aws

用户指南

AWS 最终用户消息推送

AWS 最终用户消息推送: 用户指南

Table of Contents

什么是 AWS 最终用户消息推送？

> **Note**
>
> Amazon Pinpoint 的推送通知功能现在被称为 AWS 最终用户消息。

借助 AWS 最终用户消息推送，您可以通过推送通知渠道发送推送通知，从而吸引应用程序的用户。我们支持 Apple 推送通知服务 (APNs)、Firebase 云消息 (FCM)、亚马逊设备消息 (ADM) 和百度推送。

主题

您是首次使用 AWS 最终用户消息推送的用户吗？

如果您是首次使用 AWS 最终用户消息推送的用户，我们建议您先阅读以下章节：

- 设置一个 AWS 账户
- AWS 最终用户消息推送入门
- 创建应用程序并启用推送渠道

AWS 最终用户消息推送的功能

您可以对以下推送通知服务使用单独的渠道，以将推送通知发送到您的应用程序：

- Firebase 云端消息传递 () FCM
- 苹果推送通知服务 (APNs)

> **Note**
>
> 你可以使用APNs向 iOS 设备（如 iPhones 和iPads）以及 macOS 设备（例如 Mac 笔记本电脑和台式机）上的 Safari 浏览器发送消息。

- 百度云推送
- 亚马逊设备消息 (ADM)

访问 AWS 最终用户消息推送

简要说明获取服务访问权限的不同方式，无论是通过控制台CLI、还是API。

您可以使用以下界面管理 AWS 最终用户消息推送：

AWS 最终用户消息推送控制台

用于创建和管理 AWS 最终用户消息推送资源的 Web 界面。如果您已注册 AWS 账户，则可以从访问 AWS 最终用户消息推送控制台 AWS Management Console。

AWS Command Line Interface

使用命令行 shell 中的命令与 AWS 服务进行交互。在 AWS Command Line Interface Windows、macOS 和 Linux 上都支持。有关更多信息 AWS CLI，请参阅《AWS Command Line Interface 用户指南》。您可以在《命令参考》中找到 AWS 最终用户消息推送AWS CLI 命令。

AWS SDKs

如果你是一名软件开发人员，更喜欢使用特定语言来构建应用程序，APIs而不是通过HTTP或提交请求，请 AWS 提供库HTTPS、示例代码、教程和其他资源。这些库提供了自动执行任务的基本功能，例如对请求进行加密签名、重试请求和处理错误响应。这些功能有助于提高您的入门效率。有关更多信息，请参阅用于在 AWS上进行构建的工具。

区域可用性

AWS 最终用户消息推送已 AWS 区域 在北美、欧洲、亚洲和大洋洲的多个地区推出。在每个区域中，AWS 维护多个可用区。这些可用区的物理位置是相互隔离的，但可通过私有、低延迟、高吞吐量和高度冗余的网络连接联合在一起。这些可用区域用于提供非常高的可用性和冗余性，同时还可以最大限度地减少延迟。

要了解更多信息 AWS 区域，请参阅中的指定 AWS 区域 您的账户可以使用的内容Amazon Web Services 一般参考。有关目前提供 AWS 最终用户消息推送的所有区域以及每个区域的终端节点的列表，请参阅中的 Amazon Pinpoint 和服务终端节点的终端节点API和配额以及AWS 服务终端节点。Amazon Web Services 一般参考要详细了解每个区域中可用的可用区数量，请参阅 AWS 全球基础设施。

设置一个 AWS 账户

在使用 AWS 最终用户消息推送向您的应用程序发送推送通知之前，您必须先获得 AWS 账户 具有足够IAM权限的。这 AWS 账户 也可以用于 AWS 生态系统中的其他服务。

主题

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

要注册 AWS 账户

1. 打开https://portal.aws.amazon.com/billing/注册。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

   当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 https://aws.amazon.com/ 并选择我的账户来查看当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。AWS Management Console在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的以根用户身份登录。

2. 为您的 root 用户开启多重身份验证 (MFA)。

   有关说明，请参阅《用户指南》中的 “为 AWS 账户 root 用户（控制台）启用虚拟MFA设备” IAM。

创建具有管理访问权限的用户

1. 启用IAM身份中心。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的启用 AWS IAM Identity Center。

2. 在 IAM Identity Center 中，向用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《用户指南》 IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限。

以具有管理访问权限的用户身份登录

- 要使用您的 Ident IAM ity Center 用户登录URL，请使用您在创建 Ident IAM ity Center 用户时发送到您的电子邮件地址的登录信息。

  有关使用 Ident IAM ity Center 用户登录的帮助，请参阅AWS 登录 用户指南中的登录 AWS 访问门户。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个遵循应用最低权限权限的最佳实践的权限集。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的创建权限集。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的添加组。

AWS 最终用户消息推送入门

要设置 AWS 最终用户消息推送使其能够向您的应用程序发送推送通知，您首先必须提供授权 AWS 最终用户消息推送向您的应用程序发送消息的凭据。您提供的凭证取决于您使用的推送通知系统：

- 有关 Apple 推送通知服务 (APN) 凭据，请参阅 Apple 开发者文档中的 “从 Apple 获取加密密钥和密钥 ID” 和 “从 Apple 获取提供商证书”。
- 有关可通过 Firebase 控制台获取的 Firebase 云消息 (FCM) 凭据，请参阅 Firebase 云消息传递。
- 有关百度凭证，请参阅百度。
- 有关 Amazon 设备消息 (ADM) 凭证，请参阅获取凭证。

创建应用程序并启用推送渠道

在使用 AWS 最终用户消息推送发送推送通知之前，您必须先创建一个应用程序并启用推送通知频道。

情境相关

应用程序

应用程序是所有 AWS 最终用户消息推送设置的存储容器。该应用程序还存储您的亚马逊 Pinpoint 渠道、活动和旅程设置。

密钥

AWS 最终用户消息推送使用的私有签名密钥，用于对APNs身份验证令牌进行加密签名。您可以从您的 Apple 开发人员账户中获取该签名密钥。

如果您提供签名密钥，则AWS 最终用户消息推送将使用令牌对您发送APNs的每个推送通知进行身份验证。使用您的签名密钥，您可以向APNs生产环境和沙盒环境发送推送通知。

与证书不同，签名密钥不会过期。您只需提供一次密钥即可，而且以后无需续订。您可以将同样的签名密钥用于多个应用程序。有关更多信息，请参阅 Xcode 帮助中的APNs使用身份验证令牌与通信。

证书

当您发送推送通知APNs时，“ AWS 最终用户消息推送” 用于进行身份验证的TLS证书。APNs证书可以同时支持生产环境和沙盒环境，也可以仅支持沙盒环境。您可以从 Apple 开发人员账户获取该证书。

