使用者指南

AWS 最終使用者傳訊推送

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

AWS 最終使用者傳訊推送: 使用者指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS 最終使用者傳訊推送？ 1
您是第一次 AWS 使用者傳訊推送使用者嗎？ 1
AWS 最終使用者傳訊推送的功能 1
存取 AWS 最終使用者傳訊推送 2
區域可用性 2
設定 AWS 帳戶 4
註冊 AWS 帳戶 4
建立具有管理存取權的使用者 4
開始使用 6
建立應用程式並啟用推送頻道 7
內容 7
先決條件 7
程序 8
停用推送頻道 10
傳送推送訊息 11
其他資源 24
在應用程式中接收推送通知 25
設定快速推播通知 25
使用 APNS 字符 25
設定 Android 推送通知 25
設定 Flutter 推播通知 26
設定 React Native 推播通知 26
建立應用程式 26
處理推送通知 26
刪除應用程式 27
內容 27
程序 27
最佳實務 28
傳送大量推播通知 28
安全 29
資料保護 29
資料加密 30
傳輸中加密 30
金鑰管理 31

網際網路流量隱私權 31
身分與存取管理 32
目標對象 32
使用身分驗證 33
使用政策管理存取權 35
AWS 最終使用者傳訊推送如何與 IAM 搭配使用 37
身分型政策範例 43
故障診斷 46
法規遵循驗證 48
恢復能力 48
基礎設施安全性 49
組態與漏洞分析 49
安全最佳實務 49
監控 50
使用 CloudWatch 進行監控 50
CloudTrail 日誌 50
AWS CloudTrail 中的最終使用者傳訊推送資訊 51
了解 AWS 最終使用者傳訊推送日誌檔案項目 52
AWS PrivateLink 53
考量事項 53
建立介面端點 53
建立端點政策 54
配額 55
文件歷史紀錄 56
..... lvii

什麼是 AWS 最終使用者傳訊推送？

> **Note**
>
> Amazon Pinpoint 的推送通知功能現在稱為 AWS 最終使用者傳訊。

使用 AWS 最終使用者傳訊推送，您可以透過推送通知管道傳送推送通知，以吸引應用程式的使用者。我們支援 Apple Push Notification Service (APNs)、Firebase Cloud Messaging (FCM)、Amazon Device Messaging (ADM) 和百度推送。

主題

- 您是第一次 AWS 使用者傳訊推送使用者嗎？
- AWS 最終使用者傳訊推送的功能
- 存取 AWS 最終使用者傳訊推送
- 區域可用性

您是第一次 AWS 使用者傳訊推送使用者嗎？

如果您是第一次使用 AWS 最終使用者傳訊推送，建議您先閱讀下列章節：

- 設定 AWS 帳戶
- AWS 最終使用者傳訊推送入門
- 建立應用程式並啟用推送頻道

AWS 最終使用者傳訊推送的功能

您可以針對以下推播通知服務使用單獨的管道，將推播通知傳送到應用程式：

- Firebase Cloud Messaging (FCM)
- Apple 推播通知服務 (APN)

> **Note**
>
> 您可以使用 APN 將訊息傳送到 iPhone 和 iPad 等 iOS 裝置，以及 Mac 筆記型電腦和桌上型電腦等 macOS 裝置上的 Safari 瀏覽器。

- 百度雲推送
- Amazon Device Messaging (ADM)

存取 AWS 最終使用者傳訊推送

簡要說明存取服務的不同方式，無論是透過主控台、CLI 或 API。

您可以使用下列界面管理 AWS 最終使用者傳訊推送：

AWS 最終使用者傳訊推送主控台

: 您建立和管理 AWS 最終使用者傳訊推送資源的 Web 界面。如果您已註冊 AWS 帳戶，您可以從存取 AWS 最終使用者訊息推送主控台 AWS Management Console。

AWS Command Line Interface

: 使用命令列 shell 中的命令與 AWS 服務互動。Windows、macOS 和 Linux AWS Command Line Interface 支援 。如需 的詳細資訊 AWS CLI，請參閱 AWS Command Line Interface 使用者指南。您可以在命令參考中找到 AWS 最終使用者傳訊推送命令。 AWS CLI

AWS SDKs

: 如果您是軟體開發人員，偏好使用語言特定的 APIs 來建置應用程式，而不是透過 HTTP 或 HTTPS 提交請求， AWS 會提供程式庫、範例程式碼、教學課程和其他資源。這些程式庫提供基本函數來自動化任務，例如以密碼編譯方式簽署您的請求、重試請求，以及處理錯誤回應。這些函數可協助您更有效率地開始使用。如需詳細資訊，請參閱在 AWS上建置的工具。

區域可用性

AWS 最終使用者傳訊推送可在北美洲、歐洲、亞洲和大洋洲的數個 AWS 區域 中使用。在每個區域中， AWS 會維護多個可用區域。這些可用區域各自實體隔離，但以私有、低延遲、高輸送量、高度冗餘的網路連線加以整合。這些可用區域用於提供非常高水準的可用性和備援，同時將延遲降至最低。

若要進一步了解 AWS 區域，請參閱指定 AWS 區域 您的帳戶可在 中使用哪個Amazon Web Services 一般參考。如需目前提供 AWS 最終使用者傳訊推送的所有區域清單，以及每個區域的端點，請參閱中的 Amazon Pinpoint API AWS 和服務端點的端點和配額Amazon Web Services 一般參考。如需進一步了解各區域之可用區域數量的資訊，請參閱 AWS 全球基礎設施。

設定 AWS 帳戶

您必須先取得 AWS 帳戶 具有足夠 IAM 許可的 ，才能使用 AWS 最終使用者傳訊推送將推送通知傳送至應用程式。這 AWS 帳戶 也可以用於 AWS 生態系統中的其他服務。

