Add a permission의

AWS 최종 사용자 메시징 푸시

AWS 최종 사용자 메시징 푸시: Add a permission의

Table of Contents

AWS 최종 사용자 메시징 푸시란 무엇입니까?

> **Note**
>
> Amazon Pinpoint의 푸시 알림 기능을 이제 AWS 최종 사용자 메시징이라고 합니다.

AWS 최종 사용자 메시징 푸시를 사용하면 푸시 알림 채널을 통해 푸시 알림을 전송하여 앱 사용자를 참여시킬 수 있습니다. Apple Push Notification Service(APNs), Firebase Cloud Messaging(FCM), Amazon Device Messaging(ADM) 및 Baidu Push를 지원합니다.

주제

AWS 최종 사용자 메시징 푸시를 처음 사용하십니까?

AWS 최종 사용자 메시징 푸시를 처음 사용하는 경우 먼저 다음 섹션을 읽어보는 것이 좋습니다.

- 설정 AWS 계정
- AWS 최종 사용자 메시징 푸시 시작하기
- 애플리케이션 생성 및 푸시 채널 활성화

AWS 최종 사용자 메시징 푸시의 기능

다음 푸시 알림 서비스에 대해 별도의 채널을 사용하여 앱에 푸시 알림을 보낼 수 있습니다.

- Firebase Cloud Messaging(FCM)
- Apple Push Notification service(APNs)

> **Note**
>
> APNs를 사용하여 iPhone 및 iPad 같은 iOS 디바이스뿐 아니라 Mac 랩톱 및 데스크톱 같은 macOS 디바이스의 Safari 브라우저에도 메시지를 보낼 수 있습니다.

- Baidu 클라우드 푸시
- Amazon Device Messaging(ADM)

AWS 최종 사용자 메시징 푸시 액세스

콘솔, CLI 또는 API를 통해 서비스에 액세스하는 다양한 방법을 간략하게 설명합니다.

다음 인터페이스를 사용하여 AWS 최종 사용자 메시징 푸시를 관리할 수 있습니다.

AWS 최종 사용자 메시징 푸시 콘솔

AWS 최종 사용자 메시징 푸시 리소스를 생성하고 관리하는 웹 인터페이스입니다. 에 가입한 경우 에서 AWS 최종 사용자 메시징 푸시 콘솔에 액세스할 AWS 계정수 있습니다 AWS Management Console.

AWS Command Line Interface

명령줄 셸에서 명령을 사용하여 AWS 서비스와 상호 작용합니다. AWS Command Line Interface 는 Windows, macOS 및 Linux에서 지원됩니다. 에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 AWS CLI참조하세요. AWS CLI 명령 참조에서 AWS 최종 사용자 메시징 푸시 명령을 찾을 수 있습니다.

AWS SDK

HTTP 또는 HTTPS를 통해 요청을 제출하는 대신 언어별 APIs를 사용하여 애플리케이션을 구축하는 것을 선호하는 소프트웨어 개발자인 경우는 라이브러리, 샘플 코드, 자습서 및 기타 리소스를 AWS 제공합니다. 이러한 라이브러리는 요청 암호화 서명, 요청 재시도, 오류 응답 처리와 같은 작업을 자동화하는 기본 함수를 제공합니다. 이러한 함수를 사용하면 시작하는 데 더 효율적입니다. 자세한 내용은 AWS기반의 도구를 참조하세요.

리전별 가용성

AWS 최종 사용자 메시징 푸시는 북미, 유럽, 아시아 및 오세아니아 AWS 리전 의 여러에서 사용할 수 있습니다. 각 리전에서는 여러 가용 영역을 AWS 유지합니다. 이러한 가용 영역은 물리적으로 서로 분

리되어 있지만, 지연 시간이 짧고 처리량과 중복성이 우수한 프라이빗 네트워크 연결로 통합됩니다. 이러한 가용 영역은 매우 높은 수준의 가용성과 중복성을 제공하는 동시에 지연 시간을 최소화하는 데 사용됩니다.

자세한 내용은에서 계정에서 사용할 수 있는 항목 지정을 AWS 리전참조하세요Amazon Web Services 일반 참조. AWS 리전 AWS 최종 사용자 메시징 푸시를 현재 사용할 수 있는 모든 리전과 각 리전의 엔드포인트 목록은의 Amazon Pinpoint API 및 AWS 서비스 엔드포인트에 대한 엔드포인트 및 할당량을 참조하세요Amazon Web Services 일반 참조. 각 리전에서 사용할 수 있는 가용 영역 수에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

설정 AWS 계정

AWS End User Messaging Push를 사용하여 앱에 푸시 알림을 보내려면 먼저 충분한 IAM 권한이 AWS 계정 있는를 얻어야 합니다. 이는 AWS 에코시스템의 다른 서비스에도 사용할 AWS 계정 수 있습니다.

주제

에 가입 AWS 계정

를 시작하려면이 AWS필요합니다 AWS 계정. 생성에 대한 자세한 AWS 계정내용은 AWS Account Management 참조 안내서의 시작하기 AWS 계정를 참조하세요.

AWS 최종 사용자 메시징 푸시 시작하기

앱에 푸시 알림을 보낼 수 있도록 AWS 최종 사용자 메시징 푸시를 설정하려면 먼저 AWS 최종 사용자 메시징 푸시가 앱에 메시지를 보낼 수 있도록 권한을 부여하는 자격 증명을 제공해야 합니다. 제공한 자격 증명은 사용하는 푸시 알림 시스템에 따라 다릅니다.

- Apple 푸시 알림 서비스(APN) 보안 인증 정보는 Apple 개발자 설명서의 Apple에서 암호화 키 및 키 ID 가져오기 및 Apple에서 공급자 인증서 가져오기를 참조하세요.
- Firebase 콘솔을 통해 얻을 수 있는 Firebase Cloud Messaging(FCM) 자격 증명은 Firebase Cloud Messaging을 참조하세요.
- Baidu 자격 증명은 Baidu를 참조하세요.
- Amazon Device Messaging(ADM) 보안 인증 정보는 보안 인증 획득을 참조하세요.

애플리케이션 생성 및 푸시 채널 활성화

AWS 최종 사용자 메시징 푸시를 사용하여 푸시 알림을 전송하려면 먼저 애플리케이션을 생성하고 푸시 알림 채널을 활성화해야 합니다.

컨텍스트

애플리케이션

애플리케이션은 모든 AWS 최종 사용자 메시징 푸시 설정을 위한 스토리지 컨테이너입니다. 애플리케이션은 Amazon Pinpoint 채널, 캠페인 및 여정 설정도 저장합니다.

Key(키)

AWS 최종 사용자 메시징 푸시가 APNs 인증 토큰에 암호화 방식으로 서명하는 데 사용하는 프라이빗 서명 키입니다. 이 서명 키는 Apple 개발자 계정에서 얻을 수 있습니다.

서명 키를 제공하는 경우AWS End User Messaging Push는 토큰을 사용하여 보내는 모든 푸시 알림에 대해 APNs으로 인증합니다. 이 서명 키로 APNS 프로덕션 환경 및 샌드박스 환경에 푸시 알림을 보낼 수 있습니다.

서명 키는 인증서와 달리 만료되지 않습니다. 키는 한 번만 입력하면 되고, 나중에 갱신할 필요가 없습니다. 또한 동일한 서명 키를 여러 앱에 사용할 수 있습니다. 자세한 내용은 Xcode 도움말의 인증 키로 APNs와 통신을 참조하세요.

