다.

이메일 메시지에 최대 15개의 헤더를 추가할 수 있습니다. 지원되는 헤더 목록은 Amazon Simple Email Service 개발자 안내서의 Amazon SES 헤더 필드를 참조하세요.

다음 예제에서는 AWS Command Line Interface를 사용하여 구독 취소 헤더가 포함된 이메일 메시지를 보내는 방법을 보여줍니다. 구성에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 구성을 AWS CLI AWS CLI참조하세요.

다음 명령에서 다음을 수행합니다.

- *AppId*를 애플리케이션 ID로 바꿉니다.
- *richard_roe@example.com*을 수신자의 이메일 주소로 바꿉니다.
- *https://example.com/unsub* 구독 취소 링크로 바꿉니다.
- *example123456*을 수신자의 고유 식별자로 바꿉니다.

```
aws pinpoint send-messages --application-id AppId --message-request '{
 "Addresses": {
   "richard_roe@example.com": {
     "ChannelType": "EMAIL"
   }
 },
 "MessageConfiguration": {
   "EmailMessage": {
     "Substitutions": {
       "url": [
         "https://example.com/unsub"
       ],
```

```
        "id1": [
          "/example123456"
        ]
      },
      "SimpleEmail": {
        "TextPart": {
          "Data": "Sample email message with an subscribe header",
          "Charset": "UTF-8"
        },
        "Subject": {
          "Data": "Hello",
          "Charset": "UTF-8"
        },
        "Headers": [
          {
            "Name": "List-Unsubscribe",
            "Value": "{{url}}{{id1}}"
          },
          {
            "Name": "List-Unsubscribe-Post",
            "Value": "List-Unsubscribe=One-Click"
          }
        ]
      }
    }
  }
}'
```

Amazon Pinpoint를 사용하여 트랜잭션 SMS 메시지 전송

Amazon Pinpoint API를 사용하여 특정 전화번호 또는 엔드포인트 ID에 SMS 메시지(문자 메시지)를 보낼 수 있습니다. 이 단원에는 AWS SDK를 사용하여 Amazon Pinpoint API를 통해 SMS 메시지를 보낼 때 사용할 수 있는 전체 코드 예제가 나와 있습니다. 계정이 프로덕션 상태여야 하며 SMS 메시지를 보낼 수 있는 활성 발신 ID가 있어야 합니다.

엔드포인트, 세그먼트 및 채널에 대한 추가 코드 예제는 코드 예제를 참조하세요.

C#

이 예제를 사용하여 AWS SDK for .NET를 사용해 SMS 메시지를 보냅니다. 이 예제에서는 SDK for .NET를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 AWS SDK for .NET 개발자 안내서의 시작하기를 참조하세요.

이 예제에서는 공유 자격 증명 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 보안 액세스 키를 지정한다고 가정합니다. 자세한 내용은 AWS SDK for .NET 개발자 안내서의 AWS 보안 인증 정보 구성을 참조하세요.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;
```

```
        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;

        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
 {messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
 originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";
```

```
        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

Java

이 예제를 사용하여 AWS SDK for Java를 사용해 SMS 메시지를 보냅니다. 이 예제에서는 Java용 SDK를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 AWS SDK for Java 개발자 안내서의 시작하기를 참조하세요.

이 예제에서는 공유 자격 증명 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 보안 액세스 키를 지정한다고 가정합니다. 자세한 내용은 AWS SDK for Java 개발자 안내서의 기본 보안 인증 정보 및 리전 설정을 참조하세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
```

```
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application ID
 to use when you send this message.
                                  originationNumber - The phone number or short code
 that you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
                                  destinationNumber - The recipient's phone number.
 For best results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                                  """;

                if (args.length != 4) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();
```

```
                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request = SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();
```

```
                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" + v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

전체 SDK 예제를 보려면 GitHub에서 SendMessage.java를 참조하세요.

JavaScript (Node.js)

Node.js의 JavaScript용 AWS SDK를 사용하여 SMS 메시지를 보내려면 이 예제를 사용하세요. 이 예제에서는 Node.js의 JavaScript용 SDK를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 Node.js의 JavaScript용 AWS SDK 개발자 설명서의 시작하기를 참조하세요.

이 예제에서는 공유 자격 증명 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 보안 액세스 키를 지정한다고 가정합니다. 자세한 내용은 Node.js의 JavaScript용 AWS SDK 개발자 설명서의 보안 인증 정보 설정을 참조하세요.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
```

```
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
var senderId = "MySenderID";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
```

```
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

Python

이 예제를 사용하여 AWS SDK for Python (Boto3)를 사용해 SMS 메시지를 보냅니다. 이 예제에서는 Python용 SDK를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 Python용 AWS SDK(Boto3) 시작하기의 Quickstart를 참조하세요.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
```

```
def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you send
                   this message. The SMS channel must be enabled for the project or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This phone
                               number must be associated with your Amazon Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you send
                         time-sensitive content, specify TRANSACTIONAL. If you send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
```

```
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK for
 "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

또한 다음 예제와 같이 메시지 템플릿을 사용하여 SMS 메시지를 보낼 수도 있습니다.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
```

```
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )

    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]
```

```
def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

이들 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 Python용 AWS SDK(Boto3) API 참조의 보안 인증 정보를 참조하세요.

Amazon Pinpoint를 사용하여 음성 메시지 전송

Amazon Pinpoint API를 사용하여 특정 전화번호에 음성 메시지를 보낼 수 있습니다. 이 단원에는 AWS SDK를 사용하여 Amazon Pinpoint SMS 및 음성 API를 통해 음성 메시지를 보낼 때 사용할 수 있는 전체 코드 예제가 나와 있습니다. 계정이 프로덕션 상태여야 하며 음성 메시지를 보낼 수 있는 활성 발신 ID가 있어야 합니다.

엔드포인트, 세그먼트 및 채널에 대한 추가 코드 예제는 코드 예제를 참조하세요.

Java

이 예제를 사용하여 AWS SDK for Java를 사용해 음성 메시지를 보냅니다. 이 예제에서는 Java용 SDK를 이미 설치 및 구성했다고 가정합니다. 자세한 내용은 AWS SDK for Java 개발자 안내서의 시작하기를 참조하세요.

이 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 AWS SDK for Java 개발자 안내서의 개발을 위한 AWS 보안 인증 정보 및 리전 설정을 참조하세요.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";

    // The language to use when sending the message. For a list of supported
    // languages, see
    // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
    static final String languageCode = "en-US";

    // The content of the message. This example uses SSML to customize and control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent from "
            + "<emphasis>Amazon Pinpoint</emphasis> "
            + "using the <break strength='weak'/>AWS "
            + "SDK for Java. "
            + "<amazon:effect phonation='soft'>Thank "
            + "you for listening.</amazon:effect></speak>";

    public static void main(String[] args) {

        final String usage = """
                Usage:   <originationNumber> <destinationNumber>\s

                Where:
                  originationNumber - The phone number or short code that you
 specify has to be associated with your Amazon Pinpoint account. For best results,
 specify long codes in E.164 format (for example, +1-555-555-5654).
                  destinationNumber - The recipient's phone number.  For best
 results, you should specify the phone number in E.164 format (for example,
 +1-555-555-5654).\s
                """;

        if (args.length != 2) {
```

```
            System.out.println(usage);
            System.exit(1);
        }
        String originationNumber = args[0];
        String destinationNumber = args[1];
        System.out.println("Sending a voice message");

        // Set the content type to application/json.
        List<String> listVal = new ArrayList<>();
        listVal.add("application/json");
        Map<String, List<String>> values = new HashMap<>();
        values.put("Content-Type", listVal);

        ClientOverrideConfiguration config2 = ClientOverrideConfiguration.builder()
                .headers(values)
                .build();

        PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
                .overrideConfiguration(config2)
                .region(Region.US_EAST_1)
                .build();

        sendVoiceMsg(client, originationNumber, destinationNumber);
        client.close();
    }

    public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                    String destinationNumber) {
        try {
            SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                    .languageCode(languageCode)
                    .text(ssmlMessage)
                    .voiceId(voiceName)
                    .build();

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
                    .originationPhoneNumber(originationNumber)
```

```
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

        } catch (PinpointSmsVoiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

전체 SDK 예제를 보려면 GitHub에서 SendVoiceMessage.java를 참조하세요.

JavaScript (Node.js)

Node.js의 JavaScript용 AWS SDK를 사용하여 음성 메시지를 보내려면 이 예제를 사용하세요. 이 예제에서는 Node.js의 JavaScript용 SDK를 이미 설치 및 구성했다고 가정합니다.

이 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 Node.js의 JavaScript용 AWS SDK 개발자 설명서의 보안 인증 정보 설정을 참조하세요.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best results,
 you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";
```

```
// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
```

```
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

Python

이 예제를 사용하여 AWS SDK for Python (Boto3)를 사용해 음성 메시지를 보냅니다. 이 예제에서는 Python용 SDK(Boto3)를 이미 설치 및 구성했다고 가정합니다.

이 예제에서는 공유 보안 인증 정보 파일을 사용하여 기존 IAM 사용자의 액세스 키 및 비밀 액세스 키를 지정한다고 가정합니다. 자세한 내용은 Python용 AWS SDK(Boto3) API 참조의 보안 인증 정보를 참조하세요.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
```

```
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
 message.
    :param ssml_message: The content of the message. This example uses SSML to
 control
                         certain aspects of the message, such as the volume and the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
```

```
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

AWS End User Messaging SMS and Voice API 버전 2 사용

Amazon Pinpoint에는 SMS 및 음성 메시지 전송을 위해 설계된 API(SMS and Voice API 버전 2라고 함)가 포함되어 있습니다. Amazon Pinpoint API가 예약 및 이벤트 기반 캠페인 및 여정을 통해 메시지를 보내는 데 중점을 두는 반면, SMS 및 음성 API는 SMS 메시지와 음성 메시지를 개별 수신자에게 직접 전송하는 새로운 기능을 제공합니다. SMS and Voice API를 Amazon Pinpoint 캠페인 및 여정 기능과 별도로 사용할 수도 있고, 둘 다 동시에 사용하여 다양한 사용 사례를 수용할 수도 있습니다. 이미 Amazon Pinpoint를 사용하여 SMS 메시지 또는 음성 메시지를 보내는 경우, 계정은 이미 이 API를 사용하도록 구성되어 있습니다.

이 API는 독립 소프트웨어 개발 판매 회사(ISV)와 같은 멀티 테넌트 아키텍처를 사용하는 사용자에게 적합한 솔루션입니다. 이 API를 사용하면 이벤트 데이터, 발신 전화번호, 옵트아웃 목록을 테넌트별로 구분하기가 더 쉬워집니다.

SMS 및 음성 API를 사용할 때는 구성 세트과 이벤트 대상을 설정하는 것이 좋습니다. SMS 및 음성 API는 전송하는 메시지에 대한 이벤트 데이터를 자동으로 내보내지 않습니다. 이벤트 대상을 설정하면 메시지 전송 및 실패 이벤트와 같은 중요한 이벤트 데이터를 캡처할 수 있습니다.

이 API의 버전 2는 버전 1의 후속 버전입니다. 현재 이 API의 버전 1을 사용하고 있는 경우 계속 사용할 수 있습니다. 버전 2로 마이그레이션하면 전화번호 풀을 생성하고, 프로그래밍 방식으로 새 전화번호를 요청하고, 전화번호의 특정 기능을 활성화 또는 비활성화하는 등의 추가 기능을 사용할 수 있습니다.

> **Note**
>
> 일부 작업은 Amazon Pinpoint 콘솔을 사용해야만 완료할 수 있습니다. 예를 들어 계정이 SMS 샌드박스에 있는 동안 사용할 전화번호를 확인하고 10DLC를 사용하도록 등록하는 경우가 있습니다.

Amazon Pinpoint SMS and Voice 버전 2 API에 대한 자세한 내용은 SMS and Voice, 버전 2 API 참조를 참조하세요. AWS End User Messaging SMS 및 음성 리소스를 생성, 구성 및 관리하는 방법에 대한 자세한 내용은 AWS End User Messaging SMS 사용 설명서를 참조하세요.

Amazon Pinpoint를 사용하여 일회용 암호(OTP) 생성

Amazon Pinpoint에는 새 일회용 암호를 생성하고 수신자에게 SMS 메시지로 보내는 데 사용할 수 있는 일회용 암호(OTP) 관리 기능이 포함되어 있습니다.

> **Important**
>
> 이 기능을 사용하려면 계정에 프로덕션 액세스 권한과 활성 발신 ID가 있어야 합니다. 자세한 내용은 AWS End User Messaging SMS 사용 설명서의 SMS/MMS 및 음성 샌드박스 정보 및 전화번호 요청을 참조하세요.

일부 국가 및 지역에서는 SMS 메시지를 전송하려면 먼저 전용 전화번호 또는 발신 ID를 받아야 합니다. 예를 들어, 미국의 수신자에게 메시지를 전송할 때는 전용 수신자 부담 전화번호, 10DLC 번호 또는 단축 코드가 있어야 합니다. 인도의 수신자에게 메시지를 전송할 때는 등록된 발신자 ID가 있어야 합니다. 여기에는 보안 주체 엔터티 ID(PEID), 템플릿 ID가 포함됩니다. 이러한 요구 사항은 OTP 기능을 사용하는 경우에도 적용됩니다.

이 기능을 사용하려면 OTP 메시지를 전송 및 확인할 수 있는 권한이 필요합니다(일회용 암호 참조). 권한을 결정하는 데 도움이 필요한 경우 Amazon Pinpoint의 ID 및 액세스 관리에 대한 문제 해결 섹션을 참조하세요.

Amazon Pinpoint API의 `SendOtpMessages` 작업을 사용해 애플리케이션 사용자에게 OTP 코드를 전송할 수 있습니다. 이 API를 사용하면 Amazon Pinpoint가 무작위 코드를 생성하여 사용자에게 SMS 메시지로 전송합니다. 요청은 다음 파라미터를 포함해야 합니다.

- `Channel` - OTP 코드가 전송되는 통신 채널입니다. 현재는 SMS 메시지만 지원되므로 허용되는 값은 SMS뿐입니다.
- `BrandName` - OTP 코드와 연결된 브랜드, 회사 또는 제품의 이름입니다. 이 이름은 최대 20자를 포함할 수 있습니다.

> **Note**
>
> Amazon Pinpoint가 OTP 메시지를 전송할 때 브랜드 이름이 다음 메시지 템플릿에 자동으로 삽입됩니다.
>
> ```
> This is your One Time Password: {{otp}} from {{brand}}
> ```

따라서 ExampleCorp를 브랜드 이름으로 지정하고 Amazon Pinpoint가 123456이라는 일회용 암호를 생성하면 사용자에게 다음과 같은 메시지가 전송됩니다.

```
This is your One Time Password: 123456 from ExampleCorp
```

- `CodeLength` - 수신자에게 전송되는 OTP 코드의 자릿수입니다. OTP 코드는 5~8자리 숫자를 포함할 수 있습니다.
- `ValidityPeriod` - OTP 코드가 유효한 시간(분)입니다. 유효 기간은 5~60분입니다.
- `AllowedAttempts` - 수신자가 OTP 확인을 시도할 수 있는 횟수입니다. 시도 횟수가 이 값을 초과하면 OTP는 자동으로 무효화됩니다. 허용되는 최대 시도 횟수는 5회입니다.
- `Language` - 메시지를 전송할 때 사용할 IETF BCP-47 형식의 언어입니다. 사용 가능한 값은 다음과 같습니다.
  - `de-DE` - 독일어
  - `en-GB` - 영어(영국)
  - `en-US` - 영어(미국)
  - `es-419` - 스페인어(남미)
  - `es-ES` - 스페인어
  - `fr-CA` - 프랑스어(캐나다)
  - `fr-FR` - 프랑스어
  - `it-IT` - 이탈리아어
  - `ja-JP` - 일본어
  - `ko-KR` - 한국어
  - `pt-BR` - 포르투갈어(브라질)
  - `zh-CN` - 중국어 간체
  - `zh-TW` - 중국어 번체
- `OriginationIdentity`— OTP 코드를 전송하는 데 사용되는 발신 ID(예: 긴 코드, 단축 코드 또는 발신자 ID)입니다. 긴 코드 또는 수신자 부담 전화번호를 사용하여 OTP를 전송하는 경우 전화번호는 E.164 형식이어야 합니다.
- `DestinationIdentity` - OTP 코드가 전송된 E.164 형식의 전화번호입니다.
- `ReferenceId` - 요청의 고유한 참조 ID입니다. 참조 ID는 OTP를 확인할 때 제공하는 참조 ID와 정확히 일치합니다. 참조 ID는 1~48자를 포함할 수 있습니다.

- `EntityId` - 규제 기관에 등록된 엔티티 ID입니다. 이 파라미터는 현재 인도에 있는 수신자에게 메시지를 전송할 때만 사용됩니다. 인도의 수신자에게 보내지 않는 경우에는 이 파라미터를 생략할 수 있습니다.
- `TemplateId` - 규제 기관에 등록된 템플릿 ID입니다. 이 파라미터는 현재 인도에 있는 수신자에게 메시지를 전송할 때만 사용됩니다. 인도의 수신자에게 보내지 않는 경우에는 이 파라미터를 생략할 수 있습니다.

> **Note**
>
> 인도의 수신자에게 메시지를 전송하기 위한 요구 사항에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 인도 발신자 ID 등록 프로세스를 참조하세요.

Amazon Pinpoint 계정이 OTP 메시지를 전송하도록 올바르게 구성되어 있는지 확인하려면 AWS CLI를 사용하여 테스트 메시지를 보낼 수 있습니다. 에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 AWS CLI참조하세요.

를 사용하여 테스트 OTP 메시지를 보내려면 터미널에서 send-otp-message 명령을 AWS CLI실행합니다.

```
aws pinpoint send-otp-message --application-id 7353f53e6885409fa32d07cedexample --send-
otp-message-request-parameters
 Channel=SMS,BrandName=ExampleCorp,CodeLength=5,ValidityPeriod=20,AllowedAttempts=5,Origination
```

위의 명령에서 다음을 수행합니다.

- *7353f53e6885409fa32d07cedexample*을 애플리케이션 ID로 바꿉니다.
- *ExampleCorp*를 회사 이름으로 바꿉니다.
- `CodeLegth`의 *5*를 수신자에게 전송되는 OTP 코드의 자릿수로 바꿉니다.
- `ValidityPeriod`의 *20*을 OTP 코드가 유효한 분 단위의 시간으로 바꿉니다.
- `AllowedAttempts`의 *5*를 수신자가 OTP 확인을 시도했지만 실패할 수 있는 횟수로 바꿉니다.
- `OriginationIdentity`의 *+18555550142*을 OTP 코드를 보내는 데 사용되는 발신 ID로 바꿉니다.
- `DestinationIdentity`의 *+12065550007*을 OTP 코드를 보낼 전화번호로 바꿉니다.
- `ReferenceId`의 *SampleReferenceId*를 요청에 대한 고유한 참조 ID로 바꿉니다.

SendOtpMessage 응답

OTP 메시지를 성공적으로 전송하면 다음 예시와 유사한 응답을 받게 됩니다.

```
{
    "MessageResponse": {
        "ApplicationId": "7353f53e6885409fa32d07cedexample",
        "RequestId": "255d15ea-75fe-4040-b919-096f2example",
        "Result": {
            "+12065550007": {
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "nvrmgq9kq4en96qgp0tlqli3og1at6aexample",
                "StatusCode": 200,
                "StatusMessage": "MessageId: nvrmgq9kq4en96qgp0tlqli3og1at6aexample"
            }
        }
    }
}
```

Amazon Pinpoint에서 OTP 메시지 검증

일회용 암호를 전송한 후 애플리케이션이 Amazon Pinpoint API를 직접적으로 호출하여 확인할 수 있습니다. OTP 코드를 확인하려면 `VerifyOtpMessages` API를 호출합니다. 요청은 다음 파라미터를 포함해야 합니다.

- `DestinationIdentity` - OTP 코드가 전송된 E.164 형식의 전화번호입니다.
- `ReferenceId` - 수신자에게 OTP 코드를 전송할 때 사용한 참조 ID입니다. 참조 ID는 정확히 일치해야 합니다.
- `Otp` - 확인하려는 OTP 코드입니다.

를 사용하여 검증 프로세스를 테스트 AWS CLI 할 수 있습니다. 설치 및 구성에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 AWS CLI참조하세요.

를 사용하여 OTP를 확인하려면 터미널에서 verify-otp-message 명령을 AWS CLI실행합니다.

```
aws pinpoint verify-otp-message --application-id 7353f53e6885409fa32d07cedexample --
verify-otp-message-request-parameters
 DestinationIdentity=+12065550007,ReferenceId=SampleReferenceId,Otp=01234
```

위의 명령에서 다음을 수행합니다.

- *7353f53e6885409fa32d07cedexample*을 애플리케이션 ID로 바꿉니다.
- DestinationIdentity의 *+12065550007*을 OTP 코드가 전송된 전화번호로 바꿉니다.
- ReferenceId의 *SampleReferenceId*를 요청에 대한 고유한 참조 ID로 바꿉니다. 이 값은 요청을 보내는 데 사용된 ReferenceID와 일치해야 합니다.
- Otp의 *01234*를 DestinationIdentity로 전송된 OTP로 바꿉니다.

VerifyOtpMessage 응답

VerifyOTPMessage API로 요청을 보내면 이 API는 단일 속성 Valid를 포함하는 VerificationResponse 객체를 반환합니다. 참조 ID, 전화번호 및 OTP 모두 Amazon Pinpoint가 예상하는 값과 일치하고 OTP가 만료되지 않은 경우 Valid의 값은 true이고, 그렇지 않은 경우 false입니다. 다음은 OTP 확인 성공 응답의 예입니다.

```
{
    "VerificationResponse": {
        "Valid": true
    }
}
```

Amazon Pinpoint에서 SDK for Python (Boto3)을 사용한 OTP 코드 예제

이 단원에는 Python용 SDK(Boto3)를 사용하여 OTP 코드를 전송하고 확인하는 방법을 보여주는 코드 예제가 포함되어 있습니다.

참조 ID 생성

다음 함수는 수신자의 전화번호, 수신자가 OTP를 수신하는 제품 또는 브랜드, 요청의 출처(예: 사이트 또는 앱의 페이지 이름)를 기반으로 각 수신자의 고유한 참조 ID를 생성합니다. OTP 코드를 확인할 때 동일한 참조 ID를 전달해야 확인이 성공할 수 있습니다. 전송 및 확인 코드 예제 모두 이 유틸리티 함수를 사용합니다.

이 함수가 필수는 아니지만 확인 단계에서 쉽게 다시 제출할 수 있도록 OTP 전송 및 확인 프로세스의 범위를 특정 트랜잭션으로 한정할 수 있는 유용한 방법입니다. 참조 ID는 원하는 대로 사용할 수 있습니다. 이는 기본적인 예시일 뿐입니다. 단, 이 단원의 다른 코드 예제는 이 함수를 사용합니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import hashlib

def generate_ref_id(destinationNumber,brandName,source):
    refId = brandName + source + destinationNumber
    return hashlib.md5(refId.encode()).hexdigest()
```

OTP 코드 전송

다음 코드 예제는 Python용 SDK(Boto3)를 사용하여 OTP 코드를 전송하는 방법을 보여줍니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

### Some variables that are unlikely to change from request to request. ###

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The phone number or short code to send the message from.
originationNumber = "+18555550142"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# The number of times the user can unsuccessfully enter the OTP code before it becomes
 invalid.
allowedAttempts = 3

# Function that sends the OTP as an SMS message.
def send_otp(destinationNumber,codeLength,validityPeriod,brandName,source,language):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.send_otp_message(
            ApplicationId=appId,
            SendOTPMessageRequestParameters={
                'Channel': 'SMS',
```

```
                'BrandName': brandName,
                'CodeLength': codeLength,
                'ValidityPeriod': validityPeriod,
                'AllowedAttempts': allowedAttempts,
                'Language': language,
                'OriginationIdentity': originationNumber,
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source)
            }
        )

    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Send a message to +14255550142 that contains a 6-digit OTP that is valid for 15
 minutes. The
# message will include the brand name "ExampleCorp", and the request originated from a
 part of your
# site or application called "CreateAccount". The US English message template should be
 used to
# send the message.
send_otp("+14255550142",6,15,"ExampleCorp","CreateAccount","en-US")
```

OTP 코드 확인

다음 코드 예제는 Python용 SDK(Boto3)를 사용하여 이미 전송한 OTP 코드를 확인하는 방법을 보여줍니다. 확인 단계가 성공하려면 메시지를 전송하는 데 사용된 참조 ID와 정확히 일치하는 참조 ID가 요청에 포함되어야 합니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"
```

```
# Function that verifies the OTP code.
def verify_otp(destinationNumber,otp,brandName,source):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.verify_otp_message(
            ApplicationId=appId,
            VerifyOTPMessageRequestParameters={
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source),
                'Otp': otp
            }
        )

    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Verify the OTP 012345, which was sent to +14255550142. The brand name ("ExampleCorp")
 and the
# source name ("CreateAccount") are used to generate the correct reference ID.
verify_otp("+14255550142","012345","ExampleCorp","CreateAccount")
```

Amazon Pinpoint 및 Amplify를 사용하여 인앱 메시지 사용자 지정

인앱 메시지를 사용하여 애플리케이션 사용자에게 타겟팅된 메시지를 보낼 수 있습니다. 인앱 메시지는 자유롭게 사용자 지정할 수 있습니다. 인앱 메시지에는 웹 사이트를 열거나 사용자를 앱의 특정 부분으로 이동시키는 버튼이 포함될 수 있습니다. 배경색과 텍스트 색상을 구성하고, 텍스트 위치를 조정하고, 알림에 버튼과 이미지를 추가할 수 있습니다. 단일 메시지를 보내거나 최대 5개의 고유한 메시지가 포함된 캐러셀을 만들 수 있습니다. 인앱 메시지 템플릿 생성 지침을 비롯한 인앱 메시지의 개요는 Amazon Pinpoint 사용 설명서의 인앱 템플릿 생성 섹션을 참조하세요.

AWS Amplify 를 사용하여 Amazon Pinpoint의 인앱 메시징 기능을 앱에 원활하게 통합할 수 있습니다. Amplify는 메시지를 가져오고, 메시지를 렌더링하고, 분석 데이터를 Amazon Pinpoint로 보내는 프로세스를 자동으로 처리할 수 있습니다. 이 통합은 현재 React Native 애플리케이션에서 지원됩니다. 자세한 내용은 Amplify Framework 설명서의 인앱 메시징을 참조하세요.

Amazon Pinpoint를 사용하여 프로그래밍 방식으로 엔드포인트에 대한 인앱 메시지 검색

애플리케이션은 GetInAppMessages API를 직접적으로 호출하여 주어진 엔드포인트에 권한이 부여된 모든 인앱 메시지를 검색할 수 있습니다. `GetInAppMessages` API를 직접적으로 호출할 경우 다음과 같은 파라미터를 제공하게 됩니다.

- `ApplicationId` - 인앱 메시지 캠페인이 연결된 Amazon Pinpoint 앱의 고유 ID입니다.
- `EndpointId` - 메시지를 검색할 엔드포인트의 고유 ID입니다.

이러한 값을 사용하여 API를 직접적으로 호출하면 메시지 목록이 반환됩니다. 이 작업으로 생성된 응답에 대한 자세한 내용은 `GetInAppMessages` Amazon Pinpoint API 응답 JSON 예제 섹션을 참조하세요.

AWS SDKs를 사용하여 `GetInAppMessages` 작업을 호출할 수 있습니다. 다음 코드 예제에는 인앱 메시지를 검색하는 함수가 포함되어 있습니다.

JavaScript

별도의 모듈에서 클라이언트를 생성하고 내보냅니다.

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
```

```
const REGION = "us-east-1";
const pinClient = new PinpointClient({ region: REGION });
export { pinClient };
```

엔드포인트에 대한 인앱 메시지를 검색합니다.

```
// Import required AWS SDK clients and commands for Node.js
import { PinpointClient, GetInAppMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./lib/pinClient.js";

("use strict");

//The Amazon Pinpoint application ID.
const projectId = "4c545b28d21a490cb51b0b364example";

//The ID of the endpoint to retrieve messages for.
const endpointId = "c5ac671ef67ee3ad164cf7706example";

const params = {
  ApplicationId: projectId,
  EndpointId: endpointId
};

const run = async () => {
  try {
    const data = await pinClient.send(new GetInAppMessagesCommand(params));
    console.log(JSON.stringify(data, null, 4));
    return data;
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Python

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
```

```
def retrieve_inapp_messages(
            pinpoint_client, project_id, endpoint_id):
    """
    Retrieves the in-app messages that a given endpoint is entitled to.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project ID.
    :param endpoint_id: The ID of the endpoint to retrieve messages for.
    :return: A JSON object that contains information about the in-app message.
    """

    try:
        response = pinpoint_client.get_in_app_messages(
            ApplicationId=project_id,
            EndpointId=endpoint_id)
    except ClientError:
        logger.exception("Couldn't retrieve messages.")
        raise
    else:
        return response

def main():
    project_id = "4c545b28d21a490cb51b0b364example"
    endpoint_id = "c5ac671ef67ee3ad164cf7706example"
    inapp_response = retrieve_inapp_messages(
        boto3.client('pinpoint'), project_id, endpoint_id)
    print(inapp_response)

if __name__ == '__main__':
    main()
```

GetInAppMessages Amazon Pinpoint API 응답 JSON 예제

GetInAppMessages API 작업을 직접적으로 호출하면 지정된 엔드포인트에 권한이 부여된 메시지 목록이 반환됩니다. 이렇게 되면 앱이 응답의 값을 기반으로 메시지를 렌더링할 수 있습니다.

아래는 GetInAppMessages API를 직접적으로 호출할 때 반환되는 JSON 객체의 예제입니다.

```
{
  "InAppMessagesResponse":{
    "InAppMessageCampaigns":[
      {
```

```
        "CampaignId":"inAppTestCampaign-4c545b28d21a490cb51b0b364example",
        "DailyCap":0,
        "InAppMessage":{
          "Content":[
            {
              "BackgroundColor":"#f8e71c",
              "BodyConfig":{
                "Alignment":"CENTER",
                "Body":"This is a sample in-app message sent using Amazon Pinpoint.",
                "TextColor":"#d0021b"
              },
              "HeaderConfig":{
                "Alignment":"CENTER",
                "Header":"Sample In-App Message",
                "TextColor":"#d0021b"
              },
              "ImageUrl":"https://example.com/images/thumbnail.png",
              "PrimaryBtn":{
                "DefaultConfig":{
                  "BackgroundColor":"#d0021b",
                  "BorderRadius":50,
                  "ButtonAction":"CLOSE",
                  "Text":"Dismiss",
                  "TextColor":"#f8e71c"
                }
              }
            }
          ],
          "Layout":"MIDDLE_BANNER"
        },
        "Priority":3,
        "Schedule":{
          "EndDate":"2021-11-06T00:08:05Z",
          "EventFilter":{
            "Dimensions":{
              "Attributes":{

              },
              "EventType":{
                "DimensionType":"INCLUSIVE",
                "Values":[
                  "_session.start"
                ]
              },
```

```
              "Metrics":{

              }
            }
          }
        },
        "SessionCap":0,
        "TotalCap":0,
        "TreatmentId":"0"
      }
    ]
  }
}
```

다음 섹션에서는 이 응답의 구성 요소 및 해당 속성에 대한 추가 정보를 제공합니다.

InAppMessageCampaigns 객체

각 InAppMessageCampaigns 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
CampaignId	메시지를 보낸 Amazon Pinpoint 캠페인의 이름 및 고유한 캠페인 ID가 포함된 문자열입니다. 이름은 캠페인 ID 앞에 옵니다. 두 값은 하이픈(-)으로 구분됩니다.	캠페인을 생성할 경우 Amazon Pinpoint에서 자동으로 생성합니다.
TreatmentId	이 메시지에 대한 캠페인 처리 ID를 나타내는 정수입니다. 캠페인에 처리가 한 개만 있는 경우 값은 0입니다.	
Priority	인앱 메시지의 우선순위로, 1~5 사이의 정수로 표시됩니다. 여기서 1은 가장 높은 우선순위를 나타내고, 5는 가장 낮은 우선순위를 나타냅니다.	캠페인 생성 프로세스의 1단계.

속성	설명	설정 위치
InAppMessage	메시지가 렌더링되는 방식에 대한 정보가 들어 있는 InAppMessage 객체입니다.	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.
Schedule	메시지를 보낸 시간에 대한 정보가 들어 있는 Schedule 객체입니다.	캠페인 생성 프로세스의 4단계(콘솔에서 캠페인이 생성된 경우) 또는 Schedule 객체(API 또는 SDK를 사용하여 캠페인을 생성한 경우).
DailyCap	24시간 동안 인앱 메시지가 사용자에게 표시될 수 있는 횟수(정수로 표시)입니다.	프로젝트 수준 설정에서 상속됩니다. 캠페인에 프로젝트 설정을 재정의하는 설정이 포함된 경우 해당 설정이 대신 사용됩니다.
SessionCap	애플리케이션 세션 중에 사용자에게 인앱 메시지가 표시될 수 있는 횟수(정수로 표시)입니다.	
TotalCap	캠페인당 인앱 메시지가 엔드포인트에 표시될 수 있는 총 횟수(정수로 표시)입니다.	

InAppMessage 객체

각 InAppMessage 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
Content	메시지 내용을 설명하는 InAppMessageContent 객체가 포함된 배열입니다.	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.
Layout	인앱 메시지가 수신자의 디바이스에 표시되는 방식을 설명	

속성	설명	설정 위치
	하는 문자열입니다. 가능한 값은 다음과 같습니다. • BOTTOM_BANNER - 페이지 하단에 배너로 표시되는 메시지입니다. • TOP_BANNER - 페이지 상단에 배너로 표시되는 메시지입니다. • OVERLAYS - 전체 화면을 덮는 메시지입니다. • MOBILE_FEED - 페이지 앞의 창에 표시되는 메시지입니다. • MIDDLE_BANNER - 페이지 중간에 배너로 표시되는 메시지입니다. • CAROUSEL - 최대 5개 고유 메시지의 스크롤 가능한 레이아웃입니다.	

HeaderConfig 객체

각 HeaderConfig 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
Alignment	헤더 텍스트의 텍스트 정렬을 지정하는 문자열입니다. 가능한 값은 LEFT, CENTER 및 RIGHT입니다.	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.
Header	메시지 헤더 텍스트입니다.	

속성	설명	설정 위치
TextColor	16진수 색상 코드를 설명하는 문자열로 표현되는 헤더 텍스트의 색입니다(예: 검은색의 경우 '#000000').	

BodyConfig 객체

각 BodyConfig 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
Alignment	메시지 본문의 텍스트 정렬을 지정하는 문자열입니다. 가능한 값은 LEFT, CENTER 및 RIGHT입니다.	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.
Body	메시지의 본문 텍스트입니다.	
TextColor	16진수 색상 코드가 포함된 문자열로 표현되는 본문 텍스트의 색상입니다(예: 검정색의 경우 '#000000').	

InAppMessageContent 객체

각 InAppMessageContent 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
BackgroundColor	16진수 색상 코드가 포함된 문자열로 표시되는 인앱 메시지의 배경색입니다(예: 검은색의 경우 '#000000').	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.

속성	설명	설정 위치
BodyConfig	메시지의 본문 내용과 관련된 정보가 포함되어 있는 BodyConfig 객체입니다.	
HeaderConfig	메시지의 헤더 또는 제목과 관련된 정보가 포함되어 있는 HeaderConfig 객체입니다.	
ImageUrl	메시지에 표시되는 이미지의 URL입니다.	
PrimaryBtn	메시지의 기본 버튼에 대한 정보가 포함되어 있는 InAppMessageButton 객체입니다.	
SecondaryBtn	메시지의 보조 버튼에 대한 정보가 포함되어 있는 InAppMessageButton 객체입니다. 인앱 메시지 템플릿에 보조 버튼이 지정되지 않은 경우에는 표시되지 않습니다.	

Schedule 객체

각 Schedule 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
EndDate	캠페인이 종료될 예정인 ISO 8601 형식의 예약 시간입니다.	캠페인 생성 프로세스의 4단계(콘솔에서 캠페인이 생성된 경우) 또는 Schedule 객체(API 또는 SDK를 사용하여 캠페인을 생성한 경우).
EventFilter	인앱 메시지를 표시하게 하는 이벤트에 대한 정보입니다. Amazon Pinpoint 인앱 캠페인	

속성	설명	설정 위치
	과 일치하는 이벤트를 생성하면 메시지가 표시됩니다.	

InAppMessageButton 객체

InAppMessageButton 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
DefaultConfig	인앱 메시지에 있는 버튼의 기본 설정에 대한 정보가 포함되어 있는 DefaultButtonConfig 객체입니다.	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.
Android	Android 디바이스에서 버튼이 동작하는 방식을 지정하는 OverrideButtonConfig 객체입니다. 이는 DefaultConfig 객체에 자세히 설명된 기본 버튼 구성을 재정의합니다.	
IOS	iOS 디바이스에서 버튼이 동작하는 방식을 지정하는 OverrideButtonConfig 객체입니다. 이는 DefaultConfig 객체에 자세히 설명된 기본 버튼 구성을 재정의합니다.	
Web	웹 앱에서 버튼이 동작하는 방식을 지정하는 OverrideButtonConfig 객체입니다. 이는 DefaultConfig 객체에 자세히 설명된 기본 버튼 구성을 재정의합니다.	

DefaultButtonConfig 객체

DefaultButtonConfig 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
BackgroundColor	16진수 색상 코드가 포함된 문자열로 표시되는 버튼의 배경색입니다(예: 검은색의 경우 '#000000').	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.
BorderRadius	버튼 테두리의 반경(픽셀 단위)으로, 정수로 표현됩니다. 숫자가 클수록 모서리가 더 둥글어집니다.	
ButtonAction	수신자가 인앱 메시지에서 버튼을 선택할 때 수행되는 작업을 설명하는 문자열입니다. 가능한 값은 다음과 같습니다. • LINK - 웹 대상에 대한 링크입니다. • DEEP_LINK - 애플리케이션의 특정 페이지에 대한 링크입니다. • CLOSE - 메시지를 닫습니다.	
Link	버튼의 대상 URL입니다. ButtonAction이 CLOSE인 버튼에는 표시되지 않습니다.	
Text	버튼에 표시되는 텍스트입니다.	
TextColor	16진수 색상 코드가 포함된 문자열로 표시되는 텍스트의	

속성	설명	설정 위치
	색입니다(예: 검은색의 경우 '#000000').	

OverrideButtonConfig 객체

OverrideButtonConfig 객체는 인앱 메시지 템플릿이 재정의 버튼을 사용하는 경우에만 표시됩니다. 재정의 버튼은 iOS 디바이스, Android 디바이스 또는 웹 브라우저와 같은 특정 디바이스 유형에 대한 특정한 구성이 있는 버튼입니다.

OverrideButtonConfig 객체에는 다음 속성이 포함됩니다.

속성	설명	설정 위치
ButtonAction	수신자가 인앱 메시지에서 버튼을 선택할 때 수행되는 작업입니다. 가능한 값은 다음과 같습니다. • LINK - 웹 대상에 대한 링크입니다. • DEEP_LINK - 애플리케이션의 특정 페이지에 대한 링크입니다. • CLOSE - 메시지를 닫습니다.	캠페인에 지정된 인앱 메시지 템플릿의 콘텐츠를 기반으로 합니다.
Link	버튼의 대상 URL입니다. ButtonAction 이 CLOSE인 버튼에는 표시되지 않습니다.	
Text	버튼에 표시되는 텍스트입니다.	
TextColor	16진수 색상 코드가 포함된 문자열로 표시되는 텍스트의	

속성	설명	설정 위치
	색입니다(예: 검은색의 경우 '#000000').	

Amazon Pinpoint 전화번호 검증 서비스 사용

Amazon Pinpoint에는 전화 번호가 유효한지 확인하고 전화 번호 자체에 대한 추가 정보를 얻는 데 사용할 수 있는 전화 번호 확인 서비스가 포함되어 있습니다. 예를 들어 전화 번호 확인 서비스를 사용하면 다음 정보가 반환됩니다.

- E.164 형식의 전화 번호
- 전화 번호 유형(예: 모바일, 유선 또는 VoIP)
- 해당 전화 번호의 도시 및 국가
- 전화 번호와 관련된 서비스 공급업체

전화 번호 확인 서비스 사용 시에는 추가 요금이 부과됩니다. 자세한 내용은 Amazon Pinpoint 요금을 참조하세요.

> **Important**
>
> 미국 및 캐나다에서 발신하는 전화번호의 경우 전화번호 확인 API는 더 이상 `City`, `County`, `Timezone` 및 `ZipCode`에 대한 데이터를 반환하지 않습니다.

Amazon Pinpoint 전화번호 검증 사용 사례

전화 번호 확인 서비스를 사용하여 다음을 비롯한 여러 사용 사례를 활성화할 수 있습니다.

- 웹 양식에 제공된 전화번호 확인 - 웹 기반 양식을 사용하여 고객의 연락처 정보를 수집하는 경우 양식을 제출하기 전에 고객이 제공한 전화 번호를 확인합니다. 웹 사이트의 백엔드를 활용하여 Amazon Pinpoint API로 번호를 확인합니다. API 응답에는 번호가 유효하지 않은지 여부(예: 전화번호 형식이 잘못된 경우)가 표시됩니다. 고객이 제공한 전화 번호가 유효하지 않다고 판단되면 웹 양식에서 고객에게 다른 번호를 제공하라는 메시지를 표시할 수 있습니다.
- 기존 연락처 데이터베이스 정리 - 고객 전화 번호 데이터베이스가 있는 경우 각 전화 번호를 확인하고 결과를 기반으로 데이터베이스를 업데이트할 수 있습니다. 예를 들어, SMS 메시지를 수신할 수 없는 전화 번호가 있는 엔드포인트를 찾은 경우 엔드포인트의 `ChannelType` 속성을 `SMS`에서 `VOICE`로 변경할 수 있습니다. 먼저 전화번호를 확인한 다음 단일 엔드포인트의 Amazon Pinpoint에 엔드포인트 추가 또는 여러 엔드포인트의 Amazon Pinpoint에 엔드포인트 배치 추가 내 지침에 따라 새 엔드포인트 또는 기존 엔드포인트의 `ChannelType` 속성을 업데이트할 수 있습니다.

- 메시지를 전송하기 전에 올바른 채널 선택 - SMS 메시지를 전송하려고 하는데 대상 번호가 유효하지 않다고 판단된 경우, 다른 채널을 통해 수신자에게 메시지를 전송할 수 있습니다. 예를 들어, 엔드포인트가 SMS 메시지를 수신할 수 없는 경우 대신 음성 메시지를 전송할 수 있습니다.

를 사용하여 전화번호 확인 AWS CLI

다음 예제에서는 AWS CLI를 사용하여 전화번호를 검증하는 방법을 보여줍니다. 자세한 내용은 AWS CLI 명령 참조의 phone-number-validate를 참조하세요. 예제 검증 응답은 전화 번호 검증 응답 섹션을 참조하세요. 구성에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 구성을 AWS CLI AWS CLI참조하세요.

를 사용하여 전화번호 확인 서비스를 사용하려면 AWS CLI

- 명령줄에 다음 명령을 입력합니다.

```
aws pinpoint phone-number-validate --number-validate-request
 PhoneNumber=+442079460881,IsoCountryCode=GB
```

앞의 명령에서 *+442079460881*을 검증하려는 전화번호로 바꾸고 *GB*를 두 자리 ISO 국가 또는 리전 코드로 바꿉니다.

> **Note**
>
> 전화 번호 확인 서비스에 전화 번호를 제공할 때는 항상 국가 코드를 포함해야 합니다. 국가 코드를 포함하지 않으면 서비스에서 다른 국가의 전화 번호에 대한 정보를 반환할 수 있습니다. 전화번호에 대시가 있을 수 있습니다. 예: *+44-207-946-0881*.

전화 번호 검증 응답

전화 번호 확인 서비스가 제공하는 정보는 제공한 전화 번호에 사용할 수 있는 데이터에 따라 약간 다릅니다. 이 단원에는 전화 번호 확인 서비스가 반환하는 응답의 예가 포함되어 있습니다.

> **Note**
>
> 전화 번호 확인 서비스에서 제공하는 데이터는 전 세계 통신 사업자 및 기타 기관이 제공한 정보를 기반으로 합니다. 일부 국가의 공급업체는 다른 국가의 공급업체보다 정보 업데이트 간격이 길 수 있습니다. 예를 들어, 휴대폰 번호의 확인을 요청했는데 제공한 번호가 한 이동 통신사

에서 다른 이동 통신사로 이식된 경우 전화 번호 확인 서비스의 응답에 현재 이동 통신사가 아닌 원래 이동 통신사의 이름이 포함될 수 있습니다.

유효한 휴대폰 번호

전화 번호 확인 서비스에 요청을 보냈는데 해당 전화 번호가 유효한 휴대폰 번호이면 다음 예제와 유사한 정보가 반환됩니다.

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp Mobile",
        "City": "Seattle",
        "CleansedPhoneNumberE164": "+12065550142",
        "CleansedPhoneNumberNational": "2065550142",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+12065550142",
        "PhoneType": "MOBILE",
        "PhoneTypeCode": 0,
        "Timezone": "America/Los_Angeles",
        "ZipCode": "98101"
    }
}
```

유효한 유선 전화 번호

요청에 유효한 유선 전화 번호가 포함되어 있으면 전화 번호 확인 서비스에서 다음 예제와 유사한 정보를 반환합니다.

```
{
    "CountryCodeIso2": "US",
    "CountryCodeNumeric": "1",
    "Country": "United States",
    "City": "Santa Clara",
    "ZipCode": "95037",
    "Timezone": "America/Los_Angeles",
    "CleansedPhoneNumberNational": "4085550101",
    "CleansedPhoneNumberE164": "14085550101",
    "Carrier": "AnyCompany",
    "PhoneTypeCode": 1,
```

```
    "PhoneType": "LANDLINE",
    "OriginalPhoneNumber": "+14085550101"
}
```

유효한 VoIP 전화 번호

요청에 유효한 Voice over Internet Protocol(VoIP) 전화 번호가 포함되어 있으면 전화 번호 확인 서비스에서 다음 예제와 유사한 정보를 반환합니다.

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+441514960001",
        "CleansedPhoneNumberNational": "1514960001",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+441514960001",
        "PhoneType": "VOIP",
        "PhoneTypeCode": 2
    }
}
```

잘못된 전화 번호

요청에 잘못된 전화 번호가 포함되어 있으면 전화 번호 확인 서비스에서 다음 예제와 유사한 정보를 반환합니다.

```
{
    "NumberValidateResponse": {
        "CleansedPhoneNumberE164": "+44163296076",
        "CleansedPhoneNumberNational": "163296076",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+440163296076",
        "PhoneType": "INVALID",
        "PhoneTypeCode": 3
    }
}
```

이 응답의 PhoneType 속성은 이 전화 번호가 INVALID이며 해당 전화 번호와 연결된 이동 통신사 또는 위치에 대한 정보를 포함하지 않음을 나타냅니다. PhoneType이 INVALID인 전화 번호는 실제 수신자에게 속하지 않는 것이므로 이 번호로 SMS 또는 음성 메시지를 보내지 마세요.

기타 전화 번호

전화 번호 확인 서비스의 응답에 PhoneType 값 OTHER가 포함되는 경우가 있습니다. 이 서비스는 다음과 같은 상황에서 이러한 종류의 응답을 반환할 수 있습니다.

- 전화 번호가 수신자 부담 전화 번호인 경우
- 전화 번호가 555로 시작하는 북미 전화 번호와 같이 TV 프로그램 및 영화에 사용하도록 예약된 번호인 경우
- 전화 번호에 북미 지역의 999 지역 번호와 같이 현재 사용되지 않는 지역 번호가 포함된 경우
- 전화 번호가 다른 용도로 예약되어 있는 경우

다음 예는 요청에 북미 지역의 가상 전화 번호가 포함되어 있을 때 전화 번호 확인 서비스에서 제공하는 응답을 보여줍니다.

```
{
    "NumberValidateResponse": {
        "Carrier": "Multiple OCN Listing",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "OTHER",
        "PhoneTypeCode": 4,
        "Timezone": "America/Los_Angeles"
    }
}
```

선불 전화번호

요청에 유효한 선불 전화번호가 포함되어 있으면 전화번호 확인 서비스에서 다음 예시와 유사한 정보를 반환합니다.

```
{
    "NumberValidateResponse": {
```

```
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "PREPAID",
        "PhoneTypeCode": 5
    }
}
```

이러한 응답에 포함된 정보에 대한 자세한 내용은 Amazon Pinpoint API 참조의 전화번호 확인을 참조하세요.

웹후크 또는 Lambda 함수를 사용하여 Amazon Pinpoint에서 사용자 지정 채널 생성

Amazon Pinpoint에는 푸시 알림, 이메일, SMS 및 음성 채널을 통해 메시지를 보내는 기능이 내장되어 있습니다. 사용자 지정 채널을 만들어 다른 채널을 통해 메시지를 보내도록 Amazon Pinpoint를 구성할 수도 있습니다. Amazon Pinpoint에서 사용자 지정 채널을 사용하면 타사 서비스를 비롯해 API가 있는 모든 서비스를 통해 메시지를 전송할 수 있습니다. 웹후크를 사용하거나 AWS Lambda 함수를 호출하여 APIs와 상호 작용할 수 있습니다.

사용자 지정 채널 캠페인을 보내는 세그먼트에는 모든 유형의 엔드포인트(즉, `ChannelType` 속성 값이 EMAIL, VOICE, SMS, CUSTOM 또는 다양한 푸시 알림 엔드포인트 유형 중 하나인 엔드포인트)가 포함될 수 있습니다.

웹후크 사용

웹후크를 사용하여 사용자 지정 채널 메시지를 전송하는 경우 웹후크의 URL은 “https://”로 시작해야 합니다. 웹후크 URL에는 영숫자 외에 하이픈(-), 마침표(.), 밑줄(_), 물결표(~), 물음표(?), 슬래시 또는 솔리더스(/), 파운드 또는 해시 기호(#) 및 세미콜론(:)만 포함될 수 있습니다. URL은 RFC3986을 준수해야 합니다.

웹후크 URL을 지정하는 캠페인을 만들면 Amazon Pinpoint가 해당 URL에 HTTP `HEAD`를 발행합니다. `HEAD` 요청에 대한 응답에는 `X-Amz-Pinpoint-AccountId`라는 헤더가 포함되어야 합니다. 이 헤더의 값은 AWS 계정 ID와 같아야 합니다.

Lambda 함수 사용

대신 Lambda 함수를 생성하여 사용자 지정 채널 메시지를 전송하기로 선택하는 경우 먼저 Amazon Pinpoint가 내보내는 데이터를 숙지하는 것이 가장 좋습니다. Amazon Pinpoint 캠페인은 사용자 지정 채널을 통해 메시지를 보낼 때 다음 예제와 비슷한 대상 Lambda 함수에 페이로드를 보냅니다.

```
{
  "Message":{},
  "Data":"The payload that's provided in the CustomMessage object in
 MessageConfiguration",
  "ApplicationId":"3a9b1f4e6c764ba7b031e7183example",
  "CampaignId":"13978104ce5d6017c72552257example",
```

```
  "TreatmentId":"0",
  "ActivityId":"575cb1929d5ba43e87e2478eeexample",
  "ScheduledTime":"2020-04-08T19:00:16.843Z",
  "Endpoints":{
    "1dbcd396df28ac6cf8c1c2b7fexample":{
      "ChannelType":"EMAIL",
      "Address":"mary.major@example.com",
      "EndpointStatus":"ACTIVE",
      "OptOut":"NONE",
      "Location":{
        "City":"Seattle",
        "Country":"USA"
      },
      "Demographic":{
        "Make":"OnePlus",
        "Platform":"android"
      },
      "EffectiveDate":"2020-04-01T01:05:17.267Z",
      "Attributes":{
        "CohortId":[
          "42"
        ]
      },
      "CreationDate":"2020-04-01T01:05:17.267Z"
    }
  }
}
```

이벤트 데이터는 다음 속성을 제공합니다.

- `ApplicationId` - 캠페인이 속한 Amazon Pinpoint 프로젝트의 ID입니다.
- `CampaignId` - Lambda 함수를 간접적으로 호출한 Amazon Pinpoint 캠페인의 ID입니다.
- `TreatmentId` - 캠페인 변형의 ID입니다. 표준 캠페인을 만든 경우 이 값은 항상 0입니다. A/B 테스트 캠페인을 만든 경우 이 값은 0~4 사이의 정수입니다.
- `ActivityId` - 캠페인에 의해 수행 중인 활동의 ID입니다.
- `ScheduledTime` - Amazon Pinpoint가 캠페인을 실행한 시간(ISO 8601 형식)입니다.
- `Endpoints` - 캠페인이 대상으로 지정한 엔드포인트의 목록입니다. 각 페이로드는 최대 50개의 엔드포인트를 포함할 수 있습니다. 캠페인을 보낸 세그먼트에 포함된 엔드포인트가 50개를 초과하는 경우 Amazon Pinpoint는 모든 엔드포인트가 처리될 때까지 한 번에 최대 50개의 엔드포인트를 사용하여 함수를 반복적으로 간접 호출합니다.

사용자 지정 채널 Lambda 함수를 생성하고 테스트할 경우 이 샘플 데이터를 사용할 수 있습니다.

Amazon Pinpoint API를 사용하여 개별 캠페인에 Lambda 함수 또는 웹후크 할당

개별 캠페인에 Lambda 함수 또는 웹후크를 할당하려면 Amazon Pinpoint API를 사용하여 캠페인 객체를 생성하거나 업데이트합니다.

캠페인의 `MessageConfiguration` 객체에는 `CustomMessage` 객체도 포함되어야 합니다. 이 객체에는 하나의 멤버 `Data`가 있습니다. `Data`의 값은 사용자 지정 채널로 보낼 메시지 페이로드가 포함된 JSON 문자열입니다.

캠페인은 `CustomDeliveryConfiguration` 객체를 포함해야 합니다. `CustomDeliveryConfiguration` 객체 내에서 다음을 지정합니다.

- `EndpointTypes` - 사용자 지정 채널 캠페인을 보내야 하는 모든 엔드포인트 유형을 포함하는 배열입니다. 다음 채널 유형 중 일부 또는 전부 포함할 수 있습니다.
  - `ADM`
  - `APNS`
  - `APNS_SANDBOX`
  - `APNS_VOIP`
  - `APNS_VOIP_SANDBOX`
  - `BAIDU`
  - `CUSTOM`
  - `EMAIL`
  - `GCM`
  - `SMS`
  - `VOICE`
- `DeliveryUri` - 엔드포인트를 보낼 대상입니다. 다음 중 하나를 지정할 수 있습니다.
  - 캠페인이 실행될 때 엔드포인트 데이터를 보내려는 웹후크의 URL.
  - 캠페인이 실행될 때 실행하려는 Lambda 함수의 Amazon 리소스 이름(ARN).

> **Note**
>
> Campaign 객체에는 Hook 객체도 포함될 수도 있습니다. 이 객체는 캠페인이 실행될 때 Lambda 함수에 의해 사용자 지정되는 세그먼트를 만드는 데에만 사용됩니다. 자세한 내용은 AWS Lambda 함수를 사용하여 Amazon Pinpoint 세그먼트 사용자 지정 단원을 참조하십시오.

Amazon Pinpoint 캠페인에 대한 Lambda 함수 생성 및 구성

이 섹션에서는 사용자 지정 채널을 통해 메시지를 보내는 Lambda 함수를 만드는 단계를 개략적으로 설명합니다. 먼저 함수를 만듭니다. 그런 다음 함수에 실행 정책을 추가합니다. 이 정책은 캠페인이 실행될 때 Amazon Pinpoint가 정책을 실행하도록 허용합니다.

Lambda 함수 생성에 대한 소개는 AWS Lambda 개발자 안내서의 Lambda 함수 빌드를 참조하세요.

Lambda 함수 예제

다음 코드 예제에서는 페이로드를 처리하고 CloudWatch에 각 엔드포인트 유형의 엔드포인트 수를 기록합니다.

```
import boto3
import random
import pprint
import json
import time

cloudwatch = boto3.client('cloudwatch')

def lambda_handler(event, context):
    customEndpoints = 0
    smsEndpoints = 0
    pushEndpoints = 0
    emailEndpoints = 0
    voiceEndpoints = 0
    numEndpoints = len(event['Endpoints'])

    print("Payload:\n", event)
    print("Endpoints in payload: " + str(numEndpoints))

    for key in event['Endpoints'].keys():
        if event['Endpoints'][key]['ChannelType'] == "CUSTOM":
```

```
            customEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "SMS":
            smsEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "EMAIL":
            emailEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "VOICE":
            voiceEndpoints += 1
        else:
            pushEndpoints += 1

    response = cloudwatch.put_metric_data(
        MetricData = [
            {
                'MetricName': 'EndpointCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': len(event['Endpoints'])
            },
            {
                'MetricName': 'CustomCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': customEndpoints
            },
            {
                'MetricName': 'SMSCount',
```

```
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': smsEndpoints
            },
            {
                'MetricName': 'EmailCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': emailEndpoints
            },
            {
                'MetricName': 'VoiceCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': voiceEndpoints
            },
            {
```

```
                'MetricName': 'PushCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': pushEndpoints
            },
            {
                'MetricName': 'EndpointCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': len(event['Endpoints'])
            },
            {
                'MetricName': 'CustomCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': customEndpoints
            },
            {
                'MetricName': 'SMSCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': smsEndpoints
            },
            {
                'MetricName': 'EmailCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': emailEndpoints
            },
            {
                'MetricName': 'VoiceCount',
```

```
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': voiceEndpoints
            },
            {
                'MetricName': 'PushCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': pushEndpoints
            }
        ],
        Namespace = 'PinpointCustomChannelExecution'
    )
    print("cloudwatchResponse:\n",response)
```

Amazon Pinpoint 캠페인이 이 Lambda 함수를 실행하면 Amazon Pinpoint는 세그먼트 멤버의 목록을 함수에 보냅니다. 이 함수는 각 `ChannelType`의 엔드포인트 수를 계산합니다. 그런 다음, 해당 데이터를 Amazon CloudWatch로 보냅니다. CloudWatch 콘솔의 지표 섹션에서 이러한 지표를 볼 수 있습니다. 지표는 PinpointCustomChannelExecution 네임스페이스에서 사용할 수 있습니다.

이 코드 예제를 수정하여 외부 서비스의 API에도 연결하여 해당 서비스를 통해 메시지를 보낼 수 있습니다.