主題

- 註冊 AWS 帳戶
- 建立具有管理存取權的使用者

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   註冊程序的一部分包括接聽電話或文字訊息，以及在電話鍵盤上輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 會在註冊程序完成後傳送確認電子郵件給您。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理存取權的使用者

註冊 後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center和建立管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS Management Console身分登入 。在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者（主控台）啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的啟用 AWS IAM Identity Center。

2. 在 IAM Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center《 使用者指南》中的使用預設值設定使用者存取 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

  如需使用 IAM Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

AWS 最終使用者傳訊推送入門

若要設定 AWS 最終使用者傳訊推送，以便其可以傳送推送通知到您的應用程式，您必須先提供憑證，授權 AWS 最終使用者傳訊推送將訊息傳送至應用程式。您提供的登入資料取決於您使用的推送通知系統：

- 如需 Apple Push Notification Service (APN) 登入資料，請參閱 Apple 開發人員文件中的從 Apple 取得加密金鑰和金鑰 ID，以及從 Apple 取得提供者憑證。
- 如需可透過 Firebase 主控台取得的 Firebase Cloud Messaging (FCM) 登入資料，請參閱 Firebase Cloud Messaging。
- 如需百度登入資料，請參閱百度。
- 如需 Amazon Device Messaging (ADM) 登入資料，請參閱取得登入資料。

建立應用程式並啟用推送頻道

您必須先建立應用程式並啟用推送通知管道，才能使用 AWS 最終使用者傳訊推送來傳送推送通知。

內容

Application (應用程式)

應用程式是所有 AWS 最終使用者傳訊推送設定的儲存容器。應用程式也會存放您的 Amazon Pinpoint 頻道、行銷活動和旅程設定。

索引鍵

AWS 使用者傳訊推送所使用的私有簽署金鑰，以密碼編譯方式簽署 APNs身分驗證字符。您可以透過您的 Apple 開發人員帳戶取得簽署金鑰。

如果您提供簽署金鑰，AWS 則使用者傳訊推送會使用權杖來驗證您傳送的每個推送通知APNs。您可以使用簽署金鑰，傳送推送通知至 APN 生產和沙盒環境。

與憑證不同，您的簽署金鑰不會過期。您只需提供您的簽署金鑰一次，而且之後不需要更新它。您可以將相同的簽署金鑰用於多個應用程式。詳情請參閱《Xcode 說明》中的使用驗證權杖與 APN 通訊。

憑證

當您傳送推送通知時， AWS 最終使用者傳訊推送用來向 APNs TLS 憑證。APN 憑證可同時支援製作和沙盒環境，也可以只支援沙盒環境。您可以透過您的 Apple 開發人員帳戶取得憑證。

憑證會在一年後過期。發生這種情況時，您必須建立新的憑證，然後提供給 AWS 最終使用者傳訊推送，以續約推送通知交付。詳情請參閱《Xcode 說明》中的使用 TLS 憑證與 APN 通訊。

先決條件

您必須先為推送服務取得有效的登入資料，才能使用任何推送管道。如需取得登入資料的詳細資訊，請參閱AWS 最終使用者傳訊推送入門。

程序

請依照這些指示來建立應用程式，並啟用任何推送頻道。若要完成此程序，您只需輸入應用程式名稱。您可以稍後啟用或停用任何推送頻道。

1. 在 https：//https://console.aws.amazon.com/push-notifications/ 開啟 AWS 最終使用者傳訊推送主控台。
2. 選擇建立應用程式。
3. 在應用程式名稱中，輸入應用程式的名稱。
4. （選用）請依照此選用步驟來啟用 Apple Push Notification 服務 APNs)。
   a. 對於 Apple 推送通知服務 APNs)，選取啟用。
   b. 對於預設身分驗證類型，請選擇下列其中一項：
      i. 如果您選擇金鑰登入資料，請從您的 Apple 開發人員帳戶提供以下資訊。 AWS 最終使用者訊息推送需要此資訊來建構身分驗證字符。
         - 金鑰 ID – 指派給簽署金鑰的 ID。
         - 封包識別符 – 指派給 iOS 應用程式的 ID。
         - 團隊識別符 – 指派給 Apple 開發人員帳戶團隊的 ID。
         - 驗證金鑰 – 您在建立驗證金鑰時，從 Apple 開發人員帳戶下載的 .p8 檔案。
      ii. 如果您選擇憑證登入資料，請提供下列資訊：
         - SSL 憑證 – 您的 TLS 憑證的 .p12 檔案。
         - 憑證密碼 – 如果您已為憑證指派密碼，請在此處輸入。
         - 憑證類型 – 選取要使用的憑證類型。
5. （選用）依照此選用步驟來啟用 Firebase Cloud Messaging (FCM)。
   a. 對於 Firebase Cloud Messaging (FCM)，選取啟用。
   b. 對於預設身分驗證類型，請選擇下列其中一項：
      i. 針對權杖登入資料（建議），選擇選擇檔案，然後選擇您的服務 JSON 檔案。
      ii. 對於金鑰憑證，在 API 金鑰中輸入您的金鑰。
6. （選用）依照此選用步驟來啟用百度雲端推送。
   a. 針對百度雲端推送，選取啟用。

b. 針對 API 金鑰，輸入您的 API 金鑰。

c. 對於私密金鑰，輸入您的私密金鑰。

7. （選用）遵循此選用步驟來啟用 Amazon Device Messaging。

a. 針對 Amazon Device Messaging，選取啟用。

b. 針對用戶端 ID，輸入您的用戶端 ID。

c. 對於用戶端秘密，輸入您的用戶端秘密。

8. 選擇建立應用程式。

停用推送頻道

請依照這些指示來停用任何推送頻道。

1. 在 https：//https://console.aws.amazon.com/push-notifications/ 開啟 AWS 最終使用者傳訊推送主控台。
2. 選擇包含您的推送登入資料的應用程式。
3. （選用）對於 Apple 推送通知服務 APNs)，清除啟用。
4. （選用）針對 Firebase Cloud Messaging (FCM) 清除啟用。
5. （選用）針對百度雲端推送清除啟用。
6. （選用）針對 Amazon Device Messaging 清除啟用。
7. 選擇 Save changes (儲存變更)。