Certificate

푸시 알림을 보낼 때 AWS End User Messaging Push가 APNs으로 인증하는 데 사용하는 TLS 인증서입니다. APNs 인증서는 프로덕션 환경과 샌드박스 환경을 둘 다 지원할 수도 있고, 샌드박스 환경만 지원하는 경우도 있습니다. 이 인증서는 Apple 개발자 계정에서 얻을 수 있습니다.

인증서는 1년 후 만료됩니다. 이 경우 새 인증서를 생성한 다음 AWS 최종 사용자 메시징 푸시에 제공하여 푸시 알림 전송을 갱신해야 합니다. 자세한 내용은 Xcode 도움말의 TLS 인증서를 사용하여 APN와 통신을 참조하십시오.

사전 조건

푸시 채널을 사용하려면 먼저 푸시 서비스에 대한 유효한 자격 증명이 필요합니다. 자격 증명 획득에 대한 자세한 내용은 섹션을 참조하세요AWS 최종 사용자 메시징 푸시 시작하기.

절차

다음 지침에 따라 애플리케이션을 생성하고 푸시 채널을 활성화합니다. 이 절차를 완료하려면 애플리케이션 이름만 입력하면 됩니다. 나중에 푸시 채널을 활성화하거나 비활성화할 수 있습니다.

1. https://console.aws.amazon.com/push-notifications/ AWS 최종 사용자 메시징 푸시 콘솔을 엽니다.
2. 애플리케이션 생성을 선택합니다.
3. 애플리케이션 이름에 애플리케이션의 이름을 입력합니다.
4. (선택 사항)이 선택적 단계에 따라 Apple 푸시 알림 서비스(APNs 활성화합니다.

   a. Apple 푸시 알림 서비스(APNs 선택합니다.

   b. 기본 인증 유형에서 다음 중 하나를 선택합니다.

      i. 키 자격 증명을 선택하는 경우 Apple 개발자 계정에서 다음 정보를 제공합니다. AWS End User Messaging Push는 인증 토큰을 구성하기 위해이 정보가 필요합니다.

         - 키 ID - 서명 키에 할당된 ID입니다.
         - 번들 식별자 - iOS 앱에 할당된 ID입니다.
         - 팀 식별자 – Apple 개발자 계정 팀에 할당된 ID입니다.
         - 인증 키 - 인증 키를 생성할 때 Apple 개발자 계정에서 다운로드하는 .p8 파일입니다.

      ii. 인증서 자격 증명을 선택한 경우 다음 정보를 제공합니다.

         - SSL 인증서 – TLS 인증서용 .p12 파일입니다.
         - 인증서 암호 - 인증서에 암호를 할당했으면 여기에 입력합니다.
         - 인증서 유형 - 사용할 인증서 유형을 선택합니다.

5. (선택 사항)이 선택적 단계에 따라 Firebase Cloud Messaging(FCM)을 활성화합니다.

   a. Firebase Cloud Messaging(FCM)에서 활성화를 선택합니다.

   b. 기본 인증 유형에서 다음 중 하나를 선택합니다.

      i. 토큰 자격 증명(권장)에서 파일 선택을 선택한 다음 서비스 JSON 파일을 선택합니다.

      ii. 키 자격 증명에 API 키에 키를 입력합니다.

6. (선택 사항)이 선택적 단계에 따라 Baidu 클라우드 푸시를 활성화합니다.

   a. Baidu Cloud Push에서 활성화를 선택합니다.

b. API 키에 API 키를 입력합니다.

c. 보안 암호 키에 보안 암호 키를 입력합니다.

7. (선택 사항)이 선택적 단계에 따라 Amazon Device Messaging을 활성화합니다.

a. Amazon Device Messaging에서 활성화를 선택합니다.

b. 클라이언트 ID에 클라이언트 ID를 입력합니다.

c. 클라이언트 보안 암호에 클라이언트 보안 암호를 입력합니다.

8. 애플리케이션 생성을 선택합니다.

푸시 채널 비활성화

다음 지침에 따라 푸시 채널을 비활성화합니다.

1. https://console.aws.amazon.com/push-notifications/ AWS 최종 사용자 메시징 푸시 콘솔을 엽니다.
2. 푸시 자격 증명이 포함된 애플리케이션을 선택합니다.
3. (선택 사항) Apple 푸시 알림 서비스(APNs 지웁니다.
4. (선택 사항) Firebase Cloud Messaging(FCM)에서 활성화를 선택 취소합니다.
5. (선택 사항) Baidu Cloud Push의 경우 활성화를 선택 취소합니다.
6. (선택 사항) Amazon Device Messaging에서 활성화를 지웁니다.
7. 변경 사항 저장(Save changes)을 선택합니다.

메시지 전송

AWS End User Messaging Push API는 트랜잭션 푸시 알림을 특정 디바이스 식별자로 전송할 수 있습니다. 이 섹션에는 AWS SDK를 사용하여 AWS 최종 사용자 메시징 푸시 API를 통해 푸시 알림을 보내는 데 사용할 수 있는 전체 코드 예제가 포함되어 있습니다.

다음 예제를 사용하여 AWS End User Messaging Push가 지원하는 모든 푸시 알림 서비스를 통해 푸시 알림을 보낼 수 있습니다. 현재 AWS End User Messaging Push는 Firebase Cloud Messaging(FCM), Apple Push Notification Service(APNs), Baidu Cloud Push 및 Amazon Device Messaging(ADM) 채널을 지원합니다.

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

> **Note**
>
> Firebase Cloud Messaging(FCM) 서비스를 통해 푸시 알림을 보낼 때는 AWS End User Messaging Push API 호출GCM에 서비스 이름을 사용합니다. Google에서는 2018년 4월 10일 Google Cloud Messaging(GCM) 서비스를 중단했습니다. 그러나 AWS End User Messaging Push API는 GCM GCM 서비스 중단 전에 작성된 API 코드와의 호환성을 유지하기 위해 FCM 서비스를 통해 전송하는 메시지의 서비스 이름을 사용합니다.

GCM (AWS CLI)

다음 예제에서는 send-messages를 사용하여 로 GCM 푸시 알림을 보냅니다 AWS CLI. *##*을 디바이스의 고유 토큰으로 바꾸고 *611e3e3cdd47474c9c1399a50example*을 애플리케이션 식별자로 바꿉니다.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request file://myfile.json \
--region us-west-2

Contents of myfile.json:
{
  "Addresses": {
    "token": {
      "ChannelType" : 'GCM'
    }
```

```
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "URL",
      "Body": "This is a sample message",
      "Priority": "normal",
      "SilentPush": True,
      "Title": "My sample message",
      "TimeToLive": 30,
      "Url": "https://www.example.com"
      }
  }
}
```

다음 예제에서는 send-messages를 사용하여와 함께 모든 레거시 키를 사용하여 GCM 푸시 알림을 보냅니다 AWS CLI. *##*을 디바이스의 고유 토큰으로 바꾸고 *611e3e3cdd47474c9c1399a50example*을 애플리케이션 식별자로 바꿉니다.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\"notification\": {\n \"title\": \"string\",\n \"body\":
 \"string\",\n \"android_channel_id\": \"string\",\n \"body_loc_args\": [\n \"string
\"\n ],\n \"body_loc_key\": \"string\",\n \"click_action\": \"string\",\n \"color\":
 \"string\",\n \"icon\": \"string\",\n \"sound\": \"string\",\n \"tag\": \"string
\",\n \"title_loc_args\": [\n \"string\"\n ],\n \"title_loc_key\": \"string\"\n },
\"data\":{\"message\":\"hello in data\"} }",
      "TimeToLive" : 309744
     }
   },
  "Addresses": {
    "token": {
      "ChannelType":"GCM"
      }
   }
}'
\ --region us-east-1
```