Amazon Pinpoint용 Lambda 함수 응답 형식

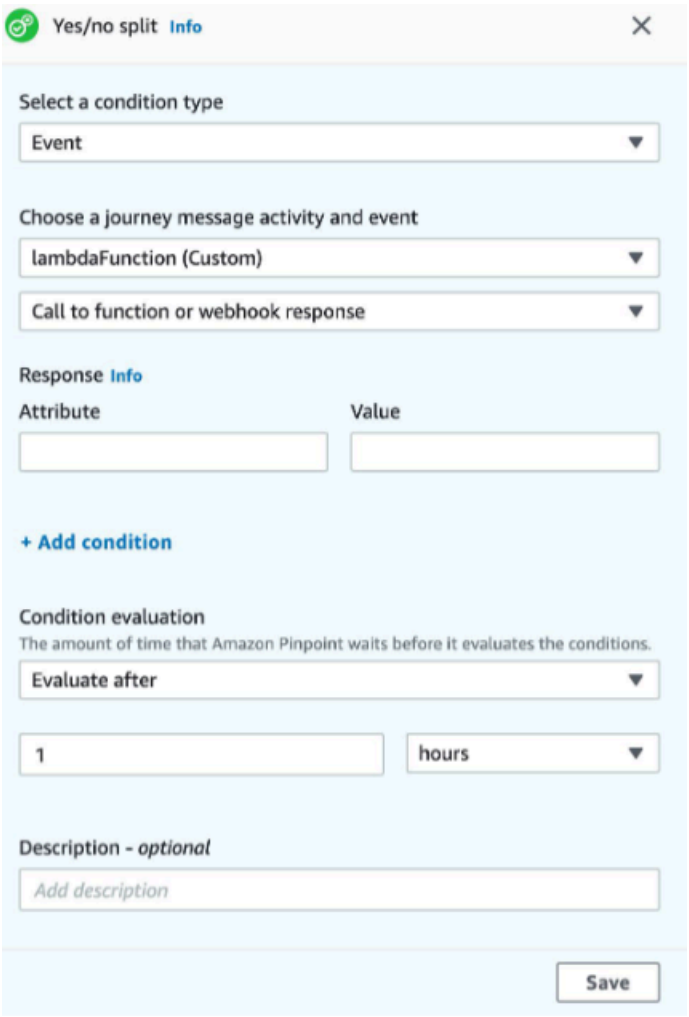
사용자 지정 채널 활동 후 여정 다변량 또는 예/아니요 분할을 사용하여 엔드포인트 경로를 결정하려면, Amazon Pinpoint가 이해할 수 있는 형식으로 Lambda 함수 응답을 구성한 다음 엔드포인트를 올바른 경로로 보내야 합니다.

응답 구조는 다음과 같은 형식이어야 합니다.

```
{
    <Endpoint ID 1>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
    <Endpoint ID 2>:{
```

```
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
...
}
```

이렇게 하면 엔드포인트 경로를 결정할 키와 값을 선택할 수 있습니다.



Amazon Pinpoint에 Lambda 함수를 간접적으로 호출할 수 있는 권한 부여

AWS Command Line Interface (AWS CLI)를 사용하여 Lambda 함수에 할당된 Lambda 함수 정책에 권한을 추가할 수 있습니다. Amazon Pinpoint가 함수를 간접적으로 호출하도록 허용하려면 다음 예제와 같이 Lambda add-permission 명령을 사용합니다.

```
aws lambda add-permission \
--function-name myFunction \
--statement-id sid0 \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:111122223333:apps/* 
--source-account 111122223333
```

위의 명령에서 다음을 수행합니다.

- *myFunction*을 Lambda 함수의 이름으로 바꿉니다.
- *us-east-1*을 Amazon Pinpoint를 사용하는 AWS 리전으로 바꿉니다.
- *111122223333*을 AWS 계정 ID로 바꿉니다.

add-permission 명령을 실행하면 Lambda에서 다음 출력이 반환됩니다.

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:myFunction\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/*\"}},
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
}
```

Statement 값은 Lambda 함수 정책에 추가된 구문의 JSON 문자열 버전입니다.

실행 정책 추가 제한

실행 정책을 특정 Amazon Pinpoint 프로젝트로 제한하여 수정할 수 있습니다. 이렇게 하려면 앞의 예제에서 *를 프로젝트의 고유 ID로 바꿉니다. 정책을 특정 캠페인으로 제한하여 추가로 제한할 수 있습니다. 예를 들어 프로젝트 ID가 dbaf6ec2226f0a9a8615e3ea5example인 프로젝트에서 캠페인 ID가 95fee4cd1d7f5cd67987c1436example인 캠페인만 허용하도록 정책을 제한하려면 source-arn 속성에 다음 값을 사용합니다.

```
arn:aws:mobiletargeting:us-east-1:111122223333:apps/dbaf6ec2226f0a9a8615e3ea5example/
campaigns/95fee4cd1d7f5cd67987c1436example
```

> **Note**
>
> Lambda 함수의 실행을 특정 캠페인으로 제한하는 경우 먼저 덜 제한적인 정책을 사용하여 함수를 만들어야 합니다. 그런 다음 Amazon Pinpoint에서 캠페인을 만들고 함수를 선택해야 합니다. 마지막으로, 지정한 캠페인을 참조하도록 실행 정책을 업데이트해야 합니다.

Amazon Pinpoint를 사용하여 Kinesis 및 Firehose를 통해 앱 이벤트 데이터 스트리밍

Amazon Pinpoint에서 이벤트는 사용자가 애플리케이션 중 하나와 상호 작용하거나, 캠페인이나 여정에서 메시지를 보내거나, 트랜잭션 SMS 또는 이메일 메시지를 보낼 때 발생하는 작업입니다. 예를 들어 이메일 메시지를 보내면 다음과 같은 여러 이벤트가 발생합니다.

- 메시지를 보내면 전송 이벤트가 발생합니다.
- 메시지가 수신자의 받은 편지함에 도달하면 배달됨 이벤트가 발생합니다.
- 수신자가 메시지를 열면 열림 이벤트가 발생합니다.

이벤트에 대한 정보를 Amazon Kinesis로 보내도록 Amazon Pinpoint를 구성할 수 있습니다. Kinesis 플랫폼은 AWS 서비스에서 실시간으로 데이터를 수집, 처리, 분석하는 데 사용할 수 있는 서비스를 제공합니다. Amazon Pinpoint는 이벤트 데이터를 Firehose로 전송할 수 있습니다. Firehose는 이러한 데이터를 AWS 데이터 스토어(예: Amazon S3 또는 Amazon Redshift)로 스트리밍합니다. 또한 Amazon Pinpoint는 Kinesis Data Streams로 데이터를 스트리밍할 수 있습니다. Kinesis Data Streams는 분석 애플리케이션에서 처리할 수 있도록 여러 데이터 스트림을 수집하고 저장합니다.

Amazon Pinpoint 이벤트 스트림에는 Amazon Pinpoint에 연결된 애플리케이션(앱)과의 사용자 상호 작용에 대한 정보가 포함됩니다. 또한 캠페인, 채널 및 여정에서 보내는 모든 메시지에 대한 정보도 포함되어 있습니다. 여기에는 정의한 모든 사용자 지정 이벤트도 포함할 수 있습니다. 마지막으로, 전송하는 모든 트랜잭션 이메일 및 SMS 메시지에 대한 정보가 포함됩니다.

> **Note**
>
> Amazon Pinpoint는 트랜잭션 푸시 알림 또는 음성 메시지에 대한 정보를 스트리밍하지 않습니다.

이 챕터에는 이벤트 데이터를 Kinesis로 스트리밍하도록 Amazon Pinpoint를 설정하는 방법에 대한 정보가 포함되어 있습니다. 또한 Amazon Pinpoint에서 스트리밍하는 이벤트 데이터의 예도 포함됩니다.

주제

- Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정
- Amazon Pinpoint의 앱 이벤트 데이터 스트림

- Amazon Pinpoint의 캠페인 이벤트 데이터 스트림
- Amazon Pinpoint의 여정 이벤트 데이터
- Amazon Pinpoint에서 이메일 이벤트 데이터 스트리밍
- Amazon Pinpoint의 SMS 이벤트 데이터 스트림
- Amazon Pinpoint에서 이벤트 스트림 삭제

Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정

Amazon Kinesis 스트림 또는Amazon Data Firehose 전송 스트림으로 이벤트를 전송하도록 Amazon Pinpoint를 설정할 수 있습니다. Amazon Pinpoint는 캠페인, 여정, 트랜잭션 이메일 및 SMS 메시지에 대한 이벤트 데이터를 전송할 수 있습니다.

이 단원에는 프로그래밍 방식의 이벤트 스트리밍 설정에 대한 정보가 포함되어 있습니다. Amazon Pinpoint 콘솔을 사용하여 이벤트 스트리밍을 설정할 수도 있습니다. Amazon Pinpoint 콘솔을 사용하여 이벤트 스트리밍을 설정하는 방법에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 이벤트 스트림 설정 섹션을 참조하세요.

사전 조건

이 단원의 예제에는 다음 입력이 필요합니다.

- Amazon Pinpoint 및 보고 이벤트와 통합된 애플리케이션의 애플리케이션 ID입니다. 통합 방법에 대한 자세한 내용은 Amazon Pinpoint를 애플리케이션과 통합 단원을 참조하십시오.
- AWS 계정에 있는 Kinesis 스트림 또는 Firehose 전송 스트림의 Amazon 리소스 이름(ARN)입니다. 이러한 리소스를 생성하는 방법에 대한 자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 생성 및 관리 또는 Amazon Data Firehose 개발자 안내서의 Amazon Data Firehose 전송 스트림 생성 섹션을 참조하세요.
- Amazon Pinpoint가 스트림으로 데이터를 전송하도록 승인하는 AWS Identity and Access Management (IAM) 역할의 ARN입니다. 역할 생성에 대한 자세한 내용은 이벤트를 Kinesis로 스트리밍하기 위한 IAM 역할 단원을 참조하십시오.

AWS CLI

다음 AWS CLI 예제에서는 put-event-stream 명령을 사용합니다. 이 명령은 Kinesis 스트림에 이벤트를 전송하도록 Amazon Pinpoint를 구성합니다.

```
aws pinpoint put-event-stream \
--application-id projectId \
--write-event-stream DestinationStreamArn=streamArn,RoleArn=roleArn
```

AWS SDK for Java

다음 Java 예제는 이벤트를 Kinesis 스트림으로 전송하도록 Amazon Pinpoint를 구성합니다.

```
public PutEventStreamResult createEventStream(AmazonPinpoint pinClient,
        String appId, String streamArn, String roleArn) {

    WriteEventStream stream = new WriteEventStream()
            .withDestinationStreamArn(streamArn)
            .withRoleArn(roleArn);

    PutEventStreamRequest request = new PutEventStreamRequest()
            .withApplicationId(appId)
            .withWriteEventStream(stream);

    return pinClient.putEventStream(request);
}
```

이 예제에서는 Kinesis 스트림 및 IAM 역할의 ARN을 저장하는 `WriteEventStream` 객체를 구성합니다. `WriteEventStream` 객체는 `PutEventStreamRequest` 객체에 전달되어 특정 애플리케이션의 이벤트를 스트리밍하도록 Amazon Pinpoint를 구성합니다. `PutEventStreamRequest` 객체는 Amazon Pinpoint 클라이언트의 `putEventStream` 메서드에 전달됩니다.

한 Kinesis 스트림을 여러 애플리케이션에 할당할 수 있습니다. 이 경우 Amazon Pinpoint가 각 애플리케이션에서 base64로 인코딩된 이벤트 데이터를 스트림으로 전송하므로 데이터를 하나의 모음으로 분석할 수 있습니다. 다음 예제 메서드는 애플리케이션(앱) ID의 목록을 허용하며, 이전 예제 메서드 `createEventStream`을 사용하여 각 애플리케이션에 스트림을 할당합니다.

```
public List<PutEventStreamResult> createEventStreamFromAppList(
        AmazonPinpoint pinClient, List<String> appIDs,
        String streamArn, String roleArn) {

    return appIDs.stream()
            .map(appId -> createEventStream(pinClient, appId, streamArn,
                    roleArn))
            .collect(Collectors.toList());
```

```
}
```

한 스트림을 여러 애플리케이션에 할당할 수 있지만 여러 스트림을 한 애플리케이션에 할당할 수는 없습니다.

Amazon Pinpoint의 앱 이벤트 데이터 스트림

애플리케이션(앱)을 Amazon Pinpoint와 통합하고 이벤트 스트리밍을 설정하면 Amazon Pinpoint는 설정 중에 지정한 대상에서 앱의 사용자 활동, 사용자 지정 이벤트 및 메시지 전송 데이터를 검색하여 볼 수 있도록 합니다. 이벤트 데이터를 볼 수 있도록 이벤트 스트리밍을 설정하는 방법에 대한 자세한 내용은 Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정 섹션을 참조하세요.

앱 이벤트 예

앱 이벤트에 대한 JSON 객체에는 다음 예제의 데이터가 포함되어 있습니다.

```
{
  "event_type": "_session.stop",
  "event_timestamp": 1487973802507,
  "arrival_timestamp": 1487973803515,
  "event_version": "3.0",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "cognito_identity_pool_id": "us-east-1:a1b2c3d4-e5f6-g7h8-i9j0-k1l2m3n4o5p6",
    "package_name": "main.page",
    "sdk": {
      "name": "aws-sdk-mobile-analytics-js",
      "version": "0.9.1:2.4.8"
    },
    "title": "title",
    "version_name": "1.0",
    "version_code": "1"
  },
  "client": {
    "client_id": "m3n4o5p6-a1b2-c3d4-e5f6-g7h8i9j0k1l2",
    "cognito_id": "us-east-1:i9j0k1l2-m3n4-o5p6-a1b2-c3d4e5f6g7h8"
  },
  "device": {
    "locale": {
      "code": "en_US",
```

```
      "country": "US",
      "language": "en"
    },
    "make": "generic web browser",
    "model": "Unknown",
    "platform": {
      "name": "android",
      "version": "10.10"
    }
  },
  "session": {
    "session_id": "f549dea9-1090-945d-c3d1-e4967example",
    "start_timestamp": 1487973202531,
    "stop_timestamp": 1487973802507
  },
  "attributes": {},
  "metrics": {}
}
```

앱 이벤트 속성

이 섹션에서는 앱 이벤트 스트림의 이전 예제에 포함된 속성을 정의합니다.

속성	설명
event_type	이벤트의 유형입니다. 가능한 값은 다음과 같습니다. • _session.start – 엔드포인트가 새 세션을 시작했습니다. • _session.stop – 엔드포인트가 세션을 종료했습니다. • _userauth.sign_in – 엔드포인트가 앱에 로그인했습니다. • _userauth.sign_up – 새로운 엔드포인트가 앱에서 등록 프로세스를 완료했습니다. • _userauth.auth_fail – 엔드포인트가 앱에 로그인하려고 했지만 프로세스를 완료할 수 없습니다.

속성	설명
	• _monetization.purchase – 엔드포인트가 앱에서 구매했습니다. • _session.pause – 엔드포인트가 세션을 일시 중지했습니다. 일시 중지된 세션을 다시 시작하면 새 세션을 처음부터 시작하지 않고도 지표를 계속 수집할 수 있습니다. • _session.resume – 엔드포인트가 세션을 재개했습니다.
`event_timestamp`	이벤트가 보고된 시간입니다(밀리초 단위 Unix 시간).
`arrival_timestamp`	Amazon Pinpoint에서 이벤트가 수신된 시간입니다(밀리초 단위 Unix 시간).
`event_version`	이벤트 JSON 스키마의 버전입니다. **Tip** 스키마 업데이트에 대응하여 언제 애플리케이션을 업데이트해야 하는지 알 수 있도록 이벤트 처리 애플리케이션에서 이 버전을 확인합니다.
`application`	이벤트와 관련된 Amazon Pinpoint 프로젝트에 대한 정보입니다. 자세한 내용은 애플리케이션 표를 참조하십시오.
`client`	이벤트를 보고한 엔드포인트에 대한 정보입니다. 자세한 내용은 클라이언트 표를 참조하십시오.
`device`	이벤트를 보고한 디바이스에 대한 정보입니다. 자세한 내용은 디바이스 표를 참조하십시오.

속성	설명
session	이벤트를 생성한 세션에 대한 정보입니다. 자세한 내용은 세션 표를 참조하십시오.
attributes	이벤트와 연결된 속성입니다. 앱에서 보고한 이벤트의 경우 이 객체에는 사용자가 정의한 사용자 지정 속성이 포함됩니다.
metrics	이벤트와 관련된 지표입니다. 선택에 따라 Amazon Pinpoint로 사용자 지정 지표를 보내도록 앱을 구성할 수 있습니다.

Application

이벤트가 연결된 Amazon Pinpoint 프로젝트에 대한 정보를 포함합니다.

속성	설명
app_id	이벤트를 보고한 Amazon Pinpoint 프로젝트의 고유 ID입니다.
cognito_identity_pool_id	엔드포인트가 연결된 Amazon Cognito 자격 증명 풀의 ID입니다.
package_name	앱 패키지의 이름(예: com.example.my_app)입니다.
sdk	이벤트를 보고하는 데 사용된 SDK에 대한 정보입니다. 자세한 내용은 SDK 표를 참조하십시오.
title	앱의 이름입니다.
version_name	앱의 버전 이름입니다(예: V2.5).
version_code	앱의 버전 번호입니다(예: 3).

SDK

이벤트를 보고하는 데 사용된 SDK에 대한 정보를 포함합니다.

속성	설명
`name`	이벤트를 보고하는 데 사용된 SDK의 이름입니다.
`version`	SDK의 버전입니다.

클라이언트

이벤트를 생성한 엔드포인트에 대한 정보를 포함합니다.

속성	설명
`client_id`	엔드포인트의 ID입니다.
`cognito_id`	엔드포인트와 연결된 Amazon Cognito ID 토큰입니다.

디바이스

이벤트를 생성한 엔드포인트의 디바이스에 대한 정보를 포함합니다.

속성	설명
`locale`	엔드포인트 디바이스의 언어 및 리전 설정에 대한 정보입니다. 자세한 내용은 로캘 표를 참조하십시오.
`make`	엔드포인트 디바이스 제조업체입니다.
`model`	엔드포인트 디바이스의 모델 식별자입니다.

속성	설명
platform	엔드포인트 디바이스의 운영 체제에 대한 정보입니다. 자세한 내용은 플랫폼 표를 참조하십시오.

Locale

엔드포인트 디바이스의 언어 및 리전 설정에 대한 정보를 포함합니다.

속성	설명
code	디바이스와 연결된 로캘 식별자입니다.
country	디바이스의 로캘과 연결된 국가 또는 리전입니다.
language	디바이스의 로캘과 연결된 언어입니다.

플랫폼

엔드포인트 디바이스의 운영 체제에 대한 정보를 포함합니다.

속성	설명
name	디바이스의 운영 체제 이름입니다.
version	디바이스의 운영 체제 버전입니다.

세션

이벤트를 생성한 세션에 대한 정보를 포함합니다.

속성	설명
session_id	세션을 식별하는 고유 ID입니다.

속성	설명
start_timestamp	세션이 시작된 날짜 및 시간입니다(밀리초 단위 Unix 시간).
stop_timestamp	세션이 종료된 날짜 및 시간입니다(밀리초 단위 Unix 시간).

Amazon Pinpoint의 캠페인 이벤트 데이터 스트림

Amazon Pinpoint를 사용하여 채널을 통해 캠페인을 전송하는 경우, Amazon Pinpoint에서 해당 캠페인에 대한 이벤트 데이터를 스트리밍할 수 있습니다. 이벤트 스트리밍을 설정하면 Amazon Pinpoint는 설정 중에 지정한 대상에서 캠페인에서 보내는 이메일 또는 SMS 메시지에 대한 앱의 이벤트 데이터를 검색하여 볼 수 있도록 합니다. 이메일 및 SMS 메시지에 대해 Amazon Pinpoint에서 스트리밍하는 데이터에 대한 자세한 내용은 the section called “Amazon Pinpoint에서 이메일 이벤트 데이터 스트리밍” 및 the section called “Amazon Pinpoint의 SMS 이벤트 데이터 스트림” 섹션을 참조하세요. 이벤트 스트리밍 설정 방법에 대한 자세한 내용은 Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정 섹션을 참조하세요.

캠페인 이벤트 예제

캠페인 이벤트에 대한 JSON 객체는 다음 예제에 나와 있는 데이터를 포함합니다.

```
{
  "event_type": "_campaign.send",
  "event_timestamp": 1562109497426,
  "arrival_timestamp": 1562109497494,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "d8dcf7c5-e81a-48ae-8313-f540cexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
```

```
  "attributes": {
    "treatment_id": "0",
    "campaign_activity_id": "5473285727f04865bc673e527example",
    "delivery_type": "GCM",
    "campaign_id": "4f8d6097c2e8400fa3081d875example",
    "campaign_send_status": "SUCCESS"
  },
  "client_context": {
    "custom": {
      "endpoint": "{\"ChannelType\":\"GCM\",\"EndpointStatus\":\"ACTIVE\",
          #\"OptOut\":\"NONE\",\"RequestId\":\"ec229696-9d1e-11e9-8bf1-85d0aexample\",
          #\"EffectiveDate\":\"2019-07-02T23:12:54.836Z\",\"User\":{}}"
    }
  },
  "awsAccountId": "123456789012"
}
```

캠페인 이벤트 속성

이 단원에서는 캠페인 이벤트 스트림에 포함된 속성을 정의합니다.

속성	설명
event_type	이벤트의 유형입니다. 가능한 값은 다음과 같습니다. • _campaign.send – Amazon Pinpoint가 캠페인을 실행했습니다. • _campaign.opened_notification – 푸시 알림 캠페인의 경우 이 이벤트 유형은 수신자가 알림을 탭하여 열었음을 나타냅니다. • _campaign.received_foreground – 푸시 알림 캠페인의 경우 이 이벤트 유형은 수신자가 포그라운드 알림으로 메시지를 수신했음을 나타냅니다. • _campaign.received_background – 푸시 알림 캠페인의 경우 이 이벤트 유형은 수신자가 백그라운드 알림으로 메시지를 수신했음을 나타냅니다.

속성	설명
	Note _campaign.opened_notification, _campaign.received_foreground, _campaign.received_background는 AWS Amplify를 사용하는 경우에만 반환됩니다. 앱을와 통합하는 방법에 대한 자세한 내용은 단원을 참조하십시오 AWS Amplify. AWS Amplify를 사용하여 프런트엔드 애플리케이션을 Amazon Pinpoint에 연결을(를) 참조하세요.
`event_timestamp`	이벤트가 보고된 시간입니다(밀리초 단위 Unix 시간).
`arrival_timestamp`	Amazon Pinpoint에서 이벤트가 수신된 시간입니다(밀리초 단위 Unix 시간).
`event_version`	이벤트 JSON 스키마의 버전입니다. **Tip** 스키마 업데이트에 대응하여 언제 애플리케이션을 업데이트해야 하는지 알 수 있도록 이벤트 처리 애플리케이션에서 이 버전을 확인합니다.
`application`	이벤트와 관련된 Amazon Pinpoint 프로젝트에 대한 정보입니다. 자세한 내용은 애플리케이션 표를 참조하십시오.

속성	설명
client	이벤트가 연결된 엔드포인트에 대한 정보입니다. 자세한 내용은 클라이언트 표를 참조하십시오.
device	이벤트를 보고한 디바이스에 대한 정보입니다. 캠페인 및 트랜잭션 메시지의 경우 이 객체가 비어 있습니다.
session	이벤트를 생성한 세션에 대한 정보입니다. 캠페인의 경우 이 객체가 비어 있습니다.
attributes	이벤트와 연결된 속성입니다. 앱 중 하나에서 보고한 이벤트의 경우 이 객체에는 앱에서 정의한 사용자 지정 속성이 포함될 수 있습니다. 캠페인을 전송할 때 생성되는 이벤트의 경우 이 객체에는 캠페인과 연결된 속성이 포함됩니다. 트랜잭션 메시지를 전송할 때 생성되는 이벤트의 경우 이 객체에는 메시지 자체와 관련된 정보가 포함됩니다. 자세한 내용은 속성 표를 참조하십시오.
client_context	endpoint 속성을 포함하는 custom 객체를 포함합니다. endpoint 속성에는 캠페인이 전송된 엔드포인트에 대한 엔드포인트 레코드의 콘텐츠가 포함됩니다.
awsAccountId	메시지를 보내는 데 사용된 AWS 계정의 ID입니다.

애플리케이션

이벤트가 연결된 Amazon Pinpoint 프로젝트에 대한 정보를 포함합니다.

속성	설명
app_id	이벤트를 보고한 Amazon Pinpoint 프로젝트의 고유 ID입니다.
sdk	이벤트를 보고하는 데 사용된 SDK입니다.

속성

이벤트를 생성한 캠페인에 대한 정보를 포함합니다.

속성	설명
treatment_id	A/B 테스트 캠페인을 사용하여 메시지를 전송한 경우 이 값은 메시지의 처리 번호를 나타냅니다. 표준 캠페인의 경우 이 값은 0입니다.
campaign_activity_id	이벤트가 발생할 때 Amazon Pinpoint에서 생성하는 고유 ID입니다.
delivery_type	캠페인의 전송 방법입니다. 이 속성을 client_context 의 endpoint 속성에 지정된 ChannelType 필드와 혼동하지 않도록 합니다. ChannelType 필드는 일반적으로 메시지가 전송되는 엔드포인트를 기반으로 합니다. 엔드포인트 유형을 하나만 지원하는 채널의 경우 delivery_type 및 ChannelType 필드의 값이 동일합니다. 예를 들어, 이메일 채널의 경우 delivery_type 및 ChannelType 필드는 EMAIL 값이 동일합니다. 하지만 사용자 지정 채널과 같이 다양한 엔드포인트 유형을 지원하는 채널의 경우, 이 조건이 항상 적용되는 것은 아닙니다. EMAIL, SMS, CUSTOM 등과 같은 다양한 엔드포인트에 사용자 지정 채널을 사용할 수 있습니다. 이 경우

속성	설명
	`delivery_type` 은 사용자 지정 배달 이벤트인 CUSTOM을 식별하고, `ChannelType` 은 캠페인이 전송된 엔드포인트 유형(예: EMAIL, SMS, CUSTOM 등)을 지정합니다. 사용자 지정 채널 생성에 대한 자세한 정보는 사용자 지정 채널 생성 섹션을 참조하세요. 가능한 값은 다음과 같습니다. • EMAIL • SMS • ADM • APNS • APNS_SANDBOX • APNS_VOIP • APNS_VOIP_샌드박스 • VOICE • GCM • BAIDU • PUSH • CUSTOM
`campaign_id`	메시지를 전송한 캠페인의 고유 ID입니다.

속성	설명
campaign_send_status	대상 엔드포인트에 대한 캠페인 상태를 나타냅니다. 가능한 값은 다음과 같습니다. • SUCCESS – 캠페인이 엔드포인트로 전송되었습니다. • FAILURE – 캠페인이 엔드포인트로 전송되지 않았습니다. • DAILY_CAP – 엔드포인트에 이미 최대 일일 메시지 수가 전송되었으므로 캠페인이 엔드포인트로 전송되지 않았습니다. • EXPIRED – 캠페인의 최대 기간 또는 전송 속도 설정을 초과하므로 캠페인이 엔드포인트로 전송되지 않았습니다. • QUIET_TIME – 전송 중단 시간 제한으로 인해 캠페인이 엔드포인트로 전송되지 않았습니다. • HOLDOUT – 엔드포인트가 거부 그룹의 멤버이므로 캠페인이 엔드포인트로 전송되지 않았습니다. • DUPLICATE_ADDRESS – 세그먼트에 중복된 엔드포인트 주소가 있습니다. 캠페인이 엔드포인트 주소로 한 번 전송되었습니다. • QUIET_TIME – 전송 중단 시간 제한으로 인해 캠페인이 엔드포인트로 전송되지 않았습니다. • CAMPAIGN_CAP – 이 캠페인에서 엔드포인트에 이미 최대 일일 메시지 수가 전송되었으므로 캠페인이 엔드포인트로 전송되지 않았습니다. • FAILURE_PERMANENT – 엔드포인트로 전송하는 동안 영구적인 오류가 발생했습니다. • TRANSIENT_FAILURE – 엔드포인트로 전송하는 동안 일시적인 오류가 발생했습니다. • THROTTLED – 전송이 조절되었습니다.

속성	설명
	• UNKNOWN – 알 수 없는 실패입니다. • HOOK_FAILURE – 캠페인 후크가 실패했습니다. • CUSTOM_DELIVERY_FAILURE – 사용자 지정 배달이 실패했습니다. • RECOMMENDATION_FAILURE – 추천이 실패했습니다. • UNSUPPORTED_CHANNEL – 채널이 지원되지 않습니다.

클라이언트

캠페인의 대상이 된 엔드포인트에 대한 정보를 포함합니다.

속성	설명
`client_id`	캠페인이 전송된 엔드포인트의 ID입니다.

Amazon Pinpoint의 여정 이벤트 데이터

여정을 게시할 때 Amazon Pinpoint는 여정에서 보내는 이메일, SMS, 푸시 및 사용자 지정 메시지에 대한 이벤트 데이터를 스트리밍할 수 있습니다. 이벤트 스트리밍을 설정한 후 Amazon Pinpoint는 설정 중에 지정한 대상에서 데이터를 검색하여 볼 수 있도록 합니다. 이메일 및 SMS 메시지에 대해 Amazon Pinpoint에서 스트리밍하는 데이터에 대한 자세한 내용은 the section called “Amazon Pinpoint에서 이메일 이벤트 데이터 스트리밍” 및 the section called “Amazon Pinpoint의 SMS 이벤트 데이터 스트림” 섹션을 참조하세요. 이벤트 스트리밍 설정 방법에 대한 자세한 내용은 Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정 섹션을 참조하세요.

여정 이벤트 예제

여정 이벤트에 대한 JSON 객체는 다음 샘플에 나와 있는 데이터를 포함합니다.

```
{
```

```
   "event_type":"_journey.send",
   "event_timestamp":1572989078843,
   "arrival_timestamp":1572989078843,
   "event_version":"3.1",
   "application":{
      "app_id":"a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
      "sdk":{

      }
   },
   "client":{
      "client_id":"d8dcf7c5-e81a-48ae-8313-f540cexample"
   },
   "device":{
      "platform":{

      }
   },
   "session":{

   },
   "attributes":{
      "journey_run_id":"edc9a0b577164d1daf72ebd15example",
      "journey_send_status":"SUCCESS",
      "journey_id":"546401670c5547b08811ac6a9example",
      "journey_activity_id":"0yKexample",
      "journey_activity_type": "EMAIL",
      "journey_send_status_message": "200",
      "journey_send_status_code": "200"
   },
   "client_context":{
      "custom":{
         "endpoint":"{\"ChannelType\":\"EMAIL\",\"EndpointStatus\":\"ACTIVE\",\"OptOut
\":\"NONE\",\"Demographic\":{\"Timezone\":\"America/Los_Angeles\"}}"
      }
   },
   "awsAccountId":"123456789012"
}
```


여정 이벤트 속성

이 섹션은 Amazon Pinpoint에서 여정에 대해 생성하는 이벤트 스트림 데이터에 포함된 속성을 정의합니다.

속성	설명
`event_type`	이벤트의 유형입니다. 여정 이벤트의 경우 이 속성의 값은 항상 `_journey.send` 이며, Amazon Pinpoint가 여정을 실행했음을 나타냅니다.
`event_timestamp`	이벤트가 보고된 시간입니다(밀리초 단위 Unix 시간).
`arrival_timestamp`	Amazon Pinpoint에서 이벤트가 수신된 시간입니다(밀리초 단위 Unix 시간).
`event_version`	이벤트 JSON 스키마의 버전입니다. **Tip** 스키마 업데이트에 대응하여 언제 애플리케이션을 업데이트해야 하는지 알 수 있도록 이벤트 처리 애플리케이션에서 이 버전을 확인합니다.
`application`	이벤트와 관련된 Amazon Pinpoint 프로젝트에 대한 정보입니다. 자세한 내용은 애플리케이션 표를 참조하십시오.
`client`	이벤트와 연결된 엔드포인트에 대한 정보입니다. 자세한 내용은 클라이언트 표를 참조하십시오.
`device`	이벤트를 보고한 디바이스에 대한 정보입니다. 여정의 경우, 이 객체는 비어 있습니다.
`session`	이벤트를 생성한 세션에 대한 정보입니다. 여정의 경우, 이 객체는 비어 있습니다.

속성	설명
attributes	이벤트를 생성한 여정 및 여정 활동과 연결된 속성입니다. 자세한 내용은 속성 표를 참조하십시오.
client_context	endpoint 속성을 포함하는 custom 객체를 포함합니다. endpoint 속성에는 이벤트와 연결된 엔드포인트에 대한 엔드포인트 레코드의 콘텐츠가 포함됩니다.
awsAccountId	여정을 실행하는 데 사용된 AWS 계정의 ID입니다.

애플리케이션

이벤트가 연결된 Amazon Pinpoint 프로젝트에 대한 정보를 포함합니다.

속성	설명
app_id	이벤트를 보고한 Amazon Pinpoint 프로젝트의 고유 ID입니다.
sdk	이벤트를 보고하는 데 사용된 SDK입니다.

클라이언트

이벤트와 연결된 엔드포인트에 대한 정보를 포함합니다.

속성	설명
client_id	엔드포인트의 ID입니다.

속성

이벤트를 생성한 여정에 대한 정보를 포함합니다.

속성	설명
journey_run_id	이벤트를 생성한 여정 실행의 고유 ID입니다. Amazon Pinpoint는 여정을 새로 실행할 때마다 이 ID를 생성하고 자동으로 할당합니다.
journey_send_status	이벤트와 연결된 메시지의 배달 상태를 나타냅니다. 가능한 값은 다음과 같습니다. • SUCCESS – 메시지가 엔드포인트로 전송되었습니다. • FAILURE – 오류가 발생하여 엔드포인트에 메시지가 전송되지 않았습니다. • CUSTOM_DELIVERY_FAILURE – 사용자 지정 배달이 실패했습니다. • FAILURE_PERMANENT – 엔드포인트로 전송하는 동안 영구적인 오류가 발생했습니다. Tip FAILURE_PERMANENT 상태의 이벤트를 필터링하고 journey_send_status_code 를 403으로 설정하여 액세스 정책 및 역할 위반이 있는지 확인할 수 있습니다. 음성을 사용하는 아웃바운드 캠페인의 경우 이러한 예외는 Amazon Pinpoint 여정을 Connect Customer 캠페인으로 바인딩하는 연결 캠페인 실행 역할이 진행 중인 여정 실행을 위해 실수로 삭제되는 인스턴스에 일반적으로 적용됩니다. • THROTTLED – 전송이 조절되었습니다. • UNSUPPORTED_CHANNEL – 채널이 지원되지 않습니다.

속성	설명
	• DAILY_CAP – 메시지를 보내면 24시간 동안 여정 또는 프로젝트가 단일 엔드포인트로 보낼 수 있는 최대 메시지 수를 초과하기 때문에 엔드포인트로 메시지가 전송되지 않았습니다. • QUIET_TIME – 여정 또는 프로젝트의 대기 시간 제한으로 인해 메시지가 전송되지 않았습니다. • QUIET_TIME_MISSING_TIMEZONE – 추정으로 엔드포인트의 시간대를 추정할 수 없고 전송 중단 시간이 활성화되어 있기 때문에 메시지가 전송되지 않았습니다.
`journey_id`	이벤트를 생성한 여정의 고유 ID입니다.
`journey_activity_id`	이벤트를 생성한 여정 활동의 고유 ID입니다.
`journey_activity_type`	이벤트의 여정 활동 유형입니다. 이는 EMAIL, SMS, PUSH, CONTACT_CENTER 또는 CUSTOM일 수 있습니다. **Note** VOICE는 지원되는 여정 활동 유형이 아닙니다. `journey_send_status` 가 QUIET_TIME_WAIT_FINISHED로 설정된 경우 `journey_activity_type` 필드가 존재하지 않습니다.
`journey_send_status_message`	전송 이벤트의 상태에 대한 설명입니다.
`journey_send_status_code`	요청의 HTTP 상태 코드입니다.

Amazon Pinpoint에서 이메일 이벤트 데이터 스트리밍

Amazon Pinpoint를 사용하여 이메일을 보내는 경우 Amazon Pinpoint는 해당 이메일에 대한 이벤트 데이터를 스트리밍할 수 있습니다. 이벤트 스트리밍을 설정한 후 Amazon Pinpoint는 설정 중에 지정한 대상에서 이벤트 데이터를 검색하여 볼 수 있도록 합니다. 이벤트 스트리밍 설정 방법에 대한 자세한 내용은 Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정 섹션을 참조하세요. Amazon Pinpoint는 이메일 메시지에 대해 다음 유형의 이벤트에 대한 데이터를 스트리밍합니다.

- 전송
- 배달
- 반송 메일
- 수신 거부
- 열기
- 클릭
- 거부
- 구독 해지
- 렌더링 오류

이러한 이벤트 유형은 이메일 이벤트 속성에 자세히 설명되어 있습니다.

이메일 메시지를 전송하는 데 사용하는 API 및 설정에 따라 추가 이벤트 유형이나 다른 데이터가 표시될 수 있습니다. 예를 들어 Amazon Simple Email Service(Amazon SES)에서 제공하는 것과 같이 이벤트 데이터를 Amazon Kinesis에 게시하는 구성 세트를 사용하여 메시지를 전송하는 경우, 데이터에는 템플릿 렌더링 실패에 대한 이벤트도 포함될 수 있습니다. 이러한 데이터에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES 이벤트 게시를 사용하여 모니터링 섹션을 참조하세요. 이벤트를 보려면 먼저 이벤트 스트리밍을 설정해야 합니다. Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정 섹션을 참조하세요. 이벤트 스트리밍을 설정할 때 이벤트 데이터를 저장할 대상을 지정한 다음 대상을 사용하여 볼 이벤트 데이터를 검색할 수 있습니다.

이메일 이벤트 예제

이메일 전송

이메일 전송 이벤트에 대한 JSON 객체에는 다음 예제의 데이터가 포함되어 있습니다.

```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
```

```
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "send": {}
      }
    }
  }
}
```

이메일 배달됨

이메일 배달됨 이벤트에 대한 JSON 객체에는 다음 예제의 데이터가 포함되어 있습니다.

```
{
  "event_type": "_email.delivered",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622690,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "e9a3000d-daa2-40dc-ac47-1cd34example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "delivered"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
```

```
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "delivery": {
          "smtp_response": "250 ok:  Message 82080542 accepted",
          "reporting_mta": "a8-53.smtp-out.amazonses.com",
          "recipients": ["recipient@example.com"],
          "processing_time_millis": 1310
        }
      }
    }
  }
}
```

이메일 클릭

이메일 클릭 이벤트에 대한 JSON 객체에는 다음 예제의 데이터가 포함되어 있습니다.

```
{
  "event_type": "_email.click",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618713751,
```

```
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "49c1413e-a69c-46dc-b1c4-6470eexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "https://aws.amazon.com/pinpoint/"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
```

```
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "click": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15",
          "link": "https://aws.amazon.com/pinpoint/"
        }
      }
    }
  }
}
```

이메일 열기

이메일 열기 이벤트에 대한 JSON 객체에는 다음 예제의 데이터가 포함되어 있습니다.

```
{
  "event_type": "_email.open",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618712316,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "8dc1f651-b3ec-46fc-9b67-2a050example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "opened"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
```

```
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "open": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko)"
        }
      }
    }
  }
}
```

이메일 이벤트 속성

이 섹션에서는 이메일 메시지를 보낼 때 Amazon Pinpoint에서 생성하는 이벤트 스트림 데이터의 이전 예제에 포함된 속성을 정의합니다.

속성	설명
`event_type`	이벤트의 유형입니다. 가능한 값은 다음과 같습니다. • _email.send – Amazon Pinpoint가 메시지를 수락하고 수신자에게 배달하려고 시도했습니다. • _email.delivered – 메시지가 수신자에게 배달되었습니다. • _email.rejected – Amazon Pinpoint가 메시지에 맬웨어가 포함되어 있다고 판단하여 전송을 시도하지 않습니다. • _email.hardbounce – 영구적인 문제로 인해 Amazon Pinpoint가 메시지를 배달하지 못했습니다. Amazon Pinpoint는 메시지 배달을 다시 시도하지 않습니다. • _email.softbounce – 일시적인 문제로 인해 Amazon Pinpoint가 메시지를 배달하지 못했습니다. Amazon Pinpoint는 일정 시간 동안 메시지 배달을 다시 시도합니다. 그래도 메시지를 배달할 수 없는 경우에는 더 이상 재시도하지 않습니다. 이렇게 되면 이메일의 최종 상태는 SOFTBOUNCE가 됩니다. • _email.complaint – 수신자는 메시지를 받은 다음 이메일 공급자에게 메시지를 스팸으로 보고했습니다(예: 이메일 클라이언트의 "스팸 보고" 기능 사용). • _email.open – 수신자가 메시지를 수신하고 열었습니다.

속성	설명
	• _email.click – 수신자가 메시지를 수신하고 메시지에 포함된 링크를 클릭했습니다. • _email.unsubscribe – 수신자가 메시지를 수신하고 메시지에 포함된 구독 해지 링크를 클릭했습니다. • _email.rendering_failure – 렌더링 실패로 인해 이메일이 전송되지 않았습니다. 이는 템플릿 데이터가 누락되었을 때 또는 템플릿 파라미터와 데이터 사이에 불일치가 있을 때 발생할 수 있습니다.
`event_timestamp`	메시지가 전송된 시간입니다(밀리초 단위 Unix 시간). 이 값은 일반적으로 메시지에 대해 생성되는 모든 이벤트에 동일합니다.
`arrival_timestamp`	Amazon Pinpoint에서 이벤트가 수신된 시간입니다(밀리초 단위 Unix 시간).
`event_version`	이벤트 JSON 스키마의 버전입니다. **Tip** 스키마 업데이트에 대응하여 언제 애플리케이션을 업데이트해야 하는지 알 수 있도록 이벤트 처리 애플리케이션에서 이 버전을 확인합니다.
`application`	이벤트와 관련된 Amazon Pinpoint 프로젝트에 대한 정보입니다. 자세한 내용은 애플리케이션 표를 참조하십시오.
`client`	이벤트를 보고한 디바이스에 설치된 앱 클라이언트에 대한 정보입니다. 자세한 내용은 클라이언트 표를 참조하십시오.

속성	설명
device	이벤트를 보고한 디바이스에 대한 정보입니다. 자세한 내용은 디바이스 표를 참조하십시오. 이메일 이벤트의 경우 이 객체가 비어 있습니다.
session	이메일 이벤트의 경우 이 객체가 비어 있습니다.
attributes	이벤트와 연결된 속성입니다. 자세한 내용은 속성 표를 참조하십시오. 앱 중 하나에서 보고한 이벤트의 경우 이 객체에는 앱에서 정의한 사용자 지정 속성이 포함될 수 있습니다. 캠페인이나 여정에서 메시지를 전송할 때 생성되는 이벤트의 경우, 이 객체에는 캠페인이나 여정과 연결된 속성이 포함됩니다. 트랜잭션 메시지를 전송할 때 생성되는 이벤트의 경우 이 객체에는 메시지 자체와 관련된 정보가 포함됩니다.
client_context	이메일 이벤트의 경우 이 객체에는 custom 객체(legacy_identifier 속성 포함)가 포함됩니다. legacy_identifier 속성 값은 메시지가 전송된 프로젝트의 ID입니다.
facets	메시지에 대한 추가 정보입니다(예: 이메일 헤더). 자세한 내용은 패싯 표를 참조하십시오.
awsAccountId	메시지를 전송하는 데 사용된 AWS 계정의 ID입니다.

Application

이벤트가 연결된 Amazon Pinpoint 프로젝트에 대한 정보를 포함합니다.

속성	설명
app_id	이벤트를 보고한 Amazon Pinpoint 프로젝트의 고유 ID입니다.
sdk	이벤트를 보고하는 데 사용된 SDK입니다. Amazon Pinpoint API를 직접 호출하거나 Amazon Pinpoint 콘솔을 사용하여 트랜잭션 이메일 메시지를 전송하는 경우 이 객체는 비어 있습니다.

속성

이벤트를 생성한 캠페인 또는 여정에 대한 정보를 포함합니다.

Campaign

이벤트를 생성한 캠페인에 대한 정보를 포함합니다.

속성	설명
feedback	_email.click 이벤트의 경우 이 속성의 값은 수신자가 이벤트를 생성하기 위해 메시지에서 클릭한 링크의 URL입니다. 다른 이벤트의 경우 이 값은 이벤트 유형을 나타냅니다(예:received, opened 또는 clicked).
treatment_id	A/B 테스트 캠페인을 사용하여 메시지를 전송한 경우 이 값은 메시지의 처리 번호를 나타냅니다. 표준 캠페인 및 트랜잭션 이메일 메시지의 경우 이 값은 0입니다.
campaign_activity_id	이벤트가 발생할 때 Amazon Pinpoint에서 생성하는 고유 ID입니다.
campaign_id	메시지를 전송한 캠페인의 고유 ID입니다.

여정

이벤트를 생성한 여정에 대한 정보를 포함합니다.

속성	설명
journey_run_id	메시지를 보낸 여정 실행의 고유 ID입니다. Amazon Pinpoint는 여정을 새로 실행할 때마다 이 ID를 생성하고 자동으로 할당합니다.
feedback	_email.click 이벤트의 경우 이 속성의 값은 수신자가 이벤트를 생성하기 위해 메시지에서 클릭한 링크의 URL입니다. 다른 이벤트의 경우 이 값은 이벤트 유형을 나타냅니다(예:received, delivered 또는 opened).
journey_id	메시지를 보낸 여정의 고유 ID입니다.
journey_activity_id	메시지를 보낸 여정 활동의 고유 ID입니다.

클라이언트

캠페인 또는 여정의 대상이 된 고객의 고유 식별자입니다.

속성	설명
client_id	클라이언트의 ID입니다. 값은 캠페인 및 여정의 경우에는 엔드포인트 ID이고, 트랜잭션 전송의 경우에는 UUID입니다.

패싯

메시지 및 이벤트 유형에 대한 정보를 포함합니다.

속성	설명
email_channel	mail_event 객체를 포함하며, 여기에는 두 개의 객체 즉, mail 및 이벤트 유형에 해당하는 객체가 포함됩니다.

Mail

이메일 메시지의 콘텐츠에 대한 정보와 메시지에 대한 메타데이터를 포함합니다.

속성	설명
message_id	메시지의 고유 ID입니다. Amazon Pinpoint는 메시지를 수락할 때 이 ID를 자동으로 생성합니다.
message_send_timestamp	RFC 822에 지정된 형식으로 메시지가 전송된 날짜 및 시간입니다.
from_address	메시지를 전송한 이메일 주소입니다.
destination	메시지를 전송한 이메일 주소가 포함된 배열입니다.
headers_truncated	이메일 헤더가 잘렸는지 여부를 나타내는 Boolean 값입니다.
headers	메시지 헤더에 해당하는 여러 이름-값 쌍을 포함하는 객체입니다. 이 객체는 일반적으로 다음 헤더에 대한 정보를 포함합니다. • From - 발신자의 이메일 주소입니다. • To - 수신자의 이메일 주소입니다. • Subject - 이메일의 제목 줄입니다.

속성	설명
	Tip 캠페인 _email.send 이벤트에는 제목 헤더가 포함되지 않습니다. • `MIME-Version` - 메시지가 MIME 형식임을 나타냅니다. 이 헤더가 있으면 값은 항상 `1.0`입니다. • `Content-Type` - 메시지 콘텐츠의 MIME 미디어 유형입니다.
`common_headers`	이메일 메시지의 몇 가지 일반적인 헤더에 대한 정보가 들어 있습니다. 이 정보는 메시지가 전송된 날짜, 메시지 받는 사람, 보낸 사람 및 제목 줄이 포함될 수 있습니다.

Amazon Pinpoint의 SMS 이벤트 데이터 스트림

프로젝트에 SMS 채널을 활성화한 경우, Amazon Pinpoint는 프로젝트의 SMS 메시지 전송에 대한 이벤트 데이터를 스트리밍할 수 있습니다. 이벤트 스트리밍을 설정한 후 Amazon Pinpoint는 설정 중에 지정한 대상에서 이벤트 데이터를 검색하여 볼 수 있도록 합니다. 이벤트 스트리밍 설정 방법에 대한 자세한 내용은 Amazon Kinesis 또는 Amazon Data Firehose를 통해 앱 이벤트 데이터를 스트리밍하도록 Amazon Pinpoint 설정 섹션을 참조하세요.

Note

통신사에서 생성한 SMS 이벤트는 수신될 때까지 최대 72시간이 걸릴 수 있으며, 아웃바운드 메시지 배달의 지연 여부를 판단하는 데 사용해서는 안 됩니다. 72시간이 지난 후에도 Amazon Pinpoint가 통신사로부터 최종 이벤트를 수신하지 않은 경우, 해당 메시지가 어떻게 되었는지 알 수 없으므로 서비스는 자동으로 UNKNOWN record_status를 반환합니다.

SMS 이벤트 예

SMS 이벤트에 대한 JSON 객체에는 다음 예제의 데이터가 포함되어 있습니다.

```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "123456789012"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
    "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
    "customer_context": "{\"userId\":\"user-id-4\"}"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
}
```

SMS 이벤트 속성

이 섹션에서는 SMS 메시지를 보낼 때 Amazon Pinpoint에서 생성하는 이벤트 스트림 데이터의 이전 예제에 포함된 속성을 정의합니다.

Event

속성	설명
`event_type`	이벤트의 유형입니다. 가능한 값은 다음과 같습니다. • _SMS.BUFFERED – 메시지가 수신자에게 배달되는 중입니다. • _SMS.SUCCESS – 메시지가 수신자에게 전달되었습니다. • _SMS.FAILURE – Amazon Pinpoint가 메시지를 수신자에게 배달할 수 없었습니다. 메시지 배달을 방해하는 오류에 대한 자세한 내용은 `attributes.record_status` 단원을 참조하십시오. • _SMS.OPTOUT – 고객이 메시지를 수신하고 옵트아웃 키워드(보통 "STOP")를 전송하여 응답했습니다.
`event_timestamp`	이벤트가 보고된 시간입니다(밀리초 단위 Unix 시간).
`arrival_timestamp`	Amazon Pinpoint에서 이벤트가 수신된 시간입니다(밀리초 단위 Unix 시간).
`event_version`	이벤트 JSON 스키마의 버전입니다. **Tip** 스키마 업데이트에 대응하여 언제 애플리케이션을 업데이트해야 하는지 알 수

속성	설명
	있도록 이벤트 처리 애플리케이션에서 이 버전을 확인합니다.
`application`	이벤트와 관련된 Amazon Pinpoint 프로젝트에 대한 정보입니다. 자세한 내용은 애플리케이션 표를 참조하십시오.
`client`	이벤트를 보고한 디바이스에 설치된 앱 클라이언트에 대한 정보입니다. 자세한 내용은 클라이언트 표를 참조하십시오.
`device`	이벤트를 보고한 디바이스에 대한 정보입니다. 자세한 내용은 디바이스 표를 참조하십시오. SMS 이벤트의 경우 이 객체가 비어 있습니다.
`session`	SMS 이벤트의 경우 이 객체가 비어 있습니다.
`attributes`	이벤트와 연결된 속성입니다. 앱 중 하나에서 보고한 이벤트의 경우 이 객체에는 앱에서 정의한 사용자 지정 속성이 포함될 수 있습니다. 캠페인을 전송할 때 생성되는 이벤트의 경우 이 객체에는 캠페인과 연결된 속성이 포함됩니다. 트랜잭션 메시지를 전송할 때 생성되는 이벤트의 경우 이 객체에는 메시지 자체와 관련된 정보가 포함됩니다. 자세한 내용은 속성 표를 참조하십시오.
`metrics`	이벤트와 연결된 추가 지표입니다. 자세한 내용은 지표 표를 참조하십시오.
`awsAccountId`	메시지를 전송하는 데 사용된 AWS 계정의 ID입니다.

Application

이벤트와 연결된 Amazon Pinpoint 프로젝트와 이벤트 보고에 사용된 SDK(해당하는 경우)에 대한 정보를 포함합니다.

속성	설명
app_id	이벤트를 보고한 Amazon Pinpoint 프로젝트의 고유 ID입니다.
sdk	이벤트를 보고하는 데 사용된 SDK입니다. Amazon Pinpoint API를 직접 호출하거나 Amazon Pinpoint 콘솔을 사용하여 트랜잭션 SMS 메시지를 전송하는 경우 이 객체는 비어 있습니다.

속성

이벤트와 연결된 속성에 대한 정보를 포함합니다.

속성	설명
sender_request_id	SMS 메시지 전송 요청과 연결된 고유 ID입니다.
campaign_activity_id	캠페인 내 활동의 고유 ID입니다.
origination_phone_number	메시지를 전송한 전화 번호입니다.
destination_phone_number	메시지 전송을 시도한 전화 번호입니다.
record_status	메시지 상태에 대한 추가 정보입니다. 가능한 값은 다음과 같습니다. • SUCCESSFUL/DELIVERED – 메시지가 성공적으로 배달되었습니다. • PENDING – 메시지가 아직 수신자의 디바이스로 배달되지 않았습니다. • INVALID – 대상 전화번호가 잘못되었습니다.

속성	설명
	• UNREACHABLE – 수신자의 디바이스가 현재 연결할 수 없거나 사용할 수 없습니다. 예를 들어, 디바이스의 전원이 꺼져 있거나 네트워크와의 연결이 끊어졌을 수 있습니다. 나중에 메시지 전송을 다시 시도할 수 있습니다. • UNKNOWN – 메시지 배달을 방해하는 오류가 발생했습니다. 이 오류는 일반적으로 일시적이며 나중에 메시지 전송을 다시 시도할 수 있습니다. • BLOCKED – 수신자의 디바이스가 발신 번호로부터의 SMS 메시지를 차단하고 있습니다. • CARRIER_UNREACHABLE – 수신자의 모바일 네트워크 문제로 인해 메시지가 배달되지 않았습니다. 이 오류는 일반적으로 일시적이며 나중에 메시지 전송을 다시 시도할 수 있습니다. • SPAM – 수신자의 통신사에서 메시지의 콘텐츠를 스팸으로 식별하여 메시지 배달을 차단했습니다. • INVALID_MESSAGE – SMS 메시지 본문이 잘못되어 배달할 수 없습니다. • CARRIER_BLOCKED – 수신자의 이동 통신 업체에서 이 메시지 배달을 차단했습니다. 이는 이동 통신 업체에서 메시지 콘텐츠를 원치 않는 메시지 또는 악성 메시지로 식별할 때 종종 발생합니다. • TTL_EXPIRED – 특정 시간 내에 SMS 메시지를 배달할 수 없었습니다. 이 오류는 일반적으로 일시적이며 나중에 메시지 전송을 다시 시도할 수 있습니다. • MAX_PRICE_EXCEEDED – 메시지를 전송할 경우 해당 계정의 월별 SMS 지출 할당량을 초과하는 요금이 발생합니다. Amazon Pinpoint

속성	설명
	사용 설명서의 월별 SMS 지출 할당량 증가 요청의 절차를 완료하여 이러한 할당량 증가를 요청할 수 있습니다. • OPTED_OUT – 수신자가 나의 메시지 수신을 옵트아웃했기 때문에 SMS 메시지가 전송되지 않았습니다. • NO_QUOTA_LEFT_ON_ACCOUNT – 계정에 남은 지출 할당량이 부족하여 메시지를 전송할 수 없습니다. AWS End User Messaging SMS 사용 설명서의 월별 SMS 지출 할당량 증가 요청의 절차를 완료하여 이러한 할당량 증가를 요청할 수 있습니다. • NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND – 계정에 대상으로 메시지를 전송하는 데 사용할 수 있는 전화번호가 포함되어 있지 않습니다. • DESTINATION_COUNTRY_NOT_SUPPORTED – 대상 국가가 차단되었습니다. 지원되는 모든 국가에 대해서는 AWS End User Messaging SMS 사용 설명서의 지원되는 국가 및 리전(SMS 채널)을 참조하세요. • ACCOUNT_IN_SANDBOX – 계정이 샌드박스에 있으며 인증된 대상 번호로만 이를 전송할 수 있습니다. Amazon Pinpoint 콘솔에서 대상 번호를 확인하거나 프로세스를 시작하여 샌드박스에서 계정을 이동할 수 있습니다. AWS End User Messaging SMS 사용 설명서의 SMS/MMS 및 음성 샌드박스 정보를 참조하세요. • RATE_EXCEEDED – 메시지를 너무 빨리 전송하려고 시도하여 조절되었습니다. 호출 속도를 낮춰야 합니다. 제한에 대한 자세한 내용은 AWS End User Messaging SMS 사용 설명

속성	설명
	서의 초당 메시지 부분(MPS) 제한을 참조하세요. • INVALID_ORIGINATION_IDENTITY – 제공된 발신 ID가 잘못되었습니다. • ORIGINATION_IDENTITY_DOES_NOT_EXIST – 제공된 발신 ID가 존재하지 않습니다. • INVALID_DLT_PARAMETERS – 잘못된 DLT 파라미터(인도 내의 대상에 필요)가 제공되었습니다. • INVALID_PARAMETERS – 잘못된 파라미터가 제공되었습니다. • ACCESS_DENIED – 계정의 메시지 전송이 차단되었습니다. 고객 지원팀에 문의하여 원인을 확인하고 문제를 해결하세요. • INVALID_KEYWORD – 제공된 키워드가 잘못되었습니다. 키워드 형식이 잘못되었거나 계정에 설정되지 않은 것일 수 있습니다. • INVALID_SENDER_ID – 제공된 발신자 ID가 잘못되었습니다. 발신자 ID의 형식이나 길이가 잘못되었을 수 있습니다. • INVALID_POOL_ID – 제공된 풀 ID가 잘못되었습니다. 풀 ID 형식이 잘못되었거나 계정에 속하지 않은 것일 수 있습니다. • SENDER_ID_NOT_SUPPORTED_FOR_DESTINATION – 대상 국가가 발신자 ID를 지원하지 않습니다. 전송하려면 전화번호 또는 다른 발신자 ID를 사용해야 합니다. • INVALID_PHONE_NUMBER – 제공된 발신자 전화번호가 잘못되었습니다. 전화번호의 형식이나 길이가 잘못되었을 수 있습니다.

속성	설명
iso_country_code	수신자의 전화 번호와 관련된 국가 이며, ISO 3166-1 alpha-2 형식으로 표시됩니다.
treatment_id	메시지가 A/B 캠페인으로 전송된 경우 메시지 처리의 ID입니다.
treatment_id	A/B 테스트 캠페인을 사용하여 메시지를 전송한 경우 이 값은 메시지의 처리 번호를 나타냅니다. 트랜잭션 SMS 메시지의 경우 이 값은 0입니다.
number_of_message_parts	Amazon Pinpoint가 메시지를 전송하기 위해 생성한 메시지 부분 수입니다. 일반적으로 SMS 메시지는 160개의 GSM-7 문자 또는 67개의 비GSM 문자만 포함할 수 있지만 이러한 한도는 국가 에 따라 다를 수 있습니다. 이 한도를 초과하는 메시지를 전송하면 Amazon Pinpoint에서 자동으로 메시지를 더 작게 분할합니다. 전송하는 메시지 부분 수에 따라 요금이 청구됩니다.
message_id	메시지를 수락할 때 Amazon Pinpoint에서 생성하는 고유 ID입니다.
message_type	메시지 유형. 가능한 값은 Promotional 및 Transactional입니다. 캠페인을 생성할 경우 또는 Amazon Pinpoint API에서 SendMessages 작업을 사용하여 트랜잭션 메시지를 전송할 경우 이 값을 지정합니다.
campaign_id	메시지를 전송한 Amazon Pinpoint 캠페인의 고유 ID입니다.
customer_context	Amazon Pinpoint SendMessages 작업에서 전송된 Context 맵의 콘텐츠 JSON 문자열입니다.

클라이언트

이벤트를 보고한 디바이스에 설치된 앱 클라이언트에 대한 정보를 포함합니다.

속성	설명
`client_id`	앱에서 생성된 이벤트의 경우 이 값은 디바이스에 설치된 앱 클라이언트의 고유 ID입니다. 이 ID는 AWS Mobile SDK for iOS 및 AWS Mobile SDK for Android에 의해 자동으로 생성됩니다. 캠페인 및 트랜잭션 메시지를 전송할 때 생성되는 이벤트의 경우 이 값은 메시지를 전송한 엔드포인트의 ID와 같습니다.
`cognito_id`	앱이 사용하는 Amazon Cognito 자격 증명 풀에서 앱 클라이언트에 지정된 고유 ID입니다.

디바이스

이벤트를 보고한 디바이스에 대한 정보를 포함합니다.

속성	설명
`locale`	디바이스 로캘입니다.
`make`	디바이스 제조사입니다(예: `Apple` 또는 `Samsung`).
`model`	디바이스 모델입니다(예: `iPhone`).
`platform`	디바이스 플랫폼입니다(예: `ios` 또는 `android`).

지표

이벤트와 연결된 지표에 대한 정보를 포함합니다.

속성	설명
price_in_millicents_usd	메시지 전송에 대해 청구된 금액입니다. 이 가격은 미국 센트의 1/1000로 표시됩니다. 예를 들어 이 속성의 값이 645인 경우, 메시지 전송에 대해 0.645¢가 청구된 것입니다(645 / 1000 = 0.645¢ = $0.00645). **Note** 이 속성은 event_type 이 _SMS.BUFFERED인 메시지에는 표시되지 않습니다.

Amazon Pinpoint에서 이벤트 스트림 삭제

한 Kinesis 스트림을 한 앱에 할당한 경우 해당 앱에 대해 이벤트 스트리밍을 비활성화할 수 있습니다. Amazon Pinpoint는 Kinesis로의 이벤트 스트리밍을 중단하지만, Amazon Pinpoint 콘솔을 사용하여 이벤트 분석을 볼 수 있습니다.

AWS CLI

delete-event-stream 명령을 사용합니다.