傳送訊息

AWS 最終使用者傳訊推送 API 可以將交易推送通知傳送至特定裝置識別符。本節包含完整的程式碼範例，您可以使用 AWS SDK，透過 AWS 最終使用者傳訊推送 API 傳送推送通知。

您可以使用這些範例，透過 AWS 最終使用者訊息推送支援的任何推送通知服務傳送推送通知。目前，AWS 最終使用者傳訊推送支援下列頻道：Firebase Cloud Messaging (FCM)、Apple Push Notification Service (APNs)、百度雲端推送和 Amazon Device Messaging (ADM)。

如需端點、區段和頻道的更多程式碼範例，請參閱程式碼範例。

> **Note**
>
> 當您透過 Firebase Cloud Messaging (FCM) 服務傳送推送通知時，請在呼叫 AWS 最終使用者訊息推送 API `GCM`時使用服務名稱。Google 已於 2018 年 4 月 10 日終止 Google Cloud Messaging (GCM) 服務。不過， AWS 最終使用者傳訊推送 API 會使用其透過 FCM `GCM`服務傳送之訊息的服務名稱，以維持與終止 GCM 服務之前寫入之 API 程式碼的相容性。

GCM (AWS CLI)

下列範例使用 send-messages 傳送 GCM Push 通知與 AWS CLI。將*##*取代為裝置的唯一字符，並將 *611e3e3cdd47474c9c1399a50example* 取代為您的應用程式識別符。

```
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"
      }
  }
}
```

下列範例使用 send-messages 傳送 GCM Push 通知，並使用所有舊版金鑰搭配 AWS CLI。將*##*取代為裝置的唯一字符，並將 *611e3e3cdd47474c9c1399a50example* 取代為您的應用程式識別符。

```
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
```

下列範例使用 send-messages，使用 傳送具有 FCMv1 訊息承載的 GCM Push 通知 AWS CLI。將*##*取代為裝置的唯一字符，並將 *611e3e3cdd47474c9c1399a50example* 取代為您的應用程式識別符。

```
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
```

如果將 `ImageUrl` 欄位用於 GCM ，則 pinpoint 會將 欄位作為資料通知傳送，金鑰為 `pinpoint.notification.imageUrl`，這可防止影像從框中轉譯出來。請使用 RawContent 或新增資料金鑰的處理，例如整合您的應用程式 AWS Amplify。

Safari (AWS CLI)

您可以使用 AWS 最終使用者傳訊推送，將訊息傳送到使用 Apple Safari Web 瀏覽器的 macOS 電腦。若要傳訊到 Safari 瀏覽器，必須指定原始訊息內容，且須在訊息承載中包含特定屬性。您可以在 Amazon Pinpoint 使用者指南中，建立具有原始訊息承載的推送通知範本，或直接在行銷活動訊息中指定原始訊息內容，以達成此目的。

> **Note**
>
> 傳送到使用 Safari 網頁瀏覽器的 macOS 筆記型電腦和桌上型電腦，需要此特殊屬性。傳送到 iPhone 和 iPad 等 iOS 裝置，不需要此屬性。

若要傳訊到 Safari 網頁瀏覽器，必須指定原始訊息承載。原始訊息承載必須在 `aps` 物件中包含一個 `url-args` 陣列。需要 `url-args` 陣列才能將推播通知傳送到 Safari 網頁瀏覽器。但可以接受陣列包含單一空白元素。

下列範例使用 send-messages 來使用 將通知傳送至 Safari Web 瀏覽器 AWS CLI。將*##*取代為裝置的唯一字符，並將 *611e3e3cdd47474c9c1399a50example* 取代為您的應用程式識別符。

```
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
```

如需 Safari 推播通知的詳細資訊，請參閱 Apple 開發人員網站上的設定 Safari 推播通知。

APNS (AWS CLI)

下列範例使用 send-messages 傳送 APNS Push 通知與 AWS CLI。將*##*取代為裝置的唯一字符，將 *611e3e3cdd47474c9c1399a50example* 取代為您的應用程式識別符，將 *GAME_INVITATION* 取代為唯一識別符。

```
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)

使用此範例，在 Node.js 中使用適用於 JavaScript 的 AWS SDK 傳送推播通知。此範例假設您已安裝並設定了 SDK for JavaScript in Node.js。

此範例也會假設您使用共用的登入資料檔案，指定現有 使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for JavaScript in Node.js 開發人員指南中的設定憑證。

```
'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

使用 適用於 Python (Boto3) 的 AWS SDK藉此範例傳送推送通知。此範例假設您已安裝並設定了 SDK for Python (Boto3)。

此範例也會假設您使用共用的登入資料檔案，指定現有 使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for Python (Boto3) API 參考中的憑證。

```
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
 "
           "### Python (Boto3) # AWS SDK.")

# 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()
```

其他資源

- 如需推送管道範本的詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的建立推送通知範本。

在應用程式中接收推送通知

下列主題說明如何修改 Swift、Android、React Native 或 Flutter 應用程式，以便接收推送通知。

主題

- 設定快速推播通知
- 設定 Android 推送通知
- 設定 Flutter 推播通知
- 設定 React Native 推播通知
- 在 AWS 最終使用者傳訊推送中建立應用程式
- 處理推送通知

設定快速推播通知

iOS 應用程式的推播通知，是使用 Apple 推播通知服務 (APN) 傳送。在您可以傳送推送通知至 iOS 裝置之前，您必須在 Apple 開發人員入口網站上建立一個應用程式 ID，並且必須建立必要的憑證。您可以在 AWS Amplify 文件的設定推送通知服務中找到有關完成這些步驟的詳細資訊。