证书在一年后过期。发生这种情况时，您必须创建一个新证书，然后将其提供给 AWS 最终用户消息推送以续订推送通知的交付。有关更多信息，请参阅 Xcode 帮助中的APNs使用TLS证书进行通信。

先决条件

在使用任何推送渠道之前，您需要有效的推送服务凭证。有关获取证书的更多信息，请参阅AWS 最终用户消息推送入门。

过程

按照以下说明创建应用程序并启用任何推送渠道。要完成此过程，您只需要输入应用程序名称即可。您可以稍后启用或禁用任何推送渠道。

1. 打开 AWS 最终用户消息推送控制台，网址为https://console.aws.amazon.com/push-notifications/。
2. 选择创建应用程序。
3. 在应用程序名称中输入应用程序的名称。
4. （可选）按照此可选步骤启用 Apple 推送通知服务（APNs）。
   a. 对于 Apple 推送通知服务 (APNs)，请选择 “启用”。
   b. 对于默认身份验证类型，请选择以下任一选项：
      i. 如果您选择密钥凭证，请提供您的 Apple 开发者帐户中的以下信息。 AWS 最终用户消息推送需要此信息来构造身份验证令牌。
         - 密钥 ID – 分配给您的签名密钥的 ID。
         - 捆绑包标识符 – 分配给您的 iOS 应用程序的 ID。
         - 团队标识符 – 分配给您的 Apple 开发人员账户团队的 ID。
         - 身份验证密钥 – 当您创建身份验证密钥时从您的 Apple 开发人员账户下载的 .p8 文件。
      ii. 如果您选择证书凭证，请提供以下信息：
         - SSL证书-您的TLS证书的.p12 文件。
         - 证书密码 – 如果您向证书分配了密码，请在此处输入。
         - 证书类型 - 选择要使用的证书类型。
5. （可选）按照此可选步骤启用 Firebase 云消息传递 (FCM)。
   a. 对于 Firebase 云消息传递 (FCM)，请选择启用。
   b. 对于默认身份验证类型，请选择以下任一选项：
      i. 对于令牌凭证（推荐），选择 “选择文件”，然后选择您的服务JSON文件。
      ii. 对于密钥凭证，请在密钥中输入您的API密钥。
6. （可选）按照此可选步骤启用百度云推送。

a. 对于百度云推送，请选择启用。

b. 对于API密钥，请输入您的API密钥。

c. 对于密钥，请输入您的密钥。

7. （可选）按照此可选步骤启用 Amazon 设备消息。

a. 对于 Amazon 设备消息，请选择 “启用”。

b. 对于客户端 ID，请输入您的客户端 ID。

c. 对于客户密钥，请输入您的客户机密钥。

8. 选择创建应用程序。

禁用推送频道

按照以下说明禁用任何推送渠道。

1. 打开 AWS 最终用户消息推送控制台，网址为https://console.aws.amazon.com/push-notifications/。
2. 选择包含您的推送凭证的应用程序。
3. （可选）对于 Apple 推送通知服务 (APNs)，清除 “启用”。
4. （可选）对于 Firebase 云消息传递 (FCM)，请清除 “启用”。
5. （可选）对于百度云推送，请清除 “启用”。
6. （可选）对于 Amazon 设备消息，清除 “启用”。
7. 选择保存更改。

发送消息

AWS 最终用户消息推送API可以向特定的设备标识符发送交易推送通知。本节包含完整的代码示例，您可以使用这些示例通过 AWS 最终用户消息推送发送推送API通知 AWS SDK。

您可以使用这些示例通过 AWS 最终用户消息推送支持的任何推送通知服务发送推送通知。目前，AWS 最终用户消息推送支持以下渠道：Firebase 云消息 (FCM)、Apple 推送通知服务 (APNs)、百度云推送和亚马逊设备消息 (ADM)。

有关端点、区段和渠道的更多代码示例，请参阅代码示例。

> **Note**
>
> 当您通过 Firebase Cloud Messaging (FCM) 服务发送推送通知时，请在调用 “ AWS 最终用户消息推送API” 时使用该服务名称GCM。谷歌于 2018 年 4 月 10 日停止了谷歌云端消息 (GCM) 服务。但是， AWS 最终用户消息推送API使用GCM服务名称来表示其通过FCM服务发送的消息，以保持与GCM服务停用之前编写的API代码的兼容性。

GCM (AWS CLI)

以下示例使用发送消息通过发送消息发送GCM推送通知。 AWS CLI Replace（替换） *token* 使用设备的唯一令牌和 *611e3e3cdd47474c9c1399a50example* 使用您的应用程序标识符。

```
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"
      }
  }
}
```

以下示例使用 send-mes sages 使用所有旧密钥发送GCM推送通知。 AWS CLI Replace（替换） *token* 使用设备的唯一令牌和 *611e3e3cdd47474c9c1399a50example* 使用您的应用程序标识符。

```
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
```

以下示例使用 send-messages 通过发送消息发送带有FCMv1消息有效负载的GCM推送通知。 AWS CLI Replace（替换） *token* 使用设备的唯一令牌和 *611e3e3cdd47474c9c1399a50example* 使用您的应用程序标识符。

```
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
```

如果使用 ImageUrl field forGCM，pinpoint 会将该字段作为数据通知发送pinpoint.notification.imageUrl，密钥为，这样可以防止图像开箱即用。请使用 RawContent 或添加对数据密钥的处理，例如将您的应用程序与集成 AWS Amplify。

Safari (AWS CLI)

你可以使用 “ AWS 最终用户消息推送” 向使用 Apple Safari 网络浏览器的 macOS 电脑发送消息。要向 Safari 浏览器发送消息，必须指定原始消息内容，且必须在消息有效负载中包含特定属性。为此，您可以在 A mazon Pinpoint 用户指南中创建带有原始消息有效负载的推送通知模板，或者直接在活动消息中指定原始消息内容。

> **Note**
>
> 在向使用 Safari web 浏览器的 macOS 笔记本电脑和台式电脑发送消息时，此特殊属性是必需的。发送到 iOS 设备（如 iPhones 和）不需要这样做iPads。

要向 Safari web 浏览器发送消息，必须指定原始消息有效负载。原始消息有效负载必须在 aps 对象中包含一个 url-args 数组。要向 Safari web 浏览器发送推送通知，该 url-args 数组是必需的。但是，数组中包含单个空元素是可以接受的。