```
aws pinpoint delete-event-stream --application-id application-id
```

AWS SDK for Java

Amazon Pinpoint 클라이언트의 deleteEventStream 메서드를 사용합니다.

```
pinClient.deleteEventStream(new DeleteEventStreamRequest().withApplicationId(appId));
```

Amazon Pinpoint 분석 데이터 쿼리

Amazon Pinpoint 콘솔에서 분석 페이지를 사용하는 것 외에도, Amazon Pinpoint Analytics API를 사용하여 사용자 참여, 캠페인 활동 등과 관련된 추세를 파악할 수 있는 표준 지표의 하위 집합에 대한 분석 데이터를 쿼리할 수 있습니다. KPI(핵심 성과 지표)라고도 하는 이 지표는 프로젝트, 캠페인 및 여정의 성과를 모니터링하고 평가하는 데 사용할 수 있는 측정 가능한 값입니다.

API를 사용하여 분석 데이터를 쿼리하는 경우 Amazon Pinpoint 콘솔에 로그인하거나 Amazon Kinesis 스트림과 같은 소스의 원시 이벤트 데이터를 분석할 필요 없이 원하는 보고 도구를 사용하여 데이터를 분석할 수 있습니다. 예를 들어 주간 캠페인 결과를 표시하거나 캠페인의 배달율을 자세히 분석하는 사용자 지정 대시보드를 만들 수 있습니다.

Amazon Pinpoint REST API, AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하여 데이터를 쿼리할 AWS 수 있습니다. 데이터를 쿼리하려면 Amazon Pinpoint API에 요청을 보내고 지원되는 파라미터를 사용하여 원하는 데이터와 적용할 필터를 지정합니다. 쿼리를 제출한 후 Amazon Pinpoint는 쿼리 결과를 JSON 응답으로 반환합니다. 그런 다음 분석, 저장 또는 보고 기능을 향상시키기 위해 결과를 다른 서비스나 애플리케이션에 전달할 수 있습니다.

Amazon Pinpoint는 지원되는 모든 지표와 모든 프로젝트, 캠페인 및 여정에 대한 데이터를 자동으로 수집하고 집계합니다. 또한 데이터가 지속적으로 업데이트되므로 데이터 지연 시간 기간이 약 2시간으로 제한됩니다. 그러나 특정 지표에 대한 추가 데이터 지연 시간이 있을 수 있습니다. 이는 일부 지표에 대한 데이터가 수신자의 이메일 공급자로부터 수신한 정보를 기반으로 하기 때문입니다. 일부 공급자는 즉시 이 정보를 보내지만 그렇지 않은 공급자도 있습니다.

Amazon Pinpoint는 90일 동안 데이터를 저장합니다. 90일 이상 데이터를 저장하거나 원시 분석 데이터에 실시간으로 액세스하려면 Amazon Pinpoint 프로젝트를 구성하여 이벤트 데이터를 Amazon Kinesis Data Streams 또는 Amazon Data Firehose로 스트리밍할 수 있습니다. 이벤트 스트림 구성에 대한 자세한 내용은 Amazon Pinpoint를 사용하여 Kinesis 및 Firehose를 통해 앱 이벤트 데이터 스트리밍 단원을 참조하십시오.

Amazon Pinpoint의 지표에 대한 구성 요소 및 파라미터 쿼리

지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 적절한 지표 리소스에 `get` 요청을 보냅니다. 요청에서 다음 쿼리 구성 요소에 대해 지원되는 파라미터를 사용하여 쿼리를 정의합니다.

- 프로젝트 - `application-id` 파라미터 값으로 프로젝트 ID를 제공하여 프로젝트를 지정합니다. 이 파라미터는 모든 지표에 필요합니다.

- 캠페인 - `campaign-id` 파라미터 값으로 캠페인 ID를 제공하여 캠페인을 지정합니다. 이 파라미터는 캠페인 지표에만 필요합니다.
- 여정 - `journey-id` 파라미터 값으로 여정 ID를 제공하여 여정을 지정합니다. 이 파라미터는 여정 참여 지표, 여정 실행 지표 및 여정 활동 실행 지표에만 필요합니다.
- 여정 활동 - `journey-activity-id` 파라미터 값으로 여정 활동 ID를 제공하여 여정 활동을 지정합니다. 이 파라미터는 여정 활동 실행 지표에만 필요합니다.
- 날짜 범위 - 날짜 범위를 기준으로 데이터를 선택적으로 필터링하려면 지원되는 시작 및 종료 시간 파라미터를 사용하여 날짜 범위의 첫 번째 및 마지막 날짜 및 시간을 제공합니다. 값은 확장된 ISO 8601 형식이어야 하며 협정 세계시(UTC)를 사용해야 합니다. 예를 들어 `2019-07-19T20:00:00Z`는 2019년 7월 19일 오후 8시 UTC입니다.

  날짜 범위는 포괄적이며 31일 이하로 제한되어야 합니다. 또한 첫 번째 날짜 및 시간은 현재 날짜로부터 90일 이내여야 합니다. 날짜 범위를 지정하지 않으면 Amazon Pinpoint에서는 이전 31일 동안의 데이터를 반환합니다. 날짜 범위 파라미터는 여정 실행 지표 및 여정 활동 실행 지표를 제외한 모든 지표에서 지원됩니다.
- 지표 - 지표를 지정하려면 지표 이름을 `kpi-name` 파라미터 값으로 제공합니다. 이 값은 연관된 지표를 설명하며 두 개 이상의 용어로 구성되며, 이 용어는 소문자 영숫자 문자로 구성되며 하이픈으로 구분됩니다. 대표적인 예는 `email-open-rate` 및 `successful-delivery-rate`입니다. 이 파라미터는 여정 실행 지표 및 여정 활동 실행 지표를 제외한 모든 지표에 필요합니다. 지원되는 지표와 각 지표에 사용할 `kpi-name` 값의 전체 목록은 프로젝트, 캠페인 및 여정에 대한 표준 지표 섹션을 참조하세요.

쿼리를 전송한 후 Amazon Pinpoint는 쿼리 결과를 JSON 응답으로 반환합니다. 응답에서 결과 구조는 쿼리한 지표에 따라 달라집니다.

일부 지표는 하나의 값만 제공합니다(예: 캠페인에 의해 배달된 메시지 수). 다른 지표는 여러 값을 제공하며, 일반적으로 해당 값을 관련 필드별로 그룹화합니다(예: 캠페인 실행마다 배달된 메시지 수, 캠페인 실행별로 그룹화됨). 지표가 여러 값을 제공하고 그룹화하는 경우 JSON 응답에는 데이터를 그룹화하는 데 사용된 필드를 나타내는 필드가 포함됩니다. 쿼리 결과의 구조에 대한 자세한 내용은 JSON 쿼리 결과 사용 단원을 참조하십시오.

Amazon Pinpoint 분석 데이터 쿼리를 위한 IAM 정책

Amazon Pinpoint API를 사용하면 Amazon Pinpoint 프로젝트, 캠페인, 여정에 적용되는 핵심 성능 지표(KPI)라고도 하는 표준 지표의 하위 집합에 대한 분석 데이터를 쿼리할 수 있습니다. 이러한 지표를 통해 프로젝트, 캠페인 및 여정의 성과를 모니터링하고 평가할 수 있습니다.

이 데이터에 대한 액세스를 관리하려면 데이터에 액세스할 권한이 있는 IAM 역할 또는 사용자에 대한 권한을 정의하는 AWS Identity and Access Management (IAM) 정책을 생성할 수 있습니다. 이 데이터에 대한 액세스를 세부적으로 제어할 수 있도록 Amazon Pinpoint에서는 IAM 정책에서 지정할 수 있는 여러 가지 고유한 작업을 제공합니다. Amazon Pinpoint 콘솔(`mobiletargeting:GetReports`)에서 분석 데이터를 보는 별도의 작업이 있으며 Amazon Pinpoint API를 사용하여 프로그래밍 방식으로 분석 데이터에 액세스하는 다른 작업이 있습니다.

분석 데이터에 대한 액세스를 관리하는 IAM 정책을 생성하려면 AWS Management Console, AWS CLI 또는 IAM API를 사용할 수 있습니다. 참고로, AWS Management Console 의 시각적 편집기 탭에는 현재 Amazon Pinpoint 분석 데이터를 보거나 쿼리하기 위한 작업이 포함되어 있지 않습니다. 그러나 콘솔의 JSON 탭을 사용하여 IAM 정책에 필요한 작업을 수동으로 추가할 수 있습니다.

예를 들어 다음 정책은 모든 AWS 리전의 모든 프로젝트, 캠페인 및 여정에 대한 모든 분석 데이터에 프로그래밍 방식으로 액세스할 수 있도록 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryAllAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:*:111122223333:apps/*/kpis/*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/campaigns/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
execution-metrics",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
activities/*/execution-metrics"
            ]
```

```
        }
    ]
}
```

여기서 *accountId*는 AWS 계정 ID입니다.

하지만 가장 좋은 방법은 최소 권한의 원칙을 따르는 정책을 만드는 것입니다. 즉, 특정 작업을 수행하는 데 필요한 권한만 포함하는 정책을 만들어야 합니다. 이러한 관행을 지원하고 보다 세분화된 제어를 구현하기 위해 특정 AWS 리전의 특정 프로젝트에 대해서만 분석 데이터에 대한 프로그래밍 방식의 액세스를 제한할 수 있습니다. 예를 들면 다음과 같습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryProjectAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/kpis/
*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/campaigns/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
execution-metrics",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

위치:

- *region*은 프로젝트를 호스팅하는 AWS 리전의 이름입니다.
- *accountId*는 AWS 계정 ID입니다.
- *projectId*는 액세스 권한을 제공하려는 프로젝트의 식별자입니다.

마찬가지로 다음 예제 정책은 특정 캠페인의 분석 데이터에만 프로그래밍 방식으로 액세스할 수 있도록 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryCampaignAnalytics",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetCampaignDateRangeKpi",
            "Resource": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/projectId/campaigns/campaignId/kpis/*"
        }
    ]
}
```

위치:

- *region*은 프로젝트를 호스팅하는 AWS 리전의 이름입니다.
- *accountId*는 AWS 계정 ID입니다.
- *projectId*는 캠페인과 연결된 프로젝트의 식별자입니다.
- *campaignId*는 액세스 권한을 제공하려는 캠페인의 식별자입니다.

다음 예제 정책은 특정 여정과 해당 여정을 구성하는 활동에 대한 모든 분석 데이터(참여 및 실행 데이터)에 프로그래밍 방식으로 액세스할 수 있도록 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryJourneyAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/kpis/*",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/execution-metrics",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/activities/*/execution-metrics"
            ]
        }
    ]
}
```

위치:

- *region*은 프로젝트를 호스팅하는 AWS 리전의 이름입니다.
- *accountId*는 AWS 계정 ID입니다.
- *projectId*는 여정과 연결된 프로젝트의 식별자입니다.
- *journeyId*는 액세스 권한을 제공하려는 여정의 식별자입니다.

IAM 정책에서 사용할 수 있는 Amazon Pinpoint API 작업의 전체 목록은 IAM 정책에 대한 Amazon Pinpoint 작업 섹션을 참조하세요. IAM 정책 생성 및 관리에 대한 자세한 내용은 IAM 사용 설명서를 참조하세요.

Amazon Pinpoint 프로젝트, 캠페인 및 여정에 적용되는 표준 지표

Amazon Pinpoint 분석 API를 사용하여 Amazon Pinpoint 프로젝트, 캠페인 및 여정에 적용되는 표준 지표의 하위 집합에 대한 분석 데이터를 쿼리할 수 있습니다. KPI(핵심 성과 지표)라고도 하는 이 지표는 프로젝트, 캠페인 및 여정의 성과를 모니터링하고 평가하는 데 사용할 수 있는 측정 가능한 값입니다.

Amazon Pinpoint는 여러 표준 지표 유형의 분석 데이터에 대한 프로그래밍 방식 액세스를 제공합니다.

- 애플리케이션 지표 - 프로젝트와 연결된 모든 캠페인 및 트랜잭션 메시지(애플리케이션이라고도 함)에 대한 추세를 파악할 수 있는 지표입니다. 예를 들어 애플리케이션 지표를 사용하면 프로젝트와 연결된 각 캠페인에 대해 수신자가 열어본 메시지 수를 분석할 수 있습니다.
- 캠페인 지표 - 개별 캠페인의 성과를 파악할 수 있는 지표입니다. 예를 들어, 캠페인 지표를 사용하여 캠페인 메시지를 전송한 엔드포인트의 수 또는 엔드포인트에 전달된 메시지의 수를 확인할 수 있습니다.
- 여정 참여 지표 - 이 지표는 개별 여정의 성과를 파악할 수 있습니다. 예를 들어 여정 참여 지표를 사용하여 각 여정 활동에서 참가자가 연 메시지 수를 분석할 수 있습니다.
- 여정 실행 지표 - 이 지표는 개별 여정의 참여 추세를 파악할 수 있습니다. 예를 들어 여정 실행 지표를 사용하여 여정을 시작한 참가자 수를 확인할 수 있습니다.
- 여정 활동 실행 지표 - 이 지표는 여정 내 개별 활동에 대한 참여 추세를 파악할 수 있습니다. 예를 들어 여정 활동 실행 지표를 사용하여 활동을 시작한 참가자 수와 활동의 각 경로를 완료한 참가자 수를 확인할 수 있습니다.

이 섹션의 항목에서는 각 지표 유형에 대해 쿼리할 수 있는 개별 지표를 나열하고 설명합니다.

주제

- 캠페인의 Amazon Pinpoint 애플리케이션 지표
- 트랜잭션 이메일 메시지의 Amazon Pinpoint 애플리케이션 지표
- 트랜잭션 SMS 메시지의 Amazon Pinpoint 애플리케이션 지표
- Amazon Pinpoint 캠페인 지표
- Amazon Pinpoint 여정 참여 지표
- Amazon Pinpoint 여정 실행 지표
- Amazon Pinpoint 여정 활동 실행 지표
- Amazon Pinpoint 여정 및 캠페인 실행 지표

캠페인의 Amazon Pinpoint 애플리케이션 지표

다음 표에는 Amazon Pinpoint 프로젝트와 연결된 모든 캠페인의 성과를 평가하기 위해 쿼리할 수 있는 표준 애플리케이션 지표 목록과 설명이 나와 있습니다. 이러한 지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 애플리케이션 지표 리소스를 사용하세요. kpi-name 열은 쿼리에서 `kpi-name` 파라미터에 사용할 값을 나타냅니다.

지표	kpi-name	설명
전송률	`successful-delivery-rate`	프로젝트와 연결된 모든 캠페인에 대해 수신자에게 배달된 메시지의 비율입니다. 이 지표는 프로젝트의 모든 캠페인에서 전송한 메시지 중 수신자에게 배달된 메시지 수를 모든 캠페인에서 전송한 메시지 수로 나눈 값으로 계산됩니다.
배달율(날짜별 그룹화)	`successful-delivery-rate-grouped-by-date`	프로젝트와 연결된 모든 캠페인에 대해 지정된 날짜 범위의 각 날짜에 수신자에게 배달된 메시지의 비율입니다. 이 지표는 프로젝트의 모든 캠페인에서 전송한 메시지 중 수신자에게 배달된 메시지 수를 지정된 날짜 범위의 각 날짜에 모든 캠페인에서 전송한 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
이메일 오픈율	`email-open-rate`	프로젝트와 연결된 모든 캠페인에 대해 수신자가 열어본 이메일 메시지의 비율입니다. 이 지표는 프로젝트의 모든 캠페인에서 전송한 이메일 메시지 중 수신자가 열어본 이메일 메시지 수를 모든 캠페인에서 전송한 이메일 메시지 중 수신자에게 배달된 이메일 메시지 수로 나눈 값으로 계산됩니다.
이메일 오픈율(캠페인별 그룹화)	`email-open-rate-grouped-by-campaign`	프로젝트와 연결된 각 캠페인에 대해 수신자가 열어본 이메일 메시지의 비율입니다. 이 지표는 한 캠페인에서 전송한 이메일 메시지 중 수신자가 열어본 이메일 메시지 수를 해당 캠페인에서 전송한 이메일 메시지 중 수신자에게 배달된 이메일 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인을 고유하게 식별하는 문자열인 캠페인 ID(`CampaignId`)별로 그룹화됩니다.
엔드포인트 전달	`unique-deliveries`	프로젝트와 연결된 모든 캠페인에 대해 메시지가 배달된 고유 엔드포인트 수입니다.

지표	kpi-name	설명
엔드포인트 배달(캠페인별 그룹화)	`unique-deliveries-grouped-by-campaign`	프로젝트와 연결된 각 캠페인에 대해 메시지가 배달된 고유 엔드포인트 수입니다. 이 지표에 대한 쿼리 결과는 캠페인을 고유하게 식별하는 문자열인 캠페인 ID(`CampaignId`)별로 그룹화됩니다.
엔드포인트 배달(날짜별 그룹화)	`unique-deliveries-grouped-by-date`	프로젝트와 연결된 모든 캠페인에 대해 지정된 날짜 범위의 각 날짜에 메시지가 배달된 고유 엔드포인트 수입니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.
배달된 메시지(캠페인별 그룹화)	`successful-deliveries-grouped-by-campaign`	프로젝트와 연결된 각 캠페인에 대해 수신자에게 배달된 메시지 수입니다. 이 지표는 한 캠페인에서 전송한 메시지 수에서, 캠페인에서 전송했지만 하드 바운스로 인해 수신자에게 배달할 수 없는 메시지 수를 뺀 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인을 고유하게 식별하는 문자열인 캠페인 ID(`CampaignId`)별로 그룹화됩니다.

지표	kpi-name	설명
푸시 오픈율	`push-open-rate`	프로젝트와 연결된 모든 캠페인에 대해 수신자가 열어본 푸시 알림의 비율입니다. 이 지표는 프로젝트의 모든 캠페인에서 전송한 푸시 알림 중 수신자가 열어본 푸시 알림 수를 모든 캠페인에서 전송한 푸시 알림 중 수신자에게 배달된 푸시 알림 수로 나눈 값으로 계산됩니다.
푸시 오픈율(캠페인별 그룹화)	`push-open-rate-grouped-by-campaign`	프로젝트와 연결된 각 캠페인에 대해 수신자가 열어본 푸시 알림의 비율입니다. 이 지표는 한 캠페인에서 전송한 푸시 알림 중 수신자가 열어본 푸시 알림 수를 해당 캠페인에서 전송한 푸시 알림 중 수신자에게 배달된 푸시 알림 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인을 고유하게 식별하는 문자열인 캠페인 ID(`CampaignId`)별로 그룹화됩니다.

트랜잭션 이메일 메시지의 Amazon Pinpoint 애플리케이션 지표

다음 표에는 Amazon Pinpoint 프로젝트와 연결된 모든 트랜잭션 이메일 메시지의 추세를 모니터링하기 위해 쿼리할 수 있는 표준 애플리케이션 지표 목록과 설명이 나와 있습니다. 이러한 지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 애플리케이션 지표 리소스를 사용하세요. kpi-name 열은 쿼리에서 `kpi-name` 파라미터에 사용할 값을 나타냅니다.

이러한 지표는 캠페인에서 전송한 이메일 메시지에 대한 데이터를 제공하지 않으며, 트랜잭션 이메일 메시지에 대한 데이터만 제공합니다. 하나 이상의 캠페인에서 전송한 메시지에 대한 데이터를 쿼리하려면 캠페인 지표 또는 캠페인의 애플리케이션 지표를 사용합니다.

지표	kpi-name	설명
클릭 수	`txn-emails-clicked`	수신자가 메시지에서 링크를 클릭한 횟수입니다. 수신자 한 명이 메시지에 있는 여러 링크를 클릭하거나 동일한 링크를 두 번 이상 클릭하는 경우에도 각 클릭이 계수됩니다.
클릭(날짜별 그룹화)	`txn-emails-clicked-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자가 메시지에서 링크를 클릭한 횟수입니다. 수신자 한 명이 메시지에 있는 여러 링크를 클릭하거나 동일한 링크를 두 번 이상 클릭하는 경우에도 각 클릭이 계수됩니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.
수신 거부율	`txn-emails-complaint-rate`	수신자가 요청되지 않은 이메일 또는 원치 않는 이메일로 보고한 메시지의 비율입니다. 이 지표는 수신자가 요청되지 않은 이메일 또는 원치 않는 이메일로 보고한 메시지 수를 전송한 메시지 수로 나눈 값으로 계산됩니다.
불만 제기 비율(날짜별 그룹화)	`txn-emails-complaint-rate-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자가 요청되지 않은 이메

지표	kpi-name	설명
		일 또는 원치 않는 이메일로 보고한 메시지의 비율입니다. 이 지표는 수신자가 요청되지 않은 이메일 또는 원치 않는 이메일로 보고한 메시지 수를 지정된 날짜 범위의 각 날짜에 전송한 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.
수신 거부	`txn-emails-with-complaints`	수신자가 요청되지 않은 이메일 또는 원치 않는 이메일로 보고한 메시지의 수입니다.
불만 제기(날짜별 그룹화)	`txn-emails-with-complaints-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자가 요청되지 않은 이메일 또는 원치 않는 이메일로 보고한 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
전달	`txn-emails-delivered`	수신자에게 배달된 메시지의 수입니다. 이 지표는 전송한 메시지 수에서 소프트 바운스나 하드 바운스로 인해 또는 메시지가 거부되어서 배달할 수 없는 메시지 수를 뺀 값으로 계산됩니다. 메시지에 맬웨어가 포함되어 있다고 Amazon Pinpoint가 판단하면 메시지가 거부됩니다. Amazon Pinpoint는 거부된 메시지를 보내려고 시도하지 않습니다.
배달(날짜별 그룹화)	`txn-emails-delivered-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자에게 배달된 메시지의 수입니다. 이 지표는 전송한 메시지 수에서 지정된 날짜 범위의 각 날짜에 소프트 바운스나 하드 바운스로 인해 또는 메시지가 거부되어서 배달할 수 없는 메시지 수를 뺀 값으로 계산됩니다. 메시지에 맬웨어가 포함되어 있다고 Amazon Pinpoint가 판단하면 메시지가 거부됩니다. Amazon Pinpoint는 거부된 메시지를 보내려고 시도하지 않습니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
전송률	`txn-emails-delivery-rate`	수신자에게 배달된 메시지의 비율입니다. 이 지표는 수신자에게 전송되고 배달된 메시지 수를 전송한 메시지 수로 나눈 값으로 계산됩니다.
배달율(날짜별 그룹화)	`txn-emails-delivery-rate-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자에게 배달된 메시지의 비율입니다. 이 지표는 수신자에게 전송되고 배달된 메시지 수를 지정된 날짜 범위의 각 날짜에 전송한 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.
하드 바운스	`txn-emails-hard-bounced`	하드 바운스로 인해 수신자에게 배달될 수 없는 메시지의 수입니다. 하드 바운스는 영구적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 수신자의 이메일 주소가 존재하지 않음).

지표	kpi-name	설명
하드 바운스(날짜별 그룹화)	`txn-emails-hard-bounced-grouped-by-date`	지정된 날짜 범위의 각 날짜에 하드 바운스로 인해 수신자에게 배달될 수 없는 메시지의 수입니다. 하드 바운스는 영구적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 수신자의 이메일 주소가 존재하지 않음). 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.
열기	`txn-emails-opened`	수신자가 열어본 메시지의 수입니다.
열기(날짜별 그룹화)	`txn-emails-opened-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자가 열어본 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.
전송	`txn-emails-sent`	전송한 메시지의 수입니다.
전송(날짜별 그룹화)	`txn-emails-sent-grouped-by-date`	지정된 날짜 범위의 각 날짜에 전송한 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
소프트 바운스	`txn-emails-soft-bounced`	소프트 바운스로 인해 수신자에게 배달될 수 없는 메시지의 수입니다. 소프트 바운스는 일시적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 수신자의 받은 편지함이 가득 찼거나 수신 서버를 일시적으로 사용할 수 없는 경우).
소프트 바운스(날짜별 그룹화)	`txn-emails-soft-bounced-grouped-by-date`	지정된 날짜 범위의 각 날짜에 소프트 바운스로 인해 수신자에게 배달될 수 없는 메시지의 수입니다. 소프트 바운스는 일시적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 수신자의 받은 편지함이 가득 찼거나 수신 서버를 일시적으로 사용할 수 없는 경우). 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
고유 사용자 클릭 이벤트	`txn-emails-unique-clicks`	메시지에 있는 링크를 클릭한 고유 수신자(엔드포인트)의 수입니다. 클릭 지표와 달리 이 지표는 발생한 클릭 이벤트 수가 아니라 링크를 클릭한 고유 수신자 수를 보고합니다. 예를 들어 수신자 한 명이 동일한 메시지에 있는 여러 링크를 클릭하거나 동일한 링크를 두 번 이상 클릭하는 경우 이 지표는 해당 수신자에 대한 유일한 클릭 이벤트를 보고합니다.
고유 사용자 클릭 이벤트(날짜별 그룹화)	`txn-emails-unique-clicks-grouped-by-date`	지정된 날짜 범위의 각 날짜에 메시지에 있는 링크를 클릭한 고유 수신자(엔드포인트)의 수입니다. 클릭(날짜별 그룹화) 지표와 달리 이 지표는 발생한 클릭 이벤트 수가 아니라 링크를 클릭한 고유 수신자 수를 보고합니다. 예를 들어 수신자 한 명이 동일한 메시지에 있는 여러 링크를 클릭하거나 동일한 링크를 두 번 이상 클릭하는 경우 이 지표는 해당 수신자에 대한 유일한 클릭 이벤트를 보고합니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
고유 사용자 열기 이벤트	`txn-emails-unique-opens`	메시지를 열어본 고유 수신자(엔드포인트)의 수입니다. 열기 지표와 달리 이 지표는 발생한 열기 이벤트 수가 아니라 메시지를 열어본 고유 수신자 수를 보고합니다. 예를 들어 수신자 한 명이 동일한 메시지를 여러 번 여는 경우 이 지표는 해당 수신자에 대한 유일한 열기 이벤트를 보고합니다.
고유 사용자 열기 이벤트(날짜별 그룹화)	`txn-emails-unique-opens-grouped-by-date`	지정된 날짜 범위의 각 날짜에 메시지를 열어본 고유 수신자(엔드포인트)의 수입니다. 열기(날짜별 그룹화) 지표와 달리 이 지표는 발생한 열기 이벤트 수가 아니라 메시지를 열어본 고유 수신자 수를 보고합니다. 예를 들어 수신자 한 명이 동일한 메시지를 여러 번 여는 경우 이 지표는 해당 수신자에 대한 유일한 열기 이벤트를 보고합니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

트랜잭션 SMS 메시지의 Amazon Pinpoint 애플리케이션 지표

다음 표에는 Amazon Pinpoint 프로젝트와 연결된 모든 트랜잭션 SMS 메시지의 추세를 모니터링하기 위해 쿼리할 수 있는 표준 애플리케이션 지표 목록과 설명이 나와 있습니다. 이러한 지표에 대한 데이

터를 쿼리하려면 Amazon Pinpoint API의 애플리케이션 지표 리소스를 사용하세요. kpi-name 열은 쿼리에서 `kpi-name` 파라미터에 사용할 값을 나타냅니다.

이러한 지표는 캠페인에서 전송한 SMS 메시지에 대한 데이터를 제공하지 않으며, 트랜잭션 SMS 메시지에 대한 데이터만 제공합니다. 하나 이상의 캠페인에서 전송한 메시지에 대한 데이터를 쿼리하려면 캠페인 지표 또는 캠페인의 애플리케이션 지표를 사용합니다.

지표	kpi-name	설명
메시지당 평균 가격(국가별 그룹화)	`txn-sms-average-price-grouped-by-country`	메시지가 전송된 각 국가 또는 리전에 대해 각 메시지 전송에 소요되는 평균 비용입니다. 가격은 미국 센트의 1/1000로 표시됩니다. 예를 들어 이 속성의 값이 645인 경우, 메시지 전송에 대해 0.645¢가 청구된 것입니다(645 / 1000 = 0.645¢ = $0.00645). 이 지표는 각 국가 또는 지역의 받는 사람에게 전송된 모든 메시지의 총 비용을 해당 국가 및 지역의 받는 사람에게 보낸 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 ISO 3166-1 alpha-2 형식으로 국가별 또는 리전별로 그룹화됩니다.
메시지 파트당 평균 가격(국가별 그룹화)	`txn-sms-average-price-by-parts-grouped-by-country`	메시지 파트가 전송된 각 국가 또는 리전에 대해 각 메시지 전송에 소요되는 평균 비용입니다. 메시지 파트는 SMS 메시지의 일부입니다. 가격은 미국 센트의 1/1000로 표시됩니다. 예를 들어 이 속성의 값

지표	kpi-name	설명
		이 645인 경우, 메시지 전송에 대해 0.645¢가 청구된 것입니다(645 / 1000 = 0.645¢ = $0.00645). 이 지표는 각 국가 또는 지역의 받는 사람에게 전송된 모든 메시지 파트의 총 비용을 해당 국가 및 지역의 받는 사람에게 보낸 메시지 파트 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 ISO 3166-1 alpha-2 형식으로 국가별 또는 리전별로 그룹화됩니다.
전달	`txn-sms-delivered`	수신자에게 배달된 메시지의 수입니다.
배달(국가별 그룹화)	`txn-sms-delivered-grouped-by-country`	메시지가 전송된 각 국가 또는 리전에 대해 수신자에게 배달된 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 ISO 3166-1 alpha-2 형식으로 국가별 또는 리전별로 그룹화됩니다.
배달(날짜별 그룹화)	`txn-sms-delivered-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자에게 배달된 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
배달 오류	`txn-sms-error-distribution`	발생한 각 유형의 오류에 대해 메시지 배달을 시도하는 동안 오류가 발생한 횟수입니다. 이 지표에 대한 쿼리 결과는 발생한 각 유형의 오류에 대해 오류 코드별로 그룹화됩니다.
전송률	`txn-sms-delivery-rate`	수신자에게 배달된 메시지의 비율입니다. 이 지표는 수신자에게 전송되고 배달된 메시지 수를 전송한 메시지 수로 나눈 값으로 계산됩니다.
배달율(날짜별 그룹화)	`txn-sms-delivery-rate-grouped-by-date`	지정된 날짜 범위의 각 날짜에 수신자에게 배달된 메시지의 비율입니다. 이 지표는 수신자에게 전송되고 배달된 메시지 수를 지정된 날짜 범위의 각 날짜에 전송한 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
배달된 메시지 파트	`txn-sms-delivered-by-parts`	배달된 메시지 파트의 수입니다. 메시지 파트는 SMS 메시지의 일부입니다. SMS 메시지에 SMS 프로토콜이 허용하는 것보다 많은 문자가 포함된 경우, Amazon Pinpoint는 메시지를 수신자에게 전송하는 데 필요한 개수만큼 메시지 파트로 분할합니다.
배달된 메시지 파트(국가별 그룹화)	`txn-sms-delivered-by-parts-grouped-by-country`	메시지가 전송된 각 국가 또는 리전에 대해 배달된 메시지 파트의 수입니다. 메시지 파트는 SMS 메시지의 일부입니다. 이 지표에 대한 쿼리 결과는 ISO 3166-1 alpha-2 형식으로 국가별 또는 리전별로 그룹화됩니다.
전송된 메시지 파트	`txn-sms-sent-by-parts`	전송된 메시지 파트의 수입니다. 메시지 파트는 SMS 메시지의 일부입니다. SMS 메시지에 SMS 프로토콜이 허용하는 것보다 많은 문자가 포함된 경우, Amazon Pinpoint는 메시지를 수신자에게 전송하는 데 필요한 개수만큼 메시지 파트로 분할합니다.

지표	kpi-name	설명
전송된 메시지 파트(국가별 그룹화)	`txn-sms-sent-by-parts-grouped-by-country`	메시지 파트가 전송된 각 국가 또는 리전에 대해 수신자에게 전송된 메시지의 수입니다. 메시지 파트는 SMS 메시지의 일부입니다. 이 지표에 대한 쿼리 결과는 ISO 3166-1 alpha-2 형식으로 국가별 또는 리전별로 그룹화됩니다.
전송한 메시지	`txn-sms-sent`	전송한 메시지의 수입니다.
전송된 메시지(국가별 그룹화)	`txn-sms-sent-grouped-by-country`	메시지가 전송된 각 국가 또는 리전에 대해 수신자에게 전송된 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 ISO 3166-1 alpha-2 형식으로 국가별 또는 리전별로 그룹화됩니다.
전송된 메시지(날짜별 그룹화)	`txn-sms-sent-grouped-by-date`	지정된 날짜 범위의 각 날짜에 전송한 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
총 가격(국가별 그룹화)	`txn-sms-total-price-grouped-by-country`	메시지가 전송된 각 국가 또는 리전에 대해 메시지 전송에 소요되는 총 비용입니다. 가격은 미국 센트의 1/1000로 표시됩니다. 예를 들어 이 속성의 값이 645인 경우, 메시지 전송에 대해 0.645¢가 청구된 것입니다(645 / 1000 = 0.645¢ = $0.00645). 이 지표에 대한 쿼리 결과는 ISO 3166-1 alpha-2 형식으로 국가별 또는 리전별로 그룹화됩니다.

Amazon Pinpoint 캠페인 지표

다음 표에는 개별 캠페인의 성과를 평가하기 위해 쿼리할 수 있는 표준 캠페인 지표 목록과 설명이 나와 있습니다. 이러한 지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 캠페인 지표 리소스를 사용하세요. 표에 나와 있는 kpi-name 열은 쿼리에서 `kpi-name` 파라미터에 사용할 값을 나타냅니다.

지표	kpi-name	설명
반송률	`hard-bounce-rate`	모든 캠페인 실행에 대해 수신자에게 배달할 수 없는 이메일 메시지의 비율입니다. 이 지표는 하드 바운스만 측정합니다. 하드 바운스란 수신자의 이메일 주소에 영구적인 문제가 있어 메시지를 배달할 수 없는 메시지입니다. 이 지표는 모든 캠페인 실행에서 전송한 이메일 메시지 중 반송된 이메일 메시지 수를 모든

지표	kpi-name	설명
		캠페인 실행에서 전송한 이메일 메시지 수로 나눈 값으로 계산됩니다.
반송률(캠페인 실행별 그룹화)	`hard-bounce-rate-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자에게 배달할 수 없는 이메일 메시지의 비율입니다. 이 지표는 하드 바운스만 측정합니다. 하드 바운스란 수신자의 이메일 주소에 영구적인 문제가 있어 메시지를 배달할 수 없는 메시지입니다. 이 지표는 한 캠페인 실행에서 전송한 이메일 메시지 중 반송된 이메일 메시지 수를 해당 캠페인 실행에서 전송한 이메일 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`) 별로 그룹화됩니다.
전송률	`successful-delivery-rate`	모든 캠페인 실행에 대해 수신자에게 배달된 메시지의 비율입니다. 이 지표는 모든 캠페인 실행에서 전송한 메시지 중 수신자에게 배달된 메시지 수를 모든 캠페인 실행에서 전송한 메시지 수로 나눈 값으로 계산됩니다.

지표	kpi-name	설명
배달율(캠페인 실행별 그룹화)	`successful-delivery-rate-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자에게 배달된 메시지의 비율입니다. 이 지표는 한 캠페인 실행에서 전송한 메시지 중 수신자에게 배달된 메시지 수를 해당 캠페인 실행에서 전송한 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`) 별로 그룹화됩니다.
배달율(날짜별 그룹화)	`successful-delivery-rate-grouped-by-date`	모든 캠페인 실행에 대해 지정된 날짜 범위의 각 날짜에 수신자에게 배달된 메시지의 비율입니다. 이 지표는 모든 캠페인 실행에서 전송한 메시지 중 수신자에게 배달된 메시지 수를 지정된 날짜 범위의 각 날짜에 모든 캠페인에서 전송한 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
이메일 오픈율	`email-open-rate`	모든 캠페인 실행에 대해 수신자가 열어본 이메일 메시지의 비율입니다. 이 지표는 모든 캠페인 실행에서 전송한 이메일 메시지 중 수신자가 열어본 이메일 메시지 수를 모든 캠페인 실행에서 전송한 이메일 메시지 중 수신자에게 배달된 이메일 메시지 수로 나눈 값으로 계산됩니다.
이메일 오픈율(캠페인 실행별 그룹화)	`email-open-rate-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자가 열어본 이메일 메시지의 비율입니다. 이 지표는 한 캠페인 실행에서 전송한 이메일 메시지 중 수신자가 열어본 이메일 메시지 수를 해당 캠페인 실행에서 전송한 이메일 메시지 중 수신자에게 배달된 이메일 메시지 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`)별로 그룹화됩니다.

지표	kpi-name	설명
열어본 이메일(캠페인 실행별 그룹화)	`direct-email-opens-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자가 열어본 이메일 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`)별로 그룹화됩니다.
엔드포인트 전달	`unique-deliveries`	모든 캠페인 실행에 대해 메시지가 배달된 고유 엔드포인트 수입니다.
엔드포인트 배달(캠페인 실행별 그룹화)	`unique-deliveries-grouped-by-campaign-activity`	각 캠페인 실행에 대해 메시지가 배달된 고유 엔드포인트 수입니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`)별로 그룹화됩니다.
엔드포인트 배달(날짜별 그룹화)	`unique-deliveries-grouped-by-date`	모든 캠페인 실행에 대해 지정된 날짜 범위의 각 날짜에 메시지가 배달된 고유 엔드포인트 수입니다. 이 지표에 대한 쿼리 결과는 확장된 ISO 8601 형식으로 날짜별로 그룹화됩니다.

지표	kpi-name	설명
클릭한 링크(캠페인 실행별 그룹화)	`clicks-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자가 이메일 메시지에서 링크를 클릭한 횟수입니다. 수신자 한 명이 메시지에 있는 여러 링크를 클릭하거나 동일한 링크를 두 번 이상 클릭하는 경우에도 각 클릭이 계수됩니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`)별로 그룹화됩니다.
배달된 메시지(캠페인 실행별 그룹화)	`successful-deliveries-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자에게 배달된 메시지의 수입니다. 이 지표는 한 캠페인 실행에서 전송한 메시지 수에서, 하드 바운스로 인해 수신자에게 배달할 수 없는 메시지 수를 뺀 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`)별로 그룹화됩니다.

지표	kpi-name	설명
전송한 메시지(캠페인 실행별 그룹화)	`attempted-deliveries-grouped-by-campaign-activity`	각 캠페인 실행에 대해 전송한 메시지의 수입니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`) 별로 그룹화됩니다.
푸시 오픈율	`push-open-rate`	모든 캠페인 실행에 대해 수신자가 열어본 푸시 알림의 비율입니다. 이 지표는 모든 캠페인 실행에서 전송한 푸시 알림 중 수신자가 열어본 푸시 알림 수를 모든 캠페인 실행에서 전송한 푸시 알림 중 수신자에게 배달된 푸시 알림 수로 나눈 값으로 계산됩니다.

지표	kpi-name	설명
푸시 오픈율(캠페인 실행별 그룹화)	`push-open-rate-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자가 열어본 푸시 알림의 비율입니다. 이 지표는 한 캠페인 실행에서 전송한 푸시 알림 중 수신자가 열어본 푸시 알림 수를 해당 캠페인 실행에서 전송한 푸시 알림 중 수신자에게 배달된 푸시 알림 수로 나눈 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`) 별로 그룹화됩니다.
열어본 총 푸시(캠페인 실행별 그룹화)	`direct-push-opens-grouped-by-campaign-activity`	각 캠페인 실행에 대해 수신자가 열어본 푸시 알림의 수입니다. 이 지표에 대한 쿼리 결과는 캠페인 실행을 고유하게 식별하는 문자열인 캠페인 활동 ID(`CampaignActivityId`) 별로 그룹화됩니다.
총 SMS 지출	`sms-spend`	모든 캠페인에 대해 SMS 전송에 지출한 총 금액(밀리센트 단위).

Amazon Pinpoint 여정 참여 지표

다음 표에서는 Amazon Pinpoint 여정에서 보낸 모든 이메일 메시지의 추세를 모니터링하기 위해 쿼리할 수 있는 표준 여정 참여 지표를 나열하고 설명합니다. 이러한 지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 여정 참여 지표 리소스를 사용하세요. kpi-name 열은 쿼리에서 `kpi-name` 파라미터에 사용할 값을 나타냅니다.

지표	kpi-name	설명
클릭 수	`journey-emails-clicked`	참가자가 메시지에서 링크를 클릭한 횟수입니다. 참가자 한 명이 메시지에 있는 여러 링크를 클릭하거나 동일한 링크를 두 번 이상 클릭하는 경우에도 각 클릭이 계수됩니다.
클릭(활동별 그룹화)	`emails-clicked-grouped-by-journey-activity`	여정의 각 활동에 대해 참가자가 메시지의 링크를 클릭한 횟수입니다. 참가자 한 명이 메시지에 있는 여러 링크를 클릭하거나 동일한 링크를 두 번 이상 클릭하는 경우에도 각 클릭이 계수됩니다. 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.
수신 거부	`journey-emails-complained`	참가자가 요청되지 않은 이메일 또는 원치 않는 이메일로 보고한 메시지의 수입니다.
불만 사항(활동별 그룹화)	`emails-complained-grouped-by-journey-activity`	여정의 각 활동에 대해 참가자가 요청되지 않은 이메일 또는 원치 않는 이메일로 보고한 메시지의 수입니다.

지표	kpi-name	설명
		이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.
전달	`journey-emails-delivered`	참가자에게 배달된 메시지의 수입니다. 이 지표는 전송한 메시지 수에서 소프트 바운스나 하드 바운스로 인해 또는 메시지가 거부되어서 배달할 수 없는 메시지 수를 뺀 값으로 계산됩니다.
배달(활동별 그룹화)	`emails-delivered-grouped-by-journey-activity`	여정의 각 활동에 대해 참가자에게 배달된 메시지 수입니다. 이 지표는 전송된 메시지 수에서 여정의 각 활동에 대해 소프트 바운스나 하드 바운스로 인해 또는 메시지가 거부되어서 배달할 수 없는 메시지 수를 뺀 값으로 계산됩니다. 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.

지표	kpi-name	설명
하드 바운스	`journey-emails-hardbounced`	하드 바운스로 인해 참가자에게 배달될 수 없는 메시지의 수입니다. 하드 바운스는 영구적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 참가자의 이메일 주소가 존재하지 않음).
하드 바운스(활동별 그룹화)	`emails-hardbounced-grouped-by-journey-activity`	여정의 각 활동에 대해 하드 바운스로 인해 참가자에게 배달할 수 없는 메시지의 수입니다. 하드 바운스는 영구적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 참가자의 이메일 주소가 존재하지 않음). 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.
열기	`journey-emails-opened`	참가자가 열어본 메시지의 수입니다.
열기(활동별 그룹화)	`emails-opened-grouped-by-journey-activity`	여정의 각 활동에 대해 참가자가 열어본 메시지 수입니다. 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.

지표	kpi-name	설명
거부	`journey-emails-rejected`	참가자가 거부하여 참가자에게 전송되지 않은 메시지 수입니다. 메시지에 맬웨어가 포함되어 있다고 Amazon Pinpoint가 판단하면 메시지가 거부됩니다. Amazon Pinpoint는 거부된 메시지를 보내려고 시도하지 않습니다.
거부(활동별 그룹화)	`emails-rejected-grouped-by-journey-activity`	여정의 각 활동에 대해 거부되었기 때문에 참가자에게 전송되지 않은 메시지 수입니다. 메시지에 맬웨어가 포함되어 있다고 Amazon Pinpoint가 판단하면 메시지가 거부됩니다. Amazon Pinpoint는 거부된 메시지를 보내려고 시도하지 않습니다. 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.
전송	`journey-emails-sent`	전송한 메시지의 수입니다.
전송(활동별 그룹화)	`emails-sent-grouped-by-journey-activity`	여정의 각 활동에 대해 전송된 메시지 수입니다. 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.

지표	kpi-name	설명
소프트 바운스	`journey-emails-softbounced`	소프트 바운스로 인해 참가자에게 배달할 수 없는 메시지의 수입니다. 소프트 바운스는 일시적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 참가자의 받은 편지함이 가득 찼거나 수신 서버를 일시적으로 사용할 수 없는 경우).
소프트 바운스(활동별 그룹화)	`emails-softbounced-grouped-by-journey-activity`	여정의 각 활동에 대해 소프트 바운스로 인해 참가자에게 배달할 수 없는 메시지의 수입니다. 소프트 바운스는 일시적인 문제로 인해 메시지가 배달되지 못할 때 발생합니다(예: 참가자의 받은 편지함이 가득 찼거나 수신 서버를 일시적으로 사용할 수 없는 경우). 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.
구독 해지	`journey-emails-unsubscribed`	참가자가 메시지에서 구독 취소 링크를 클릭한 횟수입니다. 참가자 한 명이 동일한 구독 취소 링크를 두 번 이상 클릭하는 경우에도 각 클릭이 계수됩니다.

지표	kpi-name	설명
구독 취소(활동별 그룹화)	`emails-unsubscribed-grouped-by-journey-activity`	여정의 각 활동에 대해 참가자가 메시지에서 구독 취소 링크를 클릭한 횟수입니다. 참가자 한 명이 동일한 구독 취소 링크를 두 번 이상 클릭하는 경우에도 각 클릭이 계수됩니다. 이 지표에 대한 쿼리 결과는 활동을 고유하게 식별하는 문자열인 활동 ID(`JourneyActivityId`)별로 그룹화됩니다.

Amazon Pinpoint 여정 실행 지표

다음 표에서는 Amazon Pinpoint 여정 참가자 상태를 평가하기 위해 쿼리할 수 있는 표준 실행 지표를 나열하고 설명합니다. 이러한 지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 여정 실행 지표 리소스를 사용하세요. 이 표의 필드 열은 각 지표에 대한 쿼리 결과에 나타나는 필드 이름을 식별합니다.

지표	필드	설명
활성 참가자	`ENDPOINT_ACTIVE`	여정 활동을 적극적으로 진행하고 있는 참가자의 수입니다. 이 지표는 여정에 참가한 참가자 수에서 여정을 시작한 참가자 수와 여정에서 제거된 참가자 수를 뺀 값으로 계산됩니다.
참가자 취소	`CANCELLED`	여정이 취소되어 여행을 완료하지 않은 참가자 수입니다.
이탈한 참가자	`ENDPOINT_LEFT`	여정을 떠난 참가자의 수입니다.

지표	필드	설명
시작한 참가자	ENDPOINT_ENTERED	여정을 시작한 참가자의 수입니다.
참가자 예외(재참가 제한)	REENTRY_CAP_EXCEEDED	한 명의 참가자가 여정에 다시 참가할 수 있는 최대 횟수를 초과했기 때문에 여정을 완료하지 않은 참가자 수입니다.
참가자 예외, 거부	ACTIVE_ENDPOINT_REJECTED	이미 이 여정의 활성 참가자이기 때문에 여정을 시작할 수 없는 참가자의 수입니다. 참가자가 여정을 시작한 이후, 참가자를 세그먼트(세그먼트 기준에 따라) 또는 여정(활동 조건에 따라)에 포함하는지 여부에 영향을 주는 방식으로 참가자의 엔드포인트 정의를 업데이트하면 참가자는 거부됩니다.

Amazon Pinpoint 여정 활동 실행 지표

다음 표에서는 Amazon Pinpoint 여정에 대한 각 개별 활동 유형의 참가자 상태를 평가하기 위해 쿼리할 수 있는 표준 실행 지표를 나열하고 설명합니다. 이러한 지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 여정 활동 실행 지표 리소스를 사용하세요. 이 표의 지표 열에는 각 활동 유형에 대한 쿼리 결과에 나타나는 필드가 나열됩니다. 또한 각 필드에 대한 간략한 설명을 제공합니다.

활동 유형	Metrics
예/아니요 분할(CONDITIONAL_SPLIT)	지표는 다음과 같습니다. • Branch_FALSE - 활동 조건을 충족하지 못하고 “아니요” 경로에서 활동을 진행한 참가자의 수입니다.

활동 유형	Metrics
	• Branch_TRUE - 활동 조건을 충족하고 “예” 경로에서 활동을 진행한 참가자의 수입니다. 각 경로의 활동에 대해 추가 지표를 사용할 수 있습니다. 이러한 지표에 대한 자세한 내용은 이 테이블에서 해당 활동 유형에 대한 행을 참조하십시오.
홀드아웃(HOLDOUT)	지표는 다음과 같습니다. • HOLDOUT - 활동에 대한 거부 비율의 일부로 여정에서 제거된 참가자 수입니다. • PASSED - 여정에서 다음 활동으로 진행한 참가자 수입니다.

활동 유형	Metrics
이메일(MESSAGE)	지표는 다음과 같습니다. • DAILY_CAP_EXCEEDED - 단일 참가자가 24시간 동안 받을 수 있는 최대 메시지 수를 초과했기 때문에 전송되지 않은 메시지 수입니다. • FAILURE_PERMANENT - 영구적인 문제로 인해 전송되지 않은 메시지 수입니다. • QUIET_TIME - 참가자의 표준 시간대에서 대기 시간에 배달되었기 때문에 전송되지 않은 메시지 수입니다. • SERVICE_FAILURE - Amazon Pinpoint의 문제로 인해 전송되지 않은 메시지 수입니다. • SUCCESS - 참가자에게 성공적으로 배달된 메시지 수입니다. • THROTTLED - 메시지를 전송할 경우 Amazon Pinpoint 계정의 전송 할당량을 초과하기 때문에 전송되지 않은 메시지의 수입니다. • TRANSIENT_FAILURE - 일시적인 문제로 인해 전송되지 않은 메시지 수입니다. • UNKNOWN - 알 수 없는 문제로 인해 전송되지 않은 메시지 수입니다.

활동 유형	Metrics
다변량 분할(MULTI_CONDITIONAL_SPLIT)	활동의 각 경로에 대해, 해당 경로에서 활동을 진행한 참가자 수입니다. 이 지표에 대한 쿼리 결과는 경로별로 그룹화됩니다. Branch_#에서 # 기호는 경로의 숫자 식별자입니다. 예를 들어 활동의 첫 번째 경로는 Branch_1입니다. 각 경로의 활동에 대해 추가 지표를 사용할 수 있습니다. 이러한 지표에 대한 자세한 내용은 이 테이블에서 해당 활동 유형에 대한 행을 참조하십시오.
임의 분할(RANDOM_SPLIT)	활동의 각 경로에 대해, 해당 경로에서 활동을 진행한 참가자 수입니다. 이 지표에 대한 쿼리 결과는 경로별로 그룹화됩니다. Branch_#에서 # 기호는 경로의 숫자 식별자입니다. 예를 들어 활동의 첫 번째 경로는 Branch_1입니다. 각 경로의 활동에 대해 추가 지표를 사용할 수 있습니다. 이러한 지표에 대한 자세한 내용은 이 테이블에서 해당 활동 유형에 대한 행을 참조하십시오.

활동 유형	Metrics
대기(WAIT)	지표는 다음과 같습니다. • WAIT_FINISHED - 지정된 시간 동안 대기한 참가자 수입니다. • WAIT_SKIPPED - 지정된 시간 동안 대기하지 않은 참가자 수입니다. 그 이유는 일반적으로 활동의 예약된 종료 시간 이후에 활동이나 여정을 시작했기 때문입니다. • WAIT_STARTED - 대기를 시작했지만 대기를 건너뛰거나 지정된 시간 동안 대기하지 않은 참가자 수입니다.

활동 유형	Metrics
고객 센터(CONTACT_CENTER)	지표는 다음과 같습니다. • CALL_QUEUED - 고객 연결에 전화를 걸고 대기한 참가자 수입니다. 다시 걸기 시도 횟수도 포함됩니다. • CONTINUE_WAITING - 전화 걸기 시도가 성공할 때까지 계속 기다리는 참가자 수입니다. • DIAL_FAILURE - 일일 시도에 실패한 참가자 수입니다. • DROPPED - 전송 시간에 이전 여정 활동에 정의된 조건을 더 이상 충족하지 않는 참가자 수입니다. • TIMEOUT - 여러 번의 다이얼 시도 후 Connect Customer 처리 코드를 받지 못한 참가자 수입니다. • WAIT_FINISHED - 지정된 시간 동안 대기한 참가자 수입니다. • WAIT_FOR_QUIET_HOURS - 채널에 전송하기 위해 전송 중단 시간이 끝날 때까지 기다리는 참가자 수입니다. • WAIT_STARTED - 대기를 시작했지만 대기를 건너뛰거나 지정된 시간 동안 대기하지 않은 참가자 수입니다.

Amazon Pinpoint 여정 및 캠페인 실행 지표

Amazon Pinpoint 여정 또는 캠페인에 대한 각 개별 활동 유형의 참가자 상태를 평가하기 위해 표준 실행 지표를 쿼리할 수 있습니다. 이러한 지표에 대한 데이터를 쿼리하려면 Amazon Pinpoint API의 여정 실행 활동 실행 지표 또는 캠페인 리소스를 사용하세요. 아래 표에는 각 활동 유형에 대한 쿼리 결과에 나타나는 필드가 나열됩니다.

지표 이름	여정, 캠페인 또는 둘 다에 적용	설명
ENDPOINT_PRODUCED	둘 다	필터링 전에 세그먼트 또는 이벤트에서 처음 생성된 엔드포인트 수입니다.
ENDPOINTS_FROM_USER	둘 다	고객이 사용자 ID 전용 세그먼트를 보유한 경우, 이러한 사용자의 모든 엔드포인트가 추가됩니다. 이 지표는 이러한 방식으로 추가된 엔드포인트 수를 측정합니다.
ENDPOINT_OPT_OUT	둘 다	엔드포인트가 옵트아웃되었으며 캠페인 또는 여정에 참여하지 않았습니다.
ENDPOINT_INACTIVE	둘 다	엔드포인트가 비활성화되었으며 캠페인 또는 여정에 참여하지 않았습니다.
FILTERED_OUT_BY_SEGMENT	둘 다	엔드포인트가 세그먼트 필터와 일치하지 않았으며 캠페인 또는 여정에 참여하지 않았습니다.
ENDPOINT_MISSING_ADDRESS	둘 다	엔드포인트에서 주소가 누락되었으며 캠페인 또는 여정에 참여하지 않았습니다.
ENDPOINT_MISSING_CHANNEL	둘 다	엔드포인트에서 채널이 누락되었으며 캠페인 또는 여정에 참여하지 않았습니다.
ENDPOINT_MISSING_TIMEZONE	둘 다	엔드포인트에 시간대 값이 누락되었으며 필터링되었습니다. 이는 시간대 값이 필요한 경우에만 발생합니다.

지표 이름	여정, 캠페인 또는 둘 다에 적용	설명
ENDPOINT_TIMEZONE_MISMATCH	둘 다	엔드포인트가 그 당시 실행에 포함되지 않은 시간대에 있었습니다.
ENDPOINT_CHANNEL_MISMATCH	Campaigns	캠페인에 이 엔드포인트의 채널 유형에 맞게 구성된 메시지가 없습니다.
DUPLICATE_ENDPOINT	둘 다	중복된 엔드포인트가 발견되었으며 중복 제거되었습니다.
DUPLICATE_USER	둘 다	중복된 사용자가 발견되었으며 사용자 ID 전용 세그먼트에서 중복 제거되었습니다. 사용자 ID가 동일한 경우 지표 1이 방출됩니다.
PAUSED	여정	여정이 일시 중지되었으므로 실행에서 삭제되었습니다.
ENDED	여정	여정이 종료되었으므로 실행에서 삭제되었습니다.
TREATMENT_HOLDOUT	Campaigns	이는 집단이 현재 처리와 일치하지 않는 엔드포인트에 대해 A/B 캠페인에서 방출됩니다. 예를 들어, 50/50 A/B 분할에서는 엔드포인트의 50%가 각 처리에 대해 이 지표를 방출합니다.
ENDPOINT_ESTIMATED_TIMEZONE	여정	시간대 추정을 통해 엔드포인트의 시간대를 추정할 수 있었습니다.