使用 APNS 字符

根據最佳實務，您應該開發應用程式，以便在重新安裝應用程式時重新產生客戶的裝置字符。

如果收件人將其裝置升級到新的 iOS 主要版本 (例如，從 iOS 12 升級到 iOS 13)，並在稍後重新安裝您的應用程式，則應用程式會產生新的字符。如果您的應用程式未重新整理字符，則會使用較舊的字符來傳送通知。因此，Apple 推播通知服務 (APNS) 會拒絕通知，因為字符現在無效。嘗試傳送通知時，您會收到來自 APNS 的訊息失敗通知。

設定 Android 推送通知

Android 應用程式的推播通知是使用 Firebase Cloud Messaging (FCM) 傳送 (以前使用 Google Cloud Messaging (GCM))。您必須先取得 FCM 憑證，才能將推播通知傳送到 Android 裝置。接著您可以使用那些登入資料來建立 Android 專案，並啟動可接收推送通知的範例應用程式。您可以在 AWS Amplify 文件的推送通知區段中找到有關完成這些步驟的詳細資訊。

設定 Flutter 推播通知

Flutter 應用程式的推播通知在 Android 裝置上是使用 Firebase Cloud Messaging (FCM) 傳送，在 iOS 裝置上是使用 APN 傳送。您可以在 AWS Amplify Flutter 文件的推播通知區段中，找到完成相關步驟的詳細資訊。

設定 React Native 推播通知

React Native 應用程式的推播通知在 Android 裝置上是使用 Firebase Cloud Messaging (FCM) 傳送，在 iOS 裝置上是使用 APN 傳送。您可以在 AWS Amplify JavaScript 文件的推播通知區段中，找到完成相關步驟的詳細資訊。

在 AWS 最終使用者傳訊推送中建立應用程式

若要在 AWS 最終使用者傳訊推送中開始傳送推送通知，您必須建立應用程式。接著，您必須提供適當的登入資料，以啟用您想要使用的推送通知管道。

您可以使用 AWS 最終使用者訊息推送主控台建立新的應用程式和設定推送通知管道。如需詳細資訊，請參閱建立應用程式並啟用推送頻道。

您也可以使用 API、 AWS SDK 或 AWS Command Line Interface() 來建立和設定應用程式AWS CLI。若要建立應用程式，請使用 `Apps` 資源。若要設定推送通知管道，請使用下列資源：

- APN 管道使用 Apple 推播通知服務向 iOS 裝置的使用者傳送訊息。
- ADM 管道，以傳送訊息給 Amazon Kindle Fire 裝置的使用者。
- 百度管道，以傳送訊息給百度使用者。
- GCM 管道使用 Firebase Cloud Messaging (FCM) (以前使用 Google Cloud Messaging (GCM))，向 Android 裝置傳送訊息。

處理推送通知

在您取得傳送推播通知所需的登入資料後，您可以更新您的應用程式，以便他們能夠接收推播通知。如需詳細資訊，請參閱推播通知 - 在文件中開始。 AWS Amplify

刪除應用程式

此程序會從您的帳戶和應用程式中的所有資源移除應用程式。

內容

Application (應用程式)

應用程式是所有 AWS 最終使用者傳訊推送設定的儲存容器。應用程式也會存放您的 Amazon Pinpoint 頻道、行銷活動和旅程設定。

程序

1. 開啟位於 https：//https://console.aws.amazon.com/push-notifications/ 的 AWS 終端使用者傳訊推送主控台。
2. 選擇應用程式，然後選擇刪除。
3. 在刪除應用程式視窗中，輸入 **delete**，然後選擇刪除。

> **Important**
>
> 也會刪除任何 Amazon Pinpoint 頻道、行銷活動、旅程或客群。

最佳實務

即使您已將客戶的最佳利益列入考量，仍會遇到可能影響您的訊息可交付性的情況。下節提供的建議，有助推播通訊觸及目標受眾。

傳送大量推播通知

傳送大量推送通知之前，請確定您的帳戶已設定為支援您的輸送量需求。根據預設，所有帳戶都會設定為每秒傳送 25，000 則訊息。若需要在一秒鐘內傳送超過 25,000 封郵件，可以請求增加配額。如需詳細資訊，請參閱AWS 最終使用者訊息推送配額。

請確定您的帳戶已正確設定您計劃使用的每個推送通知提供者的憑證，例如 FCM 或 APNs。

最後想出處理例外訊息的方法。每個推播通知服務提供的例外訊息都不同。關於交易傳送，如果訊息傳送期間對應的平台權杖 (例如 FCM) 或憑證 (例如 APN) 被判定無效，您會收到 API 呼叫主要狀態代碼 200，以及每個端點狀態代碼 400 永久失敗。

AWS 最終使用者傳訊推送的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。在AWS 合規計劃中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 AWS 最終使用者傳訊推送的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 AWS 最終使用者傳訊推送時套用共同責任模型。下列主題說明如何設定 AWS 最終使用者傳訊推送，以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS 最終使用者傳訊推送資源。

主題

- AWS 最終使用者傳訊推送中的資料保護
- AWS 最終使用者傳訊推送的身分和存取管理
- AWS 最終使用者傳訊推送的合規驗證
- AWS 最終使用者傳訊推送中的彈性
- AWS 最終使用者傳訊推送中的基礎設施安全性
- 組態與漏洞分析
- 安全最佳實務

AWS 最終使用者傳訊推送中的資料保護

AWS 共同責任模型適用於 AWS 最終使用者傳訊推送中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 AWS 最終使用者傳訊推送，或使用主控台、API AWS CLI或 AWS SDKs的其他 AWS 服務 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