다음 예제에서는 send-messages를 사용하여를 사용하여 FCMv1 메시지 페이로드로 GCM 푸시 알림을 보냅니다 AWS CLI. ##을 디바이스의 고유 토큰으로 바꾸고 *611e3e3cdd47474c9c1399a50example*을 애플리케이션 식별자로 바꿉니다.

```
aws pinpoint send-messages \
--application-id 6a2dafd84bec449ea75fb773f4c41fa1 \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\n \"fcmV1Message\": \n {\n \"message\" :{\n  \"notification
\": {\n \"title\": \"string\",\n \"body\": \"string\"\n },\n \"android\": {\n
 \"priority\": \"high\",\n \"notification\": {\n \"title\": \"string\",\n \"body
\": \"string\",\n \"icon\": \"string\",\n \"color\": \"string\",\n \"sound\":
 \"string\",\n \"tag\": \"string\",\n \"click_action\": \"string\",\n \"body_loc_key
\": \"string\",\n \"body_loc_args\": [\n \"string\"\n ],\n \"title_loc_key
\": \"string\",\n \"title_loc_args\": [\n \"string\"\n ],\n \"channel_id\":
 \"string\",\n \"ticker\": \"string\",\n \"sticky\": true,\n \"event_time\":
 \"2024-02-06T22:11:55Z\",\n \"local_only\": true,\n \"notification_priority\":
 \"PRIORITY_UNSPECIFIED\",\n \"default_sound\": false,\n \"default_vibrate_timings
\": true,\n \"default_light_settings\": false,\n \"vibrate_timings\": [\n \"22s
\"\n ],\n \"visibility\": \"VISIBILITY_UNSPECIFIED\",\n \"notification_count\": 5,
\n \"light_settings\": {\n \"color\": {\n \"red\": 1,\n \"green\": 2,\n \"blue\":
 3,\n \"alpha\": 6\n },\n \"light_on_duration\": \"112s\",\n \"light_off_duration
\": \"1123s\"\n },\n \"image\": \"string\"\n },\n \"data\": {\n \"dataKey1\":
 \"priority message\",\n \"data_key_3\": \"priority message\",\n \"dataKey2\":
 \"priority message\",\n \"data_key_5\": \"priority message\"\n },\n \"ttl\":
 \"10023.32s\"\n },\n \"apns\": {\n \"payload\": {\n \"aps\": {\n \"alert\": {\n
 \"subtitle\": \"string\",\n \"title-loc-args\": [\n \"string\"\n ],\n \"title-loc-
key\": \"string\",\n \"launch-image\": \"string\",\n \"subtitle-loc-key\": \"string
\",\n \"subtitle-loc-args\": [\n \"string\"\n ],\n \"loc-args\": [\n \"string
\"\n ],\n \"loc-key\": \"string\",\n \"title\": \"string\",\n \"body\": \"string
\"\n },\n \"thread-id\": \"string\",\n \"category\": \"string\",\n \"content-
available\": 1,\n \"mutable-content\": 1,\n \"target-content-id\": \"string\",\n
 \"interruption-level\": \"string\",\n \"relevance-score\": 25,\n \"filter-criteria
\": \"string\",\n \"stale-date\": 6483,\n \"content-state\": {},\n \"timestamp\":
 673634,\n \"dismissal-date\": 4,\n \"attributes-type\": \"string\",\n \"attributes
\": {},\n \"sound\": \"string\",\n \"badge\": 5\n }\n }\n },\n \"webpush\": {\n
 \"notification\": {\n \"permission\": \"granted\",\n \"maxActions\": 2,\n \"actions
\": [\n \"title\"\n ],\n \"badge\": \"URL\",\n \"body\": \"Hello\",\n \"data\": {\n
 \"hello\": \"hey\"\n },\n \"dir\": \"auto\",\n \"icon\": \"icon\",\n \"image\":
 \"image\",\n \"lang\": \"string\",\n \"renotify\": false,\n \"requireInteraction\":
 true,\n \"silent\": false,\n \"tag\": \"tag\",\n \"timestamp\": 1707259524964,\n
```

```
 \"title\": \"hello\",\n \"vibrate\": [\n 100,\n 200,\n 300\n ]\n },\n \"data\": {\n
 \"data1\": \"priority message\",\n \"data2\": \"priority message\",\n \"data12\":
 \"priority message\",\n \"data3\": \"priority message\"\n }\n },\n \"data\": {\n
 \"data7\": \"priority message\",\n \"data5\": \"priority message\",\n \"data8\":
 \"priority message\",\n \"data9\": \"priority message\"\n }\n }\n \n}\n}",
      "TimeToLive" : 309744
    }
  },
  "Addresses": {
    token: {
      "ChannelType":"GCM"
    }
   }
}'
\ --region us-east-1
```

GCM에 `ImageUrl` 필드를 사용하는 경우 Pinpoint는 필드를 데이터 알림으로 전송하고 키는 이므로 이미지가 상자에서 렌더링되지 않을 `pinpoint.notification.imageUrl`수 있습니다. RawContent를 사용하거나 앱 통합과 같은 데이터 키 처리를 추가하세요 AWS Amplify.

Safari (AWS CLI)

AWS End User Messaging Push를 사용하여 Apple의 Safari 웹 브라우저를 사용하는 macOS 컴퓨터에 메시지를 보낼 수 있습니다. Safari 브라우저로 메시지를 보내려면 원시 메시지 콘텐츠를 지정하고 메시지 페이로드에 특정 속성을 포함해야 합니다. 원시 메시지 페이로드로 푸시 알림 템플릿을 생성하거나 Amazon Pinpoint 사용 설명서의 캠페인 메시지에 원시 메시지 콘텐츠를 직접 지정하여이 작업을 수행할 수 있습니다.

> **Note**
>
> 이 특별한 속성은 Safari 웹 브라우저를 사용하는 macOS 랩톱 및 데스크톱 컴퓨터로 전송하는 데 필요합니다. iPhone 및 iPad와 같은 iOS 디바이스로 전송할 때는 필요하지 않습니다.

Safari 웹 브라우저로 메시지를 보내려면 원시 메시지 페이로드를 지정해야 합니다. 원시 메시지 페이로드의 `aps` 객체 내에 `url-args` 배열이 포함되어야 합니다. Safari 웹 브라우저에 푸시 알림을 보내려면 `url-args` 배열이 필요합니다. 하지만 배열에 비어 있는 단일 요소가 포함되어도 괜찮습니다.

다음 예제에서는 send-messages를 사용하여를 사용하여 Safari 웹 브라우저에 알림을 보냅니다 AWS CLI. *##*을 디바이스의 고유 토큰으로 바꾸고 *611e3e3cdd47474c9c1399a50example*을 애플리케이션 식별자로 바꿉니다.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
  "Addresses": {
    "token":
    {
      "ChannelType":"APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
        "RawContent":
          "{\"aps\": {\"alert\": { \"title\": \"Title of my message\", \"body\":
 \"This is a push notification for the Safari web browser.\"},\"content-available\":
 1,\"url-args\": [\"\"]}}"
     }
  }
}'
\ --region us-east-1
```

Safari 푸시 알림에 대한 자세한 내용은 Apple 개발자 웹 사이트에서 Safari 푸시 알림 구성을 참조하세요.

APNS (AWS CLI)

다음 예제에서는 send-messages를 사용하여 로 APNS 푸시 알림을 보냅니다 AWS CLI. *##*을 디바이스의 고유 토큰으로, *611e3e3cdd47474c9c1399a50example*을 애플리케이션 식별자로, *GAME_INVITATION*을 고유 식별자로 바꿉니다.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
    "Addresses": {
     "token":
    {
      "ChannelType":"APNS"
    }
```

```
  },
  "MessageConfiguration": {
    "APNSMessage": {
      "RawContent": "{\"aps\" : {\"alert\" : {\"title\" : \"Game Request\",
\"subtitle\" : \"Five Card Draw\",\"body\" : \"Bob wants to play poker\"},\"category
\" : \"GAME_INVITATION\"},\"gameID\" : \"12345678\"}"
      }
    }
}'
\ --region us-east-1
```

JavaScript (Node.js)

이 예제를 사용하여 Node.js의 JavaScript용 AWS SDK를 사용하여 푸시 알림을 보냅니다. 이 예제에서는 Node.js의 JavaScript용 SDK를 이미 설치 및 구성했다고 가정합니다.