캠페인에 대한 Amazon Pinpoint 분석 데이터 쿼리

Amazon Pinpoint 콘솔의 분석 페이지를 사용하는 것 외에도 Amazon Pinpoint Analytics API를 사용하여 분석 데이터를 쿼리하여 캠페인의 제공 및 참여 동향에 대한 통찰력을 제공하는 표준 지표의 하위 집합을 제공할 수 있습니다.

이러한 각 지표는 측정 가능한 값으로, KPI(핵심 성과 지표)라고도 하며, 하나 이상의 캠페인 성과를 모니터링하고 평가하는 데 도움이 됩니다. 예를 들어, 지표를 사용하여 캠페인 메시지를 전송한 엔드포인트의 수 또는 의도한 엔드포인트에 배달된 메시지의 수를 확인할 수 있습니다.

Amazon Pinpoint는 모든 캠페인에 대해 이 데이터를 자동으로 수집하고 집계합니다. 데이터는 90일 동안 저장됩니다. Mobile SDK를 사용하여 AWS 모바일 앱을 Amazon Pinpoint와 통합한 경우 Amazon Pinpoint는 수신자가 열어본 푸시 알림의 백분율과 같은 추가 지표를 포함하도록이 지원을 확장합니다. 모바일 앱 통합에 대한 자세한 내용은 Amazon Pinpoint를 애플리케이션과 통합 단원을 참조하십시오.

Amazon Pinpoint Analytics API를 사용하여 데이터를 쿼리하는 경우 쿼리의 범위, 데이터, 그룹화 및 필터를 정의하는 다양한 옵션을 선택할 수 있습니다. 이 작업은 적용할 날짜 기반 필터와 함께 쿼리할 프로젝트, 캠페인 및 지표를 지정하는 파라미터를 사용하여 수행할 수 있습니다.

이 항목에서는 이러한 옵션을 선택하고 하나 이상의 캠페인에 대한 데이터를 쿼리하는 방법에 대해 설명하고 예제를 제공합니다.

사전 조건

하나 이상의 캠페인에 대한 분석 데이터를 쿼리하기 전에 쿼리를 정의하는 데 사용할 다음 정보를 수집하는 것이 좋습니다.

- 프로젝트 ID - 캠페인과 연관된 프로젝트의 고유 식별자입니다. Amazon Pinpoint API에서 이 값은 `application-id` 속성에 저장됩니다. Amazon Pinpoint 콘솔에서는 이 값이 모든 프로젝트 페이지에 프로젝트 ID로 표시됩니다.
- 캠페인 ID - 하나의 캠페인에 대해서만 데이터를 쿼리하려는 경우 캠페인의 고유 식별자입니다. Amazon Pinpoint API에서 이 값은 `campaign-id` 속성에 저장됩니다. 이 값은 콘솔에 표시되지 않습니다.
- 날짜 범위 - 날짜 범위 또는 데이터를 쿼리할 일자 범위의 최초 및 최종 날짜 및 시간입니다. 날짜 범위는 포괄적이며 31일 이하로 제한되어야 합니다. 또한 현재 날짜로부터 90일 이내에 시작해야 합니다. 날짜 범위를 지정하지 않으면 Amazon Pinpoint에서 이전 31일의 데이터를 자동으로 쿼리합니다.
- 지표 유형 - 쿼리할 지표의 유형입니다. 애플리케이션 지표와 캠페인 지표의 두 가지 유형이 있습니다. 애플리케이션 지표는 프로젝트와 연관된 모든 캠페인(애플리케이션이라고도 함)에 대한 데이터를 제공합니다. 캠페인 지표는 하나의 캠페인에 대한 데이터만 제공합니다.

- 지표 - 쿼리할 지표의 이름입니다. 보다 구체적으로 말하면 지표의 값입니다. 지원되는 지표의 전체 목록과 각 지표에 대한 kpi-name 값은 프로젝트, 캠페인 및 여정에 대한 표준 지표 단원을 참조하십시오.

이는 또한 관련 필드로 데이터를 그룹화할지 여부를 결정하는 데 도움이 됩니다. 이 경우 자동으로 데이터를 그룹화하도록 설계된 지표를 선택하여 분석 및 보고를 단순화할 수 있습니다. 예를 들어 Amazon Pinpoint에서는 캠페인 수신자에게 배달된 메시지의 비율을 보고하는 몇 가지 표준 지표를 제공합니다. 이러한 지표 중 하나는 자동으로 데이터를 날짜별(successful-delivery-rate-grouped-by-date)로 그룹화합니다. 다른 지표는 자동으로 데이터를 캠페인 실행별(successful-delivery-rate-grouped-by-campaign-activity)로 그룹화합니다. 세 번째 지표는 단일 값, 즉 모든 캠페인 실행에서 수신자에게 배달된 메시지 비율을 반환합니다(successful-delivery-rate).

원하는 방식으로 데이터를 그룹화하는 표준 지표를 찾을 수 없는 경우 원하는 데이터를 반환하는 일련의 쿼리를 개발할 수 있습니다. 그런 다음 쿼리 결과를 수동으로 분류하거나 디자인 중인 사용자 지정 그룹으로 결합할 수 있습니다.

마지막으로 쿼리할 데이터에 액세스할 수 있는 권한이 있는지 확인하는 것이 중요합니다. 자세한 내용은 Amazon Pinpoint 분석 데이터 쿼리를 위한 IAM 정책 단원을 참조하십시오.

하나의 캠페인에 대한 Amazon Pinpoint 데이터 쿼리

하나의 캠페인에 대한 데이터를 쿼리하려면 캠페인 지표 API를 사용하여 다음 필수 파라미터에 대한 값을 지정합니다.

- application-id - 캠페인과 연관된 프로젝트의 고유 식별자인 프로젝트 ID입니다. Amazon Pinpoint에서 프로젝트 및 애플리케이션이라는 용어는 동일한 의미를 갖습니다.
- campaign-id - 캠페인의 고유 식별자입니다.
- kpi-name - 쿼리할 지표의 이름입니다. 이 값은 연관된 지표를 설명하며 두 개 이상의 용어로 구성되며, 이 용어는 소문자 영숫자 문자로 구성되며 하이픈으로 구분됩니다. 지원되는 지표의 전체 목록과 각 지표에 대한 kpi-name 값은 프로젝트, 캠페인 및 여정에 대한 표준 지표 단원을 참조하십시오.

특정 날짜 범위에 대해 데이터를 쿼리하는 필터를 적용할 수도 있습니다. 날짜 범위를 지정하지 않으면 Amazon Pinpoint에서는 이전 31일 동안의 데이터를 반환합니다. 다른 날짜별로 데이터를 필터링하려면 지원되는 날짜 범위 파라미터를 사용하여 날짜 범위의 첫 번째 및 마지막 날짜 및 시간을 지정합니다. 값은 확장된 ISO 8601 형식이어야 하며 협정 세계시(UTC)를 사용해야 합니다. 예를 들어

2019-07-19T20:00:00Z는 2019년 7월 19일 오후 8시 UTC입니다. 날짜 범위는 포괄적이며 31일 이하로 제한되어야 합니다. 또한 첫 번째 날짜 및 시간은 현재 날짜로부터 90일 이내여야 합니다.

다음 예제에서는 Amazon Pinpoint REST API, AWS CLI및를 사용하여 캠페인에 대한 분석 데이터를 쿼리하는 방법을 보여줍니다 AWS SDK for Java. 지원되는 AWS SDK를 사용하여 캠페인에 대한 분석 데이터를 쿼리할 수 있습니다. 이 AWS CLI 예제는 Microsoft Windows용 형식입니다. Unix, Linux 및 macOS의 경우 캐럿(^) 행 연속 문자를 백슬래시(\)로 바꿉니다.

REST API

Amazon Pinpoint REST API를 사용하여 캠페인의 분석 데이터를 쿼리하려면 HTTP(S) GET 요청을 캠페인 지표 URI에 보냅니다. URI에서 필요한 경로 파라미터에 적합한 값을 지정합니다.

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

위치:

- *#####*는 캠페인과 연결된 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- *application-id*는 캠페인과 관련된 프로젝트의 고유 식별자입니다.
- *campaign-id*는 캠페인의 고유 식별자입니다.
- *kpi-name*은 쿼리할 지표의 kpi-name 값입니다.

모든 파라미터는 URL로 인코딩해야 합니다.

특정 날짜 범위에 대한 데이터를 쿼리하는 필터를 적용하려면 start-time 및 end-time 쿼리 파라미터와 값을 URI에 추가합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 앰퍼샌드(&)를 사용하여 파라미터를 구분합니다.

예를 들어 다음 요청은 2019년 7월 19일부터 2019년 7월 26일까지 캠페인의 모든 실행에 의해 메시지가 배달된 고유 엔드포인트 수를 검색합니다.

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123example/campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

위치:

- `pinpoint.us-east-1.amazonaws.com`은 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- `12345678901234567890123456example`은 캠페인과 연결된 프로젝트의 고유 식별자입니다.
- `80b8efd84042ff8d9c96ce2f8example`은 캠페인의 고유 식별자입니다.
- `unique-deliveries`는 엔드포인트 배달 건수 캠페인 지표의 `kpi-name` 값이며, 이는 캠페인의 모든 실행에 의해 메시지가 배달된 고유한 엔드포인트 수를 보고하는 지표입니다.
- `2019-07-19T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-07-26T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

AWS CLI

를 사용하여 캠페인의 분석 데이터를 쿼리하려면 get-campaign-date-range-kpi 명령을 AWS CLI사용하고 필요한 파라미터에 적절한 값을 지정합니다.

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id application-id ^
    --campaign-id campaign-id ^
    --kpi-name kpi-name
```

위치:

- *application-id*는 캠페인과 관련된 프로젝트의 고유 식별자입니다.
- *campaign-id*는 캠페인의 고유 식별자입니다.
- *kpi-name*은 쿼리할 지표의 `kpi-name` 값입니다.

특정 날짜 범위에 대한 데이터를 쿼리하는 필터를 적용하려면 `start-time` 및 `end-time` 파라미터와 값을 쿼리에 추가합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음 요청은 2019년 7월 19일부터 2019년 7월 26일까지 캠페인의 모든 실행에 의해 메시지가 배달된 고유 엔드포인트 수를 검색합니다.

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
```

```
    --kpi-name unique-deliveries ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

위치:

- `12345678901234567890123456example`은 캠페인과 연결된 프로젝트의 고유 식별자입니다.
- `80b8efd84042ff8d9c96ce2f8example`은 캠페인의 고유 식별자입니다.
- `unique-deliveries`는 엔드포인트 배달 건수 캠페인 지표의 `kpi-name` 값이며, 이는 캠페인의 모든 실행에 의해 메시지가 배달된 고유한 엔드포인트 수를 보고하는 지표입니다.
- `2019-07-19T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-07-26T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

SDK for Java

를 사용하여 캠페인에 대한 분석 데이터를 쿼리하려면 캠페인 지표 API의 GetCampaignDateRangeKpiRequest 메서드를 AWS SDK for Java사용합니다. https://docs.aws.amazon.com/pinpoint/latest/apireference/apps-application-id-campaigns-campaign-id-kpis-daterange-kpi-name.html 필수 파라미터에 대해 적절한 값을 지정합니다.

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withCampaignId("campaignId")
        .withKpiName("kpiName")
```

위치:

- *`applicationId`*는 캠페인과 관련된 프로젝트의 고유 식별자입니다.
- *`campaignId`*는 캠페인의 고유 식별자입니다.
- *`kpiName`*은 쿼리할 지표의 `kpi-name` 값입니다.

특정 날짜 범위에 대한 데이터를 쿼리하는 필터를 적용하려면 쿼리에 `startTime` 및 `endTime` 파라미터 및 값을 포함합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음 요

청은 2019년 7월 19일부터 2019년 7월 26일까지 캠페인의 모든 실행에 의해 메시지가 배달된 고유 엔드포인트 수를 검색합니다.

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withCampaignId("80b8efd84042ff8d9c96ce2f8example")
        .withKpiName("unique-deliveries")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

위치:

- `12345678901234567890123456example`은 캠페인과 연결된 프로젝트의 고유 식별자입니다.
- `80b8efd84042ff8d9c96ce2f8example`은 캠페인의 고유 식별자입니다.
- `unique-deliveries`는 엔드포인트 배달 건수 캠페인 지표의 `kpi-name` 값이며, 이는 캠페인의 모든 실행에 의해 메시지가 배달된 고유한 엔드포인트 수를 보고하는 지표입니다.
- `2019-07-19T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-07-26T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

쿼리를 전송한 후 Amazon Pinpoint는 쿼리 결과를 JSON 응답으로 반환합니다. 결과 구조는 쿼리한 지표에 따라 다릅니다. 일부 지표는 하나의 값만 반환합니다. 예를 들어, 이전 예제에서 사용된 엔드포인트 배달 건수(`unique-deliveries`) 캠페인 지표는 하나의 값, 즉 캠페인의 모든 실행에 의해 메시지가 배달된 고유한 엔드포인트 수를 반환합니다. 이 경우 JSON 응답은 다음과 같습니다.

```
{
    "CampaignDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "CampaignId":"80b8efd84042ff8d9c96ce2f8example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
```

```
                            "Value":"123.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

다른 지표는 여러 값을 반환하고 관련 필드별로 값을 그룹화합니다. 지표가 여러 값을 반환하는 경우 JSON 응답에는 데이터를 그룹화하는 데 사용된 필드를 나타내는 필드가 포함됩니다.

쿼리 결과의 구조에 대한 자세한 내용은 JSON 쿼리 결과 사용 단원을 참조하십시오.

여러 캠페인에 대한 Amazon Pinpoint 데이터 쿼리

여러 캠페인의 데이터를 쿼리하는 방법에는 두 가지가 있습니다. 가장 좋은 방법은 모두 동일한 프로젝트와 연결된 캠페인의 데이터를 쿼리할지 여부에 따라 다릅니다. 이 경우 해당 캠페인 전체 또는 일부에 대한 데이터를 쿼리할지 아니면 해당 캠페인의 하위 집합에 대한 데이터를 쿼리할지 여부에 따라서도 달라집니다.

서로 다른 프로젝트와 연결된 캠페인 또는 동일한 프로젝트와 연결된 캠페인의 하위 집합에 대해서만 데이터를 쿼리하려면 데이터를 쿼리할 각 캠페인에 대해 하나씩 일련의 개별 쿼리를 생성하고 실행하는 것이 가장 좋습니다. 이전 섹션에서는 하나의 캠페인에 대해서만 데이터를 쿼리하는 방법에 대해 설명합니다.

동일한 프로젝트와 연결된 모든 캠페인의 데이터를 쿼리하려면 애플리케이션 지표 API를 사용하면 됩니다. 다음 필수 파라미터 값을 지정합니다.

- application-id - 프로젝트의 고유 식별자인 프로젝트 ID입니다. Amazon Pinpoint에서 프로젝트 및 애플리케이션이라는 용어는 동일한 의미를 갖습니다.
- kpi-name - 쿼리할 지표의 이름입니다. 이 값은 연관된 지표를 설명하며 두 개 이상의 용어로 구성되며, 이 용어는 소문자 영숫자 문자로 구성되며 하이픈으로 구분됩니다. 지원되는 지표의 전체 목록과 각 지표에 대한 `kpi-name` 값은 프로젝트, 캠페인 및 여정에 대한 표준 지표 단원을 참조하십시오.

날짜 범위를 기준으로 데이터를 필터링할 수도 있습니다. 날짜 범위를 지정하지 않으면 Amazon Pinpoint에서는 이전 31일 동안의 데이터를 반환합니다. 다른 날짜별로 데이터를 필터링하려면 지원되는 날짜 범위 파라미터를 사용하여 날짜 범위의 첫 번째 및 마지막 날짜 및 시간을 지정합

니다. 값은 확장된 ISO 8601 형식이어야 하며 협정 세계시(UTC)를 사용해야 합니다. 예를 들어 `2019-07-19T20:00:00Z`는 2019년 7월 19일 오후 8시 UTC입니다. 날짜 범위는 포괄적이며 31일 이하로 제한되어야 합니다. 또한 첫 번째 날짜 및 시간은 현재 날짜로부터 90일 이내여야 합니다.

다음 예제에서는 Amazon Pinpoint REST API, AWS CLI및를 사용하여 캠페인에 대한 분석 데이터를 쿼리하는 방법을 보여줍니다 AWS SDK for Java. 지원되는 AWS SDK를 사용하여 캠페인에 대한 분석 데이터를 쿼리할 수 있습니다. 이 AWS CLI 예제는 Microsoft Windows용 형식입니다. Unix, Linux 및 macOS의 경우 캐럿(^) 행 연속 문자를 백슬래시(\)로 바꿉니다.

REST API

Amazon Pinpoint REST API를 사용하여 여러 캠페인의 분석 데이터를 쿼리하려면 HTTP(S) GET 요청을 애플리케이션 지표 URI로 보냅니다. URI에서 필요한 경로 파라미터에 적합한 값을 지정합니다.

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

위치:

- *#####*는 캠페인과 연결된 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- *application-id*는 캠페인과 관련된 프로젝트의 고유 식별자입니다.
- *kpi-name*은 쿼리할 지표의 `kpi-name` 값입니다.

모든 파라미터는 URL로 인코딩해야 합니다.

특정 날짜 범위에 대한 데이터를 검색하는 필터를 적용하려면 `start-time` 및 `end-time` 쿼리 파라미터와 값을 URI에 추가합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 앰퍼샌드(&)를 사용하여 파라미터를 구분합니다.

예를 들어 다음 요청은 2019년 7월 19일부터 2019년 7월 26일까지 각 프로젝트 캠페인에 의해 메시지가 배달된 고유한 엔드포인트 수를 검색합니다.

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/unique-deliveries-grouped-by-campaign?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

위치:

- `pinpoint.us-east-1.amazonaws.com`은 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- `12345678901234567890123456example`은 캠페인과 연결된 프로젝트의 고유 식별자입니다.
- `unique-deliveries-grouped-by-campaign`은 엔드포인트 배달 건수(캠페인별 그룹화) 애플리케이션 지표의 `kpi-name` 값이며, 이는 각 캠페인에 의해 메시지가 배달된 고유 엔드포인트 수를 반환하는 지표입니다.
- `2019-07-19T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-07-26T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

AWS CLI

를 사용하여 여러 캠페인의 분석 데이터를 쿼리하려면 get-application-date-range-kpi 명령을 AWS CLI사용하고 필요한 파라미터에 적절한 값을 지정합니다.

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

위치:

- *`application-id`*는 캠페인과 관련된 프로젝트의 고유 식별자입니다.
- *`kpi-name`*은 쿼리할 지표의 `kpi-name` 값입니다.

특정 날짜 범위에 대한 데이터를 검색하는 필터를 적용하려면 쿼리에 `start-time` 및 `end-time` 파라미터 및 값을 포함합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음 요청은 2019년 7월 19일부터 2019년 7월 26일까지 각 프로젝트 캠페인에 의해 메시지가 배달된 고유한 엔드포인트 수를 검색합니다.

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name unique-deliveries-grouped-by-campaign ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

위치:

- `12345678901234567890123example`은 캠페인과 연결된 프로젝트의 고유 식별자입니다.
- `unique-deliveries-grouped-by-campaign`은 엔드포인트 배달 건수(캠페인별 그룹화) 애플리케이션 지표의 `kpi-name` 값이며, 이는 각 캠페인에 의해 메시지가 배달된 고유 엔드포인트 수를 반환하는 지표입니다.
- `2019-07-19T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-07-26T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

SDK for Java

를 사용하여 여러 캠페인의 분석 데이터를 쿼리하려면 Application Metrics API의 GetApplicationDateRangeKpiRequest 메서드를 AWS SDK for Java사용합니다. 필수 파라미터에 대해 적절한 값을 지정합니다.

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

위치:

- *applicationId*는 캠페인과 관련된 프로젝트의 고유 식별자입니다.
- *kpiName*은 쿼리할 지표의 `kpi-name` 값입니다.

특정 날짜 범위에 대한 데이터를 검색하는 필터를 적용하려면 쿼리에 `startTime` 및 `endTime` 파라미터 및 값을 포함합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음 요청은 2019년 7월 19일부터 2019년 7월 26일까지 각 프로젝트 캠페인에 의해 메시지가 배달된 고유한 엔드포인트 수를 검색합니다.

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123example")
        .withKpiName("unique-deliveries-grouped-by-campaign")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

위치:

- 12345678901234567890123456example은 캠페인과 연결된 프로젝트의 고유 식별자입니다.
- unique-deliveries-grouped-by-campaign은 엔드포인트 배달 건수(캠페인별 그룹화) 애플리케이션 지표의 kpi-name 값이며, 이는 각 캠페인에 의해 메시지가 배달된 고유 엔드포인트 수를 반환하는 지표입니다.
- 2019-07-19T00:00:00Z는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- 2019-07-26T23:59:59Z는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

쿼리를 전송한 후 Amazon Pinpoint는 쿼리 결과를 JSON 응답으로 반환합니다. 결과 구조는 쿼리한 지표에 따라 다릅니다. 일부 지표는 하나의 값만 반환합니다. 다른 지표는 여러 값을 반환하며 이러한 값은 관련 필드별로 그룹화됩니다. 지표가 여러 값을 반환하는 경우 JSON 응답에는 데이터를 그룹화하는 데 사용된 필드를 나타내는 필드가 포함됩니다.

예를 들어 이전 예제에서 사용된 엔드포인트 배달 건수(캠페인별 그룹화)(unique-deliveries-grouped-by-campaign) 애플리케이션 지표는 프로젝트와 관련된 각 캠페인에 대해 메시지가 배달된 고유한 엔드포인트 수와 같은 여러 값을 반환합니다. 이 경우 JSON 응답은 다음과 같습니다.

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
```

```
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```


이 경우 GroupedBys 필드는 값이 캠페인 ID(CampaignId)별로 그룹화됨을 나타냅니다.

쿼리 결과의 구조에 대한 자세한 내용은 JSON 쿼리 결과 사용 단원을 참조하십시오.

트랜잭션 메시지에 대한 Amazon Pinpoint 분석 데이터 쿼리

Amazon Pinpoint 콘솔의 분석 페이지를 사용하는 것 외에도 Amazon Pinpoint Analytics API를 사용하여 분석 데이터에 표준 지표의 하위 집합을 쿼리하여 프로젝트에 대해 전송된 트랜잭션 메시지의 제공 및 참여 추세를 파악할 수 있습니다.

이러한 각 지표는 측정 가능한 값으로, KPI(핵심 성과 지표)라고도 하며 트랜잭션 메시지의 성과를 모니터링하고 평가하는 데 사용할 수 있습니다. 예를 들어 지표를 사용하여 보낸 트랜잭션 이메일이나 SMS 메시지 수 또는 수신자에게 배달된 메시지 수를 확인할 수 있습니다. Amazon Pinpoint는 는 프로젝트에 보내는 모든 트랜잭션 이메일 및 SMS 메시지에 대해 이 데이터를 자동으로 수집하고 집계합니다. 데이터는 90일 동안 저장됩니다.

Amazon Pinpoint Analytics API를 사용하여 데이터를 쿼리하는 경우 쿼리의 범위, 데이터, 그룹화 및 필터를 정의하는 다양한 옵션을 선택할 수 있습니다. 이 작업은 적용할 날짜 기반 필터와 함께 쿼리할 프로젝트 및 지표를 지정하는 파라미터를 사용하여 수행할 수 있습니다.

이 항목에서는 이러한 옵션을 선택하고 프로젝트에 대한 트랜잭션 메시징 데이터를 쿼리하는 방법에 대해 설명하고 예제를 제공합니다.

사전 조건

트랜잭션 메시지에 대한 분석 데이터를 쿼리하기 전에 쿼리를 정의하는 데 사용할 다음 정보를 수집하는 것이 좋습니다.

- 프로젝트 ID - 메시지를 보낸 프로젝트의 고유 식별자입니다. Amazon Pinpoint API에서 이 값은 `application-id` 속성에 저장됩니다. Amazon Pinpoint 콘솔에서는 이 값이 모든 프로젝트 페이지에 프로젝트 ID로 표시됩니다.
- 날짜 범위 - 날짜 범위 또는 데이터를 쿼리할 일자 범위의 최초 및 최종 날짜 및 시간입니다. 날짜 범위는 포괄적이며 31일 이하로 제한되어야 합니다. 또한 현재 날짜로부터 90일 이내에 시작해야 합니다. 날짜 범위를 지정하지 않으면 Amazon Pinpoint에서 이전 31일의 데이터를 자동으로 쿼리합니다.
- 지표 - 쿼리할 지표의 이름입니다. 보다 구체적으로 말하면 지표의 값입니다. 지원되는 지표의 전체 목록과 각 지표에 대한 `kpi-name` 값은 프로젝트, 캠페인 및 여정에 대한 표준 지표 단원을 참조하십시오.

이는 또한 관련 필드로 데이터를 그룹화할지 여부를 결정하는 데 도움이 됩니다. 이 경우 자동으로 데이터를 그룹화하도록 설계된 지표를 선택하여 분석 및 보고를 단순화할 수 있습니다. 예를 들어 Amazon Pinpoint는 받는 사람에게 배달된 트랜잭션 SMS 메시지의 수를 보고하는 몇 가지 표

준 지표를 제공합니다. 이러한 지표 중 하나는 자동으로 데이터를 날짜별(txn-sms-delivered-grouped-by-date)로 그룹화합니다. 다른 지표는 자동으로 데이터를 국가 또는 리전별(txn-sms-delivered-grouped-by-country)로 그룹화합니다. 세 번째 지표는 단일 값, 즉 수신자에게 배달된 메시지 수를 반환합니다(txn-sms-delivered). 원하는 방식으로 데이터를 그룹화하는 표준 지표를 찾을 수 없는 경우 원하는 데이터를 반환하는 일련의 쿼리를 개발할 수 있습니다. 그런 다음 쿼리 결과를 수동으로 분류하거나 디자인 중인 사용자 지정 그룹으로 결합할 수 있습니다.

마지막으로 쿼리할 데이터에 액세스할 수 있는 권한이 있는지 확인하는 것이 중요합니다. 자세한 내용은 Amazon Pinpoint 분석 데이터 쿼리를 위한 IAM 정책 단원을 참조하십시오.

트랜잭션 이메일 메시지에 대한 Amazon Pinpoint 데이터 쿼리

프로젝트에 대해 전송된 트랜잭션 이메일 메시지의 데이터를 쿼리하려면 애플리케이션 지표 API를 사용하여 다음 필수 파라미터에 대한 값을 지정합니다.

- application-id - 프로젝트의 고유 식별자인 프로젝트 ID입니다. Amazon Pinpoint에서 프로젝트 및 애플리케이션이라는 용어는 동일한 의미를 갖습니다.
- kpi-name - 쿼리할 지표의 이름입니다. 이 값은 연관된 지표를 설명하며 두 개 이상의 용어로 구성되며, 이 용어는 소문자 영숫자 문자로 구성되며 하이픈으로 구분됩니다. 지원되는 지표의 전체 목록과 각 지표에 대한 kpi-name 값은 프로젝트, 캠페인 및 여정에 대한 표준 지표 단원을 참조하십시오.

특정 날짜 범위에 대해 데이터를 쿼리하는 필터를 적용할 수도 있습니다. 날짜 범위를 지정하지 않으면 Amazon Pinpoint에서는 이전 31일 동안의 데이터를 반환합니다. 다른 날짜별로 데이터를 필터링하려면 지원되는 날짜 범위 파라미터를 사용하여 날짜 범위의 첫 번째 및 마지막 날짜 및 시간을 지정합니다. 값은 확장된 ISO 8601 형식이어야 하며 협정 세계시(UTC)를 사용해야 합니다. 예를 들어 2019-09-06T20:00:00Z는 2019년 9월 6일 오후 8시 UTC입니다. 날짜 범위는 포괄적이며 31일 이하로 제한되어야 합니다. 또한 첫 번째 날짜 및 시간은 현재 날짜로부터 90일 이내여야 합니다.

다음 예제에서는 Amazon Pinpoint REST API, AWS CLI및를 사용하여 트랜잭션 이메일 메시지에 대한 분석 데이터를 쿼리하는 방법을 보여줍니다 AWS SDK for Java. 지원되는 AWS SDK를 사용하여 트랜잭션 메시지에 대한 분석 데이터를 쿼리할 수 있습니다. 이 AWS CLI 예제는 Microsoft Windows용 형식입니다. Unix, Linux 및 macOS의 경우 캐럿(^) 행 연속 문자를 백슬래시(\)로 바꿉니다.

REST API

Amazon Pinpoint REST API를 사용하여 트랜잭션 이메일 메시지에 대한 분석 데이터를 쿼리하려면 HTTP(S) GET 요청을 애플리케이션 지표 URI로 보냅니다. URI에서 필요한 경로 파라미터에 적합한 값을 지정합니다.

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

위치:

- *#####*는 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- *application-id*는 프로젝트의 고유 식별자입니다.
- *kpi-name*은 쿼리할 지표의 kpi-name 값입니다.

모든 파라미터는 URL로 인코딩해야 합니다.

특정 날짜 범위에 대한 데이터를 쿼리하는 필터를 적용하려면 start-time 및 end-time 쿼리 파라미터와 값을 URI에 추가합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 앰퍼샌드(&)를 사용하여 파라미터를 구분합니다.

예를 들어 다음 요청은 2019년 9월 6일부터 2019년 9월 13일까지 프로젝트에 대해 전송된 트랜잭션 이메일 메시지의 수를 검색합니다.

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-13T23:59:59Z
```

위치:

- pinpoint.us-east-1.amazonaws.com은 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- 12345678901234567890123456example은 프로젝트의 고유 식별자입니다.
- txn-emails-sent는 전송 수 애플리케이션 지표의 kpi-name 값으로, 이는 프로젝트에 대해 배달된 트랜잭션 이메일 메시지 수를 보고하는 지표입니다.
- 2019-09-06T00:00:00Z는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- 2019-09-13T23:59:59Z는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

AWS CLI

를 사용하여 트랜잭션 이메일 메시지에 대한 분석 데이터를 쿼리하려면 get-application-date-range-kpi 명령을 AWS CLI사용하고 필요한 파라미터에 적절한 값을 지정합니다.

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

위치:

- *application-id*는 프로젝트의 고유 식별자입니다.
- *kpi-name*은 쿼리할 지표의 kpi-name 값입니다.

특정 날짜 범위에 대한 데이터를 쿼리하는 필터를 적용하려면 start-time 및 end-time 파라미터와 값을 쿼리에 추가합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음 요청은 2019년 9월 6일부터 2019년 9월 13일까지 프로젝트에 대해 전송된 트랜잭션 이메일 메시지의 수를 검색합니다.

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-emails-sent ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-13T23:59:59Z
```

위치:

- 12345678901234567890123456example은 프로젝트의 고유 식별자입니다.
- txn-emails-sent는 전송 수 애플리케이션 지표의 kpi-name 값으로, 이는 프로젝트에 대해 배달된 트랜잭션 이메일 메시지 수를 보고하는 지표입니다.
- 2019-09-06T00:00:00Z는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- 2019-09-13T23:59:59Z는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

SDK for Java

를 사용하여 트랜잭션 이메일 메시지에 대한 분석 데이터를 쿼리하려면 Application Metrics API의 GetApplicationDateRangeKpiRequest 메서드를 AWS SDK for Java사용합니다. 필수 파라미터에 대해 적절한 값을 지정합니다.

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

위치:

- *applicationId*는 프로젝트의 고유 식별자입니다.
- *kpiName*은 쿼리할 지표의 kpi-name 값입니다.

특정 날짜 범위에 대한 데이터를 쿼리하는 필터를 적용하려면 쿼리에 startTime 및 endTime 파라미터 및 값을 포함합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음 요청은 2019년 9월 6일부터 2019년 9월 13일까지 프로젝트에 대해 전송된 트랜잭션 이메일 메시지의 수를 검색합니다.

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-emails-sent")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

위치:

- 12345678901234567890123456example은 프로젝트의 고유 식별자입니다.
- txn-emails-sent는 전송 수 애플리케이션 지표의 kpi-name 값으로, 이는 프로젝트에 대해 배달된 트랜잭션 이메일 메시지 수를 보고하는 지표입니다.
- 2019-09-06T00:00:00Z는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- 2019-09-13T23:59:59Z는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

쿼리를 전송한 후 Amazon Pinpoint는 쿼리 결과를 JSON 응답으로 반환합니다. 결과 구조는 쿼리한 지표에 따라 다릅니다. 일부 지표는 하나의 값만 반환합니다. 예를 들어, 이전 예제에서 사용된 전송 수(`txn-emails-sent`) 애플리케이션 지표는 하나의 값, 즉 프로젝트에서 전송된 트랜잭션 이메일 메시지 수를 반환합니다. 이 경우 JSON 응답은 다음과 같습니다.

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"1234567890123456789012345example",
        "EndTime":"2019-09-13T23:59:59Z",
        "KpiName":"txn-emails-sent",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"TxnEmailsSent",
                            "Type":"Double",
                            "Value":"62.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

다른 지표는 여러 값을 반환하고 관련 필드별로 값을 그룹화합니다. 지표가 여러 값을 반환하는 경우 JSON 응답에는 데이터를 그룹화하는 데 사용된 필드를 나타내는 필드가 포함됩니다.

쿼리 결과의 구조에 대한 자세한 내용은 JSON 쿼리 결과 사용 단원을 참조하십시오.

트랜잭션 SMS 메시지에 대한 Amazon Pinpoint 데이터 쿼리

프로젝트에 대해 전송된 트랜잭션 SMS 메시지의 데이터를 쿼리하려면 애플리케이션 지표 API를 사용하여 다음 필수 파라미터에 대한 값을 지정합니다.

- application-id - 프로젝트의 고유 식별자인 프로젝트 ID입니다. Amazon Pinpoint에서 프로젝트 및 애플리케이션이라는 용어는 동일한 의미를 갖습니다.

- kpi-name - 쿼리할 지표의 이름입니다. 이 값은 연관된 지표를 설명하며 두 개 이상의 용어로 구성되며, 이 용어는 소문자 영숫자 문자로 구성되며 하이픈으로 구분됩니다. 지원되는 지표의 전체 목록과 각 지표에 대한 `kpi-name` 값은 프로젝트, 캠페인 및 여정에 대한 표준 지표 단원을 참조하십시오.

특정 날짜 범위에 대해 데이터를 쿼리하는 필터를 적용할 수도 있습니다. 날짜 범위를 지정하지 않으면 Amazon Pinpoint에서는 이전 31일 동안의 데이터를 반환합니다. 다른 날짜를 기준으로 데이터를 필터링하려면 지원되는 날짜 범위 파라미터를 사용하여 날짜 범위의 첫 번째 및 마지막 날짜와 시간을 지정합니다. 값은 확장된 ISO 8601 형식이어야 하며 협정 세계시(UTC)를 사용해야 합니다. 예를 들어 `2019-09-06T20:00:00Z`는 2019년 9월 6일 오후 8시 UTC입니다. 날짜 범위는 포괄적이며 31일 이하로 제한되어야 합니다. 또한 첫 번째 날짜 및 시간은 현재 날짜로부터 90일 이내여야 합니다.

다음 예제에서는 Amazon Pinpoint REST API, AWS CLI및를 사용하여 트랜잭션 SMS 메시지에 대한 분석 데이터를 쿼리하는 방법을 보여줍니다 AWS SDK for Java. 지원되는 AWS SDK를 사용하여 트랜잭션 메시지에 대한 분석 데이터를 쿼리할 수 있습니다. 이 AWS CLI 예제는 Microsoft Windows용 형식입니다. Unix, Linux 및 macOS의 경우 캐럿(^) 행 연속 문자를 백슬래시(\)로 바꿉니다.

REST API

Amazon Pinpoint REST API를 사용하여 트랜잭션 SMS 메시지에 대한 분석 데이터를 쿼리하려면 HTTP(S) GET 요청을 애플리케이션 지표 URI로 보냅니다. URI에서 필요한 경로 파라미터에 적합한 값을 지정합니다.

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

위치:

- *#####*는 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- *application-id*는 프로젝트의 고유 식별자입니다.
- *kpi-name*은 쿼리할 지표의 `kpi-name` 값입니다.

모든 파라미터는 URL로 인코딩해야 합니다.

특정 날짜 범위에 대한 데이터를 검색하는 필터를 적용하려면 `start-time` 및 `end-time` 쿼리 파라미터와 값을 URI에 추가합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 앰퍼샌드(&)를 사용하여 파라미터를 구분합니다.

예를 들어 다음 요청은 2019년 9월 6일부터 2019년 9월 8일까지 매일 전송된 트랜잭션 SMS 메시지 수를 검색합니다.

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-time=2019-09-08T23:59:59Z
```

위치:

- `pinpoint.us-east-1.amazonaws.com`은 프로젝트를 호스팅하는 AWS 리전의 Amazon Pinpoint 엔드포인트입니다.
- `12345678901234567890123456789012345example`은 프로젝트의 고유 식별자입니다.
- `txn-sms-sent-grouped-by-date`는 날짜별로 그룹화된 전송 수 애플리케이션 지표의 `kpi-name` 값으로, 이 지표는 날짜 범위에서 매일 전송된 트랜잭션 SMS 메시지 수를 반환하는 지표입니다.
- `2019-09-06T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-09-08T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

AWS CLI

를 사용하여 트랜잭션 SMS 메시지에 대한 분석 데이터를 쿼리하려면 get-application-date-range-kpi 명령을 AWS CLI사용하고 필요한 파라미터에 적절한 값을 지정합니다.

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

위치:

- *`application-id`*는 프로젝트의 고유 식별자입니다.
- *`kpi-name`*은 쿼리할 지표의 `kpi-name` 값입니다.

특정 날짜 범위에 대한 데이터를 검색하는 필터를 적용하려면 쿼리에 `start-time` 및 `end-time` 파라미터 및 값을 포함합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음

요청은 2019년 9월 6일부터 2019년 9월 8일까지 매일 전송된 트랜잭션 SMS 메시지 수를 검색합니다.

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-sms-sent-grouped-by-date ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-08T23:59:59Z
```

위치:

- `12345678901234567890123456example`은 프로젝트의 고유 식별자입니다.
- `txn-sms-sent-grouped-by-date`는 날짜별로 그룹화된 전송 수 애플리케이션 지표의 `kpi-name` 값으로, 이 지표는 날짜 범위에서 매일 전송된 트랜잭션 SMS 메시지 수를 반환하는 지표입니다.
- `2019-09-06T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-09-08T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

SDK for Java

를 사용하여 트랜잭션 SMS 메시지에 대한 분석 데이터를 쿼리하려면 Application Metrics API의 GetApplicationDateRangeKpiRequest 메서드를 AWS SDK for Java사용하고 필요한 파라미터에 적절한 값을 지정합니다.

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

위치:

- *applicationId*는 프로젝트의 고유 식별자입니다.
- *kpiName*은 쿼리할 지표의 `kpi-name` 값입니다.

특정 날짜 범위에 대한 데이터를 검색하는 필터를 적용하려면 쿼리에 `startTime` 및 `endTime` 파라미터 및 값을 포함합니다. 이러한 파라미터를 사용하여 데이터를 검색할 포함 날짜 범위의 확장 ISO 8601 형식으로 첫 번째 날짜와 마지막 날짜 및 시간을 지정할 수 있습니다. 예를 들어 다음 요

청은 2019년 9월 6일부터 2019년 9월 8일까지 매일 전송된 트랜잭션 SMS 메시지 수를 검색합니다.

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-sms-sent-grouped-by-date")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-08T23:59:59Z")));
```

위치:

- `12345678901234567890123456example`은 프로젝트의 고유 식별자입니다.
- `txn-sms-sent-grouped-by-date`는 날짜별로 그룹화된 전송 수 애플리케이션 지표의 kpi-name 값으로, 이 지표는 날짜 범위에서 매일 전송된 트랜잭션 SMS 메시지 수를 반환하는 지표입니다.
- `2019-09-06T00:00:00Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 첫 번째 날짜 및 시간입니다.
- `2019-09-08T23:59:59Z`는 포함된 날짜 범위의 일부로 데이터를 검색할 마지막 날짜 및 시간입니다.

쿼리를 전송한 후 Amazon Pinpoint는 쿼리 결과를 JSON 응답으로 반환합니다. 결과 구조는 쿼리한 지표에 따라 다릅니다. 일부 지표는 하나의 값만 반환합니다. 다른 지표는 여러 값을 반환하고 관련 필드별로 값을 그룹화합니다. 지표가 여러 값을 반환하는 경우 JSON 응답에는 데이터를 그룹화하는 데 사용된 필드를 나타내는 필드가 포함됩니다.

예를 들어, 이전 예제에서 사용된 전송 수(날짜별로 그룹화)(`txn-sms-sent-grouped-by-date`) 애플리케이션 지표는 여러 값, 즉 지정된 날짜 범위에서 매일 전송된 트랜잭션 SMS 메시지 수를 반환합니다. 이 경우 JSON 응답은 다음과 같습니다.

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-08T23:59:59Z",
        "KpiName":"txn-sms-sent-grouped-by-date",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
```

```
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-06"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"29.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-07"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"35.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-08"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"10.0"
                    }
                ]
```

```
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

이 경우 GroupedBys 필드는 값이 달력 일(Date)별로 그룹화됨을 나타냅니다. 이는 다음을 의미합니다.

- 2019년 9월 6일에 29개의 메시지가 전송되었습니다.
- 2019년 9월 7일에 35개의 메시지가 전송되었습니다.
- 2019년 9월 8일에 10개의 메시지가 전송되었습니다.

쿼리 결과의 구조에 대한 자세한 내용은 JSON 쿼리 결과 사용 단원을 참조하십시오.

Amazon Pinpoint 분석 JSON 쿼리 결과 사용

Amazon Pinpoint Analytics API를 사용하여 분석 데이터를 쿼리하면 Amazon Pinpoint는 결과를 JSON 응답으로 반환합니다. 애플리케이션 지표, 캠페인 지표 및 여정 참여 지표의 경우 응답의 데이터는 Amazon Pinpoint 분석 데이터 보고를 위한 표준 JSON 스키마를 준수합니다.

즉, 선택한 프로그래밍 언어나 도구를 사용하여 하나 이상의 지표에 대한 데이터를 쿼리하고 각 쿼리의 결과를 캡처한 다음 결과를 테이블, 객체 또는 기타 위치에 쓰는 사용자 지정 솔루션을 구현할 수 있습니다. 그런 다음 다른 서비스 또는 애플리케이션을 사용하여 해당 위치에서 쿼리 결과를 사용할 수 있습니다.

예를 들어, 다음을 수행할 수 있습니다.

- 정기적으로 지표 집합을 쿼리하고 선호하는 데이터 시각화 프레임워크를 사용하여 결과를 표시하는 사용자 지정 대시보드를 빌드합니다.
- 적절한 지표를 쿼리하고 차트 또는 사용자가 디자인한 다른 유형의 보고서에 결과를 표시하여 참여율을 추적하는 보고서를 만듭니다.
- 분석 데이터를 구문 분석하여 특정 스토리지 형식으로 작성한 다음 결과를 장기 스토리지 솔루션으로 포트합니다.

Amazon Pinpoint Analytics API는 나중에 Amazon Pinpoint 프로젝트 또는 Amazon Pinpoint 계정에서 읽거나 사용할 수 있는 영구 객체를 만들거나 저장하도록 설계되지 않았습니다. 대신 API는 분석 데이

터를 검색하고 추가 분석, 저장 또는 보고를 위해 해당 데이터를 다른 서비스 및 애플리케이션으로 전송할 수 있도록 설계되었습니다. 애플리케이션 지표, 캠페인 지표 및 여정 참여 지표에 대해 프로그래밍 방식으로 쿼리할 수 있는 모든 분석 데이터에 대해 동일한 JSON 응답 구조 및 스키마를 사용하여 이 작업을 부분적으로 수행합니다.

이 주제에서는 애플리케이션 지표, 캠페인 지표 또는 여정 참여 지표 쿼리에 대한 JSON 응답의 구조, 객체 및 필드에 대해 설명합니다. 여정 실행 지표 또는 여정 활동 실행 지표 쿼리에 대한 JSON 응답의 필드에 대한 자세한 내용은 Amazon Pinpoint 프로젝트, 캠페인 및 여정에 적용되는 표준 지표 단원을 참조하십시오.

JSON 구조

쿼리 결과를 구문 분석하고 사용하는 데 도움이 되도록 Amazon Pinpoint 분석 API는 애플리케이션 지표, 캠페인 지표 및 여정 참여 지표에 대해 프로그래밍 방식으로 쿼리할 수 있는 모든 Amazon Pinpoint 분석 데이터에 대해 동일한 JSON 응답 구조를 사용합니다. 각 JSON 응답은 프로젝트 ID(`ApplicationId`)와 같이 쿼리를 정의한 값을 지정합니다. 응답에는 단 하나의 `KpiResult` 객체도 포함됩니다. `KpiResult` 객체에는 쿼리에 대한 전체 결과 집합이 포함됩니다.

각 `KpiResult` 객체에는 `Rows` 객체가 포함됩니다. 이는 쿼리 결과 및 해당 결과의 값에 대한 관련 메타데이터가 포함된 객체 배열입니다. `Rows` 객체의 구조와 내용에는 다음과 같은 일반적인 특성이 있습니다.

- 쿼리 결과의 각 행은 객체에서 이름이 지정된 `Values`, 별도의 JSON `Rows` 객체입니다. 예를 들어 쿼리가 세 개의 값을 반환하는 경우 `Rows` 객체에는 세 개의 `Values` 객체가 포함됩니다. 각 `Values` 객체에는 쿼리에 대한 개별 결과가 포함되어 있습니다.
- 쿼리 결과의 각 열은 적용되는 `Values` 객체의 속성입니다. 열 이름은 `Values` 객체의 `Key` 필드에 저장됩니다.
- 그룹화된 쿼리 결과의 경우 각 `Values` 객체에는 연관된 `GroupedBys` 객체가 있습니다. `GroupedBys` 객체는 결과를 그룹화하는 데 사용된 필드를 나타냅니다. 또한 연관된 `Values` 객체의 그룹화 값을 제공합니다.
- 지표에 대한 쿼리 결과가 null이면 `Rows` 객체는 비어 있습니다.

이러한 일반적인 특성 외에도 `Rows` 객체의 구조와 내용은 지표에 따라 다릅니다. Amazon Pinpoint는 단일 값 지표와 다중 값 지표라는 두 가지 종류의 지표를 지원하기 때문입니다.

단일 값 지표는 하나의 누적 값만 제공합니다. 이러한 예로는 캠페인의 모든 실행에 의해 수신자에게 배달된 메시지의 비율을 들 수 있습니다. 다중 값 지표는 둘 이상의 값을 제공하고 해당 값을 관련 필드

별로 그룹화합니다. 예를 들어 캠페인 실행마다 수신자에게 배달된 메시지의 비율이 캠페인 실행별로 그룹화되어 있습니다.

지표의 이름을 참조하여 지표가 단일 값 지표인지 또는 다중 값 지표인지 신속하게 확인할 수 있습니다. 이름에 grouped-by가 포함되어 있지 않으면 단일 값 지표입니다. 포함한다면 다중 값 지표입니다. 프로그래밍 방식으로 쿼리할 수 있는 지표의 전체 목록은 Amazon Pinpoint 프로젝트, 캠페인 및 여정에 적용되는 표준 지표 단원을 참조하십시오.

단일 값 지표

단일 값 지표의 경우 Rows 객체는 다음과 같은 Values 객체를 포함합니다.

- 쿼리된 지표의 이름을 지정합니다.
- 쿼리된 지표에 대한 값을 제공합니다.
- 반환된 값의 데이터 유형을 식별합니다.

예를 들어 다음 JSON 응답에는 단일 값 지표에 대한 쿼리 결과가 포함되어 있습니다. 이 지표는 2019년 8월 1일부터 2019년 8월 31일까지 프로젝트와 연결된 모든 캠페인에 의해 메시지가 배달된 고유한 엔드포인트 수를 보고합니다.

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"1368.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
```

```
}
```

이 예에서 응답은 모든 프로젝트의 캠페인이 2019년 8월 1일부터 2019년 8월 31일까지 1,368개의 고유한 엔드포인트에 메시지를 배달했음을 나타냅니다.

- `Key`는 `Value` 필드(`UniqueDeliveries`)에 값이 지정된 지표의 친숙한 이름입니다.
- `Type`은 `Value` 필드(`Double`) 에 지정된 값의 데이터 유형입니다.
- `Value`는 적용된 필터를 포함하여 쿼리된 지표의 실제 값입니다(`1368.0`).

단일 값 지표에 대한 쿼리 결과가 null(0보다 크거나 같지 않음)인 경우 `Rows` 객체는 비어 있습니다. Amazon Pinpoint는 어떤 지표에 대해 반환할 데이터가 없을 경우, 해당 지표에 대해 null 값을 반환합니다. 예제:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"234567890123456789012345example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[

            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

다중 값 지표

다중 값 지표에 대한 `Rows` 객체의 구조 및 내용은 대부분 단일 값 지표와 동일합니다. 다중 값 지표에 대한 `Rows` 객체에는 `Values` 객체도 포함되어 있습니다. `Values` 객체는 쿼리된 지표의 친숙한 이름을 지정하고 해당 지표에 대한 값을 제공하고 해당 값의 데이터 유형을 식별합니다.

그러나 다중 값 지표에 대한 `Rows` 객체에는 하나 이상의 `GroupedBy` 객체도 포함됩니다. 쿼리 결과에는 각 `Values` 객체에 대해 하나의 `GroupedBy` 객체가 있습니다. `GroupedBy` 객체는 결과의 데이터를 그룹화하는 데 사용된 필드와 해당 필드의 데이터 유형을 나타냅니다. 또한 연관된 `Values` 객체의 해당 필드에 대한 그룹화 값을 나타냅니다.

예를 들어 다음 JSON 응답에는 2019년 8월 1일부터 2019년 8월 31일까지 프로젝트와 연결된 각 캠페인에 대해 메시지가 배달된 고유한 엔드포인트 수를 보고하는 다중 값 지표에 대한 쿼리 결과가 포함되어 있습니다.


```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
```

```
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

이 예에서 응답은 프로젝트 캠페인 중 세 개가 2019년 8월 1일부터 2019년 8월 31일까지 고유한 엔드포인트로 메시지를 배달했음을 나타냅니다. 각 캠페인의 제공 건수 분석은 다음과 같습니다.

- `80b8efd84042ff8d9c96ce2f8example` 캠페인은 123개의 고유한 엔드포인트에 메시지를 배달했습니다.
- `810c7aab86d42fb2b56c8c966example` 캠페인은 456개의 고유한 엔드포인트로 메시지를 배달했습니다.
- `42d8c7eb0990a57ba1d5476a3example` 캠페인은 789개의 고유한 엔드포인트로 메시지를 배달했습니다.

객체 및 필드의 일반적인 구조는 다음과 같습니다.

- `GroupedBys.Key` - `GroupedBys.Value` 필드(`CampaignId`)에 지정된 그룹화 값을 저장하는 속성 또는 필드의 이름입니다.
- `GroupedBys.Type` - `GroupedBys.Value` 필드(`String`)에 지정된 값의 데이터 유형입니다.
- `GroupedBys.Value` - `GroupedBys.Key` 필드(캠페인 ID)에 지정된 대로 데이터를 그룹화하는 데 사용된 필드의 실제 값입니다.

- `Values.Key` - `Values.Value` 필드(`UniqueDeliveries`)에 값이 지정된 지표의 친숙한 이름입니다.
- `Values.Type` - `Values.Value` 필드(`Double`)에 지정된 값의 데이터 유형입니다.
- `Values.Value` - 적용된 필터를 포함하여 쿼리된 지표의 실제 값입니다.

다중 값 지표에 대한 쿼리 결과가 특정 프로젝트, 캠페인 또는 기타 리소스에 대해 null(0보다 크거나 같지 않음)인 경우 Amazon Pinpoint는 리소스에 대한 객체나 필드를 반환하지 않습니다. 다중 값 지표에 대한 쿼리 결과가 모든 리소스에 대해 null인 경우 Amazon Pinpoint는 빈 `Rows` 객체를 반환합니다.

JSON 객체 및 필드

프로젝트 ID(`ApplicationId`)와 같이 쿼리를 정의한 값을 지정하는 것 외에도 애플리케이션 지표, 캠페인 지표 또는 여정 참여 지표 쿼리에 대한 각 JSON 응답에는 `KpiResult` 객체가 포함됩니다. 이 객체에는 쿼리에 대한 전체 결과 집합이 포함되어 있습니다. 이 결과 집합을 구문 분석하여 분석 데이터를 다른 서비스 또는 애플리케이션으로 보낼 수 있습니다. 각 `KpiResult` 객체에는 지표에 따라 다음과 같은 표준 객체 및 필드 중 일부 또는 전체가 포함됩니다.

객체 또는 필드	설명
`Rows`	쿼리에 대한 결과 집합을 포함하는 객체의 배열입니다.
`Rows.GroupedBys`	다중 값 지표의 경우 쿼리 결과에서 데이터를 그룹화하는 데 사용된 필드 및 값을 정의하는 필드 배열입니다.
`Rows.GroupedBys.Key`	다중 값 지표의 경우 필드에 지정된 값을 저장하는 속성 또는 `GroupedBys.Value` 필드의 이름입니다.
`Rows.GroupedBys.Type`	다중 값 지표의 경우 `GroupedBys.Value` 필드에 지정된 값의 데이터 유형입니다.
`Rows.GroupedBys.Value`	다중 값 지표의 경우 쿼리 결과에서 데이터를 그룹화하는 데 사용된 필드의 실제 값입니다. 이 값은 연결된 `Values` 객체와 연관됩니다.
`Rows.Values`	쿼리 결과를 포함하는 필드의 배열입니다.

객체 또는 필드	설명
`Rows.Values.Key`	쿼리된 지표의 이름입니다. 지표의 값은 `Values.Value` 필드에서 지정됩니다.
`Rows.Values.Type`	`Values.Value` 필드에 지정된 값의 데이터 유형입니다.
`Rows.Values.Value`	적용된 필터를 포함하여 쿼리된 지표의 실제 값입니다.

여정 실행 지표 또는 여정 활동 실행 지표 쿼리에 대한 JSON 응답의 필드에 대한 자세한 내용은 Amazon Pinpoint 프로젝트, 캠페인 및 여정에 적용되는 표준 지표 단원을 참조하십시오.

를 사용하여 Amazon Pinpoint API 호출 로깅 AWS CloudTrail

Amazon Pinpoint는 Amazon Pinpoint에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스인와 통합됩니다. CloudTrail은 Amazon Pinpoint에 대한 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 Amazon Pinpoint 콘솔에서 수행한 호출과 Amazon Pinpoint API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면 Amazon Pinpoint에 대한 이벤트를 포함한 CloudTrail 이벤트를 Amazon Simple Storage Service(S3) 버킷에 지속적으로 전송할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail이 수집한 정보를 사용하여 Amazon Pinpoint에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

구성 및 활성화 방법을 포함하여 CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용자 안내서를 참조하세요.

CloudTrail의 Amazon Pinpoint 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. Amazon Pinpoint에서 지원되는 이벤트 활동이 발생하면 해당 활동은 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

Amazon Pinpoint에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. 자격 증명 정보를 이용하면 다음을 판단하는 데 도움이 됩니다.

- 요청이 루트 또는 AWS Identity and Access Management 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 내용은 CloudTrail userIdentity 요소를 참조하십시오.

추적을 생성하고 Amazon S3 버킷에 원하는 기간만큼 로그 파일을 저장할 수 있습니다. 또한 Amazon S3 수명 주기 규칙을 정의하여 로그 파일을 자동으로 보관하거나 삭제할 수도 있습니다. 기본적으로 로그 파일은 Amazon S3 서버 측 암호화(SSE)를 사용하여 암호화합니다.

새 로그 파일이 전달되면 Amazon SNS 알림을 게시하도록 CloudTrail을 구성하여 로그 파일 전송 시 알림을 받을 수 있습니다. 자세한 내용은 CloudTrail용 Amazon SNS 알림 구성을 참조하십시오.

여러 AWS 리전 및 AWS 여러 계정의 Amazon Pinpoint 로그 파일을 단일 Amazon S3 버킷으로 집계할 수도 있습니다. 자세한 내용은 여러 리전에서 CloudTrail 로그 파일 수신 및 여러 계정에서 CloudTrail 로그 파일 수신을 참조하세요.

CloudTrail을 사용해 다음 Amazon Pinpoint API에서 작업을 로깅할 수 있습니다.

- Amazon Pinpoint API
- Amazon Pinpoint SMS 및 음성 API

CloudTrail 로그 파일에서 지원되는 Amazon Pinpoint API 작업

Amazon Pinpoint API는 아래와 같은 작업의 로깅을 CloudTrail 로그 파일의 이벤트로 지원합니다.

- CreateApp
- CreateCampaign
- CreateEmailTemplate
- CreateExportJob
- CreateImportJob
- CreateJourney
- CreatePushTemplate
- CreateRecommenderConfiguration
- CreateSegment

- CreateSmsTemplate
- CreateVoiceTemplate
- DeleteAdmChannel
- DeleteApnsChannel
- DeleteApnsSandboxChannel
- DeleteApnsVoipChannel
- DeleteApnsVoipSandboxChannel
- DeleteApp
- DeleteBaiduChannel
- DeleteCampaign
- DeleteEmailChannel
- DeleteEmailTemplate
- DeleteEndpoint
- DeleteEventStream
- DeleteGcmChannel
- DeleteJourney
- DeletePushTemplate
- DeleteRecommenderConfiguration
- DeleteSegment
- DeleteSmsChannel
- DeleteSmsTemplate
- DeleteUserEndpoints
- DeleteVoiceChannel
- DeleteVoiceTemplate
- GetAdmChannel
- GetApnsChannel
- GetApnsSandboxChannel
- GetApnsVoipChannel
- GetApnsVoipSandboxChannel
- GetApp

- GetApplicationDateRangeKpi
- GetApplicationSettings
- GetApps
- GetBaiduChannel
- GetCampaign
- GetCampaignActivities
- GetCampaignDateRangeKpi
- GetCampaignVersion
- GetCampaignVersions
- GetCampaigns
- GetChannels
- GetEmailChannel
- GetEmailTemplate
- GetEndpoint
- GetEventStream
- GetExportJob
- GetExportJobs
- GetGcmChannel
- GetImportJob
- GetImportJobs
- GetJourney
- GetJourneydaterangeKPI
- GetJourneyExecutionActivityMetrics
- GetJourneyExecutionMetrics
- GetPushTemplate
- GetRecommenderConfiguration
- GetRecommenderConfigurations
- GetSegment
- GetSegmentExportJobs
- GetSegmentImportJobs

- GetSegmentVersion
- GetSegmentVersions
- GetSegments
- GetSmsChannel
- GetSmsTemplate
- GetUserEndpoints
- GetVoiceChannel
- GetVoiceTemplate
- ListJourneys
- ListTagsForResource
- ListTemplates
- ListTemplateVersions
- PhoneNumberValidate
- PutEvents
- PutEventStream
- RemoveAttributes
- SendMessages
- SendOTPMessage
- SendUsersMessages
- TagResource
- UntagResource
- UpdateAdmChannel
- UpdateApnsChannel
- UpdateApnsSandboxChannel
- UpdateApnsVoipChannel
- UpdateApnsVoipSandboxChannel
- UpdateApplicationSettings
- UpdateBaiduChannel
- UpdateCampaign
- UpdateEmailChannel

- UpdateEmailTemplate
- UpdateEndpoint
- UpdateEndpointsBatch
- UpdateGcmChannel
- UpdateJourney
- UpdateJourneyState
- UpdatePushTemplate
- UpdateRecommenderConfiguration
- UpdateSegment
- UpdateSmsChannel
- UpdateSmsTemplate
- UpdateTemplateActiveVersion
- UpdateVoiceChannel
- UpdateVoiceTemplate

CloudTrail 로그 파일에서 지원되는 Amazon Pinpoint 이메일 API 작업

Amazon Pinpoint 이메일 API는 아래와 같은 작업의 로깅을 CloudTrail 로그 파일의 이벤트로 지원합니다.