資料加密

AWS 最終使用者傳訊推送資料會在傳輸中和靜態時加密。當您將資料提交至 AWS 最終使用者傳訊推送時，它會在接收和儲存資料時加密資料。當您從 AWS 最終使用者傳訊推送擷取資料時，它會使用目前的安全通訊協定將資料傳輸給您。

靜態加密

AWS 最終使用者傳訊推送會加密其為您存放的所有資料。這包括組態資料、使用者和端點資料、分析資料，以及您新增或匯入 AWS 最終使用者傳訊推送的任何資料。為了加密您的資料， AWS 最終使用者傳訊推送會使用服務代表您擁有和維護的內部 AWS Key Management Service (AWS KMS) 金鑰。我們會定期輪換這些金鑰。如需 的詳細資訊 AWS KMS，請參閱 AWS Key Management Service 開發人員指南。

傳輸中加密

AWS 最終使用者傳訊推送使用 HTTPS 和 Transport Layer Security (TLS) 1.2 或更新版本與您的用戶端和應用程式通訊。若要與其他 AWS 服務通訊， AWS 最終使用者傳訊推送會使用 HTTPS 和 TLS

1.2。此外，當您使用 主控台、 AWS 開發套件或 建立和管理 AWS 最終使用者傳訊推送資源時 AWS Command Line Interface，所有通訊都會使用 HTTPS 和 TLS 1.2 進行保護。

金鑰管理

為了加密 AWS 您的最終使用者傳訊推送資料， AWS 最終使用者傳訊推送會使用服務代表您擁有和維護的內部 AWS KMS 金鑰。我們會定期輪換這些金鑰。您無法佈建和使用自己的 AWS KMS 或其他金鑰來加密存放在 AWS 最終使用者傳訊推送中的資料。

網際網路流量隱私權

網際網路流量隱私權是指保護 AWS 最終使用者傳訊推送與內部部署用戶端和應用程式之間，以及 AWS 最終使用者傳訊推送與相同 AWS 區域中其他 AWS 資源之間的連線和流量。下列功能和實務可協助您確保 AWS 最終使用者傳訊推送的網際網路流量隱私權。

AWS 最終使用者傳訊推送與內部部署用戶端和應用程式之間的流量

若要在 AWS 最終使用者傳訊推送與用戶端和現場部署網路上的應用程式之間建立私有連線，您可以使用 AWS Direct Connect。這可讓您使用標準光纖乙太網路纜線將網路連結至某個 AWS Direct Connect 位置。纜線的一端連接到路由器。另一端連接到 AWS Direct Connect 路由器。如需詳細資訊，請參閱《 AWS Direct Connect使用者指南》中的什麼是AWS Direct Connect ？。

為了協助安全存取 AWS 最終使用者傳訊推送已發佈APIs，建議您遵守 API 呼叫的最終 AWS 使用者傳訊推送要求。 AWS 最終使用者傳訊推送要求用戶端使用 Transport Layer Security (TLS) 1.2 或更新版本。用戶端也必須支援具備完整轉寄密碼 (PFS) 的密碼套件，例如暫時性 Diffie-Hellman (DHE) 或橢圓曲線 Diffie-Hellman Ephemeral (ECDHE)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，必須使用存取金鑰 ID 和與您 AWS 帳戶之 AWS Identity and Access Management (IAM) 主體相關聯的私密存取金鑰來簽署請求。或者，您可以使用 AWS Security Token Service (AWS STS) 來產生暫時安全登入資料來簽署請求。

AWS 最終使用者傳訊推送和其他 AWS 資源之間的流量

為了保護 AWS 最終使用者傳訊推送與相同 AWS 區域中其他 AWS 資源之間的通訊， AWS 最終使用者傳訊推送預設會使用 HTTPS 和 TLS 1.2。

AWS 最終使用者傳訊推送的身分和存取管理

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員控制誰可以進行身分驗證 （登入） 和授權 （具有許可），以使用 AWS 最終使用者傳訊推送資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- AWS 最終使用者傳訊推送如何與 IAM 搭配使用
- AWS 最終使用者傳訊推送的身分型政策範例
- 對 AWS 最終使用者傳訊推送身分和存取權進行故障診斷

目標對象

使用方式 AWS Identity and Access Management (IAM) 會有所不同，取決於您在 AWS 最終使用者傳訊推送中所做的工作。

服務使用者 – 如果您使用 AWS 最終使用者傳訊推送服務來執行任務，您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS 最終使用者傳訊推送功能來執行工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 AWS 最終使用者傳訊推送中的功能，請參閱 對 AWS 最終使用者傳訊推送身分和存取權進行故障診斷。

服務管理員 – 如果您在公司負責 AWS 最終使用者傳訊推送資源，您可能擁有 AWS 最終使用者傳訊推送的完整存取權。您的任務是判斷服務使用者應存取 AWS 的使用者傳訊推送功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何透過 AWS 最終使用者傳訊推送使用 IAM，請參閱 AWS 最終使用者傳訊推送如何與 IAM 搭配使用。

IAM 管理員 – 如果您是 IAM 管理員，建議您了解如何撰寫政策以管理 AWS 最終使用者傳訊推送存取權的詳細資訊。若要檢視可在 IAM 中使用的 AWS 最終使用者傳訊推送身分型政策範例，請參閱 AWS 最終使用者傳訊推送的身分型政策範例。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的登入資料以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的適用於 API 請求的AWS Signature 第 4 版。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重驗證 (MFA) 來提高帳戶的安全性。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的多重要素驗證和《IAM 使用者指南》中的 IAM 中的AWS 多重要素驗證。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取帳戶中的所有 AWS 服務 和 資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳實務是要求人類使用者，包括需要管理員存取權的使用者，使用身分提供者的聯合身分來 AWS 服務 使用臨時憑證來存取 。