以下示例使用发送消息向 S afari 网络浏览器发送通知。 AWS CLI Replace（替换） *token* 使用设备的唯一令牌和 *611e3e3cdd47474c9c1399a50example* 使用您的应用程序标识符。

```
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
```

有关 Safari 推送通知的更多信息，请参阅 Apple 开发者网站上的配置 Safari 推送通知。

APNS (AWS CLI)

以下示例使用发送消息通过发送消息发送APNS推送通知。 AWS CLI Replace（替换）*token* 使用设备的唯一令牌，*611e3e3cdd47474c9c1399a50example* 使用您的应用程序标识符，以及 *GAME_INVITATION* 带有唯一标识符。

```
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)

使用此示例 JavaScript 在 Node.js 中使用 for 发送推送通知。 AWS SDK此示例假设您已经 JavaScript 在 Node.js 中安装并配置了。SDK

此示例还假定您正在使用共享凭证文件来指定现有 用户的访问密钥和私密访问密钥。有关更多信息，请参阅 Node.js 开发人员指南 JavaScript 中的设置凭证。AWS SDK

```
'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

参考此示例，通过使用 AWS SDK for Python (Boto3)发送推送通知。此示例假设你已经安装并配置了 Python (Boto3) 的。SDK

此示例还假定您正在使用共享凭证文件来指定现有 用户的访问密钥和私密访问密钥。有关更多信息，请参阅 Python (Boto3) API 参考中的凭证。AWS SDK

```
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()
```

其他 资源

- 有关推送渠道模板的更多信息，请参阅 Amazon Pinpoint 用户指南中的创建推送通知模板。

在您的应用程序中接收推送通知

以下主题描述了如何修改你的 Swift、Android、React Native 或 Flutter 应用程序，使其能够接收推送通知。

主题

设置 Swift 推送通知

iOS 应用程序的推送通知是使用 Apple 推送通知服务 (APNs) 发送的。必须先在 Apple 开发人员门户上创建应用程序 ID，并且必须创建所需证书，然后才能将推送通知发送给 iOS 设备。您可以在 A AWS mplify 文档的设置推送通知服务中找到有关完成这些步骤的更多信息。

使用代APNs币

作为最佳实践，您应该开发自己的应用程序，以便在重新安装应用程序时重新生成客户的设备令牌。

如果接收者将其设备升级到 iOS 新的主要版本（例如，从 iOS 12 升级到 iOS 13），之后又重新安装了您的应用程序，则该应用程序会生成一个新的令牌。如果您的应用程序未刷新令牌，则会使用旧令牌来发送通知。因此，Apple 推送通知服务 (APNs) 拒绝了该通知，因为令牌现在无效。当您尝试发送通知时，您会收到来自的消息失败通知APNs。

设置 Android 推送通知

安卓应用的推送通知使用 Firebase 云端消息 (FCM) 发送，它取代了谷歌云端消息 (GCM)。必须先获取FCM凭证，然后才能向 Android 设备发送推送通知。然后，您可以使用这些凭证创建 Android 项目并启动可以接收推送通知的示例应用程序。您可以在 A AWS mplify 文档的推送通知部分中找到有关完成这些步骤的更多信息。

设置 Flutter 推送通知

Flutter 应用程序的推送通知使用适用于安卓和 i APNs OS 的 Firebase 云消息 (FCM) 发送。您可以在 AWS Amplify Flutter 文档的“推送通知”部分了解完成这些步骤的更多信息。

设置 React Native 推送通知

React Native 应用程序的推送通知使用适用于安卓和 APNs iOS 的 Firebase 云消息 (FCM) 发送。您可以在 A AWS mplify JavaScript 文档的推送通知部分找到有关完成这些步骤的更多信息。

在 “ AWS 最终用户消息推送” 中创建应用程序

要开始在 “ AWS 最终用户消息推送” 中发送推送通知，必须创建一个应用程序。接下来，您必须通过提供适当的凭证启用要使用的推送通知渠道。

您可以使用 AWS 最终用户消息推送控制台创建新应用程序并设置推送通知渠道。有关更多信息，请参阅 创建应用程序并启用推送渠道。

也可以使用AWS SDK、或 AWS Command Line Interface(AWS CLI) 来创建和设置应用程序。API要创建应用程序，请使用Apps资源。要配置推送通知渠道，请使用以下资源：

- APNs频道，使用 Apple 推送通知服务向 iOS 设备的用户发送消息。
- ADM向亚马逊 Kindle Fire 设备用户发送消息的@@ 频道。
- 百度渠道 将消息发送给百度用户。
- GCM使用 Firebase 云端消息 (FCM) 向安卓设备发送消息的@@ 频道，它取代了谷歌云端消息 (GCM)。

处理推送通知

获得发送推送通知所需的凭证后，您可以更新您的应用程序，使其能够接收推送通知。有关更多信息，请参阅文档中的推送通知——入门。 AWS Amplify

删除 应用程序

此过程将从您的账户中移除该应用程序以及该应用程序中的所有资源。

情境相关

应用程序

应用程序是所有 AWS 最终用户消息推送设置的存储容器。该应用程序还存储您的 Amazon Pinpoint 渠道、活动和旅程设置。

过程

1. 打开 “ AWS 最终用户消息推送” 控制台，网址为https://console.aws.amazon.com/push-notifications/。
2. 选择一个应用程序，然后选择 “删除”。
3. 在 “删除应用程序” 窗口中输入**`delete`**，然后选择 “删除”。

> **Important**
>
> 所有 Amazon Pinpoint 渠道、广告活动、旅程或细分也会被删除。

最佳实践

即使您已从客户的最佳利益考虑，仍可能遇到影响消息送达能力的情况。以下部分提供有助于确保您的推送通信送达目标受众的建议。

发送大量推送通知

在发送大量推送通知之前，请确保您的账户已配置为支持您的吞吐量要求。默认情况下，所有账户都配置为每秒发送 25,000 封邮件。如果您需要一秒钟发送超过 25,000 条消息，可以请求增加限额。有关更多信息，请参阅 AWS 最终用户消息推送配额。

确保您的账户正确配置了您计划使用的每个推送通知提供商的证书，例如FCM或APNs。

最后，设计一种处理异常的方法。每种推送通知服务提供不同的异常消息。对于交易发送，如果在消息发送期间确定相应的平台令牌（例如FCM）或证书（例如）无效，则每个端点的主状态代码为 200，则每个端点的状态代码为 400 永久失败。API APN

AWS 最终用户消息推送中的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 AWS 最终用户消息推送的合规性计划，请参阅AWS 按合规计划划分的范围内AWS 服务按合规计划。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 AWS 最终用户消息推送时如何应用分担责任模型。以下主题向您介绍如何配置 AWS 最终用户消息推送以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护 AWS 最终用户消息推送资源。