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- CreateDedicatedIpPool
- CreateEmailIdentity
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- DeleteDedicatedIpPool
- DeleteEmailIdentity
- GetAccount
- GetConfigurationSet

- GetConfigurationSetEventDestinations
- GetDedicatedIp
- GetDedicatedIps
- GetEmailIdentity
- ListConfigurationSets
- ListDedicatedIpPools
- ListEmailIdentities
- PutAccountDedicatedIpWarmupAttributes
- PutAccountSendingAttributes
- PutConfigurationSetDeliveryOptions
- PutConfigurationSetReputationOptions
- PutConfigurationSetSendingOptions
- PutConfigurationSetTrackingOptions
- PutDedicatedIpInPool
- PutDedicatedIpWarmupAttributes
- PutEmailIdentityDkimAttributes
- PutEmailIdentityFeedbackAttributes
- PutEmailIdentityMailFromAttributes
- UpdateConfigurationSetEventDestination

다음과 같은 Amazon Pinpoint 이메일 API 작업은 CloudTrail에 로깅되지 않습니다.

- SendEmail

CloudTrail 로그 파일에서 지원되는 Amazon Pinpoint SMS and Voice API 버전 1 작업

Amazon Pinpoint SMS 및 음성 버전 1 API는 아래와 같은 작업의 로깅을 CloudTrail 로그 파일의 이벤트로 지원합니다.

- CreateConfigurationSet
- CreateConfigurationSetEventDestination

- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- GetConfigurationSetEventDestinations
- UpdateConfigurationSetEventDestination

다음과 같은 Amazon Pinpoint SMS 및 음성 버전 1 API 작업은 CloudTrail에 로깅되지 않습니다.

- SendVoiceMessage

Amazon Pinpoint API 작업을 보여주는 CloudTrail 로그 항목 예제

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터 단일 요청을 나타냅니다. 요청된 작업, 모든 파라미터, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 Amazon Pinpoint API의 `GetCampaigns` 및 `CreateCampaign` 작업을 보여주는 CloudTrail 로그 항목을 나타낸 예제입니다.


```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
```

```
      "userAgent": "Jersey/${project.version} (HttpUrlConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        },
        "type": "Root"
      }
    },
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "CreateCampaign",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T01:05:16Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": false,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "Description": "***",
        "HoldoutPercent": 0,
        "IsPaused": false,
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
        },
        "SegmentId": "exampleda204adf991a80281aa0e591",
        "SegmentVersion": 1,
        "application-id": "example71dfa4c1aab66332a5839798f"
      },
      "responseElements": {
        "ApplicationId": "example71dfa4c1aab66332a5839798f",
```

```
        "CreationDate": "2018-02-03T01:05:16.425Z",
        "Description": "***",
        "HoldoutPercent": 0,
        "Id": "example54a654f80948680cbba240ede",
        "IsPaused": false,
        "LastModifiedDate": "2018-02-03T01:05:16.425Z",
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
        },
        "SegmentId": "example4da204adf991a80281example",
        "SegmentVersion": 1,
        "State": {
          "CampaignStatus": "SCHEDULED"
        },
        "Version": 1
      },
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/userName",
        "principalId": "AIDAIHTHRCDA62EXAMPLE",
        "type": "IAMUser",
        "userName": "userName"
      }
    }
  ]
}
```

다음은 Amazon Pinpoint SMS 및 음성 API의 `CreateConfigurationSet` 및 `CreateConfigurationSetEventDestination` 작업을 보여주는 CloudTrail 로그 항목을 나타낸 예제입니다.

```
{
  "Records": [
    {
      "eventVersion":"1.05",
```

```
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:45:55Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSet",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "ConfigurationSetName":"MyConfigurationSet"
      },
      "responseElements":null,
      "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
      "eventID":"725843fc-8846-41f4-871a-7c52dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"123456789012"
    },
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:47:08Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSetEventDestination",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "EventDestinationName":"CloudWatchEventDestination",
        "ConfigurationSetName":"MyConfigurationSet",
        "EventDestination":{
```

```
          "Enabled":true,
          "MatchingEventTypes":[
            "INITIATED_CALL",
            "INITIATED_CALL"
          ],
          "CloudWatchLogsDestination":{
            "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
            "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
          }
        }
      },
      "responseElements":null,
      "requestID":"81de1e73-e20d-11e8-b158-d5536example",
      "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"111122223333"
    }
  ]
}
```

를 사용하여 Amazon Pinpoint에서 추천 모델 사용 AWS Lambda

Amazon Pinpoint에서는 추천 모델에서 맞춤형 추천을 검색하여 캠페인 및 여정에서 보내는 메시지에 추가할 수 있습니다. 추천 모델은 데이터에서 패턴을 찾고 찾은 패턴을 기반으로 예측 및 추천을 생성하는 기계 학습(ML) 모델의 한 유형입니다. 특정 사용자가 특정 제품 또는 항목 집합에서 선호할 것을 예측하고, 해당 정보를 사용자에게 추천 집합으로 제공합니다.

Amazon Pinpoint에서 추천 모델을 사용하면 각 수신자의 속성 및 동작에 따라 메시지 수신자에게 맞춤형 추천을 보낼 수 있습니다. 를 사용하면 이러한 권장 사항을 사용자 지정하고 개선할 수도 AWS Lambda있습니다. 예를 들어 추천을 단일 텍스트 값(예: 제품 이름 또는 ID)에서 보다 정교한 콘텐트(예: 제품 이름, 설명 및 이미지)로 동적으로 변환할 수 있습니다. 또한 Amazon Pinpoint에서 메시지를 보낼 때 실시간으로 이를 수행할 수 있습니다.

이 기능은 미국 동부(버지니아 북부), 미국 서부(오레곤), 아시아 태평양(뭄바이), 아시아 태평양(시드니), 유럽(아일랜드) AWS 리전에서 사용할 수 있습니다.

Amazon Pinpoint의 메시지에 추천자 모델 추천 추가

Amazon Pinpoint에서 추천 모델을 사용하려면 먼저 Amazon Personalize 솔루션을 만들고 이 솔루션을 Amazon Personalize 캠페인으로 배포합니다. 그런 다음 Amazon Pinpoint에서 추천 모델에 대한 구성을 만듭니다. 구성에서 Amazon Personalize 캠페인에서 권장 데이터를 검색하고 처리하는 방법을 결정하는 설정을 지정합니다. 여기에는 검색된 데이터의 추가 처리를 수행하기 위해 AWS Lambda 함수를 호출할지 여부가 포함됩니다.

Personalize는 애플리케이션을 사용하는 고객에게 실시간 맞춤형 추천을 제공하는 ML 모델을 생성할 수 있도록 설계된 AWS 서비스입니다. Amazon Personalize는 ML 모델을 생성 및 교육한 다음, 해당 모델을 Amazon Personalize 캠페인으로 준비 및 배포하는 프로세스를 안내합니다. 그런 다음 캠페인에서 실시간 맞춤형 추천을 검색할 수 있습니다. Amazon Personalize에 대한 자세한 내용은 Amazon Personalize 개발자 안내서를 참조하세요.

AWS Lambda 는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행하는 데 사용할 수 있는 컴퓨팅 서비스입니다. 코드를 패키징하여에 Lambda 함수 AWS Lambda 로 업로드합니다. AWS Lambda 그런 다음 함수가 호출될 때 함수를 실행합니다. 함수는 이벤트에 대한 응답으로 또는 애플리케이션 또는 서비스(예: Amazon Pinpoint)의 요청에 대한 응답으로 사용자가 수동으로 또는 자동으로 간접 호출할 수 있습니다. Lambda 함수 생성 및 호출에 대한 자세한 내용은 AWS Lambda 개발자 안내서를 참조하세요.

추천 모델에 대한 Amazon Pinpoint 구성을 생성한 후 캠페인 및 여정에서 보내는 메시지에 모델의 권장 사항을 추가할 수 있습니다. 이렇게 하려면 추천 속성에 대한 메시지 변수가 포함된 메시지 템플릿을 사용합니다. 추천 속성은 추천 데이터를 저장하도록 설계된 동적 엔드포인트 또는 사용자 속성입니다. 추천 모델에 대한 구성을 만들 때 이러한 속성을 정의합니다.

다음과 같은 유형의 메시지 템플릿에서 추천 속성에 대한 변수를 사용할 수 있습니다.

- 이메일 템플릿 - 캠페인 또는 여정에서 보내는 이메일 메시지용입니다.
- 푸시 알림 템플릿 - 캠페인에서 보내는 푸시 알림용입니다.
- SMS 템플릿 - 캠페인에서 보내는 SMS 문자 메시지용입니다.

Amazon Pinpoint에서 추천 모델을 사용하는 방법에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 기계 학습 모델 섹션을 참조하세요.

추천 데이터를 처리하는 Lambda 함수를 간접적으로 호출하도록 Amazon Pinpoint를 구성한 경우, Amazon Pinpoint는 캠페인 또는 여정에 대한 메시지로 맞춤형 추천을 보낼 때마다 다음과 같은 일반 작업을 수행합니다.

1. 메시지 및 메시지 템플릿의 구성 설정 및 내용을 평가하고 처리합니다.
2. 메시지 템플릿이 추천 모델에 연결되어 있는지 확인합니다.
3. 모델에 연결하고 사용하기 위한 구성 설정을 평가합니다. 이는 모델의 추천 모델 리소스에 의해 정의됩니다.
4. 모델의 구성 설정에 의해 정의된 추천 속성에 대해 하나 이상의 메시지 변수를 감지합니다.
5. 모델의 구성 설정에 지정된 Amazon Personalize 캠페인에서 추천 데이터를 검색합니다. 이는 Amazon Personalize 런타임 API의 GetRecommendations 작업을 사용하여 이 태스크를 수행합니다.
6. 각 메시지 수신자에 대해 적절한 추천 데이터를 동적 추천 속성(`RecommendationItems`)에 추가합니다.
7. Lambda 함수를 간접적으로 호출하고 처리를 위해 각 수신자에 대한 추천 데이터를 해당 함수에 보냅니다.

   데이터는 각 수신자에 대한 엔드포인트 정의가 포함된 JSON 객체로 전송됩니다. 각 엔드포인트 정의에는 1~5개의 값으로 이뤄진 정렬된 배열을 포함하는 `RecommendationItems` 필드가 포함되어 있습니다. 배열의 값 수는 모델의 구성 설정에 따라 다릅니다.
8. Lambda 함수가 데이터를 처리하고 결과를 반환할 때까지 기다립니다.

결과는 각 수신자에 대해 업데이트된 엔드포인트 정의를 포함하는 JSON 객체입니다. 업데이트된 각 엔드포인트 정의에는 새 `Recommendations` 객체가 포함됩니다. 이 객체에는 모델의 구성 설정에서 정의한 각 사용자 지정 추천 속성에 대해 하나씩 1~10개의 필드가 포함되어 있습니다. 이러한 각 필드에는 엔드포인트에 대한 개선된 추천 데이터가 저장됩니다.

9. 각 수신자에 대해 업데이트된 엔드포인트 정의를 사용하여 각 메시지 변수를 해당 수신자에 대한 적절한 값으로 바꿉니다.
10. 각 메시지 수신자에 대한 맞춤형 추천이 포함된 메시지 버전을 보냅니다.

이러한 방식으로 추천을 사용자 지정하고 개선하려면 먼저 Amazon Pinpoint에서 보낸 엔드포인트 정의를 처리하고 업데이트된 엔드포인트 정의를 반환하는 Lambda 함수를 만듭니다. 다음으로 Lambda 함수 정책을 함수에 할당하고 함수를 간접적으로 호출하도록 Amazon Pinpoint에 권한을 부여합니다. 그런 다음 Amazon Pinpoint에서 추천 모델을 구성합니다. 모델을 구성할 때 간접적으로 호출할 함수를 지정하고 사용할 추천 속성을 정의합니다.

Amazon Pinpoint에 대한 Lambda 함수를 생성하여 추천자 모델에 대해 간접적으로 호출

Lambda 함수를 생성하는 방법을 알아보려면 AWS Lambda 개발자 안내서의 시작하기를 참조하세요. 함수를 설계하고 개발할 때 다음 요구 사항과 지침을 염두에 두십시오.

입력 이벤트 데이터

Amazon Pinpoint는 추천 모델에 대한 Lambda 함수를 간접적으로 호출할 때 메시지를 보내는 캠페인 또는 여정에 대한 구성 및 기타 설정이 포함된 페이로드를 보냅니다. 페이로드에는 메시지 수신자에 대한 엔드포인트 정의와 엔드포인트 ID를 연결하는 맵인 `Endpoints` 객체가 포함됩니다.

엔드포인트 정의는 Amazon Pinpoint API의 엔드포인트 리소스에서 정의한 구조를 사용합니다. 그러나 여기에는 `RecommendationItems`라는 동적 추천 속성에 대한 필드도 포함됩니다. `RecommendationItems` 필드에는 Amazon Personalize 캠페인에서 반환된 엔드포인트에 대한 추천 항목이 하나 이상 포함됩니다. 이 필드의 값은 1~5개 추천 항목(문자열)의 정렬된 배열입니다. 배열의 항목 수는 각 엔드포인트 또는 사용자에 대해 검색하도록 Amazon Pinpoint를 구성한 추천 항목 수에 따라 다릅니다.

예제:

```
"Endpoints": {
    "endpointIDexample-1":{
```

```
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z"
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
```

```
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z"
    }
}
```

앞의 예에서 관련 Amazon Pinpoint 설정은 다음과 같습니다.

- 추천 모델은 각 엔드포인트 또는 사용자에 대해 세 가지 추천 항목을 검색하도록 구성됩니다. (RecommendationsPerMessage 속성 값은 3으로 설정됩니다.) 이 설정을 사용하면 Amazon Pinpoint에서는 각 엔드포인트 또는 사용자에 대해 첫 번째, 두 번째 및 세 번째 추천 항목만 검색하고 추가합니다.
- 프로젝트는 각 사용자의 이름, 성 및 거주하는 지역을 저장하는 사용자 지정 사용자 속성을 사용하도록 구성됩니다. (UserAttributes 객체에는 이러한 속성의 값이 포함되어 있습니다.)
- 프로젝트는 해당 엔드포인트가 프로젝트에서 메시지를 수신하기 위한 사용자의 기본 주소(채널)인지 여부를 나타내는 사용자 지정 엔드포인트 속성(AddressType)을 사용하도록 구성됩니다. (Attributes 객체에는 이 속성의 값이 포함되어 있습니다.)

Amazon Pinpoint에서 Lambda 함수를 간접적으로 호출하고 이 페이로드를 이벤트 데이터로 전송하면 AWS Lambda 에서 처리를 위해 Lambda 함수에 데이터를 전달합니다.

각 페이로드는 최대 50개의 엔드포인트에 대한 데이터를 포함할 수 있습니다. 세그먼트에 50개 이상의 엔드포인트가 포함된 경우, Amazon Pinpoint는 함수가 모든 데이터를 처리할 때까지 한 번에 최대 50개의 엔드포인트에 대해 함수를 반복적으로 간접 호출합니다.

응답 데이터 및 요구 사항

Lambda 함수를 설계하고 개발할 경우 기계 학습 모델에 대한 할당량을 염두에 둡니다. 함수가 이러한 할당량에 의해 정의된 조건을 충족하지 않으면 Amazon Pinpoint에서 메시지를 처리하고 보낼 수 없습니다.

또한 다음 요구 사항을 염두에 두십시오.

- 함수는 입력 이벤트 데이터에 의해 제공된 것과 동일한 형식으로 업데이트된 엔드포인트 정의를 반환해야 합니다.
- 업데이트된 각 엔드포인트 정의에는 엔드포인트 또는 사용자에 대해 1~10개의 사용자 지정 추천 속성이 포함될 수 있습니다. 이러한 속성의 이름은 Amazon Pinpoint에서 추천 모델을 구성할 때 지정하는 속성 이름과 일치해야 합니다.
- 모든 사용자 지정 추천 속성은 각 엔드포인트 또는 사용자에 대해 단일 `Recommendations` 객체로 반환되어야 합니다. 이 요구 사항은 이름 충돌이 발생하지 않도록 하는 데 도움이 됩니다. 엔드포인트 정의의 모든 위치에 `Recommendations` 객체를 추가할 수 있습니다.
- 각 사용자 지정 추천 속성의 값은 문자열(단일 값) 또는 문자열 배열(다중 값)이어야 합니다. 값이 문자열 배열인 경우 `RecommendationItems` 필드에 표시된 대로 Amazon Personalize가 반환한 추천 항목의 순서를 유지하는 것이 좋습니다. 그렇지 않으면 콘텐츠가 엔드포인트 또는 사용자에 대한 모델의 예측을 반영하지 않을 수 있습니다.
- 함수는 엔드포인트 또는 사용자에 대한 다른 속성 값을 포함하여 이벤트 데이터의 다른 요소를 수정해서는 안 됩니다. 사용자 지정 추천 속성에 대한 값만 추가하고 반환해야 합니다. Amazon Pinpoint는 함수의 응답에 있는 다른 값에 대한 업데이트는 수락하지 않습니다.
- 함수는 함수를 호출하는 Amazon Pinpoint 프로젝트와 동일한 AWS 리전에서 호스팅되어야 합니다. 함수와 프로젝트가 동일한 리전에 있지 않으면 Amazon Pinpoint에서 이벤트 데이터를 함수로 보낼 수 없습니다.

위의 요구 사항 중 하나라도 충족되지 않으면 Amazon Pinpoint에서 메시지를 처리하고 하나 이상의 엔드포인트로 보낼 수 없습니다. 이로 인해 캠페인 또는 여정 활동이 실패할 수 있습니다.

마지막으로 함수에 대해 256개의 동시 실행을 예약하는 것이 좋습니다.

전반적으로 Lambda 함수는 Amazon Pinpoint에서 보낸 이벤트 데이터를 처리하고 수정된 엔드포인트 정의를 반환해야 합니다. `Endpoints` 객체의 각 엔드포인트를 반복하고 각 엔드포인트에 대해 사용하려는 사용자 지정 추천 속성에 대한 값을 만들고 설정하여 이 작업을 수행할 수 있습니다. Python으로 작성되어 앞의 입력 이벤트 데이터 예제를 계속 진행하는 다음 예제 핸들러가 이를 보여 줍니다.

```
import json
import string

def lambda_handler(event, context):
    print("Received event: " + json.dumps(event))
    print("Received context: " +  str(context))
    segment_endpoints = event["Endpoints"]
    new_segment = dict()
    for endpoint_id in segment_endpoints.keys():
        endpoint = segment_endpoints[endpoint_id]
        if supported_endpoint(endpoint):
            new_segment[endpoint_id] = add_recommendation(endpoint)

    print("Returning endpoints: " + json.dumps(new_segment))
    return new_segment

def supported_endpoint(endpoint):
    return True

def add_recommendation(endpoint):
    endpoint["Recommendations"] = dict()

    customTitleList = list()
    customGenreList = list()
    for i,item in enumerate(endpoint["RecommendationItems"]):
        item = int(item)
        if item == 1210:
            customTitleList.insert(i, "Hanna")
            customGenreList.insert(i, "Action")
        elif item == 1527:
            customTitleList.insert(i, "Catastrophe")
            customGenreList.insert(i, "Comedy")
        elif item == 1815:
            customTitleList.insert(i, "Fleabag")
            customGenreList.insert(i, "Comedy")
        elif item == 2009:
            customTitleList.insert(i, "Late Night")
            customGenreList.insert(i, "Drama")
        elif item == 4582:
            customTitleList.insert(i, "Agatha Christie\'s The ABC Murders")
            customGenreList.insert(i, "Crime")
        elif item == 6542:
            customTitleList.insert(i, "Hunters")
```

```
            customGenreList.insert(i, "Drama")

    endpoint["Recommendations"]["Title"] = customTitleList
    endpoint["Recommendations"]["Genre"] = customGenreList

    return endpoint
```

이전 예제에서는 이벤트 데이터를 `event` 파라미터로 핸들러에 AWS Lambda 전달합니다. 핸들러는 `Endpoints` 객체의 각 엔드포인트를 반복하고 `Recommendations.Title` 및 `Recommendations.Genre`라는 사용자 지정 추천 속성의 값을 설정합니다. `return` 문은 업데이트된 각 엔드포인트 정의를 Amazon Pinpoint로 반환합니다.

앞의 입력 이벤트 데이터 예제를 계속 진행하면서 업데이트된 엔드포인트 정의는 다음과 같습니다.

```
"Endpoints":{
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
```

```
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z",
        "Recommendations":{
            "Title":[
                "Fleabag",
                "Late Night",
                "Catastrophe"
            ],
            "Genre":[
                "Comedy",
                "Comedy",
                "Comedy"
            ]
        }
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
            "6542",
```

```
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z",
        "Recommendations":{
            "Title":[
                "Hanna",
                "Hunters",
                "Agatha Christie\'s The ABC Murders"
            ],
            "Genre":[
                "Action",
                "Drama",
                "Crime"
            ]
        }
    }
}
```

앞의 예제에서 함수는 결과를 수신하고 반환한 `Endpoints` 객체를 수정했습니다. 이제 각 엔드포인트의 `Endpoint` 객체에 `Title` 및 `Genre` 필드가 포함된 새 `Recommendations` 객체가 포함됩니다. 이러한 각 필드는 세 개의 정렬된 값 배열(문자열)을 저장합니다. 여기서 각 값은 `RecommendationItems` 필드의 해당 추천 항목에 대한 개선된 내용을 제공합니다.

Amazon Pinpoint가 추천 데이터를 처리하도록 권한을 부여하는 Lambda 함수 정책 할당

Lambda 함수를 사용하여 추천 데이터를 처리하려면 먼저 함수를 간접적으로 호출할 권한을 Amazon Pinpoint에 부여해야 합니다. 호출 권한을 부여하려면 함수에 Lambda 함수 정책을 할당합니다. Lambda 함수 정책은 리소스 기반 권한 정책으로, 함수를 사용할 수 있는 개체와 이러한 엔터티가 실행할 수 있는 작업을 지정합니다. 자세한 내용은 AWS Lambda 개발자 안내서에서 AWS Lambda에 리소스 기반 정책 사용을 참조하세요.

아래의 예제 정책은 Amazon Pinpoint 서비스 주체가 특정 Amazon Pinpoint 프로젝트(*projectId*)에서 특정 Amazon Pinpoint 캠페인(*campaignId*) 에 대한 `lambda:InvokeFunction` 작업을 사용할 수 있도록 허용합니다.

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
```

```
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

함수 정책에는 AWS:SourceArn 키를 포함하는 Condition 블록이 필요합니다. 이 키는 함수를 간접적으로 호출할 수 있는 리소스를 지정합니다. 앞의 예에서 정책은 하나의 특정 캠페인이 함수를 간접적으로 호출하도록 허용합니다.

또한 Amazon Pinpoint 서비스 주체가 특정 Amazon Pinpoint 프로젝트(*projectId*)의 모든 캠페인 및 여정에 lambda:InvokeFunction 작업을 사용할 수 있도록 허용하는 정책을 작성할 수 있습니다. 다음 예제 정책은 이를 보여 줍니다.

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

첫 번째 예제와 달리 이 예제의 Condition 블록에 있는 AWS:SourceArn 키를 사용하면 한 특정 프로젝트에서 함수를 간접적으로 호출할 수 있습니다. 이 권한은 프로젝트의 모든 캠페인 및 여정에 적용됩니다.

더 일반적인 정책을 작성하려면 다중 문자 매칭 와일드카드(*)를 사용하면 됩니다. 예를 들어 아래의 Condition 블록을 사용하여 모든 Amazon Pinpoint 프로젝트에서 함수를 간접적으로 호출할 수 있습니다.

```
"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
```

Amazon Pinpoint 계정에 대한 모든 프로젝트에서 Lambda 함수를 사용하려면 앞의 방법으로 정책의 Condition 블록을 구성하는 것이 좋습니다. 그러나 특정 리소스에 대해 특정 작업을 수행하는 데 필요한 권한만 포함하는 정책을 만드는 것이 가장 좋습니다.

AWS CLI 및 Lambda add-permission 명령을 사용하여 Amazon Pinpoint가 Lambda 함수를 호출하도록 권한 부여

Lambda 함수 정책을 함수에 할당한 후 Amazon Pinpoint에서 특정 프로젝트, 캠페인 또는 여정에 대해 함수를 간접적으로 호출할 수 있도록 하는 권한을 추가할 수 있습니다. AWS Command Line Interface (AWS CLI) 및 Lambda add-permission 명령을 사용하여이 작업을 수행할 수 있습니다. 다음 예제에서는 특정 프로젝트(*projectId*)에 대해 이 작업을 수행하는 방법을 보여 줍니다.

```
$ aws lambda add-permission \
--function-name function-name \
--statement-id sid \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*
```

앞의 예제는 Unix, Linux, macOS용 형식으로 지정된 예제입니다. Microsoft Windows의 경우 백슬래시 (\) 줄 연속 문자를 캐럿(^)으로 바꿉니다.

이 명령이 제대로 실행되면 다음과 비슷한 출력이 표시됩니다.

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
```

```
        \"arn:aws:mobiletargeting:us-east-1:111122223333:recommenders/*\"}}}"
}
```

`Statement` 값은 Lambda 함수 정책에 추가된 문의 JSON 문자열 버전입니다.

추천자 모델에 대한 Lambda 함수를 간접적으로 호출하도록 Amazon Pinpoint 구성

추천 모델에 대한 Lambda 함수를 간접적으로 호출하도록 Amazon Pinpoint를 구성하려면 모델에 대해 다음과 같은 Lambda 관련 구성 설정을 지정합니다.

- `RecommendationTransformerUri` - 이 속성은 Lambda 함수의 이름 또는 Amazon 리소스 이름(ARN)을 지정합니다.
- `Attributes` - 이 객체는 함수가 각 엔드포인트 정의에 추가하는 사용자 지정 추천 속성을 정의하는 맵입니다. 이러한 각 속성은 메시지 템플릿에서 메시지 변수로 사용할 수 있습니다.

Amazon Pinpoint API의 추천 모델 리소스(모델에 대한 구성을 생성할 경우) 또는 Amazon Pinpoint API의 추천 모델 모델 리소스(모델의 구성을 업데이트할 경우)를 사용하여 이러한 설정을 지정할 수 있습니다. Amazon Pinpoint 콘솔을 사용하여 이러한 설정을 정의할 수도 있습니다.

Amazon Pinpoint에서 추천 모델을 사용하는 방법에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 기계 학습 모델 섹션을 참조하세요.

Amazon Pinpoint 프로젝트 삭제 및 민감한 개인 데이터 제거

사용 방식에 따라 Amazon Pinpoint는 개인 데이터로 간주되는 특정 데이터를 저장할 수 있습니다. 예를 들어 Amazon Pinpoint의 엔드포인트에는 최종 사용자의 연락처 정보(그 사람의 이메일 주소나 휴대폰 번호 등)가 들어 있습니다.

콘솔이나 Amazon Pinpoint API를 사용하여 개인 데이터를 영구적으로 삭제할 수 있습니다. 이 주제에는 개인 데이터로 간주될 수 있는 다양한 유형의 데이터를 삭제하는 절차가 포함되어 있습니다.

AWS 계정을 완전히 닫을 수도 있습니다. 자세한 내용은 AWS Account Management 참조 안내서의 AWS 계정 닫기를 참조하세요.

Amazon Pinpoint 프로젝트 데이터 삭제

Amazon Pinpoint 프로젝트에 대해 저장한 모든 데이터를 영구적으로 삭제할 수 있습니다. 프로젝트를 삭제하면 됩니다.

> **Warning**
>
> 프로젝트를 삭제하면 Amazon Pinpoint는 프로젝트의 모든 프로젝트별 설정 및 데이터를 삭제합니다. 정보는 복구할 수 없습니다.

프로젝트를 삭제하면 Amazon Pinpoint가 푸시 알림 및 양방향 SMS 메시징 채널의 모든 프로젝트 관련 설정과 다음과 같이 Amazon Pinpoint에 저장된 모든 세그먼트, 캠페인, 여정 및 프로젝트 관련 분석 데이터를 삭제합니다.

- 세그먼트 - 모든 세그먼트 설정 및 데이터. 동적 세그먼트의 경우, 여기에는 정의한 세그먼트 그룹과 필터가 포함됩니다. 가져온 세그먼트의 경우, 여기에는 가져온 엔드포인트, 사용자 ID 및 기타 데이터와 적용한 필터가 포함됩니다.
- 캠페인 - 모든 메시지, 메시지 처리 및 변수, 분석 데이터, 일정, 기타 설정.
- 여정 - 모든 활동, 분석 데이터, 일정, 기타 설정.
- 분석 - 캠페인 및 여정에 대해 전송 및 전달된 메시지의 수, 모든 여정 실행 지표와 같은 모든 참여 지표에 대한 데이터. 모바일 및 웹 앱의 경우 다른 AWS 서비스로 스트리밍되지 않은 모든 이벤트 데이터(예: Amazon Kinesis, 모든 퍼널, 애플리케이션 사용량 데이터, 수익, 인구 통계 지표 등). 프로젝트를 삭제하기 전에 이 데이터를 다른 위치로 내보내는 것이 좋습니다.

Amazon Pinpoint 콘솔을 사용하여 프로젝트를 삭제할 수 있습니다. 자세한 내용은 Amazon Pinpoint 사용 설명서의 프로젝트 삭제 섹션을 참조하세요. Amazon Pinpoint API의 앱 리소스를 사용하여 프로그래밍 방식으로 프로젝트를 삭제할 수도 있습니다.

AWS SDK를 사용한 Amazon Pinpoint용 코드 예제

다음 코드 예제에서는 Amazon Pinpoint를 AWS 소프트웨어 개발 키트(SDK)와 함께 사용하는 방법을 보여줍니다.

AWS SDK 개발자 가이드 및 코드 예시의 전체 목록은 AWS SDK에서 Amazon Pinpoint 사용 섹션을 참조하세요. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

코드 예시

- AWS SDKs를 사용하는 Amazon Pinpoint의 코드 예제
  - AWS SDKs 사용한 Amazon Pinpoint의 기본 예제
    - AWS SDKs를 사용한 Amazon Pinpoint 작업
      - AWS SDK 또는 CLI와 CreateApp 함께 사용
      - AWS SDK와 CreateCampaign 함께 사용
      - AWS SDK와 CreateExportJob 함께 사용
      - AWS SDK와 CreateImportJob 함께 사용
      - AWS SDK와 CreateSegment 함께 사용
      - AWS SDK 또는 CLI와 DeleteApp 함께 사용
      - AWS SDK와 DeleteEndpoint 함께 사용
      - AWS SDK 또는 CLI와 GetEndpoint 함께 사용
      - AWS SDK와 GetSegments 함께 사용
      - AWS SDK 또는 CLI와 GetSmsChannel 함께 사용
      - AWS SDK와 GetUserEndpoints 함께 사용
      - AWS SDK 또는 CLI와 SendMessages 함께 사용
      - AWS SDK와 UpdateEndpoint 함께 사용
  - AWS SDKs를 사용한 Amazon Pinpoint 시나리오
    - End User Messaging Push 시작하기
- AWS SDKs를 사용한 Amazon Pinpoint SMS 및 음성 API의 코드 예제
  - AWS SDKs를 사용한 Amazon Pinpoint SMS 및 음성 API의 기본 예제
    - AWS SDKs를 사용하는 Amazon Pinpoint SMS 및 음성 API에 대한 작업
      - AWS SDK와 CreateConfigurationSet 함께 사용

- AWS SDK와 CreateConfigurationSetEventDestination 함께 사용
- AWS SDK와 DeleteConfigurationSet 함께 사용
- AWS SDK와 DeleteConfigurationSetEventDestination 함께 사용
- AWS SDK와 GetConfigurationSetEventDestinations 함께 사용
- AWS SDK와 ListConfigurationSets 함께 사용
- AWS SDK와 SendVoiceMessage 함께 사용
- AWS SDK와 UpdateConfigurationSetEventDestination 함께 사용

AWS SDKs를 사용하는 Amazon Pinpoint의 코드 예제

다음 코드 예제에서는 Amazon Pinpoint를 AWS 소프트웨어 개발 키트(SDK)와 함께 사용하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

Amazon Pinpoint

- AWS SDKs 사용한 Amazon Pinpoint의 기본 예제
  - AWS SDKs를 사용한 Amazon Pinpoint 작업
    - AWS SDK 또는 CLI와 CreateApp 함께 사용
    - AWS SDK와 CreateCampaign 함께 사용
    - AWS SDK와 CreateExportJob 함께 사용
    - AWS SDK와 CreateImportJob 함께 사용
    - AWS SDK와 CreateSegment 함께 사용
    - AWS SDK 또는 CLI와 DeleteApp 함께 사용
    - AWS SDK와 DeleteEndpoint 함께 사용
    - AWS SDK 또는 CLI와 GetEndpoint 함께 사용

        - AWS SDK와 GetSegments 함께 사용
        - AWS SDK 또는 CLI와 GetSmsChannel 함께 사용
        - AWS SDK와 GetUserEndpoints 함께 사용
        - AWS SDK 또는 CLI와 SendMessages 함께 사용
        - AWS SDK와 UpdateEndpoint 함께 사용
- AWS SDKs를 사용한 Amazon Pinpoint 시나리오
    - End User Messaging Push 시작하기

AWS SDKs 사용한 Amazon Pinpoint의 기본 예제

다음 코드 예제에서는 AWS SDK와 함께 Amazon Pinpoint의 기본 사항을 사용하는 방법을 보여줍니다.

예제

- AWS SDKs를 사용한 Amazon Pinpoint 작업
    - AWS SDK 또는 CLI와 CreateApp 함께 사용
    - AWS SDK와 CreateCampaign 함께 사용
    - AWS SDK와 CreateExportJob 함께 사용
    - AWS SDK와 CreateImportJob 함께 사용
    - AWS SDK와 CreateSegment 함께 사용
    - AWS SDK 또는 CLI와 DeleteApp 함께 사용
    - AWS SDK와 DeleteEndpoint 함께 사용
    - AWS SDK 또는 CLI와 GetEndpoint 함께 사용
    - AWS SDK와 GetSegments 함께 사용
    - AWS SDK 또는 CLI와 GetSmsChannel 함께 사용
    - AWS SDK와 GetUserEndpoints 함께 사용
    - AWS SDK 또는 CLI와 SendMessages 함께 사용
    - AWS SDK와 UpdateEndpoint 함께 사용

AWS SDKs를 사용한 Amazon Pinpoint 작업

다음 코드 예제에서는 AWS SDKs를 사용하여 개별 Amazon Pinpoint 작업을 수행하는 방법을 보여줍니다. 각 예제에는 GitHub에 대한 링크가 포함되어 있습니다. 여기에서 코드 설정 및 실행에 대한 지침을 찾을 수 있습니다.

이들 발췌문은 Amazon Pinpoint API를 직접적으로 호출하며, 컨텍스트에서 실행되어야 하는 더 큰 프로그램에서 발췌한 코드입니다. AWS SDKs를 사용한 Amazon Pinpoint 시나리오에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

다음 예제에는 가장 일반적으로 사용되는 작업만 포함되어 있습니다. 전체 목록은 Amazon Pinpoint API 참조를 살펴보세요.

예제

- AWS SDK 또는 CLI와 CreateApp 함께 사용
- AWS SDK와 CreateCampaign 함께 사용
- AWS SDK와 CreateExportJob 함께 사용
- AWS SDK와 CreateImportJob 함께 사용
- AWS SDK와 CreateSegment 함께 사용
- AWS SDK 또는 CLI와 DeleteApp 함께 사용
- AWS SDK와 DeleteEndpoint 함께 사용
- AWS SDK 또는 CLI와 GetEndpoint 함께 사용
- AWS SDK와 GetSegments 함께 사용
- AWS SDK 또는 CLI와 GetSmsChannel 함께 사용
- AWS SDK와 GetUserEndpoints 함께 사용
- AWS SDK 또는 CLI와 SendMessages 함께 사용
- AWS SDK와 UpdateEndpoint 함께 사용

AWS SDK 또는 CLI와 **CreateApp** 함께 사용

다음 코드 예시는 `CreateApp`의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- End User Messaging Push 시작하기

CLI

AWS CLI

예시 1: 애플리케이션 생성

다음 create-app 예시에서는 새 애플리케이션(프로젝트)을 생성합니다.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

출력:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

예시 2: 태그가 지정된 애플리케이션을 생성하는 방법

다음 create-app 예시에서는 새 애플리케이션(프로젝트)을 만들고 태그(키 및 값)를 애플리케이션에 연결합니다.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

출력:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
```

```
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateApp을 참조하세요.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateAppRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateAppResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateApplicationRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appName>

                Where:
                 appName - The name of the application to create.
```

```
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
        String appName = args[0];
        System.out.println("Creating an application with name: " + appName);

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String appID = createApplication(pinpoint, appName);
        System.out.println("App ID is: " + appID);
        pinpoint.close();
    }

    public static String createApplication(PinpointClient pinpoint, String
 appName) {
        try {
            CreateApplicationRequest appRequest =
CreateApplicationRequest.builder()
                    .name(appName)
                    .build();

            CreateAppRequest request = CreateAppRequest.builder()
                    .createApplicationRequest(appRequest)
                    .build();

            CreateAppResponse result = pinpoint.createApp(request);
            return result.applicationResponse().id();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 CreateApp을 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
suspend fun createApplication(applicationName: String?): String? {
    val createApplicationRequestOb =
        CreateApplicationRequest {
            name = applicationName
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.createApp(
                CreateAppRequest {
                    createApplicationRequest = createApplicationRequestOb
                },
            )
        return result.applicationResponse?.id
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 CreateApp을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **`CreateCampaign`** 함께 사용

다음 코드 예시는 `CreateCampaign`의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

캠페인을 생성합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>

                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
```

```
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId,
String segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
    }

    public static CampaignResponse createCampaign(PinpointClient client, String
appID, String segmentID) {

        try {
            Schedule schedule = Schedule.builder()
                    .startTime("IMMEDIATE")
                    .build();

            Message defaultMessage = Message.builder()
                    .action(Action.OPEN_APP)
                    .body("My message body.")
                    .title("My message title.")
                    .build();

            MessageConfiguration messageConfiguration =
MessageConfiguration.builder()
                    .defaultMessage(defaultMessage)
                    .build();

            WriteCampaignRequest request = WriteCampaignRequest.builder()
```

```
                    .description("My description")
                    .schedule(schedule)
                    .name("MyCampaign")
                    .segmentId(segmentID)
                    .messageConfiguration(messageConfiguration)
                    .build();

            CreateCampaignResponse result =
 client.createCampaign(CreateCampaignRequest.builder()
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());

            System.out.println("Campaign ID: " + result.campaignResponse().id());
            return result.campaignResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 CreateCampaign을 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
suspend fun createPinCampaign(
    appId: String,
    segmentIdVal: String,
) {
    val scheduleOb =
```

```
        Schedule {
            startTime = "IMMEDIATE"
        }

    val defaultMessageOb =
        Message {
            action = Action.OpenApp
            body = "My message body"
            title = "My message title"
        }

    val messageConfigurationOb =
        MessageConfiguration {
            defaultMessage = defaultMessageOb
        }

    val writeCampaign =
        WriteCampaignRequest {
            description = "My description"
            schedule = scheduleOb
            name = "MyCampaign"
            segmentId = segmentIdVal
            messageConfiguration = messageConfigurationOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result: CreateCampaignResponse =
            pinpoint.createCampaign(
                CreateCampaignRequest {
                    applicationId = appId
                    writeCampaignRequest = writeCampaign
                },
            )
        println("Campaign ID is ${result.campaignResponse?.id}")
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 CreateCampaign를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **`CreateExportJob`** 함께 사용

다음 코드 예시는 `CreateExportJob`의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

엔드포인트를 내보내세요.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
```

```
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and
 prints them.
                Usage: ExportEndpoints <applicationId> <s3BucketName>
 <iamExportRoleArn> <path>

                Where:
                  applicationId - The ID of the Amazon Pinpoint application that
 has the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the
 JSON file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
 Pinpoint write permissions to the S3 bucket.  path - The path where the files
 downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
                """;

        if (args.length != 4) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String applicationId = args[0];
        String s3BucketName = args[1];
        String iamExportRoleArn = args[2];
        String path = args[3];
        System.out.println("Deleting an application with ID: " + applicationId);

        Region region = Region.US_EAST_1;
        PinpointClient pinpoint = PinpointClient.builder()
                .region(region)
                .build();

        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
        pinpoint.close();
        s3Client.close();
    }

    public static void exportAllEndpoints(PinpointClient pinpoint,
            S3Client s3Client,
            String applicationId,
            String s3BucketName,
            String path,
            String iamExportRoleArn) {

        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
```

```
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint,
 S3Client s3Client, String s3BucketName,
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
                    .build();

            System.out.format("Exporting endpoints from Amazon Pinpoint
 application %s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);

            CreateExportJobResponse exportResult =
 pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
```

```
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                    .jobId(jobId)
                    .applicationId(applicationId)
                    .build();

            do {
                getExportJobResult =
pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
```

```
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path
 location.
    public static void downloadFromS3(S3Client s3Client, String path, String
 s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
 s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();

                // Write the data to a local file.
                String fileSuffix = new
 SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- API에 대한 세부 정보는 AWS SDK for Java 2.x API 참조의 CreateExportJob을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **`CreateImportJob`** 함께 사용

다음 코드 예시는 `CreateImportJob`의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

세그먼트를 가져오세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
```

```
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the
 segment definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon
 Pinpoint to access S3. You need to set trust management for this
 to work. See https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket,
 key, roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for
 import job status.");
        System.out.println("See application " + response.jobStatus() + " for
 import job status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {
```

```
        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse =
 client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- API에 대한 세부 정보는 AWS SDK for Java 2.x API 참조의 CreateImportJob을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **`CreateSegment`** 함께 사용

다음 코드 예시는 `CreateSegment`의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
                final String usage = """

                                Usage:   <appId>
```

```
                                Where:
                                  appId - The application ID to create a segment
for.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();
                        segmentAttributes.put("Team",
AttributeDimension.builder()
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());

                        RecencyDimension recencyDimension =
RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors =
SegmentBehaviors.builder()
                                        .recency(recencyDimension)
                                        .build();
```

```
                        SegmentDemographics segmentDemographics =
 SegmentDemographics
                                        .builder()
                                        .build();

                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
 WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
 CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();

                        CreateSegmentResponse createSegmentResult =
 client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
 createSegmentResult.segmentResponse().id());
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
        }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 CreateSegment를 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
suspend fun createPinpointSegment(applicationIdVal: String?): String? {
    val segmentAttributes = mutableMapOf<String, AttributeDimension>()
    val myList = mutableListOf<String>()
    myList.add("Lakers")

    val atts =
        AttributeDimension {
            attributeType = AttributeType.Inclusive
            values = myList
        }

    segmentAttributes["Team"] = atts
    val recencyDimension =
        RecencyDimension {
            duration = Duration.fromValue("DAY_30")
            recencyType = RecencyType.fromValue("ACTIVE")
        }

    val segmentBehaviors =
        SegmentBehaviors {
            recency = recencyDimension
        }

    val segmentLocation = SegmentLocation {}
    val dimensionsOb =
        SegmentDimensions {
            attributes = segmentAttributes
            behavior = segmentBehaviors
```

```
            demographic = SegmentDemographics {}
            location = segmentLocation
        }

    val writeSegmentRequestOb =
        WriteSegmentRequest {
            name = "MySegment101"
            dimensions = dimensionsOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val createSegmentResult: CreateSegmentResponse =
            pinpoint.createSegment(
                CreateSegmentRequest {
                    applicationId = applicationIdVal
                    writeSegmentRequest = writeSegmentRequestOb
                },
            )
        println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
        return createSegmentResult.segmentResponse?.id
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 CreateSegment를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **`DeleteApp`** 함께 사용

다음 코드 예시는 `DeleteApp`의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- End User Messaging Push 시작하기

CLI

AWS CLI

애플리케이션 삭제

다음 `delete-app` 예시에서는 애플리케이션(프로젝트)을 삭제합니다.

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

출력:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApp을 참조하세요.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

애플리케이션을 삭제합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appId>

                Where:
                 appId - The ID of the application to delete.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        System.out.println("Deleting an application with ID: " + appId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinApp(pinpoint, appId);
        System.out.println("Done");
        pinpoint.close();
    }

    public static void deletePinApp(PinpointClient pinpoint, String appId) {
        try {
            DeleteAppRequest appRequest = DeleteAppRequest.builder()
                    .applicationId(appId)
                    .build();
```

```
            DeleteAppResponse result = pinpoint.deleteApp(appRequest);
            String appName = result.applicationResponse().name();
            System.out.println("Application " + appName + " has been deleted.");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 DeleteApp을 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
suspend fun deletePinApp(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.deleteApp(
                DeleteAppRequest {
                    applicationId = appId
                },
            )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 DeleteApp를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **DeleteEndpoint** 함께 사용

다음 코드 예시는 DeleteEndpoint의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

엔드포인트를 삭제합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appName> <endpointId >

                Where:
```

```
                  appId - The id of the application to delete.
                  endpointId - The id of the endpoint to delete.
                """";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
 String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 DeleteEndpoint를 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
suspend fun deletePinEncpoint(
    appIdVal: String?,
    endpointIdVal: String?,
) {
    val deleteEndpointRequest =
        DeleteEndpointRequest {
            applicationId = appIdVal
            endpointId = endpointIdVal
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteEndpoint(deleteEndpointRequest)
        val id = result.endpointResponse?.id
        println("The deleted endpoint is  $id")
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 DeleteEndpoint를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **`GetEndpoint`** 함께 사용

다음 코드 예시는 `GetEndpoint`의 사용 방법을 보여 줍니다.

CLI

AWS CLI

애플리케이션에 대한 특정 엔드포인트의 설정 및 속성에 대한 정보 검색

다음 get-endpoint 예시에서는 애플리케이션의 특정 엔드포인트 설정 및 속성에 대한 정보를 가져옵니다.

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

출력:

```
{
    "EndpointResponse": {
        "Address": "+11234567890",
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "Attributes": {},
        "ChannelType": "SMS",
        "CohortId": "63",
        "CreationDate": "2019-01-28T23:55:11.534Z",
        "EffectiveDate": "2021-08-06T00:04:51.763Z",
        "EndpointStatus": "ACTIVE",
        "Id": "testendpoint",
        "Location": {
            "Country": "USA"
        },
        "Metrics": {
            "SmsDelivered": 1.0
        },
        "OptOut": "ALL",
        "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",
        "User": {
            "UserAttributes": {
                "Age": [
                    "24"
                ]
            },
        "UserId": "testuser"
        }
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetEndpoint를 참조하세요.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <endpoint>

                Where:
                  appId - The ID of the application to delete.
```

```
                endpoint - The ID of the endpoint.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String
appId, String endpoint) {
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
            System.out.println(endpointJson);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
```

```
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 GetEndpoint를 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
suspend fun lookupPinpointEndpoint(
    appId: String?,
    endpoint: String?,
) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.getEndpoint(
                GetEndpointRequest {
                    applicationId = appId
                    endpointId = endpoint
                },
            )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson =
            GsonBuilder()
                .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                .setPrettyPrinting()
                .create()

        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 GetEndpoint를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **GetSegments** 함께 사용

다음 코드 예시는 GetSegments의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

세그먼트를 나열하세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListSegments {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId>

                Where:
                  appId - The ID of the application that contains a segment.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listSegs(pinpoint, appId);
        pinpoint.close();
    }

    public static void listSegs(PinpointClient pinpoint, String appId) {
        try {
            GetSegmentsRequest request = GetSegmentsRequest.builder()
                    .applicationId(appId)
                    .build();

            GetSegmentsResponse response = pinpoint.getSegments(request);
            List<SegmentResponse> segments = response.segmentsResponse().item();
            for (SegmentResponse segment : segments) {
                System.out
                        .println("Segement " + segment.id() + " " +
 segment.name() + " " + segment.lastModifiedDate());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 GetSegments를 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
suspend fun listSegs(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val response =
            pinpoint.getSegments(
                GetSegmentsRequest {
                    applicationId = appId
                },
            )
        response.segmentsResponse?.item?.forEach { segment ->
            println("Segement id is ${segment.id}")
        }
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 GetSegments를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **`GetSmsChannel`** 함께 사용

다음 코드 예시는 `GetSmsChannel`의 사용 방법을 보여 줍니다.

CLI

AWS CLI

애플리케이션의 SMS 채널 상태 및 설정에 대한 정보 검색

다음 `get-sms-channel` 예시에서는 애플리케이션의 SMS 채널 상태 및 설정을 가져옵니다.

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

출력:

```
{
    "SMSChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:39:18.511Z",
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:39:18.511Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 1
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSmsChannel을 참조하세요.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelResponse;
import software.amazon.awssdk.services.pinpoint.model.GetSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateChannel {
    public static void main(String[] args) {
        final String usage = """

                Usage: CreateChannel <appId>

                Where:
                  appId - The name of the application whose channel is updated.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        SMSChannelResponse getResponse = getSMSChannel(pinpoint, appId);
        toggleSmsChannel(pinpoint, appId, getResponse);
        pinpoint.close();
    }
```

```
   private static SMSChannelResponse getSMSChannel(PinpointClient client, String
appId) {
       try {
           GetSmsChannelRequest request = GetSmsChannelRequest.builder()
                   .applicationId(appId)
                   .build();

           SMSChannelResponse response =
client.getSmsChannel(request).smsChannelResponse();
           System.out.println("Channel state is " + response.enabled());
           return response;

       } catch (PinpointException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
       return null;
   }

   private static void toggleSmsChannel(PinpointClient client, String appId,
SMSChannelResponse getResponse) {
       boolean enabled = !getResponse.enabled();
       try {
           SMSChannelRequest request = SMSChannelRequest.builder()
                   .enabled(enabled)
                   .build();

           UpdateSmsChannelRequest updateRequest =
UpdateSmsChannelRequest.builder()
                   .smsChannelRequest(request)
                   .applicationId(appId)
                   .build();

           UpdateSmsChannelResponse result =
client.updateSmsChannel(updateRequest);
           System.out.println("Channel state: " +
result.smsChannelResponse().enabled());

       } catch (PinpointException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

```
}
```

- API에 대한 세부 정보는 AWS SDK for Java 2.x API 참조의 GetSmsChannel을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **`GetUserEndpoints`** 함께 사용

다음 코드 예시는 `GetUserEndpoints`의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListEndpointIds {
```

```
   public static void main(String[] args) {
       final String usage = """

               Usage:    <applicationId> <userId>

               Where:
                  applicationId - The ID of the Amazon Pinpoint application that
has the endpoint.
                  userId - The user id applicable to the endpoints""";

       if (args.length != 2) {
           System.out.println(usage);
           System.exit(1);
       }

       String applicationId = args[0];
       String userId = args[1];
       PinpointClient pinpoint = PinpointClient.builder()
               .region(Region.US_EAST_1)
               .build();

       listAllEndpoints(pinpoint, applicationId, userId);
       pinpoint.close();
   }

   public static void listAllEndpoints(PinpointClient pinpoint,
           String applicationId,
           String userId) {

       try {
           GetUserEndpointsRequest endpointsRequest =
GetUserEndpointsRequest.builder()
                   .userId(userId)
                   .applicationId(applicationId)
                   .build();

           GetUserEndpointsResponse response =
pinpoint.getUserEndpoints(endpointsRequest);
           List<EndpointResponse> endpoints =
response.endpointsResponse().item();

           // Display the results.
           for (EndpointResponse endpoint : endpoints) {
```

```
                System.out.println("The channel type is: " +
 endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- API에 대한 세부 정보는 AWS SDK for Java 2.x API 참조의 GetUserEndpoints를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **SendMessages** 함께 사용

다음 코드 예시는 SendMessages의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- End User Messaging Push 시작하기

.NET

SDK for .NET

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

이메일 메시지를 전송합니다.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon
 Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;

        // The Amazon Pinpoint project/application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or
 application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
```

```
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<MessageResponse> SendEmailMessage(
        string region, string appId, string toAddress, string senderAddress)
    {
        var client = new
 AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        // The subject line of the email.
        string subject = "Amazon Pinpoint Email test";

        // The body of the email for recipients whose email clients don't
        // support HTML content.
        string textBody = @"Amazon Pinpoint Email Test (.NET)"
                          + "\n---------------------------------"
                          + "\nThis email was sent using the Amazon Pinpoint API
 using the AWS SDK for .NET.";

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK
 for .NET)</h1>"
                          + "\n  <p>This email was sent using the "
                          + "\n    <a href='https://aws.amazon.com/
pinpoint/'>Amazon Pinpoint</a> API "
                          + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                          + "\n  </p>"
                          + "\n</body>"
                          + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
```

```
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
                },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
                            },
                            Subject = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await
client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
```

```
        return response.MessageResponse;
    }
}
```

SMS 메시지를 전송합니다.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone
 number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify
 the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;
```

```
        // The Pinpoint project/ application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or
 application
        // that you choose.
        string appId = configuration["AppId"]!;

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;

        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
 {messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
```

```
        string region, string appId, string destinationNumber, string
originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await
client.SendMessagesAsync(sendRequest);
        return response;
    }
```

```
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 SendMessages를 참조하세요.

CLI

AWS CLI

애플리케이션의 엔드포인트를 사용하여 SMS 메시지 전송

다음 send-messages 예시에서는 엔드포인트가 있는 애플리케이션에 다이렉트 메시지를 보냅니다.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

myfile.json의 콘텐츠:

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
        "testendpoint": {}
    }
}
```

출력:

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
```

```
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

자세한 내용은 Amazon Pinpoint 사용자 안내서의 Amazon Pinpoint SMS 채널을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendMessages를 참조하세요.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

이메일 메시지를 전송합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
```

```
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendEmailMessage {

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        public static String charset = "UTF-8";

    // The body of the email for recipients whose email clients support HTML
 content.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Java 2.x

        """;

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
 <toAddress>

            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified
 in Amazon Pinpoint in the region you're using to send email\s
               toAddress - The to address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email\s
            """;
```

```
        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        System.out.println("Sending a message");
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        sendEmail(pinpoint, subject, senderAddress, toAddress);
        System.out.println("Email was sent");
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress) {
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
```

```
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

CC 값을 포함하여 이메일 메시지를 전송합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import  software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;
import java.util.ArrayList;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendEmailMessageCC {

    // The body of the email.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)
```

```
        This email was sent through the Amazon Pinpoint Email API using the AWS
SDK for Java 2.x

        """;
    public static void main(String[] args) {
        final String usage = """

            Usage:    <subject> <senderAddress> <toAddress> <ccAddress>

            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified
in Amazon Pinpoint in the region you're using to send email\s
               toAddress - The to address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
               ccAddress - The CC address.
            """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        String ccAddress = args[3];

        System.out.println("Sending a message");
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        ArrayList<String> ccList = new ArrayList<>();
        ccList.add(ccAddress);
        sendEmail(pinpoint, subject, senderAddress, toAddress, ccList);
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress, ArrayList<String> ccAddresses)
{
        try {
            Content content = Content.builder()
```

```
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .ccAddresses(ccAddresses)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            // Handle exception
            e.printStackTrace();
        }
    }
}
```

SMS 메시지를 전송합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
```

```
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
```

```
                                appId - The Amazon Pinpoint project/application
ID to use when you send this message.
                                originationNumber - The phone number or
short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format (for example,
+1-555-555-5654).
                                destinationNumber - The recipient's phone
number.  For best results, you should specify the phone number in E.164 format
(for example, +1-555-555-5654).\s
                                """;

                if (args.length != 4) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
```

```
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
 DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request =
 SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" +
 v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

SMS 메시지를 일괄 전송합니다.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessageBatch {

    // The type of SMS message that you want to send. If you plan to send
    // time-sensitive content, specify TRANSACTIONAL. If you plan to send
    // marketing-related content, specify PROMOTIONAL.
    public static String messageType = "TRANSACTIONAL";

    // The registered keyword associated with the originating short code.
    public static String registeredKeyword = "myKeyword";

    // The sender ID to use when sending the message. Support for sender ID
    // varies by country or region. For more information, see
    // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
    public static String senderId = "MySenderID";

    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:   <message> <appId> <originationNumber>
<destinationNumber> <destinationNumber1>\s

                Where:
                  message - The body of the message to send.
                  appId - The Amazon Pinpoint project/application ID to use when
you send this message.
                  originationNumber - The phone number or short code that
you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                  destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).
                  destinationNumber1 - The second recipient's phone number.  For
best results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
                """;

       if (args.length != 5) {
           System.out.println(usage);
           System.exit(1);
       }

       String message = args[0];
       String appId = args[1];
       String originationNumber = args[2];
       String destinationNumber = args[3];
       String destinationNumber1 = args[4];
       System.out.println("Sending a message");
       PinpointClient pinpoint = PinpointClient.builder()
               .region(Region.US_EAST_1)
               .build();

       sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber, destinationNumber1);
       pinpoint.close();
   }

   public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                                     String originationNumber,
                                     String destinationNumber, String
destinationNumber1) {
```

```
        try {
            Map<String, AddressConfiguration> addressMap = new HashMap<String,
AddressConfiguration>();
            AddressConfiguration addConfig = AddressConfiguration.builder()
                    .channelType(ChannelType.SMS)
                    .build();

            // Add an entry to the Map object for each number to whom you want to
send a
            // message.
            addressMap.put(destinationNumber, addConfig);
            addressMap.put(destinationNumber1, addConfig);
            SMSMessage smsMessage = SMSMessage.builder()
                    .body(message)
                    .messageType(messageType)
                    .originationNumber(originationNumber)
                    .senderId(senderId)
                    .keyword(registeredKeyword)
                    .build();

            // Create a DirectMessageConfiguration object.
            DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                    .smsMessage(smsMessage)
                    .build();

            MessageRequest msgReq = MessageRequest.builder()
                    .addresses(addressMap)
                    .messageConfiguration(direct)
                    .build();

            // Create a SendMessagesRequest object.
            SendMessagesRequest request = SendMessagesRequest.builder()
                    .applicationId(appId)
                    .messageRequest(msgReq)
                    .build();

            SendMessagesResponse response = pinpoint.sendMessages(request);
            MessageResponse msg1 = response.messageResponse();
            Map map1 = msg1.result();

            // Write out the result of sendMessage.
            map1.forEach((k, v) -> System.out.println((k + ":" + v)));
```

```
        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 SendMessages를 참조하십시오.