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

```
'use strict';

const AWS = require('aws-sdk');

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the API is available
const region = 'us-east-1';

// The title that appears at the top of the push notification.
var title = 'Test message sent from End User Messaging Push.';

// The content of the push notification.
var message = 'This is a sample message sent from End User Messaging Push by using
 the '
            + 'AWS SDK for JavaScript in Node.js';

// The application ID that you want to use when you send this
// message. Make sure that the push channel is enabled for the project that
// you choose.
var applicationId = 'ce796be37f32f178af652b26eexample';

// An object that contains the unique token of the device that you want to send
// the message to, and the push service that you want to use to send the message.
```

```
var recipient = {
  'token': 'a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0',
  'service': 'GCM'
  };

// The action that should occur when the recipient taps the message. Possible
// values are OPEN_APP (opens the app or brings it to the foreground),
// DEEP_LINK (opens the app to a specific page or interface), or URL (opens a
// specific URL in the device's web browser.)
var action = 'URL';

// This value is only required if you use the URL action. This variable contains
// the URL that opens in the recipient's web browser.
var url = 'https://www.example.com';

// The priority of the push notification. If the value is 'normal', then the
// delivery of the message is optimized for battery usage on the recipient's
// device, and could be delayed. If the value is 'high', then the notification is
// sent immediately, and might wake a sleeping device.
var priority = 'normal';

// The amount of time, in seconds, that the push notification service provider
// (such as FCM or APNS) should attempt to deliver the message before dropping
// it. Not all providers allow you specify a TTL value.
var ttl = 30;

// Boolean that specifies whether the notification is sent as a silent
// notification (a notification that doesn't display on the recipient's device).
var silent = false;

function CreateMessageRequest() {
  var token = recipient['token'];
  var service = recipient['service'];
  if (service == 'GCM') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'GCM'
        }
      },
      'MessageConfiguration': {
        'GCMMessage': {
          'Action': action,
          'Body': message,
```

```
          'Priority': priority,
          'SilentPush': silent,
          'Title': title,
          'TimeToLive': ttl,
          'Url': url
        }
      }
    };
  } else if (service == 'APNS') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'APNS'
        }
      },
      'MessageConfiguration': {
        'APNSMessage': {
          'Action': action,
          'Body': message,
          'Priority': priority,
          'SilentPush': silent,
          'Title': title,
          'TimeToLive': ttl,
          'Url': url
        }
      }
    };
  } else if (service == 'BAIDU') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'BAIDU'
        }
      },
      'MessageConfiguration': {
        'BaiduMessage': {
          'Action': action,
          'Body': message,
          'SilentPush': silent,
          'Title': title,
          'TimeToLive': ttl,
          'Url': url
        }
      }
```

```
    };
  } else if (service == 'ADM') {
    var messageRequest = {
      'Addresses': {
        [token]: {
          'ChannelType' : 'ADM'
        }
      },
      'MessageConfiguration': {
        'ADMMessage': {
          'Action': action,
          'Body': message,
          'SilentPush': silent,
          'Title': title,
          'Url': url
        }
      }
    };
  }

  return messageRequest
}

function ShowOutput(data){
  if (data["MessageResponse"]["Result"][recipient["token"]]["DeliveryStatus"]
      == "SUCCESSFUL") {
    var status = "Message sent! Response information: ";
  } else {
    var status = "The message wasn't sent. Response information: ";
  }
  console.log(status);
  console.dir(data, { depth: null });
}

function SendMessage() {
  var token = recipient['token'];
  var service = recipient['service'];
  var messageRequest = CreateMessageRequest();

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

```
  // Specify the AWS Region to use.
  AWS.config.update({ region: region });

  //Create a new Pinpoint object.
  var pinpoint = new AWS.Pinpoint();
  var params = {
    "ApplicationId": applicationId,
    "MessageRequest": messageRequest
  };

  // Try to send the message.
  pinpoint.sendMessages(params, function(err, data) {
    if (err) console.log(err);
    else     ShowOutput(data);
  });
}

SendMessage()
```

Python

AWS SDK for Python (Boto3)를 사용하여 푸시 알림을 보내려면 이 예를 사용하세요. 이 예제에서는 Python용 SDK(Boto3)를 이미 설치 및 구성했다고 가정합니다.

또한 이 예제에서는 공유 자격 증명 파일을 사용하여 기존 사용자의 액세스 키 및 보안 액세스 키를 지정한다고 가정합니다. 자세한 내용은 Python용AWS SDK(Boto3) API 참조의 보안 인증 정보를 참조하세요.

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

# The AWS Region that you want to use to send the message. For a list of
# AWS Regions where the API is available
region = "us-east-1"

# The title that appears at the top of the push notification.
title = "Test message sent from End User Messaging Push."

# The content of the push notification.
message = ("This is a sample message sent from End User Messaging Push by using the
 "
           "AWS SDK for Python (Boto3).")
```

```
# The application ID to use when you send this message.
# Make sure that the push channel is enabled for the project or application
# that you choose.
application_id = "ce796be37f32f178af652b26eexample"

# A dictionary that contains the unique token of the device that you want to send
 the
# message to, and the push service that you want to use to send the message.
recipient = {
    "token": "a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0",
    "service": "GCM"
    }

# The action that should occur when the recipient taps the message. Possible
# values are OPEN_APP (opens the app or brings it to the foreground),
# DEEP_LINK (opens the app to a specific page or interface), or URL (opens a
# specific URL in the device's web browser.)
action = "URL"

# This value is only required if you use the URL action. This variable contains
# the URL that opens in the recipient's web browser.
url = "https://www.example.com"

# The priority of the push notification. If the value is 'normal', then the
# delivery of the message is optimized for battery usage on the recipient's
# device, and could be delayed. If the value is 'high', then the notification is
# sent immediately, and might wake a sleeping device.
priority = "normal"

# The amount of time, in seconds, that the push notification service provider
# (such as FCM or APNS) should attempt to deliver the message before dropping
# it. Not all providers allow you specify a TTL value.
ttl = 30

# Boolean that specifies whether the notification is sent as a silent
# notification (a notification that doesn't display on the recipient's device).
silent = False

# Set the MessageType based on the values in the recipient variable.
def create_message_request():

    token = recipient["token"]
    service = recipient["service"]
```

```
    if service == "GCM":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'GCM'
                }
            },
            'MessageConfiguration': {
                'GCMMessage': {
                    'Action': action,
                    'Body': message,
                    'Priority' : priority,
                    'SilentPush': silent,
                    'Title': title,
                    'TimeToLive': ttl,
                    'Url': url
                }
            }
        }
    elif service == "APNS":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'APNS'
                }
            },
            'MessageConfiguration': {
                'APNSMessage': {
                    'Action': action,
                    'Body': message,
                    'Priority' : priority,
                    'SilentPush': silent,
                    'Title': title,
                    'TimeToLive': ttl,
                    'Url': url
                }
            }
        }
    elif service == "BAIDU":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'BAIDU'
```

```
                }
            },
            'MessageConfiguration': {
                'BaiduMessage': {
                    'Action': action,
                    'Body': message,
                    'SilentPush': silent,
                    'Title': title,
                    'TimeToLive': ttl,
                'Url': url
                }
            }
        }
    elif service == "ADM":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'ADM'
                }
            },
            'MessageConfiguration': {
                'ADMMessage': {
                    'Action': action,
                    'Body': message,
                    'SilentPush': silent,
                    'Title': title,
                    'Url': url
                }
            }
        }
    else:
        message_request = None

    return message_request

# Show a success or failure message, and provide the response from the API.
def show_output(response):
    if response['MessageResponse']['Result'][recipient["token"]]['DeliveryStatus']
 == "SUCCESSFUL":
        status = "Message sent! Response information:\n"
    else:
        status = "The message wasn't sent. Response information:\n"
    print(status, json.dumps(response,indent=4))
```

```
# Send the message through the appropriate channel.
def send_message():

    token = recipient["token"]
    service = recipient["service"]
    message_request = create_message_request()

    client = boto3.client('pinpoint',region_name=region)

    try:
        response = client.send_messages(
            ApplicationId=application_id,
            MessageRequest=message_request
        )
    except ClientError as e:
        print(e.response['Error']['Message'])
    else:
        show_output(response)

send_message()
```