主题

AWS 最终用户消息推送中的数据保护

分 AWS 担责任模型适用于 AWS 最终用户消息推送中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础架构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私FAQ。 有关欧洲数据保护的信息，请参阅责任AWS 共担模型和AWS安全GDPR博客上的博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭据并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用SSL/TLS与 AWS 资源通信。我们需要 TLS 1.2，建议使用 TLS 1.3。
- 使用API进行设置和用户活动记录 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或访问时需要 FIPS 140-2 经过验证的加密模块API，请使用端点。FIPS有关可用FIPS端点的更多信息，请参阅联邦信息处理标准 (FIPS) 140-2。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括 AWS 服务 使用控制台、、API或处理 AWS 最终用户消息推送或其他内容时 AWS SDKs。 AWS CLI在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您URL向外部服务器提供，我们强烈建议您不要在中包含凭据信息，URL以验证您对该服务器的请求。

数据加密

AWS 最终用户消息推送数据在传输过程中和静态时都经过加密。当您向 AWS 最终用户消息推送提交数据时，它会在接收和存储数据时对数据进行加密。当您从 “ AWS 最终用户消息推送” 中检索数据时，它会使用当前的安全协议将数据传输给您。

静态加密

AWS 最终用户消息推送会加密它为您存储的所有数据。这包括配置数据、用户和端点数据、分析数据以及您在 AWS 最终用户消息推送中添加或导入的任何数据。为了加密您的数据， AWS 最终用户消息推送使用服务代表您拥有和维护的内部 AWS Key Management Service (AWS KMS) 密钥。我们会定期轮换这些密钥。有关的信息 AWS KMS，请参阅《AWS Key Management Service 开发人员指南》。

传输中加密

AWS 最终用户消息推送使用HTTPS和传输层安全 (TLS) 1.2 或更高版本与您的客户端和应用程序通信。要与其他 AWS 服务通信， AWS 最终用户消息推送使用HTTPS和 TLS 1.2。此外，当您使用控制

台创建和管理 AWS 最终用户消息推送资源时 AWS SDK AWS Command Line Interface，所有通信都使用HTTPS和 TLS 1.2 进行保护。

密钥管理

为了加密您的 AWS 最终用户消息推送数据， AWS 最终用户消息推送使用服务代表您拥有和维护的内部 AWS KMS 密钥。我们会定期轮换这些密钥。您无法配置和使用自己的密钥 AWS KMS 或其他密钥来加密存储在 “ AWS 最终用户消息推送” 中的数据。

互联网络流量隐私

网际流量隐私是指保护 AWS 最终用户消息推送与您的本地客户端和应用程序之间，以及 AWS 最终用户消息推送与同一 AWS 区域中其他 AWS 资源之间的连接和流量。以下功能和做法可以帮助您确保 AWS 最终用户消息推送的网际流量隐私。

AWS 最终用户消息推送与本地客户端和应用程序之间的流量

要在 AWS 最终用户消息推送与本地网络上的客户端和应用程序之间建立私有连接，可以使用 AWS Direct Connect。这使您能够使用标准的光纤以太网电缆将您的网络链接到一个 AWS Direct Connect 位置。电缆的一端连接您的路由器，另一端连接到 AWS Direct Connect 路由器。有关更多信息，请参阅AWS Direct Connect 《用户指南》中的什么是 AWS Direct Connect？。

为了帮助安全访问已发布 AWS 的最终用户消息推送APIs，我们建议您遵守API呼叫 AWS 的最终用户消息推送要求。 AWS 最终用户消息推送要求客户端使用传输层安全 (TLS) 1.2 或更高版本。客户端还必须支持具有完全向前保密性的密码套件 ()，例如 Ephemeral Diffie-Hellman (PFS) 或 Elliptic Curve Diffie-Hellman Ephemeral ()。DHE ECDHE大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 AWS 账户的 AWS Identity and Access Management (IAM) 委托人关联的私有访问密钥对请求进行签名。或者，您可以使用 AWS Security Token Service (AWS STS) 生成临时安全凭证来签名请求。

AWS 最终用户消息推送与其他 AWS 资源之间的流量

为了保护 AWS 最终用户消息推送与同一 AWS 区域内其他 AWS 资源之间的通信，默认情况下， AWS 最终用户消息推送使用HTTPS和 TLS 1.2。

AWS 最终用户消息推送的身份和访问管理

AWS Identity and Access Management (IAM) AWS 服务 可以帮助管理员安全地控制对 AWS 资源的访问权限。IAM管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 AWS 最终用户消息推送资源。IAM无需支付额外费用即可使用。 AWS 服务

主题

受众

使用 AWS Identity and Access Management (IAM) 的方式会有所不同，具体取决于您在 AWS 最终用户消息推送中所做的工作。

服务用户-如果您使用 AWS 最终用户消息推送服务完成工作，则您的管理员会为您提供所需的凭据和权限。当你使用更多 AWS 的最终用户消息推送功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS 最终用户消息推送中的某项功能，请参阅对 AWS 最终用户消息推送身份和访问进行故障排除。

服务管理员-如果您负责公司 AWS 的最终用户消息推送资源，则可能拥有对 AWS 最终用户消息推送的完全访问权限。您的工作是确定您的服务用户应访问哪些 AWS 最终用户消息推送功能和资源。然后，您必须向IAM管理员提交更改服务用户权限的请求。查看此页面上的信息以了解的基本概念IAM。要详细了解贵公司如何IAM使用 AWS 最终用户消息推送，请参阅AWS 最终用户消息推送的工作原理 IAM。

IAM管理员-如果您是IAM管理员，则可能需要详细了解如何编写策略来管理对 AWS 最终用户消息推送的访问权限。要查看可在中使用 AWS 的最终用户消息推送基于身份的策略示例IAM，请参阅。AWS 最终用户消息推送的基于身份的策略示例

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 AWS 账户根用户、IAM用户身份或通过担任IAM角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （IAM身份中心）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员之前使用IAM角色设置了联合身份。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅《IAM用户指南》中的对 AWS API请求进行签名。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅用户指南中的多重身份验证和AWS IAM Identity Center 用户指南 AWS中的使用多因素身份验证 (MFA)。IAM

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建帐户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关需要您以 root 用户身份登录的任务的完整列表，请参阅《用户指南》中的 “需要根用户凭据的IAM任务”。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity C enter 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 Ident IAM ity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户