JavaScript

SDK for JavaScript (v3)

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

별도의 모듈에서 클라이언트를 생성하고 내보냅니다.

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
// Set the AWS Region.
const REGION = "us-east-1";
export const pinClient = new PinpointClient({ region: REGION });
```

이메일 메시지를 전송합니다.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
const subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";
```

```
// The email body for recipients with non-HTML email clients.
const body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https://aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
const body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint Email API</a>
 using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding for the subject line and message body of the email.
const charset = "UTF-8";

const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
```

```
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

const run = async () => {
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }

    const recipientResult = MessageResponse.Result[toAddress];

    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
    }
    console.log(recipientResult.MessageId);
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

SMS 메시지를 전송합니다.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
```

```
import { pinClient } from "./libs/pinClient.js";

/* The phone number or short code to send the message from. The phone number
 or short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number.  For best results, you should specify the phone
 number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX

// The content of the SMS message.
const message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

/*The Amazon Pinpoint project/application ID to use when you send this message.
Make sure that the SMS channel is enabled for the project or application
that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXX

/* The type of SMS message that you want to send. If you plan to send
time-sensitive content, specify TRANSACTIONAL. If you plan to send
marketing-related content, specify PROMOTIONAL.*/
const messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
const registeredKeyword = "myKeyword";

/* The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html.*/

const senderId = "MySenderID";

// Specify the parameters to pass to the API.
const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
```

```
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      `Message sent!
 ${data.MessageResponse.Result[destinationNumber].StatusMessage}`,
    );
  } catch (err) {
    console.log(err);
  }
};
run();
```

- API 세부 정보는 AWS SDK for JavaScript API 참조의 SendMessages를 참조하십시오.

SDK for JavaScript (v2)

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

이메일 메시지를 전송합니다.

```
"use strict";
```

```
const AWS = require("aws-sdk");

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using
 the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
```

```
// message body of the email.
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};
```

```
//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
  }
});
```

SMS 메시지를 전송합니다.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
```

```
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
var senderId = "MySenderID";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
```

```
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

- API 세부 정보는 AWS SDK for JavaScript API 참조의 SendMessages를 참조하십시오.

Kotlin

SDK for Kotlin

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */
```

```
val body: String =
    """
    Amazon Pinpoint test (AWS SDK for Kotlin)

    This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Kotlin.

    """.trimIndent()

suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <subject> <appId> <senderAddress> <toAddress>

    Where:
        subject - The email subject to use.
        senderAddress - The from address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email
        toAddress - The to address. This address has to be verified in Amazon
 Pinpoint in the region you're using to send email
    """

    if (args.size != 3) {
        println(usage)
        exitProcess(0)
    }

    val subject = args[0]
    val senderAddress = args[1]
    val toAddress = args[2]
    sendEmail(subject, senderAddress, toAddress)
}

suspend fun sendEmail(
    subjectVal: String?,
    senderAddress: String,
    toAddressVal: String,
) {
    var content =
        Content {
            data = body
        }
```

```
    val messageBody =
        Body {
            text = content
        }

    val subContent =
        Content {
            data = subjectVal
        }

    val message =
        Message {
            body = messageBody
            subject = subContent
        }

    val destinationOb =
        Destination {
            toAddresses = listOf(toAddressVal)
        }

    val emailContent =
        EmailContent {
            simple = message
        }

    val sendEmailRequest =
        SendEmailRequest {
            fromEmailAddress = senderAddress
            destination = destinationOb
            this.content = emailContent
        }

    PinpointEmailClient.fromEnvironment { region = "us-east-1" }.use
 { pinpointemail ->
        pinpointemail.sendEmail(sendEmailRequest)
        println("Message Sent")
    }
}
```

- API 세부 정보는 AWS SDK for Kotlin API 참조의 SendMessages를 참조하십시오.

Python

SDK for Python (Boto3)

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

이메일 메시지를 전송합니다.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
 account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and
 message
```

```
                    body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
 can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data":
 html_message},
                            "TextPart": {"Charset": char_set, "Data":
 text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    app_id = "ce796be37f32f178af652b26eexample"
```

```
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -------------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python
 (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

SMS 메시지를 전송합니다.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you
 send
                   this message. The SMS channel must be enabled for the project
 or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This
 phone
                               number must be associated with your Amazon
 Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you
 send
                         time-sensitive content, specify TRANSACTIONAL. If you
 send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
```

```
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK
 for "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

기존 이메일 템플릿을 사용하여 이메일 메시지를 전송합니다.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
                         account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress":
 sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
```

```
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

기존 SMS 템플릿을 사용하여 문자 메시지를 전송합니다.

```
import logging
import boto3
from botocore.exceptions import ClientError
```

```
logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )
```

```
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 SendMessages를 참조하세요.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

이메일 메시지를 전송합니다.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
        io_simpleemail = NEW /aws1/cl_pptsimpleemail(
          io_subject = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_subject
          )
          io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_html_message
          )
          io_textpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_text_message
          )
        )
```

```
      )
    )
  )
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Email message sent successfully.' TYPE 'I'.
```

SMS 메시지를 전송합니다.

```
" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_body = iv_message
        iv_messagetype = iv_message_type
        iv_originationnumber = iv_origination_number
      )
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
```

```
  ENDIF.
ENDLOOP.

MESSAGE 'SMS message sent successfully.' TYPE 'I'.
```

기존 이메일 템플릿을 사용하여 이메일 메시지를 전송합니다.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
      )
    )
    io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
      io_emailtemplate = NEW /aws1/cl_ppttemplate(
        iv_name = iv_template_name
        iv_version = iv_template_version
      )
    )
  )
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Templated email message sent successfully.' TYPE 'I'.
```

기존 SMS 템플릿을 사용하여 문자 메시지를 전송합니다.

```
" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_messagetype = iv_message_type
        iv_originationnumber = iv_origination_number
      )
    )
    io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
      io_smstemplate = NEW /aws1/cl_ppttemplate(
        iv_name = iv_template_name
        iv_version = iv_template_version
      )
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
  ENDIF.
ENDLOOP.

MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 SendMessages를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **`UpdateEndpoint`** 함께 사용

다음 코드 예시는 `UpdateEndpoint`의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.EndpointDemographic;
import software.amazon.awssdk.services.pinpoint.model.EndpointLocation;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.UUID;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Date;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appId>

                Where:
                  appId - The ID of the application to create an endpoint for.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        EndpointResponse response = createEndpoint(pinpoint, appId);
        System.out.println("Got Endpoint: " + response.id());
        pinpoint.close();
    }

    public static EndpointResponse createEndpoint(PinpointClient client, String
 appId) {
        String endpointId = UUID.randomUUID().toString();
        System.out.println("Endpoint ID: " + endpointId);

        try {
            EndpointRequest endpointRequest = createEndpointRequestData();
            UpdateEndpointRequest updateEndpointRequest =
 UpdateEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .endpointRequest(endpointRequest)
```

```
                    .build();

            UpdateEndpointResponse updateEndpointResponse =
client.updateEndpoint(updateEndpointRequest);
            System.out.println("Update Endpoint Response: " +
updateEndpointResponse.messageBody());

            GetEndpointRequest getEndpointRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            GetEndpointResponse getEndpointResponse =
client.getEndpoint(getEndpointRequest);
            System.out.println(getEndpointResponse.endpointResponse().address());

System.out.println(getEndpointResponse.endpointResponse().channelType());

System.out.println(getEndpointResponse.endpointResponse().applicationId());

System.out.println(getEndpointResponse.endpointResponse().endpointStatus());

System.out.println(getEndpointResponse.endpointResponse().requestId());
            System.out.println(getEndpointResponse.endpointResponse().user());

            return getEndpointResponse.endpointResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static EndpointRequest createEndpointRequestData() {
        try {
            List<String> favoriteTeams = new ArrayList<>();
            favoriteTeams.add("Lakers");
            favoriteTeams.add("Warriors");
            HashMap<String, List<String>> customAttributes = new HashMap<>();
            customAttributes.put("team", favoriteTeams);

            EndpointDemographic demographic = EndpointDemographic.builder()
                    .appVersion("1.0")
```

```
                .make("apple")
                .model("iPhone")
                .modelVersion("7")
                .platform("ios")
                .platformVersion("10.1.1")
                .timezone("America/Los_Angeles")
                .build();

        EndpointLocation location = EndpointLocation.builder()
                .city("Los Angeles")
                .country("US")
                .latitude(34.0)
                .longitude(-118.2)
                .postalCode("90068")
                .region("CA")
                .build();

        Map<String, Double> metrics = new HashMap<>();
        metrics.put("health", 100.00);
        metrics.put("luck", 75.00);

        EndpointUser user = EndpointUser.builder()
                .userId(UUID.randomUUID().toString())
                .build();

        DateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm'Z'"); //
Quoted "Z" to indicate UTC, no timezone
                                                                       //
offset
        String nowAsISO = df.format(new Date());

        return EndpointRequest.builder()
                .address(UUID.randomUUID().toString())
                .attributes(customAttributes)
                .channelType("APNS")
                .demographic(demographic)
                .effectiveDate(nowAsISO)
                .location(location)
                .metrics(metrics)
                .optOut("NONE")
                .requestId(UUID.randomUUID().toString())
                .user(user)
                .build();
```

```
        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- API에 대한 세부 정보는 AWS SDK for Java 2.x API 참조의 UpdateEndpoint를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDKs를 사용한 Amazon Pinpoint 시나리오

다음 코드 예제에서는 Amazon Pinpoint AWS SDKs에서 일반적인 시나리오를 구현하는 방법을 보여줍니다. 이러한 시나리오에서는 Amazon Pinpoint 내에서 또는 다른 함수와 결합된 여러 함수를 호출하여 특정 작업을 수행하는 방법을 보여줍니다 AWS 서비스. 각 시나리오에는 전체 소스 코드에 대한 링크가 포함되어 있습니다. 여기에서 코드를 설정 및 실행하는 방법에 대한 지침을 찾을 수 있습니다.

시나리오는 컨텍스트에 맞는 서비스 작업을 이해하는 데 도움이 되도록 중급 수준의 경험을 대상으로 합니다.

예시

- End User Messaging Push 시작하기

End User Messaging Push 시작하기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 애플리케이션 만들기
- 푸시 알림 채널 활성화
- 푸시 알림 전송
- 리소스 정리

Bash

AWS CLI Bash 스크립트 사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. 샘플 개발자 튜토리얼 리포지토리에서 전체 예제를 찾아보고 설정 및 실행 방법을 배워보세요.

```
#!/bin/bash

# AWS End User Messaging Push Getting Started Script
# This script creates an AWS End User Messaging Push application and demonstrates
# how to enable push notification channels and send a test message.
#
# Prerequisites:
# - AWS CLI installed and configured
# - Appropriate IAM permissions for Pinpoint operations
#
# Usage: ./aws-end-user-messaging-gs.sh [--auto-cleanup]

set -uo pipefail

# Security: Set secure umask for created files
umask 0077

# Set up logging
LOG_DIR="./aws-eump-logs"
mkdir -p "$LOG_DIR"
LOG_FILE="$LOG_DIR/aws-end-user-messaging-push-script-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS End User Messaging Push setup script..."
echo "Logging to $LOG_FILE"
echo "Timestamp: $(date)"

# Security: Track created resources for cleanup
declare -a TEMP_FILES=()
declare -a AWS_RESOURCES=()

# Cleanup function with improved security
```

```
cleanup() {
    local exit_code=$?
    echo "Cleaning up temporary resources..."

    # Remove temporary files securely
    for temp_file in "${TEMP_FILES[@]+"${TEMP_FILES[@]}"}"; do
        if [ -f "$temp_file" ]; then
            shred -vfz -n 3 "$temp_file" 2>/dev/null || rm -f "$temp_file"
        fi
    done

    # Optionally delete AWS resources
    if [ "${DELETE_AWS_RESOURCES:-false}" = "true" ]; then
        for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
            echo "Deleting AWS resource: $resource"
            aws pinpoint delete-app --application-id "$resource" 2>/dev/null || \
                echo "Warning: Failed to delete application $resource"
        done
    fi

    exit "$exit_code"
}

trap cleanup EXIT INT TERM

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2
    local ignore_error=${3:-false}

    if echo "$output" | grep -qi "error\|exception\|fail"; then
        echo "ERROR: Command failed: $cmd" >&2
        echo "Error details: $output" >&2

        if [ "$ignore_error" = "true" ]; then
            echo "Ignoring error and continuing..." >&2
            return 1
        else
            return 2
        fi
    fi

    return 0
```

```
}

# Function to validate AWS CLI is configured
validate_aws_cli() {
    echo "Validating AWS CLI configuration..."

    # Check if AWS CLI is installed
    if ! command -v aws &> /dev/null; then
        echo "ERROR: AWS CLI is not installed. Please install it first." >&2
        echo "Visit: https://docs.aws.amazon.com/cli/latest/userguide/getting-
started-install.html" >&2
        return 1
    fi

    # Check AWS CLI version
    AWS_VERSION=$(aws --version 2>&1 | head -n1)
    echo "AWS CLI version: $AWS_VERSION"

    # Verify credentials are set (check for credential env vars or config file)
    if ! aws sts get-caller-identity &> /dev/null; then
        echo "ERROR: AWS CLI credentials are not configured or invalid." >&2
        echo "Please configure credentials via environment variables, credential
 file, or 'aws configure'" >&2
        return 1
    fi

    # Get current AWS identity and region
    CALLER_IDENTITY=$(aws sts get-caller-identity)
    CURRENT_REGION=$(aws configure get region 2>/dev/null || echo "us-east-1")
    echo "AWS CLI configured for:"
    echo "$CALLER_IDENTITY"
    echo "Current region: $CURRENT_REGION"
    echo ""

    return 0
}

# Function to check if jq is available for JSON parsing
check_json_tools() {
    if command -v jq &> /dev/null; then
        USE_JQ=true
        echo "jq is available for JSON parsing"
    else
        USE_JQ=false
```

```
        echo "jq is not available, using grep for JSON parsing"
        echo "Consider installing jq for better JSON handling: https://
stedolan.github.io/jq/"
    fi
}

# Function to extract JSON values safely
extract_json_value() {
    local json=$1
    local key=$2

    if [ "$USE_JQ" = "true" ]; then
        echo "$json" | jq -r ".ApplicationResponse.$key // empty" 2>/dev/null ||
 echo ""
    else
        # Fallback to grep method with better validation
        echo "$json" | grep -o "\"$key\": \"[^\"]*" | cut -d'"' -f4 | head -n1 ||
 echo ""
    fi
}

# Function to validate required IAM permissions
validate_permissions() {
    echo "Validating IAM permissions..."

    # Test basic Pinpoint permissions
    if ! aws pinpoint get-apps > /dev/null 2>&1; then
        echo "WARNING: Unable to list Pinpoint applications." >&2
        echo "Please ensure you have appropriate IAM permissions for Pinpoint
 operations." >&2
        echo "Required permissions:" >&2
        echo "  - mobiletargeting:GetApps" >&2
        echo "  - mobiletargeting:CreateApp" >&2
        echo "  - mobiletargeting:DeleteApp" >&2
        echo "  - mobiletargeting:UpdateGcmChannel" >&2
        echo "  - mobiletargeting:UpdateApnsChannel" >&2
        echo "  - mobiletargeting:SendMessages" >&2
    else
        echo "Basic Pinpoint permissions validated."
    fi
}

# Function to validate input parameters
validate_input() {
```

```
    local app_name=$1

    # Validate app name length and characters
    if [ ${#app_name} -gt 64 ]; then
        echo "ERROR: Application name exceeds maximum length of 64 characters"
 >&2
        return 1
    fi

    if ! [[ "$app_name" =~ ^[a-zA-Z0-9_-]+$ ]]; then
        echo "ERROR: Application name contains invalid characters" >&2
        return 1
    fi

    return 0
}

# Function to create secure temporary files
create_temp_file() {
    local temp_file
    temp_file=$(mktemp) || {
        echo "ERROR: Failed to create temporary file" >&2
        return 1
    }
    chmod 600 "$temp_file"
    TEMP_FILES+=("$temp_file")
    echo "$temp_file"
}

# Validate prerequisites
if ! validate_aws_cli; then
    exit 1
fi

check_json_tools
validate_permissions

# Generate a random suffix for resource names to avoid conflicts
RANDOM_SUFFIX=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 8 | head -n1)
APP_NAME="PushNotificationApp-${RANDOM_SUFFIX}"

# Validate input
if ! validate_input "$APP_NAME"; then
    exit 1
```

```
fi

echo "Creating application with name: $APP_NAME"

# Step 1: Create an application
echo "Executing: aws pinpoint create-app --create-application-request Name=
${APP_NAME}"
CREATE_APP_OUTPUT=$(aws pinpoint create-app --create-application-request "Name=
${APP_NAME},tags={project=doc-smith,tutorial=aws-end-user-messaging-gs}" 2>&1)

if ! check_error "$CREATE_APP_OUTPUT" "create-app"; then
    exit 1
fi

echo "Application created successfully:"
echo "$CREATE_APP_OUTPUT"

# Extract the application ID from the output
APP_ID=$(extract_json_value "$CREATE_APP_OUTPUT" "Id")

if [ -z "$APP_ID" ] || [ "$APP_ID" = "null" ]; then
    echo "ERROR: Failed to extract application ID from output" >&2
    echo "Output was: $CREATE_APP_OUTPUT" >&2
    exit 1
fi

echo "Application ID: $APP_ID"
AWS_RESOURCES+=("$APP_ID")

# Step 2: Enable FCM (GCM) channel with a sample API key
echo ""
echo "========================================="
echo "ENABLING FCM (GCM) CHANNEL"
echo "========================================="
echo "Note: This is using a placeholder API key for demonstration purposes only."
echo "In a production environment, you should use your actual FCM API key from
 Firebase Console."
echo ""
echo "IMPORTANT: The following command will likely fail because we're using a
 placeholder API key."
echo "This is expected behavior for this demonstration script."

echo "Executing: aws pinpoint update-gcm-channel --application-id $APP_ID --gcm-
channel-request ..."
```

```
UPDATE_GCM_OUTPUT=$(aws pinpoint update-gcm-channel \
    --application-id "$APP_ID" \
    --gcm-channel-request '{"Enabled": true, "ApiKey": "sample-fcm-api-key-for-
demo-only"}' 2>&1)

# We'll ignore this specific error since we're using a placeholder API key
if check_error "$UPDATE_GCM_OUTPUT" "update-gcm-channel" "true"; then
    echo "FCM channel enabled successfully:"
    echo "$UPDATE_GCM_OUTPUT"
else
    echo "As expected, FCM channel update failed with the placeholder API key."
    echo "Error details: $UPDATE_GCM_OUTPUT"
    echo ""
    echo "To enable FCM in production:"
    echo "1. Go to Firebase Console (https://console.firebase.google.com/)"
    echo "2. Create or select your project"
    echo "3. Go to Project Settings > Cloud Messaging"
    echo "4. Copy the Server Key"
    echo "5. Replace 'sample-fcm-api-key-for-demo-only' with your actual Server
 Key"
fi

# Step 3: Try to enable APNS channel (this will also fail without real
 certificates)
echo ""
echo "========================================="
echo "ENABLING APNS CHANNEL (OPTIONAL)"
echo "========================================="
echo "Attempting to enable APNS channel with placeholder certificate..."
echo "This will also fail without real APNS certificates, which is expected."

# Create a placeholder APNS configuration
echo "Executing: aws pinpoint update-apns-channel --application-id $APP_ID --
apns-channel-request ..."
UPDATE_APNS_OUTPUT=$(aws pinpoint update-apns-channel \
    --application-id "$APP_ID" \
    --apns-channel-request '{"Enabled": true, "Certificate": "placeholder-
certificate", "PrivateKey": "placeholder-private-key"}' 2>&1)

if check_error "$UPDATE_APNS_OUTPUT" "update-apns-channel" "true"; then
    echo "APNS channel enabled successfully:"
    echo "$UPDATE_APNS_OUTPUT"
else
    echo "As expected, APNS channel update failed with placeholder certificates."
```

```
    echo "Error details: $UPDATE_APNS_OUTPUT"
    echo ""
    echo "To enable APNS in production:"
    echo "1. Generate APNS certificates from Apple Developer Console"
    echo "2. Convert certificates to PEM format"
    echo "3. Use the actual certificate and private key in the update-apns-
channel command"
fi

# Step 4: Create message files for different platforms
echo ""
echo "============================================="
echo "CREATING MESSAGE FILES"
echo "============================================="

# Create FCM message file securely
GCM_MESSAGE_FILE=$(create_temp_file)
echo "Creating FCM message file..."
cat > "$GCM_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-FCM": {
      "ChannelType": "GCM"
    }
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "OPEN_APP",
      "Body": "Hello from AWS End User Messaging Push! This is an FCM
 notification.",
      "Priority": "normal",
      "SilentPush": false,
      "Title": "My First FCM Push Notification",
      "TimeToLive": 30,
      "Data": {
        "key1": "value1",
        "key2": "value2"
      }
    }
  }
}
EOF

# Create APNS message file securely
```

```
APNS_MESSAGE_FILE=$(create_temp_file)
echo "Creating APNS message file..."
cat > "$APNS_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-APNS": {
      "ChannelType": "APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
      "Action": "OPEN_APP",
      "Body": "Hello from AWS End User Messaging Push! This is an APNS
 notification.",
      "Priority": "normal",
      "SilentPush": false,
      "Title": "My First APNS Push Notification",
      "TimeToLive": 30,
      "Badge": 1,
      "Sound": "default"
    }
  }
}
EOF

echo "Message files created:"
echo "- FCM message file (for FCM/Android)"
echo "- APNS message file (for APNS/iOS)"
echo ""
echo "Note: These messages use placeholder device tokens and will not actually be
 delivered."
echo "To send real messages, you would need to replace the sample device tokens
 with actual ones."

# Step 5: Demonstrate how to send messages (this will fail with placeholder
 tokens)
echo ""
echo "============================================="
echo "DEMONSTRATING MESSAGE SENDING"
echo "============================================="
echo "Attempting to send FCM message (will fail with placeholder token)..."

echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<gcm-message>"
```

```
SEND_FCM_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$GCM_MESSAGE_FILE" 2>&1)

if check_error "$SEND_FCM_OUTPUT" "send-messages (FCM)" "true"; then
    echo "FCM message sent successfully:"
    echo "$SEND_FCM_OUTPUT"
else
    echo "As expected, FCM message sending failed with placeholder token."
    echo "Error details: $SEND_FCM_OUTPUT"
fi

echo ""
echo "Attempting to send APNS message (will fail with placeholder token)..."

echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<apns-message>"
SEND_APNS_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$APNS_MESSAGE_FILE" 2>&1)

if check_error "$SEND_APNS_OUTPUT" "send-messages (APNS)" "true"; then
    echo "APNS message sent successfully:"
    echo "$SEND_APNS_OUTPUT"
else
    echo "As expected, APNS message sending failed with placeholder token."
    echo "Error details: $SEND_APNS_OUTPUT"
fi

# Step 6: Show application details
echo ""
echo "========================================="
echo "APPLICATION DETAILS"
echo "========================================="
echo "Retrieving application details..."

echo "Executing: aws pinpoint get-app --application-id $APP_ID"
GET_APP_OUTPUT=$(aws pinpoint get-app --application-id "$APP_ID" 2>&1)
if check_error "$GET_APP_OUTPUT" "get-app"; then
    echo "Application details:"
    echo "$GET_APP_OUTPUT"
fi

# Display summary of created resources
```

```
echo ""
echo "============================================"
echo "RESOURCES CREATED"
echo "============================================"
echo "AWS Resources:"
for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
    echo "- Application: $resource"
done

echo ""
echo "Files created:"
echo "- $LOG_FILE (script log)"

# Auto-cleanup information
echo ""
echo "============================================"
echo "CLEANUP INFORMATION"
echo "============================================"
echo "This script created AWS resources that may incur charges."
echo "AWS resources will be automatically cleaned up on script exit."
echo ""
echo "To manually delete resources later, use:"
echo "  aws pinpoint delete-app --application-id $APP_ID"

# Set flag to delete AWS resources on cleanup
DELETE_AWS_RESOURCES=true

echo ""
echo "============================================"
echo "SCRIPT COMPLETED SUCCESSFULLY"
echo "============================================"
echo "This script demonstrated:"
echo "1. Creating an AWS End User Messaging Push application"
echo "2. Attempting to enable FCM and APNS channels (with placeholder
 credentials)"
echo "3. Creating message templates for different platforms"
echo "4. Demonstrating message sending commands (with placeholder tokens)"
echo "5. Retrieving application details"
echo "6. Proper cleanup of resources"
echo ""
echo "Security best practices implemented:"
echo "- Secure temporary file handling with restricted permissions"
echo "- Input validation for application names"
echo "- Error handling with proper exit codes"
```

```
echo "- Credential validation before AWS operations"
echo "- Automatic cleanup on script exit"
echo "- Secure file destruction for sensitive data"
echo ""
echo "For production use:"
echo "- Replace placeholder API keys with real FCM server keys"
echo "- Replace placeholder certificates with real APNS certificates"
echo "- Replace placeholder device tokens with real device tokens"
echo "- Use AWS IAM roles instead of long-term credentials"
echo "- Implement comprehensive error handling"
echo "- Store sensitive credentials in AWS Secrets Manager or Parameter Store"
echo "- Enable CloudTrail for audit logging"
echo "- Use VPC endpoints for private AWS API access"
echo ""
echo "Log file: $LOG_FILE"
echo "Script completed at: $(date)"
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateApp
  - DeleteApp
  - GetApp
  - GetApps
  - GetCallerIdentity
  - SendMessages
  - UpdateApnsChannel
  - UpdateGcmChannel

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDKs를 사용한 Amazon Pinpoint SMS 및 음성 API의 코드 예제

다음 코드 예제에서는 Amazon Pinpoint SMS 및 음성 API를 AWS 소프트웨어 개발 키트(SDK)와 함께 사용하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

Amazon Pinpoint SMS 및 음성 API

- AWS SDKs를 사용한 Amazon Pinpoint SMS 및 음성 API의 기본 예제
  - AWS SDKs를 사용하는 Amazon Pinpoint SMS 및 음성 API에 대한 작업
    - AWS SDK와 CreateConfigurationSet 함께 사용
    - AWS SDK와 CreateConfigurationSetEventDestination 함께 사용
    - AWS SDK와 DeleteConfigurationSet 함께 사용
    - AWS SDK와 DeleteConfigurationSetEventDestination 함께 사용
    - AWS SDK와 GetConfigurationSetEventDestinations 함께 사용
    - AWS SDK와 ListConfigurationSets 함께 사용
    - AWS SDK와 SendVoiceMessage 함께 사용
    - AWS SDK와 UpdateConfigurationSetEventDestination 함께 사용

AWS SDKs를 사용한 Amazon Pinpoint SMS 및 음성 API의 기본 예제

다음 코드 예제에서는 Amazon Pinpoint SMS and Voice API를 AWS SDK와 함께 사용하는 방법을 보여줍니다.

예제

- AWS SDKs를 사용하는 Amazon Pinpoint SMS 및 음성 API에 대한 작업
  - AWS SDK와 CreateConfigurationSet 함께 사용
  - AWS SDK와 CreateConfigurationSetEventDestination 함께 사용
  - AWS SDK와 DeleteConfigurationSet 함께 사용
  - AWS SDK와 DeleteConfigurationSetEventDestination 함께 사용
  - AWS SDK와 GetConfigurationSetEventDestinations 함께 사용
  - AWS SDK와 ListConfigurationSets 함께 사용
  - AWS SDK와 SendVoiceMessage 함께 사용
  - AWS SDK와 UpdateConfigurationSetEventDestination 함께 사용

AWS SDKs를 사용하는 Amazon Pinpoint SMS 및 음성 API에 대한 작업

다음 코드 예제에서는 AWS SDKs를 사용하여 개별 Amazon Pinpoint SMS 및 음성 API 작업을 수행하는 방법을 보여줍니다. 각 예시에는 GitHub에 대한 링크가 포함되어 있습니다. 여기에서 코드 설정 및 실행에 대한 지침을 찾을 수 있습니다.

다음 예제에는 가장 일반적으로 사용되는 작업만 포함되어 있습니다. 전체 목록은 Amazon Pinpoint SMS 및 음성 API 참조를 살펴보세요.

예제

- AWS SDK와 CreateConfigurationSet 함께 사용
- AWS SDK와 CreateConfigurationSetEventDestination 함께 사용
- AWS SDK와 DeleteConfigurationSet 함께 사용
- AWS SDK와 DeleteConfigurationSetEventDestination 함께 사용
- AWS SDK와 GetConfigurationSetEventDestinations 함께 사용
- AWS SDK와 ListConfigurationSets 함께 사용
- AWS SDK와 SendVoiceMessage 함께 사용
- AWS SDK와 UpdateConfigurationSetEventDestination 함께 사용

AWS SDK와 **CreateConfigurationSet** 함께 사용

다음 코드 예시는 `CreateConfigurationSet`의 사용 방법을 보여 줍니다.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " Create a new configuration set
        lo_pps->createconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
```

```
        ).

        MESSAGE 'Configuration set created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateConfigurationSet를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **CreateConfigurationSetEventDestination** 함께 사용

다음 코드 예시는 CreateConfigurationSetEventDestination의 사용 방법을 보여 줍니다.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " Create event destination for the configuration set
        lo_pps->createconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateConfigurationSetEventDestination을 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **DeleteConfigurationSet** 함께 사용

다음 코드 예시는 DeleteConfigurationSet의 사용 방법을 보여 줍니다.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " Delete the configuration set
        lo_pps->deleteconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteConfigurationSet를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **DeleteConfigurationSetEventDestination** 함께 사용

다음 코드 예시는 DeleteConfigurationSetEventDestination의 사용 방법을 보여 줍니다.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " Delete the event destination
        lo_pps->deleteconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
        ).

        MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteConfigurationSetEventDestination을 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **GetConfigurationSetEventDestinations** 함께 사용

다음 코드 예시는 GetConfigurationSetEventDestinations의 사용 방법을 보여 줍니다.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " Get event destinations for the configuration set
        oo_result = lo_pps->getconfseteventdestinations(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        " Process the event destinations
        LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).
          DATA(lv_dest_name) = lo_event_dest->get_name( ).
          DATA(lv_enabled) = lo_event_dest->get_enabled( ).

          MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
 TYPE 'I'.

          " Check for CloudWatch Logs destination
          DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
          IF lo_cloudwatch_dest IS NOT INITIAL.
```

```
            DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
            MESSAGE |  CloudWatch Logs destination: { lv_log_group_arn }| TYPE
 'I'.
          ENDIF.

          " Check for Kinesis Firehose destination
          DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
          IF lo_firehose_dest IS NOT INITIAL.
            DATA(lv_delivery_stream) = lo_firehose_dest-
>get_deliverystreamarn( ).
            MESSAGE |  Kinesis Firehose destination: { lv_delivery_stream }| TYPE
 'I'.
          ENDIF.

          " Check for SNS destination
          DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
          IF lo_sns_dest IS NOT INITIAL.
            DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
            MESSAGE |  SNS destination: { lv_topic_arn }| TYPE 'I'.
          ENDIF.
        ENDLOOP.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetConfigurationSetEventDestinations를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **ListConfigurationSets** 함께 사용

다음 코드 예시는 ListConfigurationSets의 사용 방법을 보여 줍니다.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " List all configuration sets
        oo_result = lo_pps->listconfigurationsets(
          iv_nexttoken = iv_next_token    " Optional: Token for pagination
          iv_pagesize = iv_page_size      " Optional: Number of results per page,
e.g., '10'
        ).

        " Process the configuration sets
        LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
          DATA(lv_config_set_name) = lo_config_set->get_value( ).
          MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
        ENDLOOP.

        " Check if there are more results
        DATA(lv_next_token) = oo_result->get_nexttoken( ).
        IF lv_next_token IS NOT INITIAL.
          MESSAGE |More results available. Next token: { lv_next_token }| TYPE
'I'.
        ENDIF.

      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
```

```
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListConfigurationSets를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **SendVoiceMessage** 함께 사용

다음 코드 예시는 SendVoiceMessage의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a
 list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";

    // The language to use when sending the message. For a list of supported
    // languages, see
    // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
    static final String languageCode = "en-US";

    // The content of the message. This example uses SSML to customize and
 control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent from "
            + "<emphasis>Amazon Pinpoint</emphasis> "
            + "using the <break strength='weak'/>AWS "
            + "SDK for Java. "
            + "<amazon:effect phonation='soft'>Thank "
            + "you for listening.</amazon:effect></speak>";

    public static void main(String[] args) {

        final String usage = """
                Usage:   <originationNumber> <destinationNumber>\s

                Where:
```

```
                originationNumber - The phone number or short code that
you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
               """;

       if (args.length != 2) {
           System.out.println(usage);
           System.exit(1);
       }
       String originationNumber = args[0];
       String destinationNumber = args[1];
       System.out.println("Sending a voice message");

       // Set the content type to application/json.
       List<String> listVal = new ArrayList<>();
       listVal.add("application/json");
       Map<String, List<String>> values = new HashMap<>();
       values.put("Content-Type", listVal);

       ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()
               .headers(values)
               .build();

       PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
               .overrideConfiguration(config2)
               .region(Region.US_EAST_1)
               .build();

       sendVoiceMsg(client, originationNumber, destinationNumber);
       client.close();
   }

   public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                   String destinationNumber) {
       try {
           SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                   .languageCode(languageCode)
                   .text(ssmlMessage)
                   .voiceId(voiceName)
```

```
                    .build();

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
 SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
                    .originationPhoneNumber(originationNumber)
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

        } catch (PinpointSmsVoiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 SendVoiceMessage를 참조하십시오.

JavaScript

SDK for JavaScript (v2)

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
```

```
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best
 results, you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/
SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint
 account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
```

```
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

- API 세부 정보는 AWS SDK for JavaScript API 참조의 SendVoiceMessage를 참조하십시오.

Python

SDK for Python (Boto3)

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your
 Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your
 Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
```

```
    :param voice_id: The Amazon Polly voice that you want to use to send the
 message.
    :param ssml_message: The content of the message. This example uses SSML to
 control
                         certain aspects of the message, such as the volume and
 the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
```

```
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 SendVoiceMessage를 참조하세요.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " Create SSML message type object with voice parameters
        DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmlmessagetype(
          iv_languagecode = iv_language_code    " e.g., 'en-US'
          iv_voiceid = iv_voice_id              " e.g., 'Matthew'
          iv_text = iv_ssml_message             " SSML formatted message text
        ).
```

```
      " Create voice message content with the SSML message
      DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
        io_ssmlmessage = lo_ssml_message
      ).

      " Send the voice message
      DATA(lo_result) = lo_pps->sendvoicemessage(
        iv_originationphonenumber = iv_origination_number  " e.g.,
'+12065550110'
        iv_callerid = iv_caller_id                          " e.g.,
'+12065550199'
        iv_destinationphonenumber = iv_destination_number  " e.g.,
'+12065550142'
        io_content = lo_content
      ).

      " Retrieve the message ID from the response
      ov_message_id = lo_result->get_messageid( ).

      MESSAGE 'Voice message sent successfully.' TYPE 'I'.

    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SendVoiceMessage를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **UpdateConfigurationSetEventDestination** 함께 사용

다음 코드 예시는 UpdateConfigurationSetEventDestination의 사용 방법을 보여 줍니다.

SAP ABAP

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        " Update the event destination
        lo_pps->updateconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination updated successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 UpdateConfigurationSetEventDestination을 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요AWS SDK에서 Amazon Pinpoint 사용. 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

Amazon Pinpoint의 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 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. Amazon Pinpoint에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스규정 준수 프로그램 .
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon Pinpoint를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 Amazon Pinpoint를 구성하는 방법을 보여줍니다. 또한 Amazon Pinpoint 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

참조 아키텍처에 대한 자세한 내용은 Amazon Pinpoint의 복원력 있는 아키텍처 설명서를 참조하세요.

주제

- Amazon Pinpoint의 데이터 보호
- Amazon Pinpoint의 ID 및 액세스 관리
- Amazon Pinpoint의 로깅 및 모니터링
- Amazon Pinpoint의 규정 준수 검증
- Amazon Pinpoint의 복원성
- Amazon Pinpoint의 인프라 보안
- Amazon Pinpoint의 구성 및 취약성 분석
- Amazon Pinpoint의 보안 모범 사례

Amazon Pinpoint의 데이터 보호

AWS 공동 책임 모델 Amazon Pinpoint의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 일반 데이터 보호 규정(GDPR) 센터를 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon Pinpoint 또는 기타 AWS 서비스 에서 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시켜서는 안 됩니다.

서비스를 구성하고 사용하는 방법에 따라, Amazon Pinpoint는 사용자를 위해, 또는 사용자의 고객에 대해 다음과 같은 유형의 개인 데이터를 저장할 수 있습니다.

구성 데이터

여기에는 Amazon Pinpoint가 지원되는 채널을 통해 메시지를 전송하는 방법과 시기를 정의하는 보안 인증 및 설정, 메시지를 전송하는 사용자 세그먼트와 같은 프로젝트 구성 데이터가 포함됩니다. 메시지를 전송하려면 이 데이터에는 이메일 메시지의 전용 IP 주소, SMS 문자 메시지용 단축 코드 및 발신자 ID, Apple 푸시 알림 서비스(APN) 및 Firebase Cloud Messaging(FCM)과 같은 푸시 알림 서비스와 통신하기 위한 보안 인증이 포함될 수 있습니다.

사용자 및 엔드포인트 데이터

여기에는 Amazon Pinpoint 프로젝트의 사용자 및 엔드포인트에 대한 데이터를 저장하고 관리하는 데 사용하는 표준 및 사용자 정의 속성이 포함됩니다. 속성은 특정 사용자(예: 사용자 이름) 또는 사용자의 특정 엔드포인트(예: 사용자의 이메일 주소, 휴대폰 번호 또는 모바일 장치 토큰)에 대한 정보를 저장할 수 있습니다. 이 데이터에는 Amazon Pinpoint 프로젝트 사용자와 고객 관계 관리 시스템과 같은 외부 시스템의 사용자와 상호 연관시키는 외부 사용자 ID도 포함될 수 있습니다. 이 데이터에 포함될 수 있는 내용에 대한 자세한 내용은 Amazon Pinpoint API 참조의 사용자 및 엔드포인트 스키마를 참조하세요.

분석 데이터

여기에는 사용자 참여 및 구매 활동과 같은 영역에서 Amazon Pinpoint 프로젝트의 성과에 대한 인사이트를 제공하며 핵심 성능 지표(KPI)라고도 하는 지표의 데이터가 포함됩니다. 여기에는 프로젝트의 사용자 인구 통계에 대한 통찰력을 제공하는 지표에 대한 데이터도 포함됩니다. 데이터는 사용자가 거주하는 도시와 같은 사용자 및 엔드포인트에 대한 표준 및 사용자 정의 속성에서 파생될 수 있습니다. 프로젝트에 대해 전송하는 이메일 메시지에 대한 열기 및 클릭 이벤트와 같은 이벤트에서 파생될 수도 있습니다.

가져온 데이터

여기에는 외부 소스에서 추가하거나 가져와서 Amazon Pinpoint에서 사용하는 모든 사용자, 분할 및 분석 데이터가 포함됩니다. 정적 세그먼트를 빌드하기 위해 콘솔을 통해 직접, 또는 Amazon S3 버킷에서 Amazon Pinpoint로 가져온 JSON 파일을 예로 들 수 있습니다. 다른 예로는 동적 세그먼트를 빌드하기 위해 프로그래밍 방식으로 추가하는 엔드포인트 데이터, 직접 메시지를 전송하는 엔드포인트 주소, Amazon Pinpoint에 보고할 앱을 구성하는 이벤트가 있습니다.

주제

- 데이터 암호화
- 인터네트워크 트래픽 개인 정보
- Amazon Pinpoint에 대한 인터페이스 VPC 엔드포인트 생성

데이터 암호화

Amazon Pinpoint 데이터는 전송 및 저장 시 암호화됩니다. Amazon Pinpoint에 데이터를 제출하면 데이터를 수신하고 저장할 때 데이터를 암호화합니다. Amazon Pinpoint에서 데이터를 검색하면 현재 보안 프로토콜을 사용하여 데이터가 전송됩니다.

저장 시 암호화

Amazon Pinpoint는 사용자를 위해 저장하는 모든 데이터를 암호화합니다. 여기에는 구성 데이터, 사용자 및 엔드포인트 데이터, 분석 데이터, Amazon Pinpoint로 추가 또는 가져오는 모든 데이터가 포함됩니다. 데이터를 암호화하기 위해 Amazon Pinpoint는 서비스가 사용자를 대신하여 소유하고 유지 관리하는 내부 AWS Key Management Service (AWS KMS) 키를 사용합니다. 이들 키는 정기적으로 교체됩니다. 에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 AWS KMS참조하세요.

전송 중 암호화

Amazon Pinpoint는 HTTPS 및 전송 계층 보안(TLS) 1.2 이상을 사용하여 클라이언트 및 애플리케이션과 통신합니다. Amazon Pinpoint는 다른 AWS 서비스와 통신하기 위해 HTTPS 및 TLS 1.2를 사용합니다. 또한 콘솔, AWS SDK 또는를 사용하여 Amazon Pinpoint 리소스를 생성하고 관리할 때 AWS Command Line Interface모든 통신은 HTTPS 및 TLS 1.2를 사용하여 보호됩니다.

키 관리

Amazon Pinpoint 데이터를 암호화하기 위해 Amazon Pinpoint는 서비스가 사용자를 대신하여 소유하고 유지 관리하는 내부 AWS KMS 키를 사용합니다. 이들 키는 정기적으로 교체됩니다. 자체 키 AWS KMS 또는 기타 키를 프로비저닝하고 사용하여 Amazon Pinpoint에 저장하는 데이터를 암호화할 수 없습니다.

인터네트워크 트래픽 개인 정보

Internetwork 트래픽 개인 정보 보호란 Amazon Pinpoint와 온프레미스 클라이언트 및 애플리케이션 간의 연결 및 트래픽과 Amazon Pinpoint와 동일한 AWS 리전의 다른 AWS 리소스 간의 연결을 보호하는 것을 말합니다. 다음과 같은 기능 및 방법은 Amazon Pinpoint에 대한 인터네트워크 트래픽 개인 정보 보호를 보장하는 데 도움이 될 수 있습니다.

Amazon Pinpoint와 온프레미스 클라이언트 및 애플리케이션 간의 트래픽

Amazon Pinpoint와 온프레미스 네트워크의 클라이언트 및 애플리케이션 간에 프라이빗 연결을 설정하기 위해 Direct Connect를 사용할 수 있습니다. 이렇게 하면 표준 광섬유 이더넷 케이블을 사용하여

네트워크를 AWS Direct Connect 위치에 연결할 수 있습니다. 케이블의 한쪽 끝이 라우터에 연결되어 있습니다. 다른 쪽 끝은 Direct Connect 라우터에 연결됩니다. 자세한 내용은 Direct Connect사용 설명서의 Direct Connect 이란?을 참조하세요.

게시된 API를 통해 Amazon Pinpoint에 안전하게 액세스할 수 있도록 하려면 API 직접 호출에 대한 Amazon Pinpoint 요구 사항을 준수하는 것이 좋습니다. Amazon Pinpoint 클라이언트는 전송 계층 보안(TLS) 1.2 이상을 사용해야 합니다. 클라이언트는 DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Diffie-Hellman Ephemeral)와 같은 PFS(Perfect Forward Secrecy)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 액세스 키 ID와 AWS 계정의 AWS Identity and Access Management (IAM) 보안 주체와 연결된 보안 액세스 키를 사용하여 요청에 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용해 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

Amazon Pinpoint와 기타 AWS 리소스 간의 트래픽

Amazon Pinpoint와 동일한 AWS 리전의 다른 AWS 리소스 간의 통신을 보호하기 위해 Amazon Pinpoint는 기본적으로 HTTPS 및 TLS 1.2를 사용합니다.

Amazon Pinpoint에 대한 인터페이스 VPC 엔드포인트 생성

인터페이스 VPC 엔드포인트를 생성하여 Virtual Private Cloud(VPC)와 Amazon Pinpoint의 엔드포인트 간에 프라이빗 연결을 설정할 수 있습니다.

인터페이스 엔드포인트는 인터넷 게이트웨이AWS PrivateLink, NAT 디바이스, VPN 연결 또는 없이 Amazon Pinpoint APIs에 비공개로 액세스할 수 있는 기술로 구동됩니다 Direct Connect. VPC의 인스턴스는 AWS PrivateLink와 통합되는 Amazon Pinpoint API와 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다.

자세한 정보는 AWS PrivateLink 안내서를 참조하세요.

인터페이스 VPC 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS Command Line Interface ()를 사용하여 인터페이스 엔드포인트를 생성할 수 있습니다AWS CLI. 자세한 내용은 AWS PrivateLink 가이드의 인터페이스 엔드포인트 생성을 참조하세요.

Amazon Pinpoint는 다음과 같은 서비스 이름을 지원합니다.

- `com.amazonaws.region.pinpoint`

- com.amazonaws.*region*.pinpoint-sms-voice-v2

인터페이스 엔드포인트에 대해 프라이빗 DNS를 켜는 경우와 AWS 리전같은의 기본 DNS 이름을 사용하여 Amazon Pinpoint에 API 요청을 할 수 있습니다com.amazonaws.*us-east-1*.pinpoint. 자세한 내용은 AWS PrivateLink 설명서에서 DNS 호스트 이름을 참조하세요.

현재 Amazon Pinpoint를 사용할 수 있는 모든 리전 목록은 Amazon Web Services 일반 참조의 AWS 서비스 엔드포인트를 참조하세요.

VPC 엔드포인트 정책 생성

액세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스

자세한 내용은 AWS PrivateLink 안내서의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

예제: VPC 엔드포인트 정책

아래의 VPC 엔드포인트 정책은 모든 리소스의 모든 보안 주체에 대한 액세스 권한을 나열된 Amazon Pinpoint 작업에 부여합니다.

```
{
"Statement": [
    {
      "Principal": "*",
      "Action": [
        "mobiletargeting:CreateCampaign",
        "mobiletargeting:CreateApp",
        "mobiletargeting:DeleteApp",
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon Pinpoint의 ID 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 어떤 사용자가 Amazon Pinpoint 리소스를 사용할 수 있도록 인증(로그인)되고 권한이 부여(권한 있음)될 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- Amazon Pinpoint에서 IAM을 사용하는 방법
- IAM 정책에 대한 Amazon Pinpoint 작업
- Amazon Pinpoint ID 기반 정책 예제
- 일반적인 Amazon Pinpoint 태스크를 위한 IAM 역할
- Amazon Pinpoint의 ID 및 액세스 관리에 대한 문제 해결

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청(참조Amazon Pinpoint의 ID 및 액세스 관리에 대한 문제 해결)
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출(Amazon Pinpoint에서 IAM을 사용하는 방법 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성(Amazon Pinpoint ID 기반 정책 예제 참조)

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증해야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다.

로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자가 필요한 작업 목록은 IAM 사용자 설명서의 루트 사용자 자격 증명이 필요한 작업을 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 페더레이션을 사용하여 IAM 사용 설명서의 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

Amazon Pinpoint는 자격 증명 기반 정책을 사용하여 Amazon Pinpoint 리소스에 대한 액세스를 제어할 수 있습니다.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

Amazon Pinpoint는 리소스 기반 정책을 사용하여 Amazon Pinpoint 리소스에 대한 액세스를 제어할 수 있습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 위탁자(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3 AWS WAF및 Amazon VPC는 ACLs. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

Amazon Pinpoint는 ACL을 사용하여 Amazon Pinpoint 리소스에 대한 액세스를 제어할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

Amazon Pinpoint는 이러한 유형의 정책을 사용하여 Amazon Pinpoint 리소스에 대한 액세스를 제어할 수 있습니다.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

Amazon Pinpoint에서 IAM을 사용하는 방법

Amazon Pinpoint를 사용하려면 AWS 계정의 사용자에게 분석 데이터 보기, 프로젝트 생성, 사용자 세그먼트 정의, 캠페인 배포 등을 허용하는 권한이 필요합니다. 모바일 또는 웹 앱을 Amazon Pinpoint와 통합하는 경우 앱 사용자는 Amazon Pinpoint에 대한 액세스 권한도 가져야 합니다. 이러한 액세스 권한이 있어야 앱이 엔드포인트를 등록하고 Amazon Pinpoint에 사용 데이터를 보고할 수 있습니다. Amazon Pinpoint 기능에 대한 액세스 권한을 부여하려면 IAM 자격 증명 또는 Amazon Pinpoint 리소스에 대한 Amazon Pinpoint 작업을 허용하는 AWS Identity and Access Management (IAM) 정책을 생성합니다.

IAM은 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 서비스입니다. IAM 정책은 사용자가 특정 리소스에 대해 수행할 수 있는 특정 작업을 허용하거나 거부하는 명령문을 포함합니다. Amazon Pinpoint는 세분화된 권한을 지정하기 위해 IAM 정책에서 사용할 수 있는 작업 집합을 Amazon Pinpoint 사용자 및 리소스에게 제공합니다. 즉, 중요 데이터를 노출시키거나 리소스를 손상시킬 수 있는 과도하게 허용적인 정책을 생성하지 않고 Amazon Pinpoint에 대한 적정 수준의 액세스 권한을 부여할 수 있습니다. 예를 들어, Amazon Pinpoint 관리자에게는 제한 없는 액세스 권한을 부여하고 특정 프로젝트에만 액세스해야 하는 직원에게는 읽기 전용 액세스 권한만 부여할 수 있습니다.

IAM을 사용하여 Amazon Pinpoint에 대한 액세스를 관리하기 전에 Amazon Pinpoint에서 사용할 수 있는 IAM 기능을 이해해야 합니다. Amazon Pinpoint 및 기타 AWS 서비스에서 IAM을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

주제

- Amazon Pinpoint ID 기반 정책
- Amazon Pinpoint 리소스 기반 권한 정책
- Amazon Pinpoint 태그 기반 권한 부여
- Amazon Pinpoint IAM 역할

Amazon Pinpoint ID 기반 정책

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. Amazon Pinpoint는 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 살펴보세요.

작업

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

즉, 정책 작업은 사용자가 Amazon Pinpoint 콘솔에서 수행할 수 있는 작업을 제어합니다. 또한 AWS SDKs, AWS Command Line Interface (AWS CLI) 또는 Amazon Pinpoint APIs.

Amazon Pinpoint의 정책 작업은 다음 접두사를 사용합니다.

- **`mobiletargeting`** – Amazon Pinpoint API에서 파생되는 작업의 경우, 이 API는 Amazon Pinpoint의 기본 API입니다.
- **`sms-voice`** – Amazon Pinpoint SMS 및 음성 API에서 파생되는 작업의 경우, 이 API는 Amazon Pinpoint에서 SMS 및 음성 채널을 사용하고 관리하기 위한 고급 옵션을 제공하는 보조 API 입니다.

예를 들어, Amazon Pinpoint API의 `GetSegments` 작업에 해당하는 작업인 프로젝트의 모든 세그먼트에 대한 정보를 볼 수 있는 권한을 다른 사람에게 부여하려면 `mobiletargeting:GetSegments` 작업을 정책에 포함시킵니다. 정책 문에는 `Action` 또는 `NotAction` 요소가 포함되어야 합니다. Amazon Pinpoint는 이를 활용하여 수행할 수 있는 태스크를 설명하는 고유한 작업 집합을 정의합니다.

단일 구문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "mobiletargeting:action1",
      "mobiletargeting:action2"
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수도 있습니다. 예를 들어, `Get`라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "mobiletargeting:Get*"
```

하지만 가장 좋은 방법은 최소 권한의 원칙을 따르는 정책을 만드는 것입니다. 즉, 특정 작업을 수행하는 데 필요한 권한만 포함하는 정책을 생성해야 합니다.

IAM 정책에서 사용할 수 있는 Amazon Pinpoint 작업의 목록은 IAM 정책에 대한 Amazon Pinpoint 작업 섹션을 참조하세요.

리소스

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

예를 들어, `mobiletargeting:GetSegments` 작업은 특정 Amazon Pinpoint 프로젝트와 연결된 모든 세그먼트에 대한 정보를 검색합니다. ARN이 있는 프로젝트를 다음과 같은 형식으로 식별합니다.

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

ARN 형식에 대한 자세한 내용은 AWS 일반 참조의 Amazon 리소스 이름(ARN)을 참조하세요.

IAM 정책에서 다음과 같은 Amazon Pinpoint 리소스 유형에 대해 ARN을 지정할 수 있습니다.

- Campaigns
- 여정
- 메시지 템플릿(일부 컨텍스트에서는 템플릿 이라고 지칭)
- 프로젝트(일부 컨텍스트에서는 앱 또는 애플리케이션 이라고 지칭)
- 추천 모델(일부 컨텍스트에서는 추천이라고 함)
- Segments

예를 들어, 프로젝트 ID `810c7aab86d42fb2b56c8c966example`이 있는 프로젝트에 대한 정책 설명을 생성하려면 다음 ARN을 사용합니다.

```
"Resource": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

특정 계정에 속하는 모든 프로젝트를 지정하려면 와일드카드(*)를 사용합니다.

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

리소스를 생성하기 위한 특정 작업과 같은 일부 Amazon Pinpoint 작업은 특정 리소스에서 수행할 수 없습니다. 이러한 경우, 와일드카드(*)를 사용해야 합니다.

```
"Resource": "*"
```

IAM 정책에서 다음과 같은 Amazon Pinpoint SMS 및 음성 리소스 유형에 대해 ARN을 지정할 수도 있습니다.

- 구성 세트

- 수신 거부 목록
- 전화번호
- 풀
- 발신자 ID

예를 들어, 전화번호 ID가 phone-123456789012345678901234567890 12인 전화번호에 대한 정책 명령문을 생성하려면 아래의 ARN을 사용합니다.

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-123456789012345678901234567890 12"
```

특정 계정에 속하는 모든 전화번호를 지정하려면 전화번호 ID 대신 와일드카드(*)를 사용합니다.

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

일부 Amazon Pinpoint SMS 및 음성 작업은 계정 수준 설정(예: 지출 한도)을 관리하는 리소스 등의 특정 리소스에서는 수행되지 않습니다. 이러한 경우, 와일드카드(*)를 사용해야 합니다.

```
"Resource": "*"
```

일부 Amazon Pinpoint API 작업에는 여러 리소스가 관여합니다. 예를 들어 TagResource 작업을 통해 여러 프로젝트에 태그를 추가할 수 있습니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

```
"Resource": [
      "resource1",
      "resource2"
```

Amazon Pinpoint 리소스 유형 및 해당 ARN의 목록을 보려면 IAM 사용 설명서의 Amazon Pinpoint에서 정의한 리소스를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 IAM 사용 설명서의 Amazon Pinpoint에서 정의한 작업을 참조하세요.

조건 키

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

Amazon Pinpoint에서는 자체 조건 키 세트를 정의하고 일부 전역 조건 키도 지원합니다. 모든 AWS 전역 조건 키 목록을 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요. Amazon Pinpoint 조건 키 목록을 보려면 IAM 사용 설명서의 Amazon Pinpoint에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 IAM 사용 설명서의 Amazon Pinpoint에서 정의한 작업을 참조하세요.

예제

Amazon Pinpoint ID 기반 정책 예제를 보려면 Amazon Pinpoint ID 기반 정책 예제 섹션을 참조하세요.

Amazon Pinpoint 리소스 기반 권한 정책

리소스 기반 권한 정책은 지정된 주체가 Amazon Pinpoint 리소스에 대해 수행할 수 있는 작업 및 관련 조건을 지정하는 JSON 정책 문서입니다. Amazon Pinpoint는 캠페인, 여정, 메시지 템플릿(템플릿), 추천 모델(추천), 프로젝트(앱), 세그먼트에 대한 리소스 기반 권한 정책을 지원합니다.

예제

Amazon Pinpoint 리소스 기반 정책의 예제를 보려면 the section called “ID 기반 정책 예시” 섹션을 참조하세요.

Amazon Pinpoint 태그 기반 권한 부여

태그를 특정 유형의 Amazon Pinpoint 리소스와 연결하거나 요청의 태그를 Amazon Pinpoint에 전달할 수 있습니다. 태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/${TagKey}`, `aws:RequestTag/${TagKey}`또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

IAM 정책 예제를 포함해 Amazon Pinpoint 리소스 태그 지정에 대한 자세한 내용은 Amazon Pinpoint 리소스 태그 관리 섹션을 참조하세요.

Amazon Pinpoint IAM 역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정 내 엔터티입니다.

Amazon Pinpoint에서 임시 보안 인증 사용

임시 자격 증명을 사용하여 연동으로 로그인하거나 IAM 역할을 수임하거나 교차 계정 역할을 수임할 수 있습니다. AssumeRole 또는 GetFederationToken 같은 AWS Security Token Service (AWS STS) API 작업을 호출하여 임시 보안 자격 증명을 가져옵니다.

Amazon Pinpoint는 임시 보안 인증 사용을 지원합니다.

서비스 연결 역할

서비스 연결 역할을 사용하면 AWS 서비스가 다른 서비스의 리소스에 액세스하여 사용자를 대신하여 작업을 완료할 수 있습니다. 서비스 연결 역할은 IAM 계정에 나타나고 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수 없습니다.

Amazon Pinpoint는 서비스 연결 역할을 사용하지 않습니다.

서비스 역할

이 기능을 사용하면 서비스가 사용자를 대신하여 서비스 역할을 수임할 수 있습니다. 이 역할을 사용하면 서비스가 다른 서비스의 리소스에 액세스해 사용자를 대신해 작업을 완료할 수 있습니다. 서비스 역할은 IAM 계정에 나타나고, 해당 계정이 소유합니다. 즉, IAM 관리자가 이 역할에 대한 권한을 변경할 수 있습니다. 그러나 권한을 변경하면 서비스의 기능이 손상될 수 있습니다.

Amazon Pinpoint는 서비스 역할 사용을 지원합니다.

IAM 정책에 대한 Amazon Pinpoint 작업

AWS 계정의 Amazon Pinpoint 리소스에 대한 액세스를 관리하려면 AWS Identity and Access Management (IAM) 정책에 Amazon Pinpoint 작업을 추가할 수 있습니다. 정책의 작업을 사용하여 Amazon Pinpoint 콘솔에서 사용자가 수행할 수 있는 작업을 제어할 수 있습니다. 또한 AWS SDKs, AWS Command Line Interface (AWS CLI) 또는 Amazon Pinpoint APIs.