추가 리소스

- 푸시 채널 템플릿에 대한 자세한 내용은 Amazon Pinpoint 사용 설명서의 푸시 알림 템플릿 생성을 참조하세요.

애플리케이션에서 푸시 알림 수신

다음 주제에서는 푸시 알림을 수신하도록 Swift, Android, React Native 또는 Flutter 앱을 수정하는 방법을 설명합니다.

주제

Swift 푸시 알림 설정

iOS 앱용 푸시 알림은 Apple 푸시 알림 서비스(APN)를 사용하여 전송됩니다. iOS 디바이스에 푸시 알림을 전송하려면 먼저 Apple 개발자 포털에서 앱 ID를 만들고 필요한 인증서를 생성해야 합니다. 이러한 단계를 완료하는 방법에 대한 자세한 내용은 AWS Amplify 설명서의 푸시 알림 서비스 설정에서 확인할 수 있습니다.

APN 토큰 작업

앱을 다시 설치할 때 고객의 디바이스 토큰이 재생성되도록 앱을 개발하는 것이 가장 좋습니다.

수신자가 디바이스를 iOS의 새로운 메이저 버전으로 업그레이드(예: iOS 12에서 iOS 13으로 업그레이드)한 후 앱을 다시 설치하면 앱은 새 토큰을 생성합니다. 앱이 토큰을 새로 고치지 않으면 알림을 전송하는 데 이전 토큰이 사용됩니다. 결과적으로 해당 토큰이 현재 유효하지 않기 때문에 Apple 푸시 알림 서비스(APNs)는 알림을 거부합니다. 알림 전송을 시도하면 APNs로부터 메시지 실패 알림을 받게 됩니다.

Android 푸시 알림 설정

푸시 알림은 Google Cloud Messaging(GCM)을 대체하는 Firebase Cloud Messaging(FCM)을 사용하여 전송됩니다. Android 디바이스에 푸시 알림을 보내려면 먼저 FCM 보안 인증을 얻어야 합니다. 이 자격 증명을 사용하여 Android 프로젝트를 생성하고 푸시 알림을 수신할 수 있는 샘플 앱을 실행할 수

있습니다. 이러한 단계를 완료하는 방법에 대한 자세한 내용은 AWS Amplify 설명서의 푸시 알림 섹션에서 확인할 수 있습니다.

Flutter 푸시 알림 설정

Flutter 앱의 푸시 알림은 Android용 Firebase Cloud Messaging(FCM)과 iOS용 APN을 사용하여 전송됩니다. 이러한 단계의 완료에 대한 자세한 내용은 AWS Amplify Flutter 설명서의 푸시 알림 섹션을 참조하세요.

React Native 푸시 알림 설정

푸시 알림 설정 앱의 푸시 알림은 Android용 Firebase Cloud Messaging(FCM)과 iOS용 APN을 사용하여 전송됩니다. 이러한 단계의 완료에 대한 자세한 내용은 AWS Amplify JavaScript 설명서의 푸시 알림 섹션을 참조하세요.

AWS 최종 사용자 메시징 푸시에서 애플리케이션 생성

AWS 최종 사용자 메시징 푸시에서 푸시 알림 전송을 시작하려면 애플리케이션을 생성해야 합니다. 그런 다음, 해당되는 자격 증명을 제공하여 사용할 푸시 알림 채널을 활성화해야 합니다.

AWS 최종 사용자 메시징 푸시 콘솔을 사용하여 새 애플리케이션을 생성하고 푸시 알림 채널을 설정할 수 있습니다. 자세한 내용은 애플리케이션 생성 및 푸시 채널 활성화 단원을 참조하십시오.

API, AWS SDK 또는 AWS Command Line Interface ()를 사용하여 애플리케이션을 생성하고 설정할 수도 있습니다AWS CLI. 애플리케이션을 생성하려면 `Apps` 리소스를 사용합니다. 푸시 알림 채널을 구성하려면 아래 리소스를 사용합니다.

- APN 채널은 Apple 푸시 알림 서비스를 통해 iOS 디바이스 사용자에게 메시지를 전송하는 데 사용됩니다.
- ADM 채널은 Amazon Kindle Fire 디바이스 사용자에게 메시지를 전송하는 데 사용됩니다.
- Baidu 채널은 Baidu 사용자에게 메시지를 전송하는 데 사용됩니다.
- GCM 채널은 Google Cloud Messaging(GCM)을 대체하는 Firebase Cloud Messaging(FCM)을 사용하여 Android 디바이스에 메시지를 전송하는 데 사용됩니다.

푸시 알림 처리

푸시 알림을 보내는 데 필요한 자격 증명을 얻은 후 푸시 알림을 수신할 수 있도록 애플리케이션을 업데이트할 수 있습니다. 자세한 내용은 AWS Amplify 설명서의 푸시 알림 - 시작하기를 참조하세요.

애플리케이션 삭제

이 절차에서는 계정에서 애플리케이션과 애플리케이션의 모든 리소스를 제거합니다.

컨텍스트

애플리케이션

애플리케이션은 모든 AWS 최종 사용자 메시징 푸시 설정을 위한 스토리지 컨테이너입니다. 애플리케이션은 Amazon Pinpoint 채널, 캠페인 및 여정 설정도 저장합니다.

절차

1. https://console.aws.amazon.com/push-notifications/ AWS 최종 사용자 메시징 푸시 콘솔을 엽니다.
2. 애플리케이션을 선택한 다음 삭제를 선택합니다.
3. 애플리케이션 삭제 창에서를 **`delete`** 입력한 다음 삭제를 선택합니다.

> **Important**
>
> 모든 Amazon Pinpoint 채널, 캠페인, 여정 또는 세그먼트도 삭제됩니다.

모범 사례

고객의 이익을 가장 먼저 생각한다고 해도 메시지 발송률에 영향을 미치는 상황은 언제든지 발생할 수 있습니다. 다음 섹션에서는 푸시 커뮤니케이션이 목표 고객에게 도달하도록 하는 데 도움이 되는 권장 사항에 대해서 살펴봅니다.