和应用程序中使用。有关IAM身份中心的信息，请参阅什么是IAM身份中心？ 在《AWS IAM Identity Center 用户指南》中。

IAM 用户和组

IAM用户是您内部 AWS 账户 对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时证书，而不是创建拥有密码和访问密钥等长期凭证的IAM用户。但是，如果您有需要IAM用户长期凭证的特定用例，我们建议您轮换访问密钥。有关更多信息，请参阅《IAM用户指南》中的针对需要长期凭证的用例定期轮换访问密钥。

IAM群组是指定IAM用户集合的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可以拥有一个名为的组，IAMAdmins并授予该组管理IAM资源的权限。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM用户指南》中的何时创建IAM用户（而不是角色）。

IAM 角色

IAM角色是您内部具有特定权限 AWS 账户 的身份。它与IAM用户类似，但与特定人员无关。您可以 AWS Management Console 通过切换IAM角色在中临时扮演角色。您可以通过调用 AWS CLI 或 AWS API操作或使用自定义操作来代入角色URL。有关使用角色的方法的更多信息，请参阅IAM用户指南中的使用IAM角色。

IAM具有临时证书的角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关用于联合身份验证的角色的信息，请参阅《IAM用户指南》中的为第三方身份提供商创建角色。如果您使用 IAM Identity Center，则需要配置权限集。为了控制您的身份在进行身份验证后可以访问的内容，Ident IAM ity Center 会将权限集关联到中的IAM角色。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的 权限集。
- 临时IAM用户权限-IAM 用户或角色可以代入一个IAM角色，为特定任务临时获得不同的权限。
- 跨账户访问-您可以使用IAM角色允许其他账户中的某人（受信任的委托人）访问您账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解角色和基于资源的跨账户访问策略之间的区别，请参阅IAM用户指南IAM中的跨账户资源访问权限。

- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在服务中拨打电话时，该服务通常会在 Amazon 中运行应用程序EC2或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS)-当您使用IAM用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS使用调用委托人的权限 AWS 服务以及 AWS 服务 向下游服务发出请求的请求。FAS只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出请求。在这种情况下，您必须具有执行这两个操作的权限。有关提出FAS请求时的政策详情，请参阅转发访问会话。
  - 服务角色-服务IAM角色是服务代替您执行操作的角色。IAM管理员可以在内部创建、修改和删除服务角色IAM。有关更多信息，请参阅《IAM用户指南》 AWS 服务中的创建角色以向委派权限。
  - 服务相关角色-服务相关角色是一种与服务相关联的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon 上运行的应用程序 EC2 — 您可以使用IAM角色管理在EC2实例上运行并发出 AWS CLI 或 AWS API请求的应用程序的临时证书。这比在EC2实例中存储访问密钥更可取。要为EC2实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建一个附加到该实例的实例配置文件。实例配置文件包含角色并允许在EC2实例上运行的程序获得临时证书。有关更多信息，请参阅IAM用户指南中的使用IAM角色向在 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用IAM角色还是使用IAM用户，请参阅《用户指南》中的何时创建IAM角色（而不是IAM用户）。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以JSON文档的 AWS 形式存储在中。有关JSON策略文档结构和内容的更多信息，请参阅《IAM用户指南》中的JSON策略概述。

管理员可以使用 AWS JSON策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对其所需资源执行操作的权限，IAM管理员可以创建IAM策略。然后，管理员可以将IAM策略添加到角色中，用户可以代入这些角色。

IAM无论您使用何种方法执行操作，策略都会定义该操作的权限。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或中获取角色信息 AWS API。

基于身份的策略

基于身份的策略是可以附加到身份（例如IAM用户、用户组或角色）的JSON权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅IAM用户指南中的创建IAM策略。

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管策略或内联策略之间进行选择，请参阅《IAM用户指南》中的在托管策略和内联策略之间进行选择。

基于资源的策略

基于资源的JSON策略是您附加到资源的策略文档。基于资源的策略的示例包括IAM角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略IAM中使用 AWS 托管策略。

访问控制列表 (ACLs)

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。ACLs与基于资源的策略类似，尽管它们不使用JSON策略文档格式。

Amazon S3 AWS WAF、和亚马逊VPC就是支持的服务示例ACLs。要了解更多信息ACLs，请参阅《亚马逊简单存储服务开发者指南》中的访问控制列表 (ACL) 概述。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界-权限边界是一项高级功能，您可以在其中设置基于身份的策略可以向IAM实体（IAM用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中

的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM用户指南》中的IAM实体的权限边界。

- 服务控制策略 (SCPs)-SCPs 是为中的组织或组织单位 (OU) 指定最大权限的JSON策略 AWS Organizations。 AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果您启用组织中的所有功能，则可以将服务控制策略 (SCPs) 应用于您的任何或所有帐户。对成员账户中的实体（包括每个实体）的权限进行了SCP限制 AWS 账户根用户。有关 Organizations 和的更多信息SCPs，请参阅《AWS Organizations 用户指南》中的SCPs工作原理。
- 会话策略 – 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅IAM用户指南中的策略评估逻辑。

AWS 最终用户消息推送的工作原理 IAM

在使用管理IAM对 AWS 最终用户消息推送的访问权限之前，请先了解哪些IAM功能可用于 AWS 最终用户消息推送。

IAM可用于 “ AWS 最终用户消息推送” 的功能

IAM 功能	AWS 最终用户消息推送支持
基于身份的策略	是
基于资源的策略	是
策略操作	是
策略资源	是
策略条件键	是
ACLs	不支持

IAM 功能	AWS 最终用户消息推送支持
ABAC（策略中的标签）	部分
临时凭证	是
主体权限	是
服务角色	是
服务相关角色	否

要全面了解 AWS 最终用户消息推送和其他 AWS 服务如何与大多数IAM功能配合使用，请参阅《IAM用户指南》IAM中与之配合使用的AWS 服务。

AWS 最终用户消息推送的基于身份的策略

支持基于身份的策略：是

基于身份的策略是可以附加到身份（例如IAM用户、用户组或角色）的JSON权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅IAM用户指南中的创建IAM策略。

使用IAM基于身份的策略，您可以指定允许或拒绝的操作和资源，以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可以在JSON策略中使用的所有元素，请参阅IAM用户指南中的IAMJSON策略元素参考。

AWS 最终用户消息推送的基于身份的策略示例

要查看基于 AWS 最终用户消息推送身份的策略示例，请参阅。AWS 最终用户消息推送的基于身份的策略示例

AWS 最终用户消息推送中基于资源的策略

支持基于资源的策略：是

基于资源的JSON策略是您附加到资源的策略文档。基于资源的策略的示例包括IAM角色信任策略和Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件

下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或另一个账户中的IAM实体指定为基于资源的策略中的委托人。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的IAM管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM用户指南》IAM中的跨账户资源访问权限。