정책에서는 적절한 Amazon Pinpoint 네임스페이스와 콜론, 작업 이름(예: `GetSegments`)으로 각 작업을 지정합니다. 대부분의 작업은 특정 URI 및 HTTP 메서드를 사용하는 Amazon Pinpoint API에 대한 요청에 해당합니다. 예를 들어 사용자 정책에서 `mobiletargeting:GetSegments` 작업을 허용하면 HTTP GET 요청을 `/apps/projectId/segments` URI에 제출하여 프로젝트의 모든 세그먼트에 대한 정보를 검색할 수 있습니다. 또한이 정책은 사용자가 콘솔에서 해당 정보를 보고 AWS SDK 또는를 사용하여 해당 정보를 검색할 수 있도록 허용합니다 AWS CLI.

각 작업은 정책 설명에서 Amazon 리소스 이름(ARN)으로 식별되는 특정 Amazon Pinpoint 리소스에 대해 수행됩니다. 예를 들어, `mobiletargeting:GetSegments` 작업은 ARN

`arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*로 식별되는 특정 프로젝트에 대해 수행됩니다.

이 주제에서는 AWS 계정의 IAM 정책에 추가할 수 있는 Amazon Pinpoint 작업을 알아봅니다. 정책에서 작업을 사용하여 Amazon Pinpoint 리소스에 대한 액세스를 관리하는 방법을 보여주는 예제는 Amazon Pinpoint ID 기반 정책 예제 섹션을 참조하세요.

주제

- Amazon Pinpoint API 작업
- Amazon Pinpoint SMS 및 음성 버전 1 API 작업

Amazon Pinpoint API 작업

이 섹션에서는 Amazon Pinpoint의 기본 API인 Amazon Pinpoint API에서 사용할 수 있는 기능에 대한 작업을 알아봅니다. 이 API를 자세히 알아보려면 Amazon Pinpoint API 참조를 살펴보세요.

범주:

- 분석 및 지표
- Campaigns
- 채널
- 엔드포인트
- 이벤트 스트림
- 이벤트
- 내보내기 작업
- 가져오기 작업
- 여정
- 메시지 템플릿
- 메시지
- 일회용 암호
- 전화 번호 확인
- Projects
- 추천 모델
- Segments
- Tags

- Users

분석 및 지표

다음 권한은 Amazon Pinpoint 콘솔에서 분석 데이터를 보는 작업과 관련이 있습니다. 또한 프로젝트, 캠페인 및 여정에 적용되는 핵심 성과 지표(KPI)라고도 하는 표준 지표에 대한 집계 데이터의 검색(쿼리)과 관련이 있습니다.

mobiletargeting:GetReports

Amazon Pinpoint 콘솔에서 분석 데이터를 봅니다. Amazon Pinpoint 콘솔을 사용하여 사용자 지정 속성이 포함된 세그먼트를 생성할 때도 이 권한이 필요합니다. Amazon Pinpoint 콘솔에서 세그먼트 크기의 추정치를 구하는 데에도 필요합니다.

- URI – 해당 사항 없음
- 메서드 – 해당 사항 없음
- 리소스 ARN – `arn:aws:mobiletargeting:region:accountId:*`

mobiletargeting:GetApplicationDateRangeKpi

표준 애플리케이션 지표에 대해 집계된 데이터를 검색(쿼리)합니다. 이는 프로젝트와 연관된 모든 캠페인 또는 트랜잭션 메시지에 적용되는 지표입니다.

- URI – /apps/*projectId*/kpis/daterange/*kpi-name*
- 매서드 – GET
- 리소스 ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/daterange/kpi-name`

mobiletargeting:GetCampaignDateRangeKpi

표준 캠페인 지표에 대해 집계된 데이터를 검색(쿼리)합니다. 이는 개별 캠페인에 적용되는 지표입니다.

- URI – /apps/*projectId*/campaigns/*campaignId*/kpis/daterange/*kpi-name*
- 매서드 – GET
- 리소스 ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`

mobiletargeting:GetJourneyDateRangeKpi

표준 여정 참여 지표에 대해 집계된 데이터를 검색(쿼리)합니다. 이는 개별 여정에 적용되는 참여 지표입니다(예: 여정의 모든 활동에 대해 참가자가 열어 본 메시지 수).

- URI – /apps/*projectId*/journeys/*journeyId*/kpis/daterange/*kpi-name*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/kpis/daterange/*kpi-name*

mobiletargeting:GetJourneyExecutionMetrics

개별 여정에 적용되는 표준 실행 지표(예: 여정의 모든 활동을 적극적으로 진행 중인 참가자 수)에 대한 집계 데이터를 검색(쿼리)합니다.

- URI – /apps/*projectId*/journeys/*journeyId*/execution-metrics
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/execution-metrics

mobiletargeting:GetJourneyExecutionActivityMetrics

여정의 개별 활동에 적용되는 표준 실행 지표(예: 활동을 시작하거나 완료한 참가자 수)에 대한 집계 데이터를 검색(쿼리)합니다.

- URI – /apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics

Campaigns

다음 권한은 Amazon Pinpoint 계정의 캠페인 관리에 관여합니다.

mobiletargeting:CreateCampaign

프로젝트 캠페인을 생성합니다.

- URI – /apps/*projectId*/campaigns
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns

mobiletargeting:DeleteCampaign

특정 캠페인을 삭제합니다.

- URI – /apps/*projectId*/campaigns/*campaignId*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaign

특정 캠페인에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/campaigns/*campaignId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignActivities

캠페인이 수행하는 활동에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/campaigns/*campaignId*/activities
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaigns

프로젝트의 모든 캠페인에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/campaigns
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetCampaignVersion

특정 캠페인 버전에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/campaigns/*campaignId*/versions/*versionId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignVersions

캠페인의 현재 및 이전 버전에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/campaigns/*campaignId*/versions
- 메서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:UpdateCampaign

특정 캠페인을 업데이트합니다.

- URI – /apps/*projectId*/campaigns/*campaignId*
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

채널

다음 권한은 Amazon Pinpoint 계정의 채널 관리에 관여합니다. Amazon Pinpoint에서 채널은 이메일, SMS 메시지 또는 푸시 알림 전송 등 사용자가 고객과 소통하기 위해 사용하는 방법을 말합니다.

mobiletargeting:DeleteAdmChannel

프로젝트의 Amazon Device Messaging(ADM) 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/adm
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:GetAdmChannel

프로젝트의 ADM 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/adm
- 메서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:UpdateAdmChannel

프로젝트의 ADM 채널을 활성화 또는 업데이트합니다.

- URI – /apps/*projectId*/channels/adm
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:DeleteApnsChannel

프로젝트의 Apple 푸시 알림 서비스(APNs) 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/apns
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:GetApnsChannel

프로젝트의 APN 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/apns
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:UpdateApnsChannel

프로젝트의 APN 채널을 활성화하거나 업데이트합니다.

- URI – /apps/*projectId*/channels/apns
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:DeleteApnsSandboxChannel

프로젝트의 APNs 샌드박스 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/apns_sandbox
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:GetApnsSandboxChannel

프로젝트의 APN 샌드박스 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/apns_sandbox
- 메서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:UpdateApnsSandboxChannel

프로젝트의 APN 샌드박스 채널을 활성화하거나 업데이트합니다.

- URI – /apps/*projectId*/channels/apns_sandbox
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:DeleteApnsVoipChannel

프로젝트의 APNs VoIP 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/apns_voip
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip

mobiletargeting:GetApnsVoipChannel

프로젝트의 APN VoIP 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/apns_voip
- 메서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip

mobiletargeting:UpdateApnsVoipChannel

프로젝트의 APN VoIP 채널을 활성화하거나 업데이트합니다.

- URI – /apps/*projectId*/channels/apns_voip
- 메서드 – PUT

- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip

mobiletargeting:DeleteApnsVoipSandboxChannel

프로젝트의 APNs VoIP 샌드박스 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip_sandbox

mobiletargeting:GetApnsVoipSandboxChannel

프로젝트의 APN VoIP 샌드박스 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip_sandbox

mobiletargeting:UpdateApnsVoipSandboxChannel

프로젝트의 APN VoIP 샌드박스 채널을 활성화하거나 업데이트합니다.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip_sandbox

mobiletargeting:DeleteBaiduChannel

프로젝트의 Baidu 클라우드 푸시 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/baidu
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:GetBaiduChannel

프로젝트의 Baidu 클라우드 푸시 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/baidu
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:UpdateBaiduChannel

프로젝트의 Baidu 클라우드 푸시 채널을 활성화하거나 업데이트합니다.

- URI – /apps/*projectId*/channels/baidu
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:DeleteEmailChannel

프로젝트의 이메일 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/email
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:GetEmailChannel

프로젝트의 이메일 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/email
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:UpdateEmailChannel

프로젝트의 이메일 채널을 활성화하거나 업데이트합니다.

- URI – /apps/*projectId*/channels/email
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:DeleteGcmChannel

프로젝트의 Firebase Cloud Messaging(FCM)을 비활성화합니다. 이 채널을 통해 Amazon Pinpoint는 Google Cloud Messaging(GCM) 서비스를 대체하는 FCM 서비스를 통해 Android 앱에 푸시 알림을 보낼 수 있습니다.

- URI – /apps/*projectId*/channels/gcm
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:GetGcmChannel

프로젝트의 FCM 채널에 대한 정보를 검색합니다. 이 채널을 통해 Amazon Pinpoint는 Google Cloud Messaging(GCM) 서비스를 대체하는 FCM 서비스를 통해 Android 앱에 푸시 알림을 보낼 수 있습니다.

- URI – /apps/*projectId*/channels/gcm
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:UpdateGcmChannel

프로젝트의 FCM 채널을 활성화하거나 업데이트합니다. 이 채널을 통해 Amazon Pinpoint는 Google Cloud Messaging(GCM) 서비스를 대체하는 FCM 서비스를 통해 Android 앱에 푸시 알림을 보낼 수 있습니다.

- URI – /apps/*projectId*/channels/gcm
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:DeleteSmsChannel

프로젝트의 SMS 채널을 비활성화합니다.

- URI – /apps/*projectId*/channels/sms
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetSmsChannel

프로젝트의 SMS 채널에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/channels/sms
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:UpdateSmsChannel

프로젝트의 SMS 채널을 활성화하거나 업데이트합니다.

- URI – /apps/*projectId*/channels/sms
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetChannels

애플리케이션의 각 채널 기록 및 상태에 대한 정보를 검색합니다.

- URI – /apps/*application-id*/channels
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/channels

mobiletargeting:DeleteVoiceChannel

애플리케이션의 음성 채널을 비활성화하고 채널의 기존 설정을 삭제합니다.

- URI – /apps/*application-id*/channels/voice
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:GetVoiceChannel

애플리케이션의 음성 채널 상태 및 설정에 대한 정보를 검색합니다.

- URI – /apps/*application-id*/channels/voice
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:UpdateVoiceChannel

애플리케이션에 대해 음성 채널을 활성화하거나 애플리케이션용 음성 채널의 상태 및 설정을 업데이트합니다.

- URI – /apps/*application-id*/channels/voice
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

엔드포인트

다음 권한은 Amazon Pinpoint 계정의 엔드포인트 관리에 관여합니다. Amazon Pinpoint에서 엔드포인트는 메시지가 향하는 단일 대상입니다. 예를 들어 고객의 이메일 주소, 전화 번호 또는 모바일 디바이스 토큰이 엔드포인트가 될 수 있습니다.

mobiletargeting:DeleteEndpoint

엔드포인트를 삭제합니다.

- URI – /apps/*projectId*/endpoints/*endpointId*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:GetEndpoint

특정 엔드포인트에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/endpoints/*endpointId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:RemoveAttributes

애플리케이션과 연결된 모든 엔드포인트에서 속성 유형이 동일한 하나 이상의 속성을 제거합니다.

- URI – apps/*application-id*/attributes/*attribute-type*
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/attributes/*attribute-type*

mobiletargeting:UpdateEndpoint

엔드포인트를 생성하거나 엔드포인트 정보를 업데이트합니다.

- URI – /apps/*projectId*/endpoints/*endpointId*
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:UpdateEndpointsBatch

배치 작업으로 엔드포인트를 생성하거나 업데이트합니다.

- URI – /apps/*projectId*/endpoints
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

이벤트 스트림

다음 권한은 Amazon Pinpoint 계정의 이벤트 스트림 관리에 관여합니다.

mobiletargeting:DeleteEventStream

프로젝트의 이벤트 스트림을 삭제합니다.

- URI – /apps/*projectId*/eventstream/
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:GetEventStream

프로젝트의 이벤트 스트림에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/eventstream/
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:PutEventStream

프로젝트의 이벤트 스트림을 생성 또는 업데이트합니다.

- URI – /apps/*projectId*/eventstream/
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

이벤트

다음 권한은 Amazon Pinpoint 계정의 이벤트 작업 관리에 관여합니다. Amazon Pinpoint에서 가져오기 작업을 생성하여 Amazon S3 버킷에 저장된 엔드포인트 정의를 기반으로 세그먼트를 생성합니다.

mobiletargeting:PutEvents

엔드포인트를 기록할 새 이벤트를 생성하거나, 기존 이벤트가 연결된 엔드포인트 데이터를 생성 또는 업데이트합니다.

- URI – /apps/*application-id*/events
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/events

내보내기 작업

다음 권한은 Amazon Pinpoint 계정의 내보내기 작업 관리에 관여합니다. Amazon Pinpoint에서 내보내기 작업을 생성하면 엔드포인트에 관한 정보를 Amazon S3 버킷으로 보내어 저장하거나 분석할 수 있습니다.

mobiletargeting:CreateExportJob

엔드포인트 정의를 Amazon S3로 내보낼 내보내기 작업을 생성합니다.

- URI – /apps/*projectId*/jobs/export
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/export

mobiletargeting:GetExportJob

프로젝트의 특정 내보내기 작업에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/jobs/export/*jobId*

- 매서드 – GET
- 리소스 ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

`mobiletargeting:GetExportJobs`

프로젝트의 모든 내보내기 작업의 목록을 검색합니다.

- URI – `/apps/projectId/jobs/export`
- 매서드 – GET
- 리소스 ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

가져오기 작업

다음 권한은 Amazon Pinpoint 계정의 가져오기 작업 관리에 관여합니다. Amazon Pinpoint에서 가져오기 작업을 생성하여 Amazon S3 버킷에 저장된 엔드포인트 정의를 기반으로 세그먼트를 생성합니다.

`mobiletargeting:CreateImportJob`

Amazon S3에서 엔드포인트 정의를 가져와 세그먼트를 생성합니다.

- URI – `/apps/projectId/jobs/import`
- 메서드 – POST
- 리소스 ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetImportJob`

프로젝트의 특정 가져오기 작업에 대한 정보를 검색합니다.

- URI – `/apps/projectId/jobs/import/jobId`
- 매서드 – GET
- 리소스 ARN – `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/import/jobId`

`mobiletargeting:GetImportJobs`

프로젝트의 모든 가져오기 작업에 대한 정보를 검색합니다.

- URI – `/apps/projectId/jobs/import`
- 매서드 – GET

- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

여정

다음 권한은 Amazon Pinpoint 계정의 여정 관리와 관련이 있습니다.

mobiletargeting:CreateJourney

프로젝트에 대한 여정을 만듭니다.

- URI – /apps/*projectId*/journeys
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys

mobiletargeting:GetJourney

특정 여정에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/journeys/*journeyId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:ListJourneys

프로젝트의 모든 여정에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/journeys
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys

mobiletargeting:UpdateJourney

특정 여정에 대한 구성 및 기타 설정을 업데이트합니다.

- URI – /apps/*projectId*/journeys/*journeyId*
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:UpdateJourneyState

활성 여정을 취소합니다.

- URI – /apps/*projectId*/journeys/*journeyId*/state
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/state

mobiletargeting:DeleteJourney

특정 여정을 삭제합니다.

- URI – /apps/*projectId*/journeys/*journeyId*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

메시지 템플릿

다음 사용 권한은 Amazon Pinpoint 계정에 대한 메시지 템플릿을 만들고 관리하는 것과 관련이 있습니다. 메시지 템플릿은 모든 Amazon Pinpoint 프로젝트에 대해 보내는 메시지에서 정의, 저장 및 재사용할 수 있는 콘텐츠 및 설정 집합입니다.

mobiletargeting:ListTemplates

Amazon Pinpoint 계정과 연결된 모든 메시지 템플릿에 대한 정보를 검색합니다.

- URI – /templates
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates

mobiletargeting:ListTemplateVersions

특정 메시지 템플릿의 모든 버전에 대한 정보를 검색합니다.

- URI – /templates/*template-name*/*template-type*/versions
- 매서드 – GET
- 리소스 ARN – 해당 사항 없음

mobiletargeting:UpdateTemplateActiveVersion

메시지 템플릿의 특정 버전을 활성 버전 템플릿으로 지정합니다.

- URI – /templates/*template-name*/*template-type*/active-version
- 매서드 – GET
- 리소스 ARN – 해당 사항 없음

mobiletargeting:GetEmailTemplate

이메일 채널을 통해 전송되는 메시지에 대한 메시지 템플릿에 대한 정보를 검색합니다.

- URI – /templates/*template-name*/email
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:CreateEmailTemplate

이메일 채널을 통해 전송되는 메시지에 대한 메시지 템플릿을 만듭니다.

- URI – /templates/*template-name*/email
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:UpdateEmailTemplate

이메일 채널을 통해 전송되는 메시지에 대한 기존 메시지 템플릿을 업데이트합니다.

- URI – /templates/*template-name*/email
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:DeleteEmailTemplate

이메일 채널을 통해 보낸 메시지의 메시지 템플릿을 삭제합니다.

- URI – /templates/*template-name*/email
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:GetPushTemplate

푸시 알림 채널을 통해 전송되는 메시지에 대한 메시지 템플릿에 대한 정보를 검색합니다.

- URI – /templates/*template-name*/push
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:CreatePushTemplate

푸시 알림 채널을 통해 전송되는 메시지에 대한 메시지 템플릿을 만듭니다.

- URI – /templates/*template-name*/push
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:UpdatePushTemplate

푸시 알림 채널을 통해 전송되는 메시지에 대한 기존 메시지 템플릿을 업데이트합니다.

- URI – /templates/*template-name*/push
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:DeletePushTemplate

푸시 알림 채널을 통해 전송된 메시지에 대한 메시지 템플릿을 삭제합니다.

- URI – /templates/*template-name*/push
- 메서드 – DELETE

- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:GetSmsTemplate

SMS 채널을 통해 전송되는 메시지에 대한 메시지 템플릿에 대한 정보를 검색합니다.

- URI – /templates/*template-name*/sms
- 메서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:CreateSmsTemplate

SMS 채널을 통해 전송되는 메시지에 대한 메시지 템플릿을 만듭니다.

- URI – /templates/*template-name*/sms
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:UpdateSmsTemplate

SMS 채널을 통해 전송되는 메시지에 대한 기존 메시지 템플릿을 업데이트합니다.

- URI – /templates/*template-name*/sms
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:DeleteSmsTemplate

SMS 채널을 통해 전송된 메시지의 메시지 템플릿을 삭제합니다.

- URI – /templates/*template-name*/sms
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:GetVoiceTemplate

음성 채널을 통해 전송되는 메시지에 대한 메시지 템플릿에 대한 정보를 검색합니다.

- URI – /templates/*template-name*/voice
- 메서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:CreateVoiceTemplate

음성 채널을 통해 전송되는 메시지에 대한 메시지 템플릿을 만듭니다.

- URI – /templates/*template-name*/voice
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:UpdateVoiceTemplate

음성 채널을 통해 전송되는 메시지에 대한 기존 메시지 템플릿을 업데이트합니다.

- URI – /templates/*template-name*/voice
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:DeleteVoiceTemplate

음성 채널을 통해 보낸 메시지의 메시지 템플릿을 삭제합니다.

- URI – /templates/*template-name*/voice
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

메시지

다음 권한은 Amazon Pinpoint 계정의 메시지 및 푸시 알림 전송과 관련됩니다. SendMessages 및 SendUsersMessages 작업을 사용하면 세그먼트와 캠페인을 먼저 생성할 필요 없이 특정 엔드포인트로 메시지를 전송할 수 있습니다.

mobiletargeting:SendMessages

특정 엔드포인트로 메시지 또는 푸시 알림을 전송합니다.

- URI – /apps/*projectId*/messages
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

mobiletargeting:SendUsersMessages

특정 사용자 ID와 연결된 모든 엔드포인트로 메시지 또는 푸시 알림을 전송합니다.

- URI – /apps/*projectId*/users-messages
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

일회용 암호

다음 권한은 Amazon Pinpoint에서의 일회용 암호(OTP) 전송 및 확인에 관여합니다.

mobiletargeting:SendOTPMessage

일회용 암호가 포함된 문자 메시지를 전송합니다.

- URI – /apps/*projectId*/otp
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/otp

mobiletargeting:VerifyOTPMessage

SendOTPMessage 작업을 사용하여 생성된 일회용 암호(OTP)의 유효성을 확인합니다.

- URI – /apps/*projectId*/verify-otp
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/verify-otp

전화 번호 확인

다음 권한은 Amazon Pinpoint의 전화번호 확인 서비스 사용과 관련됩니다.

mobiletargeting:PhoneNumberValidate

전화번호 관련 정보를 검색합니다.

- URI – /phone/number/validate
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:phone/number/validate

Projects

다음 권한은 Amazon Pinpoint 계정의 프로젝트 관리에 관여합니다. 원래 프로젝트를 애플리케이션이라고 했습니다. 이러한 작업에서는 Amazon Pinpoint 애플리케이션이 Amazon Pinpoint 프로젝트와 동일합니다.

mobiletargeting:CreateApp

Amazon Pinpoint 프로젝트를 생성합니다.

- URI – /apps
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps

mobiletargeting:DeleteApp

Amazon Pinpoint 프로젝트를 삭제합니다.

- URI – /apps/*projectId*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetApp

Amazon Pinpoint 프로젝트에 대한 정보를 검색합니다.

- URI – /apps/*projectId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetApps

Amazon Pinpoint 계정과 연결된 모든 프로젝트에 대한 정보를 검색합니다.

- URI – /apps
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps

mobiletargeting:GetApplicationSettings

Amazon Pinpoint 프로젝트의 기본 설정을 검색합니다.

- URI – /apps/*projectId*/settings
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:UpdateApplicationSettings

Amazon Pinpoint 프로젝트의 기본 설정을 업데이트합니다.

- URI – /apps/*projectId*/settings
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

추천 모델

다음 권한은 추천 모델에서 추천 데이터를 검색 및 처리하기 위해 Amazon Pinpoint 구성을 관리하는 것과 관련이 있습니다. 추천 모델은 데이터에서 패턴을 찾아 맞춤형 추천을 예측 및 생성하는 기계 학습 모델의 한 유형입니다.

mobiletargeting:CreateRecommenderConfiguration

추천 모델에 대한 Amazon Pinpoint 구성을 생성합니다.

- URI – /recommenders
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders

mobiletargeting:GetRecommenderConfigurations

Amazon Pinpoint 계정과 연결된 모든 추천 모델 구성 정보를 검색합니다.

- URI – /recommenders
- 매서드 – GET

- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders

mobiletargeting:GetRecommenderConfiguration

추천 모델의 개별 Amazon Pinpoint 구성에 대한 정보를 검색합니다.

- URI – /recommenders/*recommenderId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

mobiletargeting:UpdateRecommenderConfiguration

추천 모델에 대한 Amazon Pinpoint 구성을 업데이트합니다.

- URI – /recommenders/*recommenderId*
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

mobiletargeting:DeleteRecommenderConfiguration

추천 모델에 대한 Amazon Pinpoint 구성을 삭제합니다.

- URI – /recommenders/*recommenderId*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

Segments

다음 권한은 Amazon Pinpoint 계정의 세그먼트 관리에 관여합니다. Amazon Pinpoint에서 세그먼트는 사용자가 정의하는 특정 속성을 공유하는 캠페인에 대한 수신자 그룹입니다.

mobiletargeting:CreateSegment

세그먼트를 생성합니다. 사용자가 Amazon Pinpoint 외부에서 엔드포인트 데이터를 가져와 세그먼트를 생성하도록 허용하려면 mobiletargeting:CreateImportJob 작업을 허용합니다.

- URI – /apps/*projectId*/segments
- 메서드 – POST

- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:DeleteSegment

세그먼트를 삭제합니다.

- URI – /apps/*projectId*/segments/*segmentId*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegment

특정 세그먼트에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/segments/*segmentId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentExportJobs

세그먼트의 엔드포인트 정의를 가져오는 작업에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/segments/*segmentId*/jobs/export
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*/jobs/export

mobiletargeting:GetSegments

프로젝트의 모든 세그먼트에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/segments
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetSegmentImportJobs

Amazon S3에서 엔드포인트 정의를 가져와 세그먼트를 생성하는 작업에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/segments/*segmentId*/jobs/import
- 매서드 – GET

- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersion

특정 세그먼트 버전에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/segments/*segmentId*/versions/*versionId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersions

세그먼트의 현재 및 이전 버전에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/segments/*segmentId*/versions
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:UpdateSegment

특정 세그먼트를 업데이트합니다.

- URI – /apps/*projectId*/segments/*segmentId*
- 메서드 – PUT
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

Tags

다음 권한은 Amazon Pinpoint 리소스에 대한 태그의 조회 및 관리와 관련이 있습니다.

mobiletargeting:ListTagsForResource

프로젝트, 캠페인, 메시지 템플릿 또는 세그먼트와 관련된 태그에 대한 정보를 검색합니다.

- URI – /tags/*resource-arn*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:*

mobiletargeting:TagResource

프로젝트, 캠페인, 메시지 템플릿 또는 세그먼트에 하나 이상의 태그를 추가합니다.

- URI – /tags/*resource-arn*
- 메서드 – POST
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:*

mobiletargeting:UntagResource

프로젝트, 캠페인, 메시지 템플릿 또는 세그먼트에서 하나 이상의 태그를 제거합니다.

- URI – /tags/*resource-arn*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:*

Users

다음 권한은 사용자 관리에 관여합니다. Amazon Pinpoint에서 사용자는 메시지를 수신하는 개인에 해당됩니다. 단일 사용자는 하나 이상의 엔드포인트와 연결되었을 수 있습니다.

mobiletargeting:DeleteUserEndpoints

사용자 ID와 연결된 모든 엔드포인트를 삭제합니다.

- URI – /apps/*projectId*/users/*userId*
- 메서드 – DELETE
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/users/*userId*

mobiletargeting:GetUserEndpoints

사용자 ID와 연결된 모든 엔드포인트에 대한 정보를 검색합니다.

- URI – /apps/*projectId*/users/*userId*
- 매서드 – GET
- 리소스 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/users/*userId*

Amazon Pinpoint SMS 및 음성 버전 1 API 작업

이 섹션에서는 Amazon Pinpoint SMS 및 음성 API에서 사용할 수 있는 기능에 대한 작업을 알아봅니다. 이 API는 Amazon Pinpoint에서 SMS 및 음성 채널을 사용하고 관리하기 위한 고급 옵션을 제공하는 보충 API입니다. 이 API를 자세히 알아보려면 Amazon Pinpoint SMS 및 음성 API 참조를 살펴보세요.

`sms-voice:CreateConfigurationSet`

음성 메시지를 전송하기 위한 구성 집합을 생성합니다.

- URI – `/sms-voice/configuration-sets`
- 메서드 – POST
- 리소스 ARN – 사용할 수 없습니다. *를 사용합니다.

`sms-voice:DeleteConfigurationSet`

음성 메시지를 전송하기 위한 구성 집합을 삭제합니다.

- URI – /sms-voice/configuration-sets/*`ConfigurationSetName`*
- 메서드 – DELETE
- 리소스 ARN – 사용할 수 없습니다. *를 사용합니다.

`sms-voice:GetConfigurationSetEventDestinations`

구성 집합과 여기에 포함된 이벤트 대상에 대한 정보를 검색합니다.

- URI – /sms-voice/configuration-sets/*`ConfigurationSetName`*/event-destinations
- 매서드 – GET
- 리소스 ARN – 사용할 수 없습니다. *를 사용합니다.

`sms-voice:CreateConfigurationSetEventDestination`

음성 이벤트에 대한 이벤트 대상을 생성합니다.

- URI – /sms-voice/configuration-sets/*`ConfigurationSetName`*/event-destinations
- 메서드 – POST
- 리소스 ARN – 사용할 수 없습니다. *를 사용합니다.

`sms-voice:UpdateConfigurationSetEventDestination`

음성 이벤트에 대한 이벤트 대상을 업데이트합니다.

- URI – /sms-voice/configuration-sets/*ConfigurationSetName*/event-destinations/*EventDestinationName*
- 메서드 – PUT
- 리소스 ARN – 사용할 수 없습니다. *를 사용합니다.

`sms-voice:DeleteConfigurationSetEventDestination`

음성 이벤트에 대한 이벤트 대상을 삭제합니다.

- URI – /sms-voice/configuration-sets/*ConfigurationSetName*/event-destinations/*EventDestinationName*
- 메서드 – DELETE
- 리소스 ARN – 사용할 수 없습니다. *를 사용합니다.

`sms-voice:SendVoiceMessage`

음성 메시지를 생성해서 전송합니다.

- URI – /sms-voice/voice/message
- 메서드 – POST
- 리소스 ARN – 사용할 수 없습니다. *를 사용합니다.

Amazon Pinpoint ID 기반 정책 예제

기본적으로 사용자 및 역할은 Amazon Pinpoint 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console AWS CLI또는 AWS API를 사용하여 작업을 수행할 수 없습니다. IAM 관리자는 필요한 리소스에서 특정 API 작업을 수행할 수 있는 권한을 사용자와 역할에게 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 사용자 또는 그룹에 이러한 정책을 연결해야 합니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 JSON 탭에서 정책 생성을 참조하세요.

주제

- 정책 모범 사례
- Amazon Pinpoint 콘솔 사용
- 예: 단일 Amazon Pinpoint 프로젝트에 액세스
- 예제: 태그를 기반으로 Amazon Pinpoint 리소스 보기
- 예제: 사용자가 자체 권한을 볼 수 있도록 허용
- 예제: Amazon Pinpoint API 작업에 대한 액세스 제공
- 예제: Amazon Pinpoint SMS 및 음성 API 작업에 대한 액세스 제공
- 예제: 특정 IP 주소에 대한 Amazon Pinpoint 프로젝트 액세스 제한
- 예제: 태그를 기반으로 Amazon Pinpoint 액세스 제한
- 예: Amazon Pinpoint가 Amazon SES에서 확인된 ID를 사용하여 이메일을 보내도록 허용

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Amazon Pinpoint 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하

여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.

- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

Amazon Pinpoint 콘솔 사용

Amazon Pinpoint 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은 AWS 계정의 Amazon Pinpoint 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적으로 권한을 적용하는 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다. 이러한 엔터티가 Amazon Pinpoint 콘솔을 사용할 수 있도록 하려면 엔터티에 정책을 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

다음 예제 정책은 특정 AWS 리전의 Amazon Pinpoint 콘솔에 대한 읽기 전용 액세스를 제공합니다. 여기에는 Amazon Simple Email Service(Amazon SES), IAM, Amazon Kinesis처럼 Amazon Pinpoint 콘솔에서 사용하는 다른 서비스에 대한 읽기 전용 액세스가 포함됩니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UseConsole",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
```

```
                "kinesis:ListStreams",
                "s3:List*",
                "ses:Describe*",
                "ses:Get*",
                "ses:List*",
                "sns:ListTopics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

이전 정책 예제에서 *region*을 AWS 리전 이름으로 바꾸고 *accountId*를 AWS 계정 ID로 바꿉니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

예: 단일 Amazon Pinpoint 프로젝트에 액세스

특정 프로젝트에 대한 액세스만 제공하는 읽기 전용 정책을 만들 수도 있습니다. 다음 예제 정책은 사용자가 콘솔에 로그인하여 프로젝트 목록을 볼 수 있도록 허용합니다. 또한 사용자가 Amazon Pinpoint 콘솔에서 사용되는 다른 AWS 서비스(예: Amazon SES, IAM, Amazon Kinesis)에서 관련 리소스에 대한 정보를 볼 수 있도록 허용합니다. 하지만 사용자에게 정책에 지정된 프로젝트에 대한 추가 정보만 볼 수 있도록 허용합니다. 이 정책을 수정하여 추가 프로젝트 또는 AWS 리전에 대한 액세스를 허용할 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

앞의 예에서 *region*을 AWS 리전 이름으로 바꾸고, *accountId*를 AWS 계정 ID로 바꾸고, *projectId*를 액세스를 제공하려는 Amazon Pinpoint 프로젝트의 ID로 바꿉니다.

마찬가지로 프로젝트 `810c7aab86d42fb2b56c8c966example` ID가 있는 프로젝트와 같이 Amazon Pinpoint 프로젝트 중 하나에 대한 쓰기 액세스가 제한된 AWS 계정의 사용자에게 권한을 부여하는 정책을 생성할 수 있습니다. 이 경우 사용자는 세그먼트 및 캠페인과 같은 프로젝트 구성 요소를 확인, 추가 및 업데이트할 수 있지만 구성 요소는 삭제할 수 없습니다.

mobiletargeting:Get 및 mobiletargeting:List 작업에 대한 권한을 부여하는 것 외에도 mobiletargeting:Create, mobiletargeting:Update 및 mobiletargeting:Put 작업에 대한 권한을 부여하는 정책을 생성합니다. 대부분의 프로젝트 구성 요소를 생성 및 관리하는 데 필요한 추가적인 사용 권한입니다. 예제:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LimitedWriteProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*",
                "mobiletargeting:Create*",
                "mobiletargeting:Update*",
                "mobiletargeting:Put*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example",
    "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example/*",
                "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
```

```
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

예제: 태그를 기반으로 Amazon Pinpoint 리소스 보기

자격 증명 기반 정책의 조건을 사용하여 태그를 기반으로 Amazon Pinpoint 리소스에 대한 액세스를 제어할 수 있습니다. 이 예제 정책은 Amazon Pinpoint 리소스 보기를 허용하기 위해 이러한 종류의 정책을 생성하는 방법을 보여 줍니다. 하지만 `Owner` 리소스 태그가 해당 사용자의 사용자 이름 값을 가지고 있는 경우에만 권한이 부여됩니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewResourceIfOwner",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
```

```
            ],
            "Resource": "arn:aws:mobiletargeting:*:*:*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:*"
                }
            }
        }
    ]
}
```

이러한 유형의 정책을 계정의 사용자에게 연결할 수 있습니다. `richard-roe`라는 사용자가 Amazon Pinpoint 리소스를 보려고 하면 리소스에 `Owner=richard-roe` 또는 `owner=richard-roe` 태그를 지정해야 합니다. 그렇지 않으면 액세스가 거부됩니다. 조건 키 이름은 대소문자를 구분하지 않기 때문에 조건 태그 키 `Owner`는 `Owner` 및 `owner` 모두와 일치합니다. 자세한 내용은 IAM 사용자 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.

예제: 사용자가 자체 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여 줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
```

```
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

예제: Amazon Pinpoint API 작업에 대한 액세스 제공

이 섹션에서는 Amazon Pinpoint의 기본 API인 Amazon Pinpoint API에서 사용할 수 있는 기능에 대한 액세스를 허용하는 정책 예제를 제공합니다. 이 API를 자세히 알아보려면 Amazon Pinpoint API 참조를 살펴보세요.

읽기 전용 액세스

다음 예제 정책은 특정 AWS 리전의 Amazon Pinpoint 계정에 있는 모든 리소스에 대한 읽기 전용 액세스를 허용합니다.

JSON

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

```
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

앞의 예에서 *region*을 AWS 리전 이름으로 바꾸고 *accountId*를 계정 ID로 바꿉니다 AWS .

관리자 액세스

다음 예제 정책은 모든 Amazon Pinpoint 계정 내 모든 Amazon Pinpoint 작업 및 리소스에 대해 전체 액세스를 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

앞의 예제에서 *accountId*를 해당되는 AWS 계정 ID로 바꿉니다.

예제: Amazon Pinpoint SMS 및 음성 API 작업에 대한 액세스 제공

이 섹션에서는 Amazon Pinpoint SMS 및 음성 API에서 사용할 수 있는 기능에 대한 액세스를 허용하는 정책 예제를 제공합니다. 이 API는 Amazon Pinpoint에서 SMS 및 음성 채널을 사용하고 관리하기 위한 고급 옵션을 제공하는 보충 API입니다. 이 API를 자세히 알아보려면 Amazon Pinpoint SMS 및 음성 API 참조를 살펴보세요.

읽기 전용 액세스

다음 예제 정책은 AWS 계정의 모든 Amazon Pinpoint SMS 및 음성 API 작업과 리소스에 대한 읽기 전용 액세스를 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SMSVoiceReadOnly",
            "Effect": "Allow",
            "Action": [
                "sms-voice:Get*",
                "sms-voice:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

관리자 액세스

다음 예제 정책은 AWS 계정의 모든 Amazon Pinpoint SMS 및 음성 API 작업과 리소스에 대한 전체 액세스를 허용합니다.

JSON

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

```
        {
            "Sid": "SMSVoiceFullAccess",
            "Effect": "Allow",
            "Action": [
                "sms-voice:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

예제: 특정 IP 주소에 대한 Amazon Pinpoint 프로젝트 액세스 제한

다음 예제 정책은 지정된 프로젝트(*projectId*)에서 Amazon Pinpoint 작업을 수행할 수 있는 권한을 모든 사용자에게 부여합니다. 하지만 요청은 조건에 지정된 IP 주소 범위에서만 시작되어야 합니다.

이 문의 조건은 허용되는 인터넷 프로토콜 버전 4(IPv4) 주소의 `54.240.143.*` 범위를 식별합니다(한 가지 예외: `54.240.143.188`). `Condition` 블록은 `IpAddress` 및 `NotIpAddress` 조건과 AWS전체 `aws:SourceIp` 조건 키인 조건 키를 사용합니다. 이러한 조건 키에 대한 자세한 내용은 IAM 사용 설명서의 정책에서 조건 지정 섹션을 참조하세요. `aws:SourceIp` IPv4 값은 표준 CIDR 표기법을 사용합니다. 자세한 내용은 IAM 사용 설명서의 IP 주소 조건 연산자를 참조하세요.

JSON

```
{
    "Version":"2012-10-17",
    "Id":"AMZPinpointPolicyId1",
    "Statement":[
        {
            "Sid":"IPAllow",
            "Effect":"Allow",
            "Principal":"*",
            "Action":"mobiletargeting:*",
```

```
            "Resource":[
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
            ],
            "Condition":{
                "IpAddress":{
                    "aws:SourceIp":"54.240.143.0/24"
                },
                "NotIpAddress":{
                    "aws:SourceIp":"54.240.143.188/32"
                }
            }
        }
    ]
}
```

예제: 태그를 기반으로 Amazon Pinpoint 액세스 제한

다음 예제 정책은 지정된 프로젝트(*projectId*)에서 Amazon Pinpoint 작업을 수행할 수 있는 권한을 부여합니다. 그러나 조건에 지정된 대로 이름이 프로젝트의 `Owner` 리소스 태그의 값인 사용자로부터 요청이 시작된 경우에만 권한이 부여됩니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ModifyResourceIfOwner",
            "Effect": "Allow",
            "Action": "mobiletargeting:*",
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                }
            }
        }
```

```
    ]
}
```

예: Amazon Pinpoint가 Amazon SES에서 확인된 ID를 사용하여 이메일을 보내도록 허용

Amazon Pinpoint 콘솔을 통해 이메일 ID(예: 이메일 주소 또는 도메인)를 확인할 경우, Amazon Pinpoint와 Amazon SES 양쪽 모두에서 사용할 수 있도록 해당 ID가 자동으로 구성됩니다. 하지만 Amazon SES를 통해 이메일 ID를 확인하고 Amazon Pinpoint에서 해당 ID를 사용하려는 경우에는 이러한 ID에 정책을 적용해야 합니다.

아래의 예제 정책은 Amazon SES를 통해 확인된 이메일 ID를 사용하여 이메일을 보낼 수 있는 권한을 Amazon Pinpoint에 부여합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws:ses:us-east-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:mobiletargeting:us-
east-1:111122223333:apps/*"
                }
            }
        }
    ]
}
```

AWS GovCloud(미국 서부) 리전에서 Amazon Pinpoint를 사용하는 경우 대신 다음 정책 예제를 사용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws-us-gov:ses:us-gov-
west-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
                    "aws:SourceArn":"arn:aws-us-gov:mobiletargeting:us-gov-
west-1:111122223333:apps/*"
                }
            }
        }
    ]
}
```

일반적인 Amazon Pinpoint 태스크를 위한 IAM 역할

IAM 역할은 AWS 계정에서 생성하고 특정 권한을 부여할 수 있는 AWS Identity and Access Management (IAM) 자격 증명입니다. IAM 역할은 자격 AWS 증명이 수행할 수 있는 작업과 수행할 수 없는 작업을 결정하는 권한 정책이 있는 자격 증명입니다 AWS. 그러나 역할은 한 사람에게만 연결되지 않으며, 해당 역할이 필요한 사람이라면 누구든지 수임할 수 있습니다.

또한 역할에는 그와 연결된 표준 장기 보안 인증이 없습니다. 대신에 세션에 대한 임시 보안 자격 증명을 제공합니다. IAM 역할을 사용하여 일반적으로 AWS 리소스에 액세스할 수 없는 사용자, 앱, 애플리케이션 또는 서비스에 대한 액세스를 위임할 수 있습니다.

이러한 이유로 IAM 역할을 사용하여 계정의 특정 AWS 서비스 및 리소스와 Amazon Pinpoint를 통합할 수 있습니다. 예를 들어 Amazon Simple Storage Service(S3) 버킷에 저장하고 세그먼트에 사용하려는 엔드포인트 정의에 Amazon Pinpoint가 액세스하도록 허용할 수 있습니다. 또는 Amazon Pinpoint가 이벤트 데이터를 계정의 Amazon Kinesis 스트림으로 스트리밍하도록 허용할 수도 있습니다. 마찬가지로 IAM 역할을 사용하여 웹 또는 모바일 앱이 앱에 AWS 키를 포함하지 않고(교체하기 어렵고 사용자가 잠재적으로 추출할 수 있는 경우) 엔드포인트를 등록하거나 Amazon Pinpoint 프로젝트의 사용 데이터를 보고하도록 허용할 수 있습니다.

이러한 시나리오의 경우 IAM 역할을 사용하여 Amazon Pinpoint에 대한 액세스 권한을 위임할 수 있습니다. 이 섹션에서는 IAM 역할을 사용하여 다른 AWS 서비스를 사용하는 일반적인 Amazon Pinpoint 태스크의 예제를 설명하고 제공합니다. 웹 및 모바일 앱에서 IAM 역할을 사용하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.

주제

- 엔드포인트 또는 세그먼트를 가져오기 위한 IAM 역할
- 엔드포인트 또는 세그먼트를 내보내기 위한 IAM 역할
- Amazon Personalize에서 권장 사항을 검색하기 위한 IAM 역할
- 이벤트를 Kinesis로 스트리밍하기 위한 IAM 역할
- Amazon SES로 이메일 보내기 위한 IAM 역할

엔드포인트 또는 세그먼트를 가져오기 위한 IAM 역할

Amazon Pinpoint를 사용하면 AWS 계정의 Amazon Simple Storage Service(Amazon S3) 버킷에서 엔드포인트 정의를 가져와 사용자 세그먼트를 정의할 수 있습니다. 가져오기 전에 Amazon Pinpoint에 필요한 권한을 위임해야 합니다. 이렇게 하려면 AWS Identity and Access Management (IAM) 역할을 생성하고 다음 정책을 역할에 연결합니다.

- `AmazonS3ReadOnlyAccess` AWS 관리형 정책. 이 정책은에서 생성 AWS및 관리하며 Amazon S3 버킷에 대한 읽기 전용 액세스 권한을 부여합니다.
- Amazon Pinpoint가 역할을 수임하도록 허용하는 신뢰 정책.

역할을 생성한 후에는 Amazon Pinpoint를 사용하여 Amazon S3 버킷에서 세그먼트를 가져올 수 있습니다. 버킷 생성, 엔드포인트 파일 생성, 콘솔을 사용하여 세그먼트 가져오기에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 세그먼트 가져오기 섹션을 참조하세요. 를 사용하여 프로그래밍 방식

으로 세그먼트를 가져오는 방법의 예는이 가이드Amazon Pinpoint에서 세그먼트 가져오기의 섹션을 AWS SDK for Java참조하세요.

IAM 역할(AWS CLI) 생성

AWS Command Line Interface ()를 사용하여 IAM 역할을 생성하려면 다음 단계를 완료합니다AWS CLI. 를 설치하지 않은 경우 AWS Command Line Interface 사용 설명서의 설치를 AWS CLI AWS CLI 참조하세요.

를 사용하여 IAM 역할을 생성하려면 AWS CLI

1. 역할에 대한 신뢰 정책을 포함하는 JSON 파일을 생성하여 로컬 위치에 저장합니다. 다음 신뢰 정책을 사용할 수 있습니다.

   JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
          {
               "Action": "sts:AssumeRole",
               "Effect": "Allow",
               "Principal": {
                    "Service": "pinpoint.amazonaws.com"
               },
               "Condition": {
                    "StringEquals": {
                         "aws:SourceAccount": "111122223333"
                    },
               "ArnLike": {
               "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/application-id"
                    }
               }
          }
     ]
}
```

   이전 예제에서 다음을 수행합니다.

   - *region*을 Amazon Pinpoint를 사용하는 AWS 리전으로 바꿉니다.

- *accountId*를 AWS 계정의 고유 ID로 바꿉니다.
- *application-id*를 프로젝트의 고유 ID로 바꿉니다.

2. 명령줄에서 create-role 명령을 사용하여 역할을 생성한 후 신뢰 정책을 연결합니다.

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
 file://PinpointImportTrustPolicy.json
```

file:// 접두사 다음에 신뢰 정책을 포함하는 JSON 파일의 경로를 지정합니다.

이 명령을 실행하면 터미널에서 다음과 유사한 출력이 표시됩니다.

3. attach-role-policy 명령을 사용하여 AmazonS3ReadOnlyAccess AWS 관리형 정책을 역할에 연결합니다.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

엔드포인트 또는 세그먼트를 내보내기 위한 IAM 역할

내보내기 작업을 생성하여 엔드포인트 목록을 얻을 수 있습니다. 내보내기 작업을 생성할 경우 프로젝트 ID를 지정해야 하며, 선택에 따라 세그먼트 ID를 지정할 수 있습니다. 이렇게 하면 Amazon Pinpoint는 프로젝트 또는 세그먼트와 연결된 엔드포인트 목록을 Amazon Simple Storage Service(S3) 버킷으로 내보내게 됩니다. 이 결과 만들어지는 파일에는 JSON 형식의 엔드포인트와 그 속성(예: 채널, 주소, 옵트인/옵트아웃 상태, 생성일, 엔드포인트 ID)의 목록이 들어 있습니다.

내보내기 작업을 생성하려면 Amazon Pinpoint에서 Amazon S3 버킷에 쓸 수 있도록 허용하는 IAM 역할을 구성해야 합니다. 역할 구성 프로세스는 두 단계로 구성됩니다.

1. 엔터티(이 경우 Amazon Pinpoint)가 특정 Amazon S3 버킷에 쓰도록 허용하는 IAM 정책을 생성합니다.
2. IAM 역할을 생성하여 여기에 정책을 연결합니다.

이 주제에는 이러한 두 단계를 완료하는 절차가 포함되어 있습니다. 이 절차에서는 이미 Amazon S3 버킷을 생성했고, 내보낸 세그먼트를 저장할 폴더가 그 버킷 안에 있다고 가정합니다. 버킷 생성에 대한 내용을 알아보려면 Amazon Simple Storage Service 사용 설명서의 버킷 생성을 참조하세요.

또한 이 절차에서는 AWS Command Line Interface (AWS CLI)를 이미 설치하여 구성했다고 가정합니다. 설정에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 설치를 AWS CLI AWS CLI참조하세요.

1단계: IAM 정책 생성

IAM 정책은 자격 증명 또는 리소스와 같은 엔터티에 대한 권한을 정의합니다. Amazon Pinpoint 엔드포인트를 내보내는 역할을 생성하려면 특정 Amazon S3 버킷 내의 특정 폴더로 쓰기를 허용하는 정책을 생성해야 합니다. 아래의 정책 예제는 최소 권한, 즉 단일 태스크를 수행하는 데 필요한 권한만 부여하는 보안 사례를 따릅니다.

IAM 정책을 생성하려면

1. 텍스트 편집기에서 새로운 파일을 생성합니다. 다음 코드를 파일에 붙여넣습니다.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::*" ]
        },
        {
            "Sid": "AllowRootAndHomeListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
            "Condition": {
                "StringEquals": {
                    "s3:delimiter": [ "/" ],
                    "s3:prefix": [
                        "",
```

```
                    "Exports/"
                ]
            }
        }
    },
    {
        "Sid": "AllowListingOfUserFolder",
        "Action": [
            "s3:ListBucket"
        ],
        "Effect": "Allow",
        "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
        "Condition": {
            "StringLike": {
                "s3:prefix": [
                    "Exports/*"
                ]
            }
        }
    },
    {
        "Sid": "AllowAllS3ActionsInUserFolder",
        "Action": [ "s3:*" ],
        "Effect": "Allow",
        "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket/Exports/*" ]
    }
  ]
}
```

앞 코드에서 *amzn-s3-demo-bucket-example-bucket*의 모든 인스턴스를 세그먼트 정보를 내보낼 폴더가 포함된 Amazon S3 버킷의 이름으로 바꿉니다. 또한 *####*의 모든 인스턴스를 폴더 자체의 이름으로 바꿉니다.

작업을 마치면 파일 이름을 s3policy.json(으)로 저장합니다.

2. 를 사용하여 s3policy.json 파일이 있는 디렉터리로 AWS CLI이동합니다. 그리고 다음 명령을 사용하여 정책을 생성합니다.

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
 file://s3policy.json
```

정책이 무사히 생성되면 다음과 비슷한 출력이 표시됩니다.

```
{
    "Policy": {
        "CreateDate": "2018-04-11T18:44:34.805Z",
        "IsAttachable": true,
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PolicyId": "ANPAJ2YJQRJCG3EXAMPLE",
        "UpdateDate": "2018-04-11T18:44:34.805Z",
        "Arn": "arn:aws:iam::123456789012:policy/s3ExportPolicy",
        "PolicyName": "s3ExportPolicy",
        "Path": "/"
    }
}
```

정책의 Amazon 리소스 이름(ARN)을 복사합니다(앞 예에서 `arn:aws:iam::123456789012:policy/s3ExportPolicy`). 다음 섹션에서 새 역할을 만들 때 이 ARN을 공급해야 합니다.

> **Note**
>
> `CreatePolicy` 작업을 수행할 권한이 없는 계정이라는 메시지가 보이면 새 IAM 정책과 역할을 생성할 수 있는 사용자에게 정책을 연결해야 합니다. 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거를 참조하세요.

2단계: IAM 역할 생성

IAM 정책을 작성했으므로 역할을 생성하여 여기에 정책을 연결할 수 있습니다. 각 IAM 역할에는 신뢰 정책이 포함되어 있습니다. 이 정책은 어떤 엔터티가 역할을 수임하도록 허용되는지 지정하는 규칙 집합입니다. 이 섹션에서는 Amazon Pinpoint에 역할을 수임하도록 허용하는 신뢰 정책을 생성합니다. 다음에는 역할 자체를 생성한 다음 이전 단계에서 만든 정책을 연결합니다.

IAM 역할을 만들려면

1. 텍스트 편집기에서 새로운 파일을 생성합니다. 다음 코드를 파일에 붙여넣습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
                }
            }
        }
    ]
}
```

파일을 `trustpolicy.json`(으)로 저장합니다.

2. 를 사용하여 `trustpolicy.json` 파일이 있는 디렉터리로 AWS CLI이동합니다. 다음 명령을 입력하여 새 역할을 생성합니다.

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
 file://trustpolicy.json
```

3. 명령줄에서 다음 명령을 실행하여 이전 단계에서 생성한 정책을 방금 생성한 역할에 연결합니다.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

앞 명령에서 *arn:aws:iam::123456789012:policy/s3ExportPolicy*를 이전 섹션에서 만든 정책의 ARN으로 바꿉니다.

Amazon Personalize에서 권장 사항을 검색하기 위한 IAM 역할

Amazon Personalize 캠페인으로 배포된 Amazon Pinpoint 솔루션에서 권장 데이터를 검색하도록 Amazon Pinpoint를 구성할 수 있습니다. 이 데이터를 사용하여 각 수신자의 속성 및 동작에 따라 메시지 수신자에게 맞춤형 추천을 보낼 수 있습니다. 자세한 내용은 Amazon Pinpoint 사용 설명서의 기계 학습 모델을 참조하세요.

Amazon Personalize 캠페인에서 권장 데이터를 검색하려면 먼저 Amazon Pinpoint가 캠페인에서 데이터를 검색할 수 있도록 하는 AWS Identity and Access Management (IAM) 역할을 생성해야 합니다. Amazon Pinpoint는 콘솔을 사용하여 Amazon Pinpoint에서 추천 모델을 설정할 때 자동으로 이 역할을 생성할 수 있습니다. 또는 이 역할을 수동으로 만들 수 있습니다.

역할을 수동으로 생성하려면 IAM API를 사용하여 다음 단계를 완료합니다.

1. 엔터티(이 경우 Amazon Pinpoint)가 Amazon Personalize 캠페인에서 권장 데이터를 검색할 수 있도록 허용하는 IAM 정책을 생성합니다.
2. IAM 역할을 생성하여 여기에 IAM 정책을 연결합니다.

이 주제에서는 AWS Command Line Interface ()를 사용하여 이러한 단계를 완료하는 방법을 설명합니다AWS CLI. 이미 Amazon Personalize 솔루션을 생성한 후 이 솔루션 Amazon Personalize 캠페인으로 배포했다고 가정합니다. 캠페인 생성 및 배포에 대한 자세한 내용은 Amazon Personalize 개발자 안내서의 캠페인 생성을 참조하세요.

이 주제는 AWS CLI를 이미 설치하여 구성한 상태를 전제로 설명합니다. 설정에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 설치를 AWS CLI AWS CLI참조하세요.

1단계: IAM 정책 생성

IAM 정책은 ID 또는 리소스와 같은 엔터티에 대한 권한을 정의합니다. Amazon Pinpoint가 Amazon Personalize 캠페인에서 권장 데이터를 검색할 수 있도록 하는 역할을 생성하려면 먼저 해당 역할에 대한 IAM 정책을 생성해야 합니다. 이 정책은 Amazon Pinpoint가 다음 작업을 수행할 수 있도록 허용해야 합니다.

- 캠페인에서 배포하는 솔루션에 대한 구성 정보 검색(`DescribeSolution`)
- 캠페인의 상태 확인(`DescribeCampaign`)
- 캠페인에서 추천 데이터 검색(`GetRecommendations`)

다음 절차의 정책 예제에서는 특정 Amazon Personalize 캠페인에서 배포한 특정 Amazon Personalize 솔루션에 대해 이 액세스를 허용합니다.

IAM 정책을 생성하려면

1. 텍스트 편집기에서 새로운 파일을 생성합니다. 다음 코드를 파일에 붙여넣습니다.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "RetrieveRecommendationsOneCampaign",
            "Effect": "Allow",
            "Action": [
                "personalize:DescribeSolution",
                "personalize:DescribeCampaign",
                "personalize:GetRecommendations"
            ],
            "Resource": [
                "arn:aws:personalize:us-
east-1:111122223333:solution/solutionId",
                "arn:aws:personalize:us-
east-1:111122223333:campaign/campaignId"
            ]
        }
    ]
}
```

   위의 예제에서 *####* 텍스트를 해당 정보로 바꿉니다.

   - *region* – Amazon Personalize 솔루션과 캠페인을 호스트하는 AWS 리전의 이름입니다.
   - *## ID* – 사용자의 AWS 계정 ID입니다.
   - *solutionId* – 캠페인에서 배포한 Amazon Personalize 솔루션의 고유한 리소스 ID입니다.
   - *campaignId* - 권장 데이터를 검색할 Amazon Personalize 캠페인의 고유한 리소스 ID입니다.
2. 작업을 마치면 파일 이름을 `RetrieveRecommendationsPolicy.json`(으)로 저장합니다.
3. 명령줄 인터페이스를 사용하여 `RetrieveRecommendationsPolicy.json` 파일을 저장한 디렉터리로 이동합니다.

4. 다음 명령을 입력하여 정책을 만들고 이름을 RetrieveRecommendationsPolicy로 지정합니다. 다른 이름을 사용하려면 *RetrieveRecommendationsPolicy*를 원하는 이름으로 변경합니다.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
 file://RetrieveRecommendationsPolicy.json
```

> **Note**
>
> CreatePolicy 작업을 수행할 권한이 없는 계정이라는 메시지를 수신하면 계정에 대한 새 IAM 정책과 역할을 생성할 수 있는 사용자에게 정책을 연결해야 합니다. 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거를 참조하세요.

5. 정책의 Amazon 리소스 이름(ARN)을 복사합니다(앞 예에서 arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy). 다음 섹션에서 IAM 역할을 생성하려면 이 ARN이 필요합니다.

2단계: IAM 역할 생성

IAM 정책을 생성한 후 IAM 역할을 생성하여 여기에 정책을 연결할 수 있습니다.

각 IAM 역할에는 신뢰 정책이 포함되어 있습니다. 이 정책은 어떤 엔터티가 역할을 수임하도록 허용되는지 지정하는 규칙 집합입니다. 이 섹션에서는 Amazon Pinpoint에 역할을 수임하도록 허용하는 신뢰 정책을 생성합니다. 다음으로 역할 자체를 만듭니다. 그런 다음 정책을 역할에 연결합니다.

IAM 역할을 만들려면

1. 텍스트 편집기에서 새로운 파일을 생성합니다. 다음 코드를 파일에 붙여넣습니다.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
```

```
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "AWS:SourceArn": "arn:aws:mobiletargeting:us-
east-1:444455556666:apps/*"
                }
            }
        }
    ]
}
```

2. 파일을 RecommendationsTrustPolicy.json(으)로 저장합니다.
3. 명령줄 인터페이스를 사용하여 RecommendationsTrustPolicy.json 파일을 저장한 디렉터리로 이동합니다.
4. 다음 명령을 입력하여 새 역할을 만들고 이름을 PinpointRoleforPersonalize로 지정합니다. 다른 이름을 사용하려면 *PinpointRoleforPersonalize*를 원하는 이름으로 변경합니다.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-
document file://RecommendationsTrustPolicy.json
```

5. 다음 명령을 입력하여 이전 단원에서 생성한 정책을 방금 생성한 역할에 연결합니다.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

앞 명령에서 *arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy*를 이전 섹션에서 만든 정책의 ARN으로 바꿉니다. 또한 역할에 대해 다른 이름을 지정한 경우 *PinpointRoleforPersonalize*를 4단계에서 지정한 역할의 이름으로 바꿉니다.