대량의 푸시 알림 보내기

대량의 푸시 알림을 보내기 전에 처리량 요구 사항을 지원하도록 계정이 구성되어 있는지 확인합니다. 기본적으로 모든 계정은 초당 25,000개의 메시지를 보내도록 구성됩니다. 1초에 25,000개 이상의 메시지를 보내야 하는 경우, 할당량 증가를 요청할 수 있습니다. 자세한 내용은 AWS 최종 사용자 메시징 푸시 할당량 단원을 참조하십시오.

계정이 FCM 또는 APNs과 같이 사용하려는 각 푸시 알림 공급자의 자격 증명으로 올바르게 구성되었는지 확인합니다.

마지막으로, 예외를 처리하는 방법을 고안하세요. 푸시 알림 서비스마다 제공하는 예외 메시지가 다릅니다. 트랜잭션 전송의 경우 API 호출은 기본 상태 코드 200이 수신되며, 메시지 전송 중에 해당 플랫폼 토큰(예: FCM)이나 인증서(예: APNs)가 유효하지 않은 것으로 확인되면 엔드포인트당 상태 코드 400 영구 실패가 수신됩니다.

AWS 최종 사용자 메시징 푸시의 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 는에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. AWS 최종 사용자 메시징 푸시에 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 AWS 최종 사용자 메시징 푸시를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 AWS 최종 사용자 메시징 푸시를 구성하는 방법을 보여줍니다. 또한 AWS 최종 사용자 메시징 푸시 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- AWS 최종 사용자 메시징 푸시의 데이터 보호
- AWS 최종 사용자 메시징 푸시를 위한 자격 증명 및 액세스 관리
- AWS 최종 사용자 메시징 푸시에 대한 규정 준수 검증
- AWS 최종 사용자 메시징 푸시의 복원력
- AWS 최종 사용자 메시징 푸시의 인프라 보안
- 구성 및 취약성 분석
- 보안 모범 사례

AWS 최종 사용자 메시징 푸시의 데이터 보호

AWS 공동 책임 모델 AWS 최종 사용자 메시징 푸시의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이 인

프라에 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.
- 내의 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 AWS 최종 사용자 메시징 푸시 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 증명을 URL에 포함해서는 안 됩니다.

데이터 암호화

AWS 최종 사용자 메시징 푸시 데이터는 전송 중 및 저장 시 암호화됩니다. AWS 최종 사용자 메시징 푸시에 데이터를 제출하면 데이터를 수신하고 저장할 때 데이터가 암호화됩니다. AWS 최종 사용자 메시징 푸시에서 데이터를 검색하면 현재 보안 프로토콜을 사용하여 데이터를 전송합니다.

저장 시 암호화

AWS End User Messaging Push는 저장된 모든 데이터를 암호화합니다. 여기에는 구성 데이터, 사용자 및 엔드포인트 데이터, 분석 데이터, AWS 최종 사용자 메시징 푸시로 추가하거나 가져오는 모든 데이터가 포함됩니다. 데이터를 암호화하기 위해 AWS End User Messaging Push는 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

AWS 최종 사용자 메시징 푸시는 HTTPS 및 TLS(전송 계층 보안) 1.2 이상을 사용하여 클라이언트 및 애플리케이션과 통신합니다. 다른 AWS 서비스와 통신하기 위해 AWS End User Messaging Push는 HTTPS 및 TLS 1.2를 사용합니다. 또한 콘솔, AWS SDK 또는를 사용하여 AWS 최종 사용자 메시징 푸시 리소스를 생성하고 관리할 때 AWS Command Line Interface모든 통신은 HTTPS 및 TLS 1.2를 사용하여 보호됩니다.

키 관리

AWS End User Messaging Push 데이터를 암호화하기 위해 AWS End User Messaging Push는 서비스가 사용자를 대신하여 소유하고 유지 관리하는 내부 AWS KMS 키를 사용합니다. 이들 키는 정기적으로 교체됩니다. 자체 키 AWS KMS 또는 기타 키를 프로비저닝하고 사용하여 AWS 최종 사용자 메시징 푸시에 저장하는 데이터를 암호화할 수 없습니다.

인터네트워크 트래픽 개인 정보 보호

Internetwork 트래픽 개인 정보 보호는 AWS End User Messaging Push와 온프레미스 클라이언트 및 애플리케이션 간의 연결 및 트래픽과 AWS End User Messaging Push와 동일한 AWS 리전의 기타 AWS 리소스 간의 연결을 보호하는 것을 말합니다. 다음 기능 및 관행은 AWS 최종 사용자 메시징 푸시에 대한 인터넷 작업 트래픽 개인 정보 보호를 보장하는 데 도움이 될 수 있습니다.

AWS 최종 사용자 메시징 푸시와 온프레미스 클라이언트 및 애플리케이션 간의 트래픽

End AWS User Messaging Push와 온프레미스 네트워크의 클라이언트 및 애플리케이션 간에 프라이빗 연결을 설정하려면를 사용할 수 있습니다 Direct Connect. 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 APIs를 통해 AWS 최종 사용자 메시징 푸시에 안전하게 액세스하려면 API 호출에 대한 AWS 최종 사용자 메시징 푸시 요구 사항을 준수하는 것이 좋습니다. AWS 최종 사용자 메시징 푸시를 사용하려면 클라이언트가 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.

AWS 최종 사용자 메시징 푸시와 기타 AWS 리소스 간의 트래픽

AWS End User Messaging Push와 동일한 AWS 리전의 다른 AWS 리소스 간의 통신을 보호하기 위해 AWS End User Messaging Push는 기본적으로 HTTPS 및 TLS 1.2를 사용합니다.

AWS 최종 사용자 메시징 푸시를 위한 자격 증명 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 AWS 최종 사용자 메시징 푸시 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청(참조AWS 최종 사용자 메시징 푸시 자격 증명 및 액세스 문제 해결)

- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출(AWS 최종 사용자 메시징 푸시가 IAM과 작동하는 방식 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성(AWS 최종 사용자 메시징 푸시에 대한 자격 증명 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 인증되어야 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 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자가 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?를 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 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 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.

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.

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.

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.

AWS 최종 사용자 메시징 푸시가 IAM과 작동하는 방식

IAM을 사용하여 AWS 최종 사용자 메시징 푸시에 대한 액세스를 관리하기 전에 AWS 최종 사용자 메시징 푸시와 함께 사용할 수 있는 IAM 기능을 알아봅니다.

AWS 최종 사용자 메시징 푸시와 함께 사용할 수 있는 IAM 기능

IAM 특성	AWS 최종 사용자 메시징 푸시 지원
자격 증명 기반 정책	예
리소스 기반 정책	예
정책 작업	예
정책 리소스	예
정책 조건 키	예
ACL	아니요
ABAC(정책 내 태그)	부분적
임시 자격 증명	예
엔터티 권한	예
서비스 역할	예
서비스 연결 역할	아니요

AWS 최종 사용자 메시징 푸시 및 기타 AWS 서비스에서 대부분의 IAM 기능을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

AWS 최종 사용자 메시징 푸시에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

AWS 최종 사용자 메시징 푸시에 대한 자격 증명 기반 정책 예제

AWS End User Messaging Push 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS 최종 사용자 메시징 푸시에 대한 자격 증명 기반 정책 예제.

AWS 최종 사용자 메시징 푸시 내의 리소스 기반 정책

리소스 기반 정책 지원: 예

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는가 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM에서 교차 계정 리소스 액세스를 참조하세요.

AWS 최종 사용자 메시징 푸시에 대한 정책 작업

정책 작업 지원: 예

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.