聯合身分是您企業使用者目錄、Web 身分提供者、 AWS Directory Service、Identity Center 目錄，或 AWS 服務 是透過身分來源提供的登入資料存取的任何使用者。當聯合身分存取時 AWS 帳戶，它們會擔任 角色，而角色會提供臨時登入資料。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群組，以便

在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是 中的身分 AWS 帳戶 ，具有單一人員或應用程式的特定許可。建議您盡可能依賴臨時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色（主控台）。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱《IAM 使用者指南》中的為第三方身分提供者 (聯合) 建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以直接將政策連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。
  - 服務連結角色 – 服務連結角色是一種連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件， AWS 當與身分或資源相關聯時， 會定義其許可。當委託人（使用者、根使用者或角色工作階段）發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 的形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，請參閱《IAM 使用者指南》中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交

集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。

- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 的多個 的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括支援 RCPs AWS 服務 的 清單，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 (RCPs)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS 最終使用者傳訊推送如何與 IAM 搭配使用

在您使用 IAM 管理 AWS 最終使用者傳訊推送的存取權之前，請先了解哪些 IAM 功能可與 AWS 最終使用者傳訊推送搭配使用。

您可以搭配 AWS 最終使用者傳訊推送使用的 IAM 功能

IAM 功能	AWS 最終使用者傳訊推送支援
身分型政策	是
資源型政策	是

IAM 功能	AWS 最終使用者傳訊推送支援
政策動作	是
政策資源	是
政策條件索引鍵	是
ACL	否
ABAC(政策中的標籤)	部分
臨時憑證	是
主體許可	是
服務角色	是
服務連結角色	否

若要全面了解 AWS 最終使用者傳訊推送和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱《IAM 使用者指南》中的與 IAM AWS 搭配使用的 服務。

AWS 最終使用者傳訊推送的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

AWS 最終使用者傳訊推送的身分型政策範例

若要檢視 AWS 最終使用者傳訊推送身分型政策的範例，請參閱 AWS 最終使用者傳訊推送的身分型政策範例。

AWS 最終使用者傳訊推送中的資源型政策

支援資源型政策：是

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當委託人和資源位於不同位置時 AWS 帳戶，信任帳戶中的 IAM 管理員也必須授予委託人實體（使用者或角色）存取資源的許可。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

AWS 最終使用者傳訊推送的政策動作

支援政策動作：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS 最終使用者傳訊推送動作的清單，請參閱服務授權參考中的AWS 最終使用者傳訊推送定義的動作。

AWS 最終使用者傳訊推送中的政策動作在動作之前使用下列字首：

```
mobiletargeting
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

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