이벤트를 Kinesis로 스트리밍하기 위한 IAM 역할

Amazon Pinpoint는 앱에서 계정의 Amazon Kinesis 데이터 스트림 또는 Amazon Data Firehose 전송 스트림으로 앱 사용 데이터 또는 이벤트 데이터를 자동으로 전송할 수 있습니다 AWS . Amazon Pinpoint가 이벤트 데이터를 스트리밍할 수 있으려면 먼저 필요한 권한을 Amazon Pinpoint에 위임해야 합니다.

콘솔을 사용하여 이벤트 스트리밍을 설정할 경우 Amazon Pinpoint가 자동으로 필요한 권한을 가진 AWS Identity and Access Management (IAM) 역할을 생성합니다. 자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint 이벤트를 Kinesis에 스트리밍 섹션을 참조하세요.

수동으로 역할을 생성하려면 다음 정책을 역할에 연결합니다.

- Amazon Pinpoint가 이벤트 데이터를 스트림에 전송하도록 허용하는 권한 정책.
- Amazon Pinpoint가 역할을 수임하도록 허용하는 신뢰 정책.

역할을 생성한 후에는 자동으로 이벤트를 스트림으로 전송하도록 Amazon Pinpoint를 구성할 수 있습니다. 자세한 내용은 이 안내서의 Amazon Pinpoint를 사용하여 Kinesis 및 Firehose를 통해 앱 이벤트 데이터 스트리밍 섹션을 참조하세요.

IAM 역할(AWS CLI) 생성

AWS Command Line Interface (AWS CLI)를 사용하여 IAM 역할을 생성하려면 다음 단계를 수행합니다. Amazon Pinpoint 콘솔을 사용하여 역할을 생성하는 방법을 알아보려면 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint 이벤트를 Kinesis에 스트리밍 섹션을 참조하세요.

를 설치하지 않은 경우 AWS Command Line Interface 사용 설명서의 설치를 AWS CLI AWS CLI참조하세요. 또한 Kinesis 스트림 또는 Firehose 스트림을 생성해야 합니다. 이러한 리소스를 생성하는 방법에 대한 자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 생성 및 관리 또는 Amazon Data Firehose 개발자 안내서의 Amazon Data Firehose 전송 스트림 생성 섹션을 참조하세요.

를 사용하여 IAM 역할을 생성하려면 AWS CLI

1. 새 파일을 만듭니다. 다음 정책을 문서에 붙여 넣고 다음과 같이 변경합니다.

   - *region*을 Amazon Pinpoint를 사용하는 AWS 리전으로 바꿉니다.
   - *accountId*를 AWS 계정의 고유 ID로 바꿉니다.
   - *applicationId*를 프로젝트의 고유 ID로 바꿉니다.

   JSON

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

```
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
                }
            }
        }
    ]
}
```

작업을 마치면 파일 이름을 PinpointEventStreamTrustPolicy.json(으)로 저장합니다.

2. create-role 명령을 사용하여 역할을 생성한 후 신뢰 정책에 연결합니다.

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

3. 역할에 대한 권한 정책을 포함하는 새 파일을 생성합니다.

Kinesis 스트림으로 데이터를 전송하도록 Amazon Pinpoint를 구성할 경우 아래의 정책을 파일에 붙여 넣고 다음을 바꿉니다.

- *region*을 Amazon Pinpoint를 사용하는 AWS 리전으로 바꿉니다.
- *accountId*를 AWS 계정의 고유 ID로 바꿉니다.
- *streamName*을 Kinesis 스트림의 이름으로 바꿉니다.

JSON

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

```
            "kinesis:PutRecords",
            "kinesis:DescribeStream"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:kinesis:us-east-1:111122223333:stream/streamName"
        ]
    }
}
```

또는 Firehose 스트림으로 데이터를 전송하도록 Amazon Pinpoint를 구성할 경우 아래의 정책을 파일에 붙여 넣고 다음을 바꿉니다.

- *region*을 Amazon Pinpoint를 사용하는 AWS 리전으로 바꿉니다.
- *accountId*를 AWS 계정의 고유 ID로 바꿉니다.
- *delivery-stream-name*을 Firehose 스트림의 이름으로 바꿉니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "firehose:PutRecordBatch",
            "firehose:DescribeDeliveryStream"
        ],
        "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/delivery-
stream-name"
        ]
    }
}
```

작업을 마치면 파일 이름을 `PinpointEventStreamPermissionsPolicy.json`(으)로 저장합니다.

4. [put-role-policy](#) 명령을 사용하여 권한 정책을 역할에 연결합니다.

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json
```

Amazon SES로 이메일 보내기 위한 IAM 역할

Amazon Pinpoint는 Amazon SES 리소스를 사용하여 캠페인 또는 여정에 대한 이메일을 보냅니다. Amazon Pinpoint가 Amazon SES 리소스를 사용하여 이메일을 보내려면 먼저 Amazon Pinpoint에 필요한 권한을 부여해야 합니다. IAM 역할을 업데이트하거나 생성하려면 계정에 `iam:PutRolePolicy` 및 `iam:UpdateAssumeRolePolicy` 권한이 있어야 합니다.

Amazon Pinpoint 콘솔은 필요한 권한이 있는 AWS Identity and Access Management (IAM) 역할을 자동으로 생성할 수 있습니다. 자세한 내용은 Amazon Pinpoint 사용 설명서의 이메일 오케스트레이션 전송 역할 생성을 참조하세요.

수동으로 역할을 생성하려면 다음 정책을 역할에 연결합니다.

- Amazon Pinpoint에 Amazon SES 리소스에 대한 액세스 권한을 부여하는 권한 정책입니다.
- Amazon Pinpoint가 역할을 수임하도록 허용하는 신뢰 정책.

역할을 생성한 후 Amazon Pinpoint가 Amazon SES 리소스를 사용하도록 구성할 수 있습니다.

IAM 정책은 IAM 정책 시뮬레이터로 테스트할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 시뮬레이터로 IAM 정책 테스트를 참조하세요.

IAM 역할(AWS Management Console) 생성

다음 단계를 완료하여 캠페인 또는 여정에서 이메일을 보낼 IAM 역할을 수동으로 생성합니다.

1. IAM 사용 설명서의 JSON 편집기를 사용하여 정책 생성의 지침에 따라 새로운 권한 정책을 생성합니다.
    - 5단계에서는 IAM 역할에 대해 다음 권한 정책을 사용합니다.
        - *###*을 리소스가 있는 파티션으로 바꿉니다. 표준 AWS 리전의 경우 파티션은 입니다`aws`. 리소스가 다른 파티션에 있는 경우 파티션은 `aws-partitionname`입니다. 예를 들어 AWS , GovCloud(미국 서부)의 리소스에 대한 파티션은 입니다`aws-us-gov`.
        - *`region`*을 Amazon Pinpoint 프로젝트를 호스팅 AWS 리전 하는의 이름으로 바꿉니다.

- *accountId*를 해당 AWS 계정의 고유한 ID로 바꿉니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PinpointUsesSESForEmailSends",
            "Effect": "Allow",
            "Action": [
                "ses:SendEmail",
                "ses:SendRawEmail"
            ],
            "Resource": [
            "arn:aws:ses:us-east-1:111122223333:identity/*",
    "arn:aws:ses:us-east-1:111122223333:configuration-set/*"
            ]
        }
    ]
}
```

2. IAM 사용 설명서의 사용자 지정 신뢰 정책을 사용하여 역할 생성의 지침에 따라 새로운 신뢰 정책을 생성합니다.

   a. 4단계에서 다음 신뢰 정책을 사용합니다.

   - *accountId*를 해당 AWS 계정의 고유한 ID로 바꿉니다.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPinpoint",
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
```

```
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

b. 11단계에서 이전 단계에서 만든 권한 정책을 추가합니다.

Amazon Pinpoint의 ID 및 액세스 관리에 대한 문제 해결

다음 정보를 사용하여 Amazon Pinpoint 및 IAM에서 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- Amazon Pinpoint에서 태스크를 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 AWS 계정 외부의 사람이 내 Amazon Pinpoint 리소스에 액세스하도록 허용하고 싶습니다.

Amazon Pinpoint에서 태스크를 수행할 권한이 없음

에서 작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 문의하여 도움을 받아야 합니다. 관리자는 로그인 보안 인증 정보를 제공한 사람입니다.

다음 예제 오류는 `mateojackson` 사용자가 콘솔을 사용하여 프로젝트에 대한 세부 정보를 보려고 하지만 `mobiletargeting:GetApp` 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mobiletargeting:GetApp on resource: my-example-project
```

이 경우 Mateo는 *my-example-project* 태스크를 사용하여 `mobiletargeting:GetApp` 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon Pinpoint에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예제 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 Amazon Pinpoint에서 태스크를 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 AWS 계정 외부의 사람이 내 Amazon Pinpoint 리소스에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- Amazon Pinpoint에서 이러한 기능을 지원하는지 여부를 알아보려면 Amazon Pinpoint에서 IAM을 사용하는 방법 섹션을 참조하세요.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 타사에 리소스에 대한 액세스를 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

Amazon Pinpoint의 로깅 및 모니터링

로깅 및 모니터링은 Amazon Pinpoint 프로젝트 및 기타 유형의 Amazon Pinpoint 리소스의 신뢰성, 가용성, 성능을 유지하는 데 중요한 부분입니다. 다중 지점 실패가 발생할 경우 보다 쉽게 디버깅하려면 Amazon Pinpoint 프로젝트 및 리소스의 모든 부분에서 모니터링 데이터를 로깅하고 수집해야 합니다. 는이 데이터를 로깅 및 수집하고 잠재적 인시던트에 대응하는 데 도움이 되는 몇 가지 도구를 AWS 제공합니다.

AWS CloudTrail

Amazon Pinpoint는 사용자 AWS CloudTrail, 역할 또는 다른 서비스가 Amazon Pinpoint에서 수행한 작업에 대한 레코드를 제공하는 AWS 서비스인와 통합됩니다. 여기에는 Amazon Pinpoint 콘솔의 작업과 Amazon Pinpoint API 작업에 대한 프로그래밍 방식 직접 호출이 포함됩니다. CloudTrail에서 수집한 정보를 사용하여 Amazon Pinpoint에 대해 어떤 요청이 수행되었는지 확인할 수 있습니다. 각 요청에 대해 요청이 이루어진 시기, 요청이 이루어진 IP 주소, 요청한 사람 및 추가 세부 정보를 식별할 수 있습니다. 자세한 내용은 이 안내서의 를 사용하여 Amazon Pinpoint API 호출 로깅 AWS CloudTrail 섹션을 참조하세요.

Amazon CloudWatch

이제 Amazon CloudWatch를 사용하여 Amazon Pinpoint 계정 및 프로젝트와 관련된 중요한 여러 지표를 수집, 확인 및 분석할 수 있습니다. CloudWatch를 사용하여 지표에 대한 값이 특정 조건을 충족하고 사용자가 정의한 임계값 내에 있거나 초과할 경우 이를 알리는 경보를 생성할 수도 있습니다. 경보를 생성하면 CloudWatch에서는 사용자가 지정한 Amazon Simple Notification Service(SNS) 주제에 대한 알림을 전송합니다. 자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon CloudWatch를 사용하여 Amazon Pinpoint 모니터링 섹션을 참조하세요.

AWS Health 대시보드

AWS Health 대시보드를 사용하면 Amazon Pinpoint 환경의 상태를 확인하고 모니터링할 수 있습니다. Amazon Pinpoint 서비스의 상태를 전체적으로 확인하려면 AWS 서비스 상태 대시보드를 사용합니다. AWS 환경에 더 구체적으로 영향을 미칠 수 있는 이벤트 또는 문제에 대한 기록 데이터를 확인, 모니터링 및 보려면 AWS Personal Health Dashboard를 사용합니다. 이러한 대시보드에 관한 자세한 내용은 AWS Health 사용 설명서를 참조하세요.

AWS Trusted Advisor

AWS Trusted Advisor 는 AWS 환경을 검사하고 보안 격차를 해결하고, 시스템 가용성과 성능을 개선하고, 비용을 절감할 수 있는 기회에 대한 권장 사항을 제공합니다. 모든 AWS 고객은 핵심 Trusted Advisor 검사 세트에 액세스할 수 있습니다. 비즈니스 또는 엔터프라이즈 지원 플랜을 보유한 고객은 추가 Trusted Advisor 검사를 받을 수 있습니다.

이러한 검사 중 다수는 전체 AWS 계정의 일부로 Amazon Pinpoint 리소스의 보안 태세를 평가하는 데 도움이 될 수 있습니다. 예를 들어, 핵심 Trusted Advisor 점검 항목에는 다음이 포함됩니다.

- 지원되는 각 AWS 리전에 대한 AWS 계정의 구성 로깅.
- Amazon Simple Storage Service(S3) 버킷에 대한 액세스 권한(세그먼트를 빌드하기 위해 Amazon Pinpoint로 가져오는 파일이 포함될 수 있음).
- AWS Identity and Access Management 사용자, 그룹 및 역할을 사용하여 Amazon Pinpoint 리소스에 대한 액세스를 제어합니다.
- AWS 환경 및 Amazon Pinpoint 리소스의 보안을 손상시킬 수 있는 IAM 구성 및 정책 설정입니다.

자세한 내용은지원 사용 설명서의 AWS Trusted Advisor를 참조하세요.

Amazon Pinpoint의 규정 준수 검증

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon Pinpoint의 보안 및 규정 준수를 평가합니다. 여기에는 보안 관리 제어를 위한 AWS 시스템 및 조직 제어(SOC), FedRAMP, HIPAA, ISO/IEC 27001:2013, 클라우드별 제어를 위한 ISO/IEC 27017:2015, 개인 데이터 보호를 위한 ISO/IEC 27018:2014, 품질 관리 시스템을 위한 ISO/IEC 9001:2015 등이 포함됩니다.

특정 규정 준수 프로그램의 범위에 속하는 AWS 서비스 목록은 규정 준수 프로그램 AWS 범위에 속하는 서비스규정 준수 프로그램 . 일반 정보는 AWS 규정 준수 프로그램 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 AWS Artifact에서 보고서 다운로드.

Amazon Pinpoint 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다.는 규정 준수를 지원하기 위해 다음 리소스를 AWS 제공합니다.

- 보안 및 규정 준수 빠른 시작 가이드 -이 배포 가이드에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수 중심 기준 환경을 배포하기 위한 단계를 제공합니다 AWS.
- HIPAA 보안 및 규정 준수를 위한 설계 백서 -이 백서에서는 기업이를 사용하여 HIPAA 준수 애플리케이션을 AWS 생성하는 방법을 설명합니다.
- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 산업 및 위치에 적용될 수 있습니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 -이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub CSPM -이 AWS 서비스는 보안 업계 표준 및 모범 사례 준수 여부를 확인하는 데 도움이 AWS 되는 내 보안 상태에 대한 포괄적인 보기를 제공합니다.

Amazon Pinpoint는 고객이 적절한 통신 채널을 사용할 때 AWS HIPAA 적격 서비스입니다. Amazon Pinpoint를 사용하여 HIPAA와 관련 법률 및 규정에 정의된 대로 개인 건강 정보(PHI)가 포함된 워크로드를 실행하려면 이메일 채널, 푸시 알림 채널 또는 SMS 채널을 사용하여 PHI가 포함된 메시지를 전송해야 합니다. SMS 채널을 사용하여 PHI가 포함된 메시지를 전송하는 경우 PHI가 포함되거나 포함될 수 있는 메시지를 전송할 명시적 목적으로 AWS 계정에 요청한 전용 단축 코드에서 해당 메시지를 보내야 합니다. 음성 채널은 AWS HIPAA 자격이 없으므로 음성 채널을 사용하여 PHI가 포함된 메시지를 보내지 마십시오.

Amazon Pinpoint의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며,이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

참조 아키텍처에 대한 자세한 내용은 Amazon Pinpoint의 복원력 있는 아키텍처 설명서를 참조하세요.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

Amazon Pinpoint의 인프라 보안

관리형 서비스인 Amazon Pinpoint는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요 AWS .

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon Pinpoint에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

모든 네트워크 위치에서 이러한 API 직접 호출을 수행할 수 있지만 Amazon Pinpoint는 리소스 기반 액세스 정책을 지원합니다. 이러한 정책에는 소스 IP 주소에 따른 제한이 포함될 수 있습니다. 이러한 유형의 정책에 대한 자세한 내용은 정책을 사용하여 액세스 관리 단원을 참조하십시오.

또한 다양한 AWS 보안 기능을 구성하고 사용하여 Amazon Pinpoint와 통합하는 모든 모바일 또는 웹 앱에서 Amazon Pinpoint 리소스에 대한 액세스를 제어할 수 있습니다. 여기에는 엔드포인트 추가, 엔드포인트 데이터 업데이트, 이벤트 데이터 제출, 사용 데이터 보고와 같은 작업에 대한 API 직접 호출 제한이 포함됩니다.

이러한 기능을 사용하려면 AWS 모바일 SDKs 또는 AWS Amplify JavaScript 라이브러리를 사용하여 모바일 및 웹 앱을 Amazon Pinpoint와 통합하는 것이 좋습니다. Android 또는 iOS 앱의 경우 AWS Mobile SDK for iOS각각 AWS Mobile SDK for Android 또는를 사용하는 것이 좋습니다. JavaScript 기반 모바일 또는 웹 앱의 경우 웹용 AWS Amplify JavaScript 라이브러리 또는 React Native용 AWS Amplify JavaScript 라이브러리를 사용하는 것이 좋습니다. 이러한 리소스에 대한 자세한 내용은 AWS 모바일 SDKs, 웹용 AWS Amplify 라이브러리 시작하기, 반응 네이티브용 AWS Amplify 라이브러리 시작하기를 참조하세요.

Amazon Pinpoint의 구성 및 취약성 분석

관리형 서비스인 Amazon Pinpoint는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다. https://d0.awsstatic.com/whitepapers/Security/AWS_Security_Whitepaper.pdf 즉,는 Amazon Pinpoint 계정 및 리소스의 기본 인프라를 강화, 패치, 업데이트 및 유지 관리하기 위한 기본 보안 작업 및 절차를 AWS 관리하고 수행합니다. 적합한 제3자가 이 절차를 검토하고 인증하였습니다.

자세한 정보는 다음 자료를 참조하세요.

- Amazon Pinpoint의 규정 준수 검증
- 공동 책임 모델
- Amazon Web Services: 보안 프로세스의 개요(백서)

Amazon Pinpoint의 보안 모범 사례

AWS Identity and Access Management(IAM) 계정을 사용하여 API 작업, 특히 Amazon Pinpoint 리소스를 생성, 수정 또는 삭제하는 작업에 대한 Amazon Pinpoint 액세스를 제어합니다. Amazon Pinpoint API의 경우 이러한 리소스에는 프로젝트, 캠페인 및 여정이 포함됩니다. Amazon Pinpoint SMS 및 Voice API의 경우 이러한 리소스에는 전화번호, 풀, 구성 세트가 포함됩니다.

- 자신을 포함하여 Amazon Pinpoint 리소스를 관리하는 각 사람에 대해 개별 사용자를 생성합니다. AWS 루트 자격 증명을 사용하여 Amazon Pinpoint 리소스를 관리하지 마세요.
- 각 사용자에게 각자의 임무를 수행하는 데 필요한 최소 권한 집합을 부여합니다.

- IAM 그룹을 사용해 여러 사용자에 대한 권한을 효과적으로 관리합니다.
- IAM 자격 증명을 정기적으로 순환합니다.

Amazon Pinpoint 보안에 대한 자세한 내용은 Amazon Pinpoint의 보안을 참조하세요. IAM에 대한 자세한 내용은 AWS ID 및 액세스 관리 섹션을 참조하세요. IAM 모범 사례에 대한 자세한 내용은 IAM 모범 사례 단원을 참조하십시오.

Amazon Pinpoint 할당량

다음 단원에서는 Amazon Pinpoint 리소스 및 작업에 적용되는 할당량(이전에는 한도라고 함)을 나열하고 설명합니다. 일부 할당량만 늘릴 수 있습니다. 할당량 증가를 요청할 수 있는지 여부를 확인하려면 각 단원의 증가 가능 여부 열을 참조하세요.

주제

- 프로젝트 할당량
- API 요청 할당량
- 캠페인 할당량
- 이메일 할당량
- 엔드포인트 할당량
- 엔드포인트 가져오기 할당량
- 이벤트 수집 할당량
- 여정 할당량
- Lambda 할당량
- 기계 학습 할당량
- 메시지 템플릿 할당량
- 푸시 알림 할당량
- 인앱 메시지 할당량
- 세그먼트 할당량
- SMS 할당량
- 10DLC 할당량
- 음성 할당량
- 할당량 증가 요청

프로젝트 할당량

다음 표에는 Amazon Pinpoint 프로젝트와 관련된 할당량이 나와 있습니다.

Resource	기본 할당량	증가 가능 여부
Projects	각각 최대 100개의 프로젝트를 보유 AWS 리전할 수 있습니다.	아니요

API 요청 할당량

Amazon Pinpoint는 AWS 계정에서 Amazon Pinpoint API에 수행할 수 있는 요청의 크기와 수를 제한하는 할당량을 구현합니다.

특정 유형의 리소스에 대해 별도로 지정하지 않는 한 호출(요청 및 응답) 페이로드의 최대 크기는 7MB입니다. 리소스에 다른 할당량이 있는지 확인하려면 해당 리소스 유형에 대해 이 주제에 해당하는 섹션을 참조하세요.

최대 요청 수는 할당량 유형 및 API 작업에 따라 다릅니다. Amazon Pinpoint는 API 요청에 대해 두 가지 유형의 할당량을 구현합니다.

- 속도 할당량 - 속도 제한이라고도 하는 이 유형의 할당량은 특정 작업에 대해 초당 수행할 수 있는 최대 요청 수를 정의합니다. 계정별로 송수신되는 요청의 속도를 제어합니다.
- 버스트 할당량 - 버스트 제한 또는 버스트 용량이라고도 하는 이 유형의 할당량은 계정에서 동시에 처리되는 최대 요청 수를 정의합니다.

다음 표에는 Amazon Pinpoint API의 속도 및 버스트 할당량이 나와 있습니다.

연산	기본 버스트/속도 할당량(초당 요청 수)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300
CreatePushTemplate	10
CreateSegment	25

연산	기본 버스트/속도 할당량(초당 요청 수)
CreateSmsTemplate	10
CreateVoiceTemplate	10
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4,000
SendUsersMessages	6,000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25
UpdateSmsTemplate	10
UpdateVoiceTemplate	10
기타 모든 작업	300

다음 표에는 CreateImportJob의 파일 가져오기 할당량이 나와 있습니다.

연산	기본 할당량	증가 가능 여부
최대 가져오기 파일 수	가져오기 작업당 파일 10,000개	아니요

이러한 할당량 중 하나를 초과하면 Amazon Pinpoint가 요청을 제한합니다. 즉, 유효한 요청을 거부하고 TooManyRequests 오류를 반환합니다. 제한은 특정 AWS 리전의 특정 작업에 대해 계정에서 요청한 총 요청 수를 기준으로 합니다. 또한 조절 결정은 각 작업에 대해 독립적으로 계산됩니다. 예를 들어 Amazon Pinpoint가 SendMessages 작업에 대한 요청을 제한하는 경우, UpdateEndpoint 작업에 대한 동시 요청을 성공적으로 완료할 수 있습니다.

캠페인 할당량

다음 할당량은 Amazon Pinpoint API의 캠페인 리소스에 적용됩니다.

다음 할당량은 별로 적용되며 AWS 리전 일부는 늘릴 수 있습니다. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

리소스	기본 할당량	증가 가능 여부
활성 캠페인	계정당 200개 **Note** 활성 캠페인은 완료되거나 실패하지 않은 캠페인입니다. 활성 캠페인의 상태는 SCHEDULED, EXECUTING 또는 PENDING_NEXT_RUN 입니다.	아니요
최대 세그먼트 크기	가져온 세그먼트의 경우: 캠페인당 100,000,000개	아니요

리소스	기본 할당량	증가 가능 여부
	동적 세그먼트의 경우: 무제한	
이벤트 기반 캠페인	각 프로젝트에는 이벤트 발생 시 전송되는 캠페인이 최대 25개 포함될 수 있습니다. 이벤트 기반 트리거를 사용하는 캠페인은 동적 세그먼트를 사용해야 합니다. 가져온 세그먼트는 사용할 수 없습니다. AWS Mobile SDK를 사용하여 앱을 Amazon Pinpoint와 통합하는 경우 이벤트 기반 캠페인의 메시지는 앱이 AWS Mobile SDK for Android 버전 2.7.2 이상 또는 AWS Mobile SDK for iOS 버전 2.6.30 이상을 실행하는 고객에게만 전송됩니다. Amazon Pinpoint는 5분 내에 이벤트 기반 캠페인에서 메시지를 전달할 수 없으면 해당 메시지를 삭제하고 다시 전달을 시도하지 않습니다.	아니요

이메일 할당량

이메일 채널에는 다음 섹션의 할당량이 적용됩니다.

이메일 메시지 할당량

Resource	기본 할당량	증가 가능 여부
최대 메시지 크기(첨부 파일 포함)	메시지당 10MB	아니요
확인된 ID 수	ID 10,000개 **Note** ID란 이메일 주소나 도메인 또는 이 두 가지의 조합을 가리킵니다. Amazon Pinpoint를 통해 보내는 이메일은 모두 확인된 ID로 전송해야 합니다.	아니요

이메일 발신자 및 수신자 할당량

Resource	기본 할당량	증가 가능 여부
발신자 주소	모든 발신 주소 또는 도메인을 확인해야 합니다.	아니요
수신자 주소	계정이 샌드박스에 있을 경우 모든 수신자 이메일 주소 또는 도메인을 확인해야 합니다. 계정이 샌드박스에서 나온 경우 임의의 유효한 주소로 이메일을 보낼 수 있습니다.	예
메시지당 수신자 수	메시지당 수신자 50명	아니요

Resource	기본 할당량	증가 가능 여부
확인할 수 있는 ID 수	AWS 리전당 10,000개의 자격 증명 **Note** ID란 이메일 주소나 도메인 또는 이 두 가지의 조합을 가리킵니다. Amazon Pinpoint를 통해 보내는 이메일은 모두 확인된 ID로 전송해야 합니다.	아니요

이메일 전송 할당량

전송 할당량, 전송 속도, 샌드박스 한도는 동일한 리전의 두 서비스 간에 공유됩니다. us-east-1에서 Amazon SES를 사용하고 있고 샌드박스에서 제거되어 전송 할당량/속도가 증가한 경우, 이러한 변경 사항은 us-east-1의 Pinpoint 계정에 모두 적용됩니다.

Resource	기본 할당량	증가 가능 여부
24시간의 기간당 보낼 수 있는 이메일 수(발신 할당량)	계정이 샌드박스에 있는 경우: 24시간의 기간당 이메일 200개입니다. 계정이 샌드박스 외부에 있는 경우, 할당량은 구체적인 사용 사례에 따라 달라집니다. **Note** 이 할당량은 고유한 메시지 전송 수와 반대로 수신자 수에 따라 결정	예

Resource	기본 할당량	증가 가능 여부
	됩니다. 수신자는 To: 행의 모든 이메일 주소입니다.	
초당 보낼 수 있는 이메일 수(발신 속도)	계정이 샌드박스에 있는 경우: 초당 이메일 1개입니다. 계정이 샌드박스 외부에 있는 경우, 속도는 구체적인 사용 사례에 따라 달라집니다. **Note** 이 속도는 고유한 메시지 전송 수와 반대로 수신자 수에 따라 결정됩니다. 수신자는 To: 행의 모든 이메일 주소입니다.	예

엔드포인트 할당량

다음 할당량은 Amazon Pinpoint API의 엔드포인트 리소스에 적용됩니다.

엔드포인트당 지원되는 최대 속성 수는 250개이고 최대 엔드포인트 크기는 15KB입니다. 그러나 속성 수는 모든 속성을 포함하는 엔드포인트의 전체 크기에 따라 제한될 수 있습니다. 템플릿에 속성을 추가할 때 오류가 발생하는 경우 각 속성의 데이터 양을 줄이거나 속성 수를 줄이는 것이 좋습니다.

Resource	기본 할당량	증가 가능 여부
엔드포인트 크기	최대 크기: 15KB	아니요

Resource	기본 할당량	증가 가능 여부
`Attributes` , `Metrics` 및 `UserAttributes` 파라미터에 단체 할당된 속성	애플리케이션당 모든 속성 파라미터에 대해 250개	아니요
`Attributes` 파라미터에 할당된 속성	애플리케이션당 모든 속성 파라미터에 대해 250개	아니요
`Metrics` 파라미터에 할당된 속성	애플리케이션당 모든 속성 파라미터에 대해 250개	아니요
`UserAttributes` 파라미터에 할당된 속성	애플리케이션당 모든 속성 파라미터에 대해 250개	아니요
속성 이름 길이	50자	아니요
속성 값 길이	100자	아니요
`EndpointBatchRequest` 페이로드의 `EndpointBatchItem` 객체	페이로드당 100개. 페이로드 크기는 7MB 이하여야 합니다.	아니요
사용자 ID가 동일한 엔드포인트	사용자 ID당 15개의 고유 엔드포인트	아니요
`Attributes` 파라미터 속성에 할당된 값	속성당 50개	아니요
`UserAttributes` 파라미터 속성에 할당된 값	속성당 50개	아니요

엔드포인트 가져오기 할당량

다음 할당량은 엔드포인트를 Amazon Pinpoint로 가져올 때 적용됩니다.

Resource	기본 할당량	증가 가능 여부
활성 가져오기 작업	계정당 10	아니요

Resource	기본 할당량	증가 가능 여부
	가져오기 작업은 실행 중인 경우에만 이 할당량에 포함됩니다. 가져오기 작업이 완료되면 더 이상 이 할당량에 포함되지 않습니다.	
가져오기 크기	가져오기 작업당 1GB 예를 들어 엔드포인트 한 개가 4KB 이하라면 250,000개를 가져올 수 있습니다.	아니요

이벤트 수집 할당량

다음 할당량은 AWS Mobile SDKs Amazon Pinpoint API의 이벤트 리소스를 사용하여 이벤트를 수집하는 데 적용됩니다.

Resource	기본 할당량	증가 가능 여부
최대 사용자 지정 이벤트 유형 수	앱당 1,500개	아니요
최대 사용자 지정 속성 키 수	앱당 500개	아니요
속성 키당 최대 사용자 지정 속성 값 수	100,000. 100,000개를 초과하는 개수는 여전히 등록되지만 Amazon Pinpoint 분석 콘솔에서는 사용할 수 없습니다.	아니요
속성 키당 최대 문자 수	50	아니요
속성 값당 최대 문자 수	200. 문자 수가 200자를 초과하면 이벤트가 삭제됩니다.	아니요
최대 사용자 지정 지표 키 수	앱당 500개	아니요

Resource	기본 할당량	증가 가능 여부
요청의 최대 이벤트 수	요청당 100	아니요
최대 요청 크기	4MB	아니요
개별 이벤트의 최대 크기	1,000KB	아니요
각 이벤트에 대한 속성 키 및 지표 키의 최대 수	요청당 40	아니요

여정 할당량

다음 할당량은 여정에 적용됩니다.

다음 할당량은 별로 적용되며 AWS 리전 일부는 늘릴 수 있습니다. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

리소스	기본 할당량	증가 가능 여부
최대 활성 여정 수	계정당 50개	아니요
최대 활성 이벤트 트리거 여정 수	계정당 20개	아니요
최대 여정 활동 수	여정당 40	아니요
최대 세그먼트 크기	가져온 세그먼트의 경우: 여정당 100,000,000 동적 세그먼트의 경우: 무제한	아니요
최대 컨택 센터 활동	여정당 3	아니요
최대 휴무일 규칙 수	채널당 20개	아니요
휴무일 규칙 이름의 최대 길이	150자	아니요

리소스	기본 할당량	증가 가능 여부
휴무일 규칙의 시작 시간과 종료 시간 사이의 최대 일수	7일	아니요
최대 영업 시간 규칙 수	하루당 4개	아니요

Lambda 할당량

다음 할당량은 Lambda에서 데이터를 검색 및 처리하기 위한 Amazon Pinpoint 구성에 적용됩니다.

Resource	기본 할당량	증가 가능 여부
Lambda 함수에 대한 간접 호출 페이로드(요청 및 응답)의 최대 크기	6MB	아니요
Lambda 함수가 데이터를 처리할 때까지 대기하는 최대 시간	15초	아니요
엔드포인트당 최대 이벤트 속성 수	5	아니요
최대 이벤트 속성 이름 문자 수	128자	아니요
최대 이벤트 속성 값 문자 수	128자	아니요
여정을 실행할 수 있는 최대 일수	540일	아니요

기계 학습 할당량

다음 할당량은 기계 학습(ML) 모델에서 데이터를 검색 및 처리하기 위한 Amazon Pinpoint 구성에 적용됩니다.

Resource	기본 할당량	증가 가능 여부
최대 모델 구성 수	메시지 템플릿당 1개 계정당 100개	아니요
최대 추천 수	엔드포인트 또는 사용자당 5개	아니요
엔드포인트 또는 사용자당 최대 추천 속성 수	1 - 속성 값이 AWS Lambda 함수에 의해 처리되지 않는 경우 10 - 속성 값이 AWS Lambda 함수에 의해 처리되는 경우	아니요
추천 속성 이름의 최대 길이	속성 이름의 경우 50자 속성 표시 이름 25자(콘솔의 속성 찾기에 나타나는 이름)	아니요
Amazon Personalize에서 검색되는 추천 속성 값의 최대 길이	100자	아니요
Lambda 함수에 대한 간접 호출 페이로드(요청 및 응답)의 최대 크기	6MB	아니요
Lambda 함수가 데이터를 처리할 때까지 대기하는 최대 시간	15초	아니요
Lambda 함수를 간접 호출하기 위한 최대 시도 횟수	3회	아니요

ML 모델을 사용하도록 Amazon Pinpoint를 구성하는 방법에 따라 추가 할당량이 적용될 수 있습니다. Amazon Personalize 할당량에 대해 자세히 알아보려면 Amazon Personalize 개발자 안내서의 할당량을 참조하세요. AWS Lambda 할당량에 대해 자세히 알아보려면 AWS Lambda 개발자 안내서의 할당량을 참조하세요.

메시지 템플릿 할당량

다음 할당량은 Amazon Pinpoint 계정의 메시지 템플릿에 적용됩니다.

Resource	기본 할당량	증가 가능 여부
최대 메시지 템플릿 수	계정당 20,000개	아니요
최대 버전 수	템플릿당 5,000개	아니요
이메일 템플릿의 최대 문자 수	60만 자	아니요
인앱 템플릿의 최대 문자 수	20만 자	아니요
푸시 알림 템플릿의 기본 템플릿 부분에 있는 최대 문자 수	4,000자	아니요
푸시 알림 템플릿의 ADM 관련 템플릿 부분에 있는 최대 문자 수	6,000자	아니요
푸시 알림 템플릿의 APNs 전용 템플릿 부분에 있는 최대 문자 수	4,000자	아니요
푸시 알림 템플릿의 BAIDU 특정 템플릿 부분에 있는 최대 문자 수	4,000자	아니요
푸시 알림 템플릿의 FCM 특정 템플릿 부분에 있는 최대 문자 수	4,000자	아니요
SMS 템플릿의 최대 문자 수	1,600자	아니요
음성 템플릿의 최대 문자 수	10,000자	아니요

푸시 알림 할당량

다음 할당량은 푸시 알림 채널을 통해 Amazon Pinpoint가 보내는 메시지에 적용됩니다.

Resource	기본 할당량	증가 가능 여부
단일 캠페인에서 초당 전송 가능한 최대 푸시 알림 수	초당 25,000개	예
ADM(Amazon Device Messaging) 메시지 페이로드 크기	메시지당 6KB	아니요
Apple 푸시 알림 서비스(APNs) 메시지 페이로드 크기	메시지당 4KB	아니요
APNs 샌드박스 메시지 페이로드 크기	메시지당 4KB	아니요
Baidu Cloud Push 메시지 페이로드 크기	메시지당 4KB	아니요
Firebase Cloud Messaging (FCM) 메시지 페이로드 크기	메시지당 4KB	아니요

인앱 메시지 할당량

다음 할당량은 Amazon Pinpoint로 관리하는 인앱 메시지에 적용됩니다.

다음 할당량은 별로 적용되며 AWS 리전 일부는 늘릴 수 있습니다. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

리소스	기본 할당량	증가 가능 여부
초당 `GetInAppMessages` API를 호출할 수 있는 최대 횟수.	초당 5,000개의 요청	예

리소스	기본 할당량	증가 가능 여부
인앱 메시징 캠페인	각 프로젝트에는 인앱 메시징 채널을 사용하는 캠페인을 최대 25개까지 포함할 수 있습니다.	예. Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

세그먼트 할당량

다음 할당량은 Amazon Pinpoint API의 세그먼트 리소스에 적용됩니다.

Resource	기본 할당량	증가 가능 여부
세그먼트를 생성하는 데 사용할 수 있는 최대 차원 수	세그먼트당 100개	아니요
세그먼트당 최대 세그먼트 그룹 수	5	아니요
세그먼트당 최대 소스 세그먼트 수	5	아니요
소스 세그먼트의 최대 깊이. 예를 들어 세그먼트에 있는 소스 세그먼트에도 소스 세그먼트가 있는 경우 깊이 체인은 이 한도를 초과하지 않습니다.	5	아니요

SMS 할당량

SMS 할당량은 AWS End User Messaging SMS 사용 설명서의 SMS 할당량을 참조하세요.

10DLC 할당량

10DLC 할당량은 AWS End User Messaging SMS 사용 설명서의 10DLC 할당량을 참조하세요.

음성 할당량

음성 할당량은 AWS End User Messaging SMS 사용 설명서의 음성 할당량을 참조하세요.

할당량 증가 요청

앞의 표에서 증가 가능 여부 열의 값이 예일 경우 해당 할당량에 대해 증가를 요청할 수 있습니다.

할당량 증가 요청

1. https://console.aws.amazon.com/ AWS Management Console 에 로그인합니다.
2. https://console.aws.amazon.com/support/home#/case/create 새 AWS 지원 사례를 생성합니다.
3. 지원 사례 창에서 사례 생성을 선택합니다.
4. 서비스 한도 증가를 원하십니까? 링크를 선택합니다.
5. 서비스 할당량 증가에서 서비스에 대해 다음 옵션 중 하나를 선택합니다.
   - 이메일 채널과 관련된 할당량 증가를 요청하려면 Pinpoint 이메일을 선택합니다.
   - SMS 지출 한도 또는 SMS 전송 속도에 대한 할당량 증가를 요청하려면 Pinpoint SMS를 선택합니다. 다른 모든 SMS 할당량 증가의 경우 Pinpoint를 선택합니다.
   - 음성 채널과 관련된 할당량 증가를 요청하려면 Pinpoint 음성를 선택합니다.
   - 다른 Amazon Pinpoint 기능과 관련된 할당량 증가를 요청하려면 Pinpoint를 선택합니다.
6. 선택한 서비스에 따라 다음을 입력하라는 메시지가 표시될 수 있습니다.
   - (선택 사항) SMS 메시지를 보낼 사이트 또는 앱의 링크 제공에 SMS 메시지를 보낼 웹 사이트, 애플리케이션 또는 서비스에 대한 정보를 입력합니다.
   - (선택 사항) 보낼 메시지의 유형에서 긴 코드를 사용하여 보낼 메시지의 유형을 선택합니다.
     - 일회용 암호 - 고객이 웹 사이트 또는 애플리케이션에서 인증을 위해 사용해야 하는 암호가 담긴 메시지입니다.
     - 프로모션 - 비즈니스 또는 서비스(예: 특가 행사, 공지 사항)를 홍보하는 중요하지 않은 메시지입니다.
     - 트랜잭션 - 고객 트랜잭션을 지원하는 중요한 정보 메시지(예: 주문 확인, 계정 알림)입니다. 트랜잭션 메시지에 홍보 또는 마케팅 콘텐츠를 포함해서는 안 됩니다.
   - (선택 사항) 메시지를 보낼 AWS 리전에서 메시지를 보낼 리전을 선택합니다.
   - (선택 사항) 메시지를 보낼 국가에 단축 코드를 구입할 국가 또는 리전을 입력합니다.

- (선택 사항) 고객이 메시지 수신에 옵트인하는 방법에서 옵트인 프로세스에 대한 세부 정보를 제공합니다.
- (선택 사항) 고객에게 메시지를 보내는 데 사용할 메시지 템플릿을 제공합니다 필드에 사용할 메시지 템플릿을 포함시킵니다.

7. 요청에서 다음과 같이 하세요.

   - 리전의 경우 AWS 리전을 선택합니다.
   - 리소스 유형에서 일반 한도를 선택합니다. 리소스 유형 필드는 일부 서비스에만 있습니다.
   - 할당량에서 변경할 할당량을 선택합니다.
   - 새 할당량 값에 새 할당량 값을 입력합니다.
   - 추가에서 동일한 할당량 증가를 요청하려면 다른 요청 추가를 AWS 리전선택한 다음 추가를 선택하고 새 요청을 AWS 리전 작성합니다.

8. 증가시킬 할당량을 선택한 다음 할당량에 대해 원하는 새 값을 입력합니다.
9. 사례 설명에서 할당량 증가를 요청하는 이유를 설명합니다.
10. 연락처 옵션의 기본 연락처 언어에서 AWS 지원 팀과 통신할 때 사용할 언어를 선택합니다.
11. 연락 방법에서 AWS 지원 팀과 통신하는 원하는 방법을 선택합니다.
12. 제출을 선택합니다.

AWS 지원 팀은 24시간 이내에 요청에 대한 초기 응답을 제공합니다.

시스템이 원치 않는 콘텐츠 또는 악성 콘텐츠를 전송하지 않도록 하기 위해 각 요청을 신중하게 고려해야 합니다. 가능한 경우 이 24시간 이내에 요청에 대한 권한을 부여합니다. 그러나 사용자의 추가 정보가 필요한 경우, 요청을 해결하는 데 시간이 더 오래 걸릴 수도 있습니다.

정책에 맞지 않는 사용 사례인 경우, 요청에 대한 권한을 부여하지 않을 수도 있습니다.

Amazon Pinpoint 문서 기록

다음 표에서는 2018년 12월 이후 Amazon Pinpoint 개발자 안내서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

- 최신 설명서 업데이트: 2023년 11월 16일

변경 사항	설명	날짜
지원 종료 알림	지원 종료 공지: 2026년 10월 30일에 AWS 는 Amazon Pinpoint에 대한 지원을 종료합니다. 2026년 10월 30일 이후에는 Amazon Pinpoint 콘솔 또는 Amazon Pinpoint 리소스(엔드포인트, 세그먼트, 캠페인, 여정, 분석)에 더 이상 액세스할 수 없습니다. 자세한 내용은 Amazon Pinpoint 지원 종료를 참조하세요. 참고: SMS, 음성, 모바일 푸시, OTP 및 전화번호 검증과 관련된 APIs는이 변경의 영향을 받지 않으며 AWS 최종 사용자 메시징에서 지원됩니다.	2025년 5월 20일
메시지 전송을 위한 CloudTrail 지원 추가	PutEvents, SendUserMessages, SendMessages API 직접 호출에 대한 CloudTrail 로깅 지원이 추가되었습니다. CloudTrail 로그 파일에서 지원되는 Amazon Pinpoint API 작업을 참조하세요.	2025년 2월 17일
Amazon Pinpoint는 사용 설명서 문서를 업데이트했습니다	푸시 리소스를 생성, 구성 및 관리하는 방법에 대한 최	2024년 7월 22일

	신 정보는 새로운 End User Messaging Push 사용 설명서를 참조하세요.	
이메일 헤더	이메일 메시지에 이메일 헤더를 추가할 수 있습니다. 자세한 내용은 구독 취소 헤더가 포함된 이메일 전송을 참조하세요.	2024년 5월 7일
이메일 오케스트레이션	Amazon Pinpoint는 Amazon SES 리소스를 사용하여 이메일을 보내는 방법을 업데이트했습니다. 자세한 내용은 Amazon SES로 이메일 보내기 위한 IAM 역할을 참조하세요.	2024년 4월 30일
Amazon Pinpoint는 사용 설명서 문서를 업데이트했습니다	이제 SMS and Voice 리소스 관리 주제가 AWS End User Messaging SMS 사용 설명서로 리디렉션됩니다. 자세한 내용은 AWS End User Messaging SMS 사용 설명서 섹션을 참조하세요.	2024년 2월 8일
Amazon Pinpoint 할당량	최대 휴무일 규칙 수, 휴무일 규칙 이름의 최대 길이, 휴무일 규칙의 시작과 종료 시간 사이의 최대 일수 및 최대 영업 시간 규칙에 대한 할당량이 추가되었습니다. 자세한 내용은 Amazon Pinpoint 할당량을 참조하세요.	2023년 12월 19일

Amazon Pinpoint는 사용 설명서 문서를 업데이트했습니다	AWS End User Messaging SMS 및 음성 리소스를 생성, 구성 및 관리하는 방법에 대한 최신 정보는 새로운 AWS End User Messaging SMS 사용 설명서를 참조하세요.	2023년 11월 16일
Amazon Pinpoint 할당량	UpdateEndpointsBatch, UpdateEndpoint, PutEvents, DeleteEndpoint, GetEndpoint에 대한 할당량을 업데이트했습니다. 자세한 내용은 Amazon Pinpoint 할당량 섹션을 참조하세요.	2023년 9월 22일
Amazon Pinpoint 할당량	CreateEmailTemplate, CreateSmsTemplate, CreatePushTemplate, CreateInAppTemplate, CreateVoiceTemplate, UpdateEmailTemplate, UpdateSmsTemplate, UpdatePushTemplate, UpdateInAppTemplate, UpdateVoiceTemplate, CreateImportJob에 대한 할당량을 업데이트했습니다. 자세한 내용은 Amazon Pinpoint 할당량 섹션을 참조하세요.	2023년 9월 12일
여정 및 캠페인 실행 지표	여정 및 캠페인에 대한 새로운 분석 지표가 추가되었습니다. 자세한 내용은 여정 및 캠페인 실행 지표 섹션을 참조하세요.	2023년 4월 25일

Amazon Pinpoint에 대한 인터페이스 VPC 엔드포인트 생성	이제 Amazon Pinpoint는 인터페이스 VPC 엔드포인트를 지원합니다. 자세한 내용은 Amazon Pinpoint에 대한 인터페이스 VPC 엔드포인트 생성 섹션을 참조하세요.	2023년 4월 11일
전송 중 데이터 암호화	2023년 3월 22일부터 Amazon Pinpoint에서 TLS 1.0을 더 이상 지원하지 않지만, TLS 1.2 이상은 계속 사용할 수 있습니다. 자세한 내용은 전송 중 데이터 암호화를 참조하세요.	2023년 3월 20일
Amazon Pinpoint 할당량	캠페인, 여정, 인앱 메시지에 대한 할당량 증가를 요청하는 프로세스를 업데이트했습니다. 자세한 내용은 Amazon Pinpoint 할당량 섹션을 참조하세요.	2022년 12월 16일
지역별 가용성	이제 미국 동부(오하이오) 리전에서 Amazon Pinpoint를 사용할 수 있습니다.	2022년 10월 5일
IAM 역할 예제 업데이트	보안 모범 사례와 더욱 부합하도록 문서 전체의 여러 가지 IAM 역할 예제를 업데이트했습니다.	2022년 5월 27일

SMS 및 음성 API, 버전 2	이제 Amazon Pinpoint에 SMS 및 음성 메시지를 전송하기 위한 전용 API가 포함됩니다. 이 API에는 SMS 및 음성 메시지를 트랜잭션 방식으로 전송하는 고객에게 도움이 될 구성 세트, 풀, 옵트아웃 목록 등의 새로운 기능이 포함되어 있습니다. 자세한 내용은 Amazon Pinpoint SMS 및 음성 API 사용 섹션을 참조하세요.	2022년 4월 1일
일회성 암호 생성 및 검증	이제 Amazon Pinpoint에 일회용 암호(OTP)를 생성하고 이를 사용자에게 SMS 메시지로 보내는 기능이 포함됩니다. 사용자가 애플리케이션 또는 사이트에 OTP 코드를 입력할 때 OTP 코드를 검증하기 위한 API도 포함됩니다. 자세한 내용은 일회용 암호(OTP) 전송 및 검증 섹션을 참조하세요.	2021년 11월 26일
인앱 메시지	Amazon Pinpoint의 인앱 메시징 기능을 앱과 통합하는 방법에 대한 정보가 추가되었습니다.	2021년 11월 10일
코드 예제	일반적인 Amazon Pinpoint 작업을 위한 코드 예제 라이브러리가 추가되었습니다.	2021년 11월 3일
프로젝트 할당량	Amazon Pinpoint 프로젝트의 최대 수는 100개로 유지되지만 이제를 사용하여 서비스 한도 증가 요청을 열어이 할당량을 늘릴 수 있습니다 지원.	2021년 10월 11일

Lambda 정책 업데이트.	이제 특정 Lambda 권한 정책에 `AWS:SourceAccount` 조건을 포함해야 합니다. Amazon Pinpoint에서 사용자 지정 채널 생성 및 AWS Lambda으로 세그먼트 사용자 지정 주제의 정책 샘플을 업데이트했습니다.	2021년 10월 7일
UpdateEndpoint	이제 Amazon Pinpoint UpdateEndpoint API가 CloudTrail에 의해 기록됩니다.	2020년 11월 16일
사용자 지정 속성	이제 Amazon Pinpoint는 이메일 메시징 템플릿에 대해 최대 250개의 속성을 지원합니다. 할당량을 참조하세요.	2020년 9월 18일
지역별 가용성	이제 아시아 태평양(도쿄) 리전, 유럽(런던) 리전, 캐나다(중부) 리전에서 Amazon Pinpoint를 사용할 수 있습니다. Amazon Pinpoint SMS 및 음성 API는 이러한 리전에서 사용할 수 없습니다.	2020년 9월 10일
리전별 가용성	이제 아시아 태평양(도쿄) 리전에서 Amazon Pinpoint를 사용할 수 있습니다. Amazon Pinpoint SMS 및 음성 API는 이 리전에서 사용할 수 없습니다.	2020년 9월 2일
캠페인 이벤트	새 캠페인 이벤트 `delivery_type` 파라미터에 대한 정보를 캠페인 이벤트 섹션에 추가했습니다.	2023년 8월 2일

지역별 가용성	이제 아시아 태평양(서울) 리전에서 Amazon Pinpoint를 사용할 수 있습니다. Amazon Pinpoint API는 이 리전에서 음성 또는 SMS를 지원하지 않습니다.	2020년 7월 31일
리전별 가용성	이제 AWS GovCloud (US) 리전에서 Amazon Pinpoint를 사용할 수 있습니다.	2020년 4월 30일
사용자 지정 채널	Lambda 함수 또는 웹후크를 사용하여 사용자 지정 채널 생성에 대한 정보를 업데이트했습니다.	2020년 4월 23일
기계 학습	추천 모델에서 개인화된 추천을 검색하고 선택적으로 AWS Lambda 함수를 사용하여 이러한 추천을 개선하는 방법에 대한 정보가 추가되었습니다.	2020년 3월 4일
보안	Amazon Pinpoint의 다양한 보안 제어 및 기능에 대한 정보를 제공하는 보안 챕터를 추가했습니다.	2020년 2월 4일
여정	Amazon Pinpoint 여정을 사용하여 프로젝트에 대한 메시징 활동을 수행하는 자동화된 워크플로우 개발에 대한 정보를 추가했습니다. 또한 여정에 적용되는 지표의 하위 집합에 대한 분석 데이터 쿼리 정보도 추가했습니다.	2019년 10월 31일

분석	캠페인 및 트랜잭션 메시지에 대한 분석 데이터를 쿼리하는 방법을 설명하는 절차와 쿼리 결과 사용에 대한 정보가 추가되었습니다.	2019년 10월 17일
분석	트랜잭션 이메일 및 SMS 메시지에 적용되는 지표의 하위 집합에 대한 분석 데이터 쿼리 정보를 추가했습니다.	2019년 9월 6일
코드 예제	Amazon Pinpoint가 지원하는 모든 서비스를 사용하여 트랜잭션 푸시 알림을 보내는 데 사용할 수 있는 코드 예제가 추가되었습니다.	2019년 7월 30일
분석	프로젝트(애플리케이션) 및 캠페인에 적용되는 지표의 하위 집합에 대한 분석 데이터 쿼리 정보를 추가했습니다.	2019년 7월 24일
세그먼트	Salesforce 또는 Marketo 같은 외부 시스템에서 Amazon Pinpoint로 고객 데이터를 가져오는 솔루션을 설명하는 튜토리얼이 추가되었습니다.	2019년 5월 14일
지역별 가용성	이제 AWS 아시아 태평양(뭄바이) 및 아시아 태평양(시드니) 리전에서 Amazon Pinpoint를 사용할 수 있습니다.	2019년 4월 25일
Amazon Pinpoint와 함께 Postman 사용	Postman을 사용하여 Amazon Pinpoint API와 상호 작용하는 방법을 설명하는 튜토리얼이 추가되었습니다.	2019년 4월 8일

태그 지정	Amazon Pinpoint 리소스 태그 지정에 대한 정보가 추가되었습니다.	2019년 2월 27일
SMS 등록	자습서 단원과 SMS 사용자 등록을 처리하는 솔루션을 생성하는 방법을 설명하는 자습서가 추가되었습니다.	2019년 2월 27일
코드 예제	프로그래밍 방식으로 이메일, SMS, 음성 메시지를 보내는 방법을 보여주는 여러 프로그래밍 언어의 코드 예제가 추가되었습니다.	2019년 2월 6일

이전 업데이트

다음 표에서는 2018년 12월까지 Amazon Pinpoint 개발자 안내서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다.

변경	설명	Date
리전별 가용성	이제 AWS 미국 서부(오레곤) 및 유럽(프랑크푸르트) 리전에서 Amazon Pinpoint를 사용할 수 있습니다.	2018년 12월 21일
음성 채널	새로운 Amazon Pinpoint 음성 채널을 사용해 음성 메시지를 생성한 다음 휴대폰을 통해 고객에게 전달할 수 있습니다. 현재는 Amazon Pinpoint SMS 및 음성 API를 사용하여 음성 메시지만 전송할 수 있습니다.	2018년 11월 15일

변경	설명	Date
유럽(아일랜드) 가용성	이제 AWS 유럽(아일랜드) 리전에서 Amazon Pinpoint를 사용할 수 있습니다.	2018년 10월 25일
이벤트 API	Amazon Pinpoint API를 사용하여 이벤트를 기록하고 이벤트를 엔드포인트와 연결합니다.	2018년 8월 7일
엔드포인트를 정의하고 조회하기 위한 코드 예제	엔드포인트를 정의, 업데이트, 삭제 및 조회하는 방법을 보여주는 코드 예제가 추가됩니다. AWS CLI AWS SDK for Java 및 Amazon Pinpoint API에 대한 예제가 제공됩니다. 자세한 내용은 엔드포인트를 사용하여 Amazon Pinpoint에서 대상을 나타냄 단원을 참조하십시오.	2018년 8월 7일
엔드포인트 내보내기 권한	Amazon Pinpoint 엔드포인트를 Amazon S3 버킷으로 내보낼 수 있는 IAM 정책을 구성합니다.	2018년 5월 1일
SMS용 전화번호 확인	Amazon Pinpoint API를 사용하여 전화번호를 확인하고 그 번호가 SMS 메시지의 유효한 대상인지 판단합니다.	2018년 4월 23일
Amazon Pinpoint 통합에 대한 주제 업데이트	SDK 또는 라이브러리를 사용하여 Amazon Pinpoint를 Android, iOS 또는 JavaScript 애플리케이션과 통합 AWS SDKs. JavaScript	2018년 3월 23일

변경	설명	Date
AWS CloudTrail 로깅	CloudTrail을 사용하여 Amazon Pinpoint API 직접 호출 로깅에 대한 정보를 추가했습니다.	2018년 2월 6일
업데이트된 Service Quotas	할당량를 이메일 할당량에 대한 추가 정보로 업데이트했습니다.	2018년 1월 19일
Amazon Pinpoint 확장용 공개 베타 버전	AWS Lambda 함수를 사용하여 세그먼트를 사용자 지정하거나 사용자 지정 메시징 채널을 생성합니다.	2017년 11월 28일
푸시 알림 페이로드 할당량	할당량에는 모바일 푸시 메시지에 대한 페이로드 크기가 포함됩니다.	2017년 10월 25일
업데이트된 Service Quotas	할당량에 SMS 및 이메일 채널 정보가 추가되었습니다.	2017년 10월 9일
ADM 및 Baidu 모바일 푸시	Baidu 및 ADM 모바일 푸시 채널의 푸시 알림을 처리하도록 앱 코드를 업데이트합니다.	2017년 9월 27일
Amazon Cognito 사용자 풀의 사용자 ID 및 인증 이벤트.	Amazon Cognito 사용자 풀을 사용하여 모바일 앱의 사용자 로그인을 관리하는 경우 Amazon Cognito는 엔드포인트에 사용자 ID를 할당하며 Amazon Pinpoint에 인증 이벤트를 보고합니다.	2017년 9월 26일

변경	설명	Date
사용자 ID	엔드포인트에 사용자 ID를 할당하여 개별 사용자의 앱 사용을 모니터링합니다. AWS Mobile SDK 및 Java용 SDK에 대한 예제를 제시했습니다.	2017년 8월 31일
인증 이벤트	인증 이벤트를 보고하여 사용자가 앱에서 인증하는 빈도를 알아봅니다. 예제는 애플리케이션에서 Amazon Pinpoint 이벤트 보고 단원에 있습니다.	2017년 8월 31일
샘플 이벤트 업데이트	예제 이벤트에는 Amazon Pinpoint가 이메일 및 SMS 활동에 대해 스트리밍하는 이벤트가 포함됩니다.	2017년 08월 6일
Android 세션 관리	AWS Mobile Hub 샘플 앱이 제공하는 클래스를 사용하여 Android 앱에서 세션을 관리합니다.	2017년 20월 4일
업데이트된 수익 창출 이벤트 샘플	수익 창출 이벤트 보고를 위한 샘플 코드를 업데이트했습니다..	2017년 3월 31일
이벤트 스트림	앱 및 캠페인 이벤트를 Amazon Kinesis 스트림으로 전송하도록 Amazon Pinpoint를 구성할 수 있습니다.	2017년 3월 24일

변경	설명	Date
권한	계정의 AWS 사용자와 모바일 앱의 사용자에게 Amazon Pinpoint에 대한 액세스 권한을 부여하는 방법에 대한 Amazon Pinpoint에서 IAM을 사용하는 방법 자세한 내용은 섹션을 참조하세요.	2017년 1월 12일
Amazon Pinpoint 일반 가용성	이 릴리스에서 Amazon Pinpoint가 도입되었습니다.	2016년 12월 1일