AWS 최종 사용자 메시징 푸시 작업 목록을 보려면 서비스 승인 참조의 AWS 최종 사용자 메시징 푸시에서 정의한 작업을 참조하세요.

AWS End User Messaging Push의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
mobiletargeting
```

단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

```
"Action": [
```

```
      "mobiletargeting:action1",
      "mobiletargeting:action2"
         ]
```

AWS End User Messaging Push 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS 최종 사용자 메시징 푸시에 대한 자격 증명 기반 정책 예제.

AWS 최종 사용자 메시징 푸시에 대한 정책 리소스

정책 리소스 지원: 예

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": "*"
```

AWS End User Messaging Push 리소스 유형 및 해당 ARNs 목록을 보려면 서비스 권한 부여 참조의 AWS End User Messaging Push에서 정의한 리소스를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 AWS End User Messaging Push에서 정의한 작업을 참조하세요.

AWS End User Messaging Push 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS 최종 사용자 메시징 푸시에 대한 자격 증명 기반 정책 예제.

AWS End User Messaging Push에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

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.

AWS End User Messaging 푸시 조건 키 목록을 보려면 서비스 승인 참조의 AWS End User Messaging 푸시에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 AWS 최종 사용자 메시징 푸시에서 정의한 작업을 참조하세요.

AWS End User Messaging Push 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS 최종 사용자 메시징 푸시에 대한 자격 증명 기반 정책 예제.

AWS 최종 사용자 메시징 푸시의 ACLs

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(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.

AWS 최종 사용자 메시징 푸시가 포함된 ABAC

ABAC 지원(정책의 태그): 부분적

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

AWS 최종 사용자 메시징 푸시와 함께 임시 자격 증명 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션을 사용하거나 역할을 전환할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 임시 보안 자격 증명 및 IAM으로 작업하는AWS 서비스 섹션을 참조하세요.

AWS 최종 사용자 메시징 푸시에 대한 교차 서비스 보안 주체 권한

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

AWS 최종 사용자 메시징 푸시에 대한 서비스 역할

서비스 역할 지원: 예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스 AWS에 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 AWS 최종 사용자 메시징 푸시 기능이 중단될 수 있습니다. AWS End User Messaging Push가 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

AWS 최종 사용자 메시징 푸시에 대한 서비스 연결 역할

서비스 연결 역할 지원: 아니요

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 태스크를 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스를 참조하세요. 서비스 연결 역할 열에서 `Yes`가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

AWS 최종 사용자 메시징 푸시에 대한 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할에는 AWS End User Messaging Push 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARNs 형식을 포함하여 AWS End User Messaging Push에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 AWS End User Messaging Push에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- AWS 최종 사용자 메시징 푸시 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS End User Messaging Push 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.

AWS 최종 사용자 메시징 푸시 콘솔 사용

AWS End User Messaging Push 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은에서 AWS End User Messaging Push 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 AWS End User Messaging Push 콘솔을 계속 사용할 수 있도록 하려면 `AWSEndUserMessaging` AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AWSEndUserMessaging",
   "Effect": "Allow",
   "Action": [
    "mobiletargeting:CreateApp",
                "mobiletargeting:GetApp",
                "mobiletargeting:GetApps",
                            "mobiletargeting:DeleteApp",
                            "mobiletargeting:GetChannels",
                            "mobiletargeting:GetApnsChannel",
                            "mobiletargeting:GetApnsVoipChannel",
                            "mobiletargeting:GetApnsVoipSandboxChannel",
                            "mobiletargeting:GetApnsSandboxChannel",
```

```
                "mobiletargeting:GetAdmChannel",
                "mobiletargeting:GetBaiduChannel",
                "mobiletargeting:GetGcmChannel",
                "mobiletargeting:UpdateApnsChannel",
                "mobiletargeting:UpdateApnsVoipChannel",
                "mobiletargeting:UpdateApnsVoipSandboxChannel",
                "mobiletargeting:UpdateBaiduChannel",
                "mobiletargeting:UpdateGcmChannel",
                "mobiletargeting:UpdateAdmChannel"
            ],
   "Resource": [
    "*"
   ]
  }
 ]
}
```

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 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": "*"
        }
    ]
}
```

AWS 최종 사용자 메시징 푸시 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 AWS 최종 사용자 메시징 푸시 및 IAM 작업 시 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- AWS 최종 사용자 메시징 푸시에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 AWS 최종 사용자 메시징 푸시 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