AWS 最终用户消息推送的策略操作

支持策略操作：是

管理员可以使用 AWS JSON策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件下执行。

JSON策略`Action`元素描述了可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API操作同名。也有一些例外，例如没有匹配API操作的仅限权限的操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

要查看 AWS 最终用户消息推送操作列表，请参阅《服务授权参考》中的 “ AWS 最终用户消息推送定义的操作”。

AWS 最终用户消息推送中的策略操作在操作前使用以下前缀：

```
mobiletargeting
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

要查看基于 AWS 最终用户消息推送身份的策略示例，请参阅。AWS 最终用户消息推送的基于身份的策略示例

AWS 最终用户消息推送的策略资源

支持策略资源：是

管理员可以使用 AWS JSON策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件下执行。

ResourceJSON策略元素指定要应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。最佳做法是，使用资源的 Amazon 资源名称 (ARN) 来指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

要查看 AWS 最终用户消息推送资源类型及其列表ARNs，请参阅《服务授权参考》中的“ AWS 最终用户消息推送定义的资源”。要了解您可以为每ARN种资源指定哪些操作，请参阅 AWS 最终用户消息推送定义的操作。

要查看基于 AWS 最终用户消息推送身份的策略示例，请参阅。AWS 最终用户消息推送的基于身份的策略示例

AWS 最终用户消息推送的策略条件密钥

支持服务特定的策略条件键：是

管理员可以使用 AWS JSON策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在资源上标有IAM用户的用户名时，您才能向IAM用户授予访问该资源的权限。有关更多信息，请参阅《IAM用户指南》中的IAM策略元素：变量和标签。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅《IAM用户指南》中的AWS 全局条件上下文密钥。

要查看 AWS 最终用户消息推送条件键列表，请参阅《服务授权参考》中的 “AWS 最终用户消息推送条件密钥”。要了解可以使用条件键的操作和资源，请参阅AWS 最终用户消息推送定义的操作。

要查看基于 AWS 最终用户消息推送身份的策略示例，请参阅。AWS 最终用户消息推送的基于身份的策略示例

ACLs在 AWS 最终用户消息推送中

支持ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。ACLs与基于资源的策略类似，尽管它们不使用JSON策略文档格式。

ABAC使用 AWS 最终用户消息推送

支持ABAC（策略中的标签）：部分

基于属性的访问控制 (ABAC) 是一种基于属性定义权限的授权策略。在中 AWS，这些属性称为标签。您可以将标签附加到IAM实体（用户或角色）和许多 AWS 资源。为实体和资源添加标签是的第一步。ABAC然后，您可以设计ABAC策略，允许在委托人的标签与他们尝试访问的资源上的标签匹配时进行操作。

ABAC在快速增长的环境中很有用，也有助于解决策略管理变得繁琐的情况。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关的更多信息ABAC，请参阅什么是ABAC？ 在《IAM用户指南》中。要查看包含设置步骤的教程ABAC，请参阅IAM用户指南中的使用基于属性的访问控制 (ABAC)。

在 “ AWS 最终用户消息推送” 中使用临时证书

支持临时凭证：是

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关其他信息，包括哪些 AWS 服务 适用于临时证书 AWS 服务 ，请参阅《IAM用户指南》IAM中的 “适用于临时证书”。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM用户指南》中的切换到角色（控制台）。

您可以使用 AWS CLI 或手动创建临时证书 AWS API。然后，您可以使用这些临时证书进行访问 AWS。 AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅中的临时安全证书IAM。

AWS 最终用户消息推送的跨服务主体权限

支持转发访问会话 (FAS)：是

当您使用IAM用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS使用调用委托人的权限 AWS 服务以及 AWS 服务 向下游服务发出请求的请求。FAS只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出请求。在这种情况下，您必须具有执行这两个操作的权限。有关提出FAS请求时的政策详情，请参阅转发访问会话。

AWS 最终用户消息推送的服务角色

支持服务角色：是

服务IAM角色是服务代替您执行操作的角色。IAM管理员 可以在内部创建、修改和删除服务角色IAM。有关更多信息，请参阅《IAM用户指南》 AWS 服务中的创建角色以向委派权限。

> **Warning**
>
> 更改服务角色的权限可能会中断 “ AWS 最终用户消息推送” 功能。只有在 “ AWS 最终用户消息推送” 提供相关指导时才编辑服务角色。

AWS 最终用户消息推送的服务相关角色

支持服务相关角色：否

服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM管理员 可以查看但不能编辑服务相关角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅与之配合IAM使用的AWS 服务。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

AWS 最终用户消息推送的基于身份的策略示例

默认情况下，用户和角色无权创建或修改 AWS 最终用户消息推送资源。他们也无法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或来执行任务 AWS API。要授予用户对其所需资源执行操作的权限，IAM管理员可以创建IAM策略。然后，管理员可以将IAM策略添加到角色中，用户可以代入这些角色。

要了解如何使用这些示例策略文档创建IAM基于身份的JSON策略，请参阅IAM用户指南中的创建IAM策略。

有关 AWS 最终用户消息推送定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》中的 “AWS 最终用户消息推送的操作、资源和条件密钥”。ARNs

主题

策略最佳实践

基于身份的策略决定是否有人可以在您的账户中创建、访问或删除 AWS 最终用户消息推送资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM用户指南》中的AWS 托AWS 管策略或工作职能托管策略。
- 应用最低权限权限-使用IAM策略设置权限时，仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用应用权限IAM的更多信息，请参阅《IAM用户指南》IAM中的策略和权限。
- 使用IAM策略中的条件进一步限制访问权限-您可以在策略中添加条件以限制对操作和资源的访问权限。例如，您可以编写一个策略条件来指定所有请求都必须使用发送SSL。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM用户指南》中的IAMJSON策略元素：条件。
- 使用 A IAM ccess Analyzer 验证您的IAM策略以确保权限的安全性和功能性 — A IAM ccess Analyzer 会验证新的和现有的策略，以便策略符合IAM策略语言 (JSON) 和IAM最佳实

践。IAMAccess Analyzer 提供了 100 多项策略检查和可行的建议，可帮助您制定安全和实用的策略。有关更多信息，请参阅《IAM用户指南》中的 IAMAccess Analyzer 策略验证。

- 需要多重身份验证 (MFA)-如果您的场景需要IAM用户或 root 用户 AWS 账户，请打开MFA以提高安全性。要要求MFA何时调用API操作，请在策略中添加MFA条件。有关更多信息，请参阅《IAM用户指南》中的配置MFA受保护的API访问权限。

有关中最佳做法的更多信息IAM，请参阅《IAM用户指南》IAM中的安全最佳实践。

使用 AWS 最终用户消息推送控制台

要访问 AWS 最终用户消息推送控制台，您必须拥有一组最低权限。这些权限必须允许您在中列出和查看有关 AWS 最终用户消息推送资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

您无需为仅拨打 AWS CLI 或的用户设置最低控制台权限 AWS API。相反，只允许访问与他们尝试执行的API操作相匹配的操作。

为确保用户和角色仍然可以使用 AWS 最终用户消息推送控制台，还需要将AWSEndUserMessaging AWS 托管策略附加到实体。有关更多信息，请参阅《用户指南》中的向IAM用户添加权限。