```
        "mobiletargeting:action2"
           ]
```

若要檢視 AWS 最終使用者傳訊推送身分型政策的範例，請參閱 AWS 最終使用者傳訊推送的身分型政策範例。

AWS 最終使用者傳訊推送的政策資源

支援政策資源：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。陳述式必須包含 `Resource` 或 `NotResource` 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

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

若要查看 AWS 最終使用者傳訊推送資源類型及其 ARNs的清單，請參閱服務授權參考中的AWS 最終使用者傳訊推送定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱AWS 最終使用者傳訊推送 定義的動作。

若要檢視 AWS 最終使用者傳訊推送身分型政策的範例，請參閱 AWS 最終使用者傳訊推送的身分型政策範例。

AWS 最終使用者傳訊推送的政策條件索引鍵

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素 (或 `Condition` 區塊) 可讓您指定使陳述式生效的條件。`Condition` 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 `Condition` 元素，或是在單一 `Condition` 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值， 會使用邏輯OR操作 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定的條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 AWS 最終使用者傳訊推送條件金鑰的清單，請參閱服務授權參考中的AWS 最終使用者傳訊推送的條件金鑰。若要了解您可以使用條件金鑰的動作和資源，請參閱AWS 最終使用者傳訊推送 定義的動作。

若要檢視 AWS 最終使用者傳訊推送身分型政策的範例，請參閱 AWS 最終使用者傳訊推送的身分型政策範例。

AWS 最終使用者傳訊推送中的 ACLs

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 搭配 AWS 最終使用者傳訊推送

支援 ABAC (政策中的標籤)：部分

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在 中 AWS，這些屬性稱為標籤。您可以將標籤連接到 IAM 實體 （使用者或角色） 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的使用屬性型存取控制 (ABAC)。

搭配 AWS 最終使用者傳訊推送使用臨時登入資料

支援臨時憑證：是

當您使用臨時登入資料登入時，有些 AWS 服務 無法運作。如需詳細資訊，包括哪些 AWS 服務 使用臨時登入資料，請參閱《AWS 服務 IAM 使用者指南》中的使用 IAM 的 。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則會使用暫時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的從使用者切換至 IAM 角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

AWS 使用者傳訊推送的跨服務主體許可

支援轉寄存取工作階段 (FAS)：是

當您使用 IAM 使用者或角色在 中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱轉發存取工作階段。

AWS 最終使用者傳訊推送的服務角色

支援服務角色：是

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。

> **Warning**
>
> 變更服務角色的許可可能會中斷 AWS 最終使用者傳訊推送功能。只有在 AWS 最終使用者傳訊推送提供指引時，才能編輯服務角色。

AWS 最終使用者傳訊推送的服務連結角色

支援服務連結角色：否

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱可搭配 IAM 運作的AWS 服務。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

AWS 最終使用者傳訊推送的身分型政策範例

根據預設，使用者和角色沒有建立或修改 AWS 最終使用者傳訊推送資源的許可。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行任務。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需 AWS 最終使用者傳訊推送所定義之動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱服務授權參考中的AWS 最終使用者傳訊推送的動作、資源和條件金鑰。

主題

- 政策最佳實務
- 使用 AWS 最終使用者傳訊推送主控台
- 允許使用者檢視他們自己的許可

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 AWS 最終使用者傳訊推送資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。

- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_configure-api-require.html中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

使用 AWS 最終使用者傳訊推送主控台

若要存取 AWS 最終使用者傳訊推送主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 AWS 最終使用者傳訊推送資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

為了確保使用者和角色仍然可以使用 AWS 最終使用者傳訊推送主控台，也請將 `AWSEndUserMessaging` AWS 受管政策連接到實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

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

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

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

對 AWS 最終使用者傳訊推送身分和存取權進行故障診斷

使用以下資訊來協助您診斷和修正在使用 AWS 最終使用者傳訊推送和 IAM 時可能遇到的常見問題。

主題

- 我無權在 AWS 最終使用者傳訊推送中執行動作
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 以外的人員 AWS 帳戶 存取我的 AWS 最終使用者傳訊推送資源

我無權在 AWS 最終使用者傳訊推送中執行動作

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在`mateojackson` IAM 使用者嘗試使用主控台檢視一個虛構 *`my-example-widget`* 資源的詳細資訊，但卻無虛構 `mobiletargeting:`*`GetWidget`* 許可時發生。

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

在此情況下，必須更新 `mateojackson` 使用者的政策，允許使用 `mobiletargeting:`*`GetWidget`* 動作存取 *`my-example-widget`* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行 `iam:PassRole`動作，您的政策必須更新，以允許您將角色傳遞給 AWS 最終使用者傳訊推送。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為 的 IAM `marymajor` 使用者嘗試使用主控台在 AWS 最終使用者傳訊推送中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

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

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 AWS 最終使用者傳訊推送資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 AWS 最終使用者傳訊推送是否支援這些功能，請參閱 AWS 最終使用者傳訊推送如何與 IAM 搭配使用。
- 若要了解如何提供您擁有 AWS 帳戶 的資源存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 中提供存取權給 IAM 使用者。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。

- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

AWS 最終使用者傳訊推送的合規驗證

若要了解 是否 AWS 服務 在特定合規計劃範圍內，請參閱AWS 服務 合規計劃然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- HIPAA 合格服務參考 – 列出 HIPAA 合格服務。並非所有 AWS 服務 都符合 HIPAA 資格。
- AWS 合規資源 – 此工作手冊和指南集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同責任模型。本指南摘要說明跨多個架構 （包括國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 保護 AWS 服務 和映射指南至安全控制的最佳實務。
- 《 AWS Config 開發人員指南》中的使用 規則評估資源 – AWS Config 服務會評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub – 這 AWS 服務 可讓您全面檢視其中的安全狀態 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱「Security Hub 控制參考」。
- Amazon GuardDuty – 這會監控您的環境是否有可疑和惡意活動，以 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。
- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和業界標準的方式。