AWS 최종 사용자 메시징 푸시에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 mobiletargeting:*GetWidget* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mobiletargeting:GetWidget on resource: my-example-widget
```

이 경우, mobiletargeting:*GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 권한이 없다는 오류가 수신되면 AWS 최종 사용자 메시징 푸시에 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 라는 IAM 사용자가 콘솔을 사용하여 AWS End User Messaging Push에서 작업을 수행하려고 `marymajor` 할 때 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 외부의 사람이 내 AWS 최종 사용자 메시징 푸시 리소스 AWS 계정 에 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.

- AWS 최종 사용자 메시징 푸시가 이러한 기능을 지원하는지 여부를 알아보려면 섹션을 참조하세요AWS 최종 사용자 메시징 푸시가 IAM과 작동하는 방식.
- 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 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 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.

AWS 최종 사용자 메시징 푸시에 대한 규정 준수 검증

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 제공 범위 내를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서를 AWS 서비스참조하세요.

AWS 최종 사용자 메시징 푸시의 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 지연 시간이 짧고 처리량이 많으며 중복성이 높은 네트워킹과 연결된 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공합니다. 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.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

AWS 글로벌 인프라 외에도 AWS End User Messaging Push는 데이터 복원력 및 백업 요구 사항을 지원하는 몇 가지 기능을 제공합니다.

AWS 최종 사용자 메시징 푸시의 인프라 보안

관리형 서비스인 AWS End User Messaging Push는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 AWS 최종 사용자 메시징 푸시에 액세스합니다. 클라이언트가 전송 계층 보안(TLS) 1.2 이상을 지원해야 합니다. 클라이언트는 DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 위탁자와 관련된 시크릿 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

구성 및 취약성 분석

관리형 서비스인 AWS End User Messaging Push는 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 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.

보안 모범 사례

AWS Identity and Access Management(IAM) 계정을 사용하여 API 작업, 특히 리소스를 생성, 수정 또는 삭제하는 작업에 대한 액세스를 제어합니다. API의 경우 이러한 리소스에는 프로젝트, 캠페인, 여정이 포함됩니다.

- 본인을 포함하여 리소스를 관리하는 각 개인에 대해 개별 사용자를 생성합니다. AWS 루트 자격 증명을 사용하여 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.

보안에 대한 자세한 내용은 AWS 최종 사용자 메시징 푸시의 보안을 참조하세요. IAM에 대한 자세한 내용은 AWS ID 및 액세스 관리 섹션을 참조하세요. IAM 모범 사례에 대한 자세한 내용은 IAM 모범 사례 단원을 참조하십시오.

AWS 최종 사용자 메시징 푸시 모니터링

모니터링은 AWS 최종 사용자 메시징 푸시 및 기타 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. AWS는 AWS 최종 사용자 메시징 푸시를 관찰하고, 문제가 있을 때 보고하고, 적절한 경우 자동 조치를 취할 수 있는 다음과 같은 모니터링 도구를 제공합니다.

- Amazon CloudWatch는 AWS 리소스와 실행 중인 애플리케이션을 AWS 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임곗값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 CloudWatch에서 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 추적하고 필요할 때 자동으로 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 CloudWatch 사용 설명서를 참조하세요.
- Amazon CloudWatch Logs로 Amazon EC2 인스턴스, CloudTrail, 기타 소스의 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다. CloudWatch Logs는 로그 파일의 정보를 모니터링하고 특정 임곗값에 도달하면 사용자에게 알릴 수 있습니다. 또한 매우 내구력 있는 스토리지에 로그 데이터를 저장할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서를 참조하세요.
- Amazon EventBridge를 사용하여 AWS 서비스를 자동화하고 애플리케이션 가용성 문제 또는 리소스 변경과 같은 시스템 이벤트에 자동으로 대응할 수 있습니다. AWS 서비스의 이벤트는 거의 실시간으로 EventBridge로 전달됩니다. 원하는 이벤트만 표시하도록 간단한 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.
- AWS CloudTrail는 AWS 계정에서 또는 계정을 대신하여 수행된 API 호출 및 관련 이벤트를 캡처하고 사용자가 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 호출된 사용자 및 계정 AWS, 호출이 수행된 원본 IP 주소, 호출이 발생한 시기를 식별할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

Amazon CloudWatch를 사용하여 AWS 최종 사용자 메시징 푸시 모니터링

원시 데이터를 수집하고 읽기 가능하며 실시간에 가까운 지표로 처리하는 CloudWatch를 사용하여 AWS 최종 사용자 메시징 푸시를 모니터링할 수 있습니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임곗값을 주시하다가 해당 임곗값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.

지표 및 차원 목록은 Amazon Pinpoint 사용 설명서의 CloudWatch를 사용하여 Amazon Pinpoint 모니터링을 참조하세요. Amazon Pinpoint

를 사용하여 AWS 최종 사용자 메시징 푸시 API 호출 로깅 AWS CloudTrail

AWS 최종 사용자 메시징 푸시는 AWS 최종 사용자 메시징 푸시에서 사용자, 역할 또는 AWS CloudTrail서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스인와 통합됩니다. CloudTrail은 AWS 최종 사용자 메시징 푸시에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 AWS 최종 사용자 메시징 푸시 콘솔의 호출과 AWS 최종 사용자 메시징 푸시 API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 AWS 최종 사용자 메시징 푸시 이벤트를 포함하여 CloudTrail 이벤트를 Amazon S3 버킷으로 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS 최종 사용자 메시징 푸시에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

AWS CloudTrail의 최종 사용자 메시징 푸시 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. AWS 최종 사용자 메시징 푸시에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

AWS 최종 사용자 메시징 푸시에 대한 이벤트를 AWS 계정포함하여에서 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 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 AWS 최종 사용자 메시징 푸시 작업은 CloudTrail에서 로깅되며 AWS 최종 사용자 메시징 푸시 API 참조에 문서화됩니다. 예를 들어 `GetAdmChannel`, `UpdateApnsChannel`, `GetApnsVoipChannel` 작업을 직접 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에 의해 이루어졌는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

AWS 최종 사용자 메시징 푸시 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 간접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

인터페이스 엔드포인트를 사용하여 AWS 최종 사용자 메시징 푸시 액세스(AWS PrivateLink)

AWS PrivateLink 를 사용하여 VPC와 AWS 최종 사용자 메시징 푸시 간에 프라이빗 연결을 생성할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 AWS 최종 사용자 메시징 푸시에 액세스할 수 있습니다. VPC의 인스턴스는 AWS 최종 사용자 메시징 푸시에 액세스하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

AWS PrivateLink에서 제공되는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 AWS 최종 사용자 메시징 푸시로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은 AWS PrivateLink 가이드의를 AWS 서비스 통한 액세스를 AWS PrivateLink 참조하세요.

AWS 최종 사용자 메시징 푸시 고려 사항

AWS 최종 사용자 메시징 푸시에 대한 인터페이스 엔드포인트를 설정하기 전에 AWS PrivateLink 안내서의 고려 사항을 검토하세요.

AWS 최종 사용자 메시징 푸시는 인터페이스 엔드포인트를 통해 모든 API 작업에 대한 호출을 지원합니다.

VPC 엔드포인트 정책은 AWS 최종 사용자 메시징 푸시에 지원되지 않습니다. 기본적으로 인터페이스 엔드포인트를 통해 AWS 최종 사용자 메시징 푸시에 대한 전체 액세스가 허용됩니다. 또는 보안 그룹을 엔드포인트 네트워크 인터페이스와 연결하여 인터페이스 엔드포인트를 통해 AWS 최종 사용자 메시징 푸시에 대한 트래픽을 제어할 수 있습니다.

AWS 최종 사용자 메시징 푸시를 위한 인터페이스 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS Command Line Interface ()를 사용하여 AWS 최종 사용자 메시징 푸시에 대한 인터페이스 엔드포인트를 생성할 수 있습니다AWS CLI. 자세한 내용은 AWS PrivateLink 설명서의 인터페이스 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 AWS 최종 사용자 메시징 푸시에 대한 인터페이스 엔드포인트를 생성합니다.

```
com.amazonaws.region.pinpoint
```

인터페이스 엔드포인트에 프라이빗 DNS를 활성화하면 기본 리전 DNS 이름을 사용하여 AWS 최종 사용자 메시징 푸시에 API 요청을 할 수 있습니다. 예: `com.amazonaws.us-east-1.pinpoint`.

엔드포인트의 엔드포인트 정책 생성

엔드포인트 정책은 인터페이스 인터페이스 엔드포인트에 연결할 수 있는 IAM 리소스입니다. 기본 엔드포인트 정책은 인터페이스 엔드포인트를 통해 AWS 최종 사용자 메시징 푸시에 대한 전체 액세스를 허용합니다. VPC에서 AWS 최종 사용자 메시징 푸시에 허용되는 액세스를 제어하려면 인터페이스 엔드포인트에 사용자 지정 엔드포인트 정책을 연결합니다.

엔드포인트 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 보안 주체 (AWS 계정, IAM 사용자, IAM 역할)
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

예: AWS 최종 사용자 메시징 푸시 작업에 대한 VPC 엔드포인트 정책

다음은 사용자 지정 엔드포인트 정책의 예입니다. 이 정책을 인터페이스 엔드포인트에 연결하면 모든 리소스의 모든 보안 주체에 대해 나열된 AWS 최종 사용자 메시징 푸시 작업에 대한 액세스 권한을 부여합니다.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
             "mobiletargeting:CreateApp",
             "mobiletargeting:DeleteApp"
         ],
         "Resource":"*"
      }
   ]
```

```
}
```

AWS 최종 사용자 메시징 푸시 할당량

AWS 계정 에는 각 AWS 서비스에 대해 이전에 제한이라고 하는 기본 할당량이 있습니다. 다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다. 일부 할당량에 대한 증가를 요청할 수 있으며 다른 할당량은 늘릴 수 없습니다.

AWS 최종 사용자 메시징 푸시의 할당량을 보려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스를 선택하고 Amazon Pinpoint를 선택합니다.

AWS 계정에는 AWS 최종 사용자 메시징 푸시와 관련된 다음과 같은 할당량이 있습니다.

리소스	기본 할당량	증가 가능 여부
단일 캠페인에서 초당 전송 가능한 최대 푸시 알림 수	초당 25,000개	예, Service Quotas 콘솔을 사용합니다.
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

AWS 최종 사용자 메시징 푸시 사용 설명서의 문서 기록

다음 표에서는 AWS 최종 사용자 메시징 푸시에 대한 설명서 릴리스를 설명합니다.

변경 사항	설명	날짜
최초 릴리스	AWS 최종 사용자 메시징 푸시 사용 설명서의 최초 릴리스	2024년 7월 24일

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