```
{
 "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": [
    "*"
   ]
  }
 ]
}
```

允许用户查看他们自己的权限

此示例说明如何创建允许IAM用户查看附加到其用户身份的内联和托管策略的策略。此策略包括在控制台上或使用或以编程方式完成此操作的 AWS CLI 权限。 AWS API

```
{
    "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": "*"
        }
    ]
}
```

对 AWS 最终用户消息推送身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS 最终用户消息推送时可能遇到的常见问题，以及IAM。

主题

- 我无权在 “ AWS 最终用户消息推送” 中执行操作
- 我无权执行 iam：PassRole
- 我想允许我以外的人访问我的 AWS 账户 “ AWS 最终用户消息推送” 资源

我无权在 “ AWS 最终用户消息推送” 中执行操作

如果您收到错误提示，表明您无权执行某个操作，则您必须更新策略以允许执行该操作。

当mateojacksonIAM用户尝试使用控制台查看虚构*my-example-widget*资源的详细信息但没有虚构权限时，就会出现以下示例错误。mobiletargeting:*GetWidget*

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

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 mobiletargeting:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到一条错误消息，指出您无权执行该iam:PassRole操作，则必须更新您的策略以允许您将角色传递给 “ AWS 最终用户消息推送”。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为的用户marymajor尝试使用控制台在 “ AWS 最终IAM用户消息推送” 中执行操作时，会出现以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人访问我的 AWS 账户 “ AWS 最终用户消息推送” 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解 AWS 最终用户消息推送是否支持这些功能，请参阅AWS 最终用户消息推送的工作原理IAM。
- 要了解如何提供对您拥有的资源的访问权限，请参阅《IAM用户指南》中的 AWS 账户 向其他IAM用户提供访问权限。 AWS 账户
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅IAM用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过联合身份验证提供访问权限，请参阅《用户指南》中的向经过外部身份验证的用户提供访问权限（联合身份验证）。IAM
- 要了解使用角色和基于资源的策略进行跨账户访问的区别，请参阅IAM用户指南IAM中的跨账户资源访问权限。

AWS 最终用户消息推送的合规性验证

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了在这些基础上 AWS 部署以安全性和合规性为重点的基准环境的步骤。
- 在 Amazon Web Services 上进行HIPAA安全与合规架构 — 本白皮书描述了各公司如何使用 AWS 来创建HIPAA符合条件的应用程序。

> **Note**
>
> 并非所有 AWS 服务 人都有HIPAA资格。有关更多信息，请参阅《HIPAA合格服务参考》。

- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控件的列表，请参阅 Security Hub 控件参考。
- Amazon GuardDuty — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务 检测您的工作负载、容器和数据面临的潜在威胁。 GuardDuty 可以帮助您满足各种合规性要求 PCIDSS，例如满足某些合规性框架规定的入侵检测要求。
- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

AWS 最终用户消息推送的弹性

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。 AWS 区域 提供多个物理分隔和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础设施。

除了 AWS 全球基础架构外， AWS 最终用户消息推送还提供多项功能，以帮助支持您的数据弹性和备份需求。

AWS 最终用户消息推送中的基础设施安全

作为一项托管服务， AWS 最终用户消息推送受《Amazon Web Services：安全流程概述》白皮书中描述的 AWS 全球网络安全程序保护。

您可以使用 AWS 已发布的API呼叫通过网络访问 AWS 最终用户消息推送。客户端必须支持传输层安全 (TLS) 1.2 或更高版本。客户还必须支持具有完全向前保密性的密码套件 ()，例如（Ephemeral Diffie-HellmanPFS）或（Elliptic Cur DHE ve Ephemeral Diffie-Hellman）。ECDHE大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与IAM委托人关联的私有访问密钥对请求进行签名。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

配置和漏洞分析

作为一项托管服务， AWS 最终用户消息推送受《Amazon Web Services：安全流程概述》白皮书中描述的 AWS 全球网络安全程序保护。这意味着 AWS 它管理和执行基本的安全任务和程序，以强化、修补、更新和以其他方式维护您的账户和资源的底层基础架构。这些流程已通过相应第三方审核和认证。

安全最佳实操

使用 AWS Identity and Access Management (IAM) 帐户来API控制对操作的访问权限，尤其是创建、修改或删除资源的操作。对于API，此类资源包括项目、活动和旅程。

- 为管理 资源的每个人（包括您自己）创建一个单独的用户。不要使用 AWS 根凭证来管理资源。
- 授予每位用户执行其职责所需的最低权限集。
- 使用IAM群组有效管理多个用户的权限。
- 定期轮换 IAM 凭证。

有关安全性的更多信息，请参阅AWS 最终用户消息推送中的安全性。有关的更多信息IAM，请参阅Ident AWS ity and Access 管理。有关IAM最佳实践的信息，请参阅IAM最佳实践。

监控 AWS 最终用户消息推送

监控是维护 AWS 最终用户消息推送和其他AWS解决方案的可靠性、可用性和性能的重要组成部分。AWS提供了以下监控工具，用于监视 AWS 最终用户消息推送、报告出现问题并在适当时自动采取措施：

- Amazon 会实时 CloudWatch监控您的 AWS 资源和您运行 AWS 的应用程序。您可以收集和跟踪指标，创建自定义的控制平面，以及 设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。例如，您可以 CloudWatch 跟踪您的 Amazon EC2 实例的CPU使用情况或其他指标，并在需要时自动启动新实例。有关更多信息，请参阅 Amazon CloudWatch 用户指南。
- Amazon Lo CloudWatch gs 使您能够监控、存储和访问来自亚马逊EC2实例和其他来源的日志文件。 CloudTrail CloudWatch 日志可以监视日志文件中的信息，并在达到特定阈值时通知您。您还可以在高持久性存储中检索您的日志数据。有关更多信息，请参阅 Amazon CloudWatch 日志用户指南。
- Amazon EventBridge 可用于实现 AWS 服务自动化，并自动响应系统事件，例如应用程序可用性问题或资源更改。来自 AWS 服务的事件几乎实时 EventBridge 地传送到。您可以编写简单的规则来指示您关注的事件，并指示要在事件匹配规则时执行的自动化操作。有关更多信息，请参阅 Amazon EventBridge 用户指南。