AWS 最終使用者傳訊推送中的彈性

AWS 全球基礎設施是以 AWS 區域 和 可用區域為基礎建置。 AWS 區域 提供多個實體分隔和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

除了 AWS 全球基礎設施之外， AWS 最終使用者傳訊推送還提供數種功能，以協助支援您的資料彈性和備份需求。

AWS 最終使用者傳訊推送中的基礎設施安全性

作為受管服務， AWS 最終使用者傳訊推送受到 Amazon Web Services：安全程序概觀白皮書中所述的 AWS 全球網路安全程序的保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 AWS 最終使用者傳訊推送。用戶端必須支援 Transport Layer Security (TLS) 1.2 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

組態與漏洞分析

作為受管服務， AWS 最終使用者傳訊推送受到 Amazon Web Services：安全程序概觀白皮書中所述的 AWS 全球網路安全程序的保護。這表示 會 AWS 管理和執行基本安全任務和程序，以強化、修補、更新和以其他方式維護您帳戶和資源的基礎基礎設施。這些程序已由適當的第三方進行檢閱並認證。

安全最佳實務

使用 AWS Identity and Access Management (IAM) 帳戶來控制對 API 操作的存取，尤其是建立、修改或刪除資源的操作。 API 這類資源包括專案、行銷活動和旅程。

- 為管理 資源的每個人 (包括您自己)，建立個人使用者。請勿使用 AWS 根登入資料來管理資源。
- 授予每個使用者執行其職責所需最低程度的許可。
- 使用 IAM 群組來有效管理多個使用者的許可。
- 定期輪替您的 IAM 登入資料。

如需詳細資訊，請參閱 AWS 最終使用者傳訊推送的安全性。如需 IAM 的相關資訊，請參閱 AWS Identity and Access Management。如需 IAM 最佳實務的資訊，請參閱 IAM 最佳實務。

監控 AWS 最終使用者訊息推送

監控是維護 AWS 最終使用者傳訊推送和其他 AWS 解決方案的可靠性、可用性和效能的重要部分。AWS 提供下列監控工具來監看 AWS 最終使用者傳訊推送、報告錯誤，以及適時採取自動動作：

- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。
- Amazon CloudWatch Logs 可讓您監控、存放和存取來自 Amazon EC2 執行個體、CloudTrail 及其他來源的日誌檔案。CloudWatch Logs 可監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。
- Amazon EventBridge 可用來自動化您的 AWS 服務，並自動回應系統事件，例如應用程式可用性問題或資源變更。來自 AWS 服務的事件會以近乎即時的方式交付至 EventBridge。您可編寫簡單的規則，來指示您在意的事件，以及當事件符合規則時所要自動執行的動作。如需詳細資訊，請參閱 Amazon EventBridge 使用者指南。
- AWS CloudTrail 會擷取由您的帳戶或代表 AWS 您的帳戶發出的 API 呼叫和相關事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/。

使用 Amazon CloudWatch 監控 AWS 最終使用者訊息推送

您可以使用 CloudWatch 監控 AWS 最終使用者傳訊推送，它會收集原始資料並將其處理為可讀且近乎即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

如需指標和維度的清單，請參閱《Amazon Pinpoint 使用者指南》中的使用 CloudWatch 監控 Amazon Pinpoint。 Amazon Pinpoint

使用 記錄 AWS 最終使用者傳訊推送 API 呼叫 AWS CloudTrail

AWS 最終使用者傳訊推送已與 整合 AWS CloudTrail，此服務提供 AWS 最終使用者傳訊推送 AWS 中使用者、角色或服務所採取動作的記錄。CloudTrail 會將 AWS 最終使用者傳訊推送的所有 API 呼叫擷

取為事件。擷取的呼叫包括來自 AWS 最終使用者訊息推送主控台的呼叫，以及最終 AWS 使用者訊息推送 API 操作的程式碼呼叫。如果您建立線索，您可以啟用 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 AWS 最終使用者傳訊推送的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。使用 CloudTrail 收集的資訊，您可以判斷對 AWS 最終使用者傳訊推送提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

AWS CloudTrail 中的最終使用者傳訊推送資訊

建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。當活動在 AWS 最終使用者傳訊推送中發生時，該活動會記錄在 CloudTrail 事件中，以及事件歷史記錄中的其他服務 AWS 事件。您可以在 中檢視、搜尋和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱「使用 CloudTrail 事件歷史記錄檢視事件」。

若要持續記錄 中的事件 AWS 帳戶，包括 AWS 最終使用者傳訊推送的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。追蹤會記錄 AWS 分割區中所有 區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析 CloudTrail 日誌中收集的事件資料並對其採取行動。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案

所有 AWS 最終使用者傳訊推送動作都會由 CloudTrail 記錄，並記錄在AWS 最終使用者傳訊推送 API 參考中。例如，對 `GetAdmChannel`、`UpdateApnsChannel` 和 `GetApnsVoipChannel` 動作發出的呼叫會在 CloudTrail 記錄檔案中產生專案。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 請求是使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 AWS 最終使用者傳訊推送日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

使用界面端點存取 AWS 最終使用者傳訊推送 (AWS PrivateLink)

您可以使用 在 VPC 和 AWS 最終使用者傳訊推送之間 AWS PrivateLink 建立私有連線。您可以像在 VPC 中一樣存取 AWS 最終使用者傳訊推送，而無需使用網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址，即可存取 AWS 最終使用者傳訊推送。

您可以建立由 AWS PrivateLink提供支援的介面端點來建立此私有連線。我們會在您為介面端點啟用的每個子網中建立端點網路介面。這些是請求者管理的網路介面，可做為最終 AWS 使用者訊息推送目的地流量的進入點。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的AWS 服務 透過 存取 AWS PrivateLink 。

AWS 最終使用者傳訊推送的考量

設定 AWS 最終使用者傳訊推送的介面端點之前，請檢閱 AWS PrivateLink 指南中的考量事項。

AWS 最終使用者傳訊推送支援透過介面端點呼叫其所有 API 動作。

AWS 最終使用者傳訊推送不支援 VPC 端點政策。根據預設，透過介面端點允許完整存取 AWS 最終使用者傳訊推送。或者，您可以將安全群組與端點網路介面建立關聯，以控制使用者 AWS 傳訊推送至介面端點的流量。

為 AWS 最終使用者傳訊推送建立介面端點

您可以使用 Amazon VPC 主控台或 AWS Command Line Interface () 為 AWS 最終使用者傳訊推送建立介面端點AWS CLI。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的建立介面端點。

使用下列服務名稱建立 AWS 最終使用者傳訊推送的介面端點：

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

如果您為介面端點啟用私有 DNS，您可以使用其預設的區域 DNS 名稱向 AWS 最終使用者傳訊推送提出 API 請求。例如：`com.amazonaws.us-east-1.pinpoint`。

為您的介面端點建立端點政策

端點政策為 IAM 資源，您可將其連接至介面端點。預設端點政策允許透過介面端點完整存取 AWS 最終使用者傳訊推送。若要控制最終 AWS 使用者傳訊 從您的 VPC 推送的允許存取，請將自訂端點政策連接至介面端點。

端點政策會指定以下資訊：

- 可執行動作 (AWS 帳戶、IAM 使用者和 IAM 角色) 的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱「AWS PrivateLink 指南」中的使用端點政策控制對服務的存取。

範例： AWS 最終使用者傳訊推送動作的 VPC 端點政策

以下是自訂端點政策的範例。當您將此政策連接到介面端點時，它會授予所有資源上所有主體所列出的 AWS 最終使用者訊息推送動作的存取權。

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

AWS 最終使用者訊息推送配額

您的 AWS 帳戶 具有每個 AWS 服務的預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定規定。您可以請求提高某些配額，而其他配額無法提高。

若要檢視 AWS 最終使用者訊息推送的配額，請開啟 Service Quotas 主控台。在導覽窗格中，選擇 AWS 服務，然後選取 Amazon Pinpoint。

您的 AWS 帳戶具有與 AWS 最終使用者傳訊推送相關的下列配額。

資源	預設配額	是否可增加？
行銷活動中每秒可傳送的推播通知數目上限	每秒 25,000 個通知	是，使用 Service Quotas 主控台
Amazon Device Messaging (ADM) 訊息承載大小	每則訊息 6 KB	否
Apple 推播通知服務 (APN) 訊息承載大小	每則訊息 4 KB	否
APN 沙盒訊息承載大小	每則訊息 4 KB	否
百度雲推送訊息承載大小	每則訊息 4 KB	否
Firebase Cloud Messaging (FCM) 訊息承載大小	每則訊息 4 KB	否

AWS 最終使用者傳訊推送使用者指南的文件歷史記錄

下表說明 AWS 最終使用者傳訊推送的文件版本。

變更	描述	日期
初始版本	AWS 使用者傳訊推送使用者指南的初始版本	2024 年 7 月 24 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。