- AWS CloudTrail捕获由您的账户或代表您的 AWS 账户发出的API调用和相关事件，并将日志文件传输到您指定的 Amazon S3 存储桶。您可以识别哪些用户和帐户拨打了电话 AWS、发出呼叫的源 IP 地址以及呼叫发生的时间。有关更多信息，请参阅 《AWS CloudTrail 用户指南》。

使用 Amazon 监控 AWS 最终用户消息推送 CloudWatch

您可以使用监控 AWS 最终用户消息推送 CloudWatch，它会收集原始数据并将其处理为可读的近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

有关指标和维度的列表，请参阅亚马逊 Pinp oint 用户指南 CloudWatch中的监控亚马逊 Pinpoin t。

使用记录 AWS 最终用户消息推送API呼叫 AWS CloudTrail

AWS 最终用户消息推送与 AWS CloudTrail一项服务集成，该服务提供用户、角色或 AWS 服务在 AWS 最终用户消息推送中采取的操作的记录。 CloudTrail 将所有对 AWS 最终用户消息推送的API呼

叫捕获为事件。捕获的呼叫包括来自 AWS 最终用户消息推送控制台的呼叫和对 AWS 最终用户消息推送API操作的代码调用。如果您创建跟踪，则可以将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 AWS 最终用户消息推送的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “事件历史记录” 中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向 AWS 最终用户消息推送发出的请求、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅《AWS CloudTrail 用户指南》。

AWS 最终用户消息推送信息 CloudTrail

CloudTrail 在您创建账户 AWS 账户 时已在您的账户上启用。当 AWS 最终用户消息推送中发生活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在中查看、搜索和下载最近发生的事件 AWS 账户。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录您的事件 AWS 账户，包括 AWS 最终用户消息推送的事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 为以下各项配置亚马逊SNS通知 CloudTrail
- 接收来自多个地区的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

所有 AWS 最终用户消息推送操作均由《最终用户消息推送参考》记录 CloudTrail 并记录在《AWS 最终用户消息推送API参考》中。例如，调用`UpdateApnsChannel`和`GetApnsVoipChannel`操作会在 CloudTrail 日志文件中生成条目。`GetAdmChannel`

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用 root 还是 AWS Identity and Access Management (IAM) 用户凭据发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅CloudTrail userIdentity元素。

了解 AWS 最终用户消息推送日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共API调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

使用接口 AWS 端点访问最终用户消息推送 (AWS PrivateLink)

您可以使用 AWS PrivateLink 在您的VPC和 AWS 最终用户消息推送之间创建私有连接。无需使用 Internet 网关、NAT设备VPC、连接或VPN AWS Direct Connect 连接，即可像访问 AWS 最终用户消息推送一样访问最终用户消息推送。您中的实例VPC不需要公有 IP 地址即可访问 AWS 最终用户消息推送。

您可以通过创建由 AWS PrivateLink提供支持的接口端点来建立此私有连接。我们将在您为接口端点启用的每个子网中创建一个端点网络接口。这些是请求者管理的网络接口，是发往 AWS 最终用户消息推送的流量的入口点。

有关更多信息，请参阅AWS PrivateLink 指南 AWS PrivateLink中的AWS 服务 直通访问。

AWS 最终用户消息推送的注意事项

在为 AWS 最终用户消息推送设置接口端点之前，请查看AWS PrivateLink 指南中的注意事项。

AWS 最终用户消息推送支持通过接口端点调用其所有API操作。

VPC终端用户消息推送不支持 AWS 端点策略。默认情况下，允许通过接口 AWS 端点对最终用户消息推送进行完全访问。或者，您可以将安全组与端点网络接口关联，以控制通过接口端点发送到 AWS 最终用户消息推送的流量。

为 AWS 最终用户消息推送创建接口端点

您可以使用 Amazon VPC 控制台或 AWS Command Line Interface (AWS CLI) 为 AWS 最终用户消息推送创建接口终端节点。有关更多信息，请参阅《AWS PrivateLink 指南》中的创建接口端点。

使用以下服务名称为 AWS 最终用户消息推送创建接口端点：

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

如果您DNS为接口终端节点启用私有，则可以使用其默认区域DNS名称向 AWS 最终用户消息推送API发出请求。例如，`com.amazonaws.us-east-1.pinpoint`。

为 VPC 端点创建端点策略

终端节点策略是您可以附加到接口终端节点的IAM资源。默认端点策略允许通过接口端点对 AWS 最终用户消息推送进行完全访问权限。要控制允许从您的访问 AWS 最终用户消息推送的权限VPC，请将自定义终端节点策略附加到接口终端节点。

端点策略指定以下信息：

- 可以执行操作的委托人（AWS 账户、IAM用户和IAM角色）。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅《AWS PrivateLink 指南》中的使用端点策略控制对服务的访问权限。

示例： AWS 最终用户消息推送操作的VPC端点策略

以下是自定义端点策略的示例。当您将此策略附加到接口终端节点时，它会向所有资源的所有委托人授予访问列出 AWS 的最终用户消息推送操作的权限。

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

AWS 最终用户消息推送配额

您的每项 AWS 服务 AWS 账户 都有默认配额，以前称为限制。除非另有说明，否则，每个限额是区域特定的。您可以请求增加某些配额，但其他一些配额无法增加。

要查看 AWS 最终用户消息推送的配额，请打开 S ervice Quotas 控制台。在导航窗格中，选择AWS服务，然后选择 Amazon Pinpoint。

您的AWS账户具有以下与 AWS 最终用户消息推送相关的配额。

资源	默认限额	是否符合提高配额的条件？
一个活动中每秒可发送的推送通知的最大数量	每秒 25,000 个通知	是的，请使用 S ervice Quotas 控制台
Amazon 设备消息 (ADM) 消息有效负载大小	每个消息 6 KB	否
Apple 推送通知服务 (APNs) 消息有效载荷大小	每封邮件 4 KB	否
APNs 沙盒消息负载大小	每封邮件 4 KB	否
百度云推送消息有效负载大小	每封邮件 4 KB	否
Firebase 云消息 (FCM) 消息有效负载大小	每封邮件 4 KB	否

《 AWS 最终用户消息推送用户指南》的文档历史记录

下表描述了 AWS 最终用户消息推送的文档版本。

变更	说明	日期
初始版本	《 AWS 最终用户消息推送用户指南》的初始版本	2024 年 7 月 24 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。