aws

Guía del usuario

# AWS Envío push de mensajería para el usuario final

# AWS Envío push de mensajería para el usuario final: Guía del usuario

# Table of Contents

# ¿Qué es AWS End User Messaging Push?

> **Note**
>
> Las funciones de notificaciones push de Amazon Pinpoint ahora se denominan AWS End User Messaging.

Con la mensajería push para el usuario AWS final, puede captar la atención de los usuarios de sus aplicaciones mediante el envío de notificaciones push a través de un canal de notificaciones push. Admitimos Apple Push Notification Service (APNs), Firebase Cloud Messaging (FCM), Amazon Device Messaging (ADM) y Baidu Push.

Temas



## ¿Es la primera vez que utiliza AWS End User Messaging Push?

Si es la primera vez que utiliza AWS End User Messaging Push, le recomendamos que comience por leer las siguientes secciones:

- Configuración de un Cuenta de AWS
- Cómo empezar con AWS End User Messaging Push
- Crear una aplicación y habilitar los canales push

## Características de la mensajería AWS push para usuarios finales

Puede enviar notificaciones de inserción a las aplicaciones con canales independientes para los siguientes servicios de notificaciones de inserción:

- Firebase Cloud Messaging () FCM
- Servicio de notificaciones push de Apple () APNs

> **Note**
>
> Se puede utilizar APNs para enviar mensajes a dispositivos iOS como iPhones yiPads, así como al navegador Safari en dispositivos macOS, como ordenadores portátiles y de sobremesa Mac.

- Baidu Cloud Push
- Mensajería para dispositivos Amazon (ADM)

## Acceso a la mensajería push para usuarios AWS finales

Explique brevemente las diferentes formas de acceder al servicio, ya sea mediante consola oAPI. CLI

Puede administrar la mensajería push para el usuario AWS final mediante las siguientes interfaces:

AWS Consola push de mensajería para el usuario final

: La interfaz web en la que se crean y administran los recursos de mensajería push para el usuario AWS final. Si se ha registrado en una Cuenta de AWS, puede acceder a la consola push de mensajería para el usuario AWS final desde el AWS Management Console.

AWS Command Line Interface

: Interactúa con AWS los servicios mediante los comandos de la consola de la línea de comandos. AWS Command Line Interface Es compatible con Windows, macOS y Linux. Para obtener más información sobre el AWS CLI, consulte la Guía AWS Command Line Interface del usuario. Puede encontrar los comandos push de mensajería para el usuario AWS final en la AWS CLI Referencia de comandos.

AWS SDKs

: Si es un desarrollador de software que prefiere crear aplicaciones con un lenguaje específico APIs en lugar de enviar una solicitud en lugar de enviar una HTTP solicitudHTTPS, AWS proporciona bibliotecas, códigos de muestra, tutoriales y otros recursos. Estas bibliotecas proporcionan funciones básicas que automatizan las tareas, como la firma criptográfica de las solicitudes, el reintento de las solicitudes y la gestión de las respuestas a los errores. Estas funciones le ayudan a empezar de forma más eficiente. Para obtener más información, consulte Herramientas para crear en AWS.

# Disponibilidad regional

AWS End User Messaging Push está disponible Regiones de AWS en varios países de Norteamérica, Europa, Asia y Oceanía. En cada región, AWS mantiene varias zonas de disponibilidad. Estas zonas de disponibilidad están físicamente aisladas entre sí, pero están unidas mediante conexiones de red privadas con un alto nivel de rendimiento y redundancia y con baja latencia. Estas zonas de disponibilidad se utilizan para proporcionar niveles muy altos de disponibilidad y redundancia y, al mismo tiempo, minimizar la latencia.

Para obtener más información Regiones de AWS, consulte Especificar qué Regiones de AWS cuenta puede usar en el Referencia general de Amazon Web Services. Para obtener una lista de todas las regiones en las que la mensajería push para usuarios AWS finales está disponible actualmente y los puntos de enlace de cada región, consulte los puntos de enlace y las cuotas de Amazon API Pinpoint AWS y los puntos de enlace de servicio en. Referencia general de Amazon Web Services Para obtener más información sobre la cantidad de zonas de disponibilidad de cada región, consulte Infraestructura global de AWS.

# Configuración de un Cuenta de AWS

Antes de que puedas usar AWS Mensajería automática para el usuario final Para enviar notificaciones push a su aplicación, primero debe obtener un Cuenta de AWS con IAM permisos suficientes. Este Cuenta de AWS también se puede utilizar para otros servicios en el AWS ecosistema.

Temas



## Inscríbase en un Cuenta de AWS

Si no tienes un Cuenta de AWS, complete los pasos siguientes para crear uno.

Para suscribirte a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en una Cuenta de AWS, un Usuario raíz de la cuenta de AWSse crea. El usuario root tiene acceso a todos Servicios de AWS y los recursos de la cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

# Creación de un usuario con acceso administrativo

Después de suscribirse a una Cuenta de AWS, asegure su Usuario raíz de la cuenta de AWS, habilitar AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión en la AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su Cuenta de AWS dirección de correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con un usuario root, consulte Iniciar sesión como usuario root en AWS Sign-In Guía del usuario.
2. Activa la autenticación multifactorial (MFA) para tu usuario root.

   Para obtener instrucciones, consulte Habilitar un MFA dispositivo virtual para su Cuenta de AWS usuario root (consola) en la Guía IAM del usuario.

Creación de un usuario con acceso administrativo

1. Habilite IAM Identity Center.

   Para obtener instrucciones, consulte Habilitar AWS IAM Identity Center en la AWS IAM Identity Center Guía del usuario.
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre el uso de Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la AWS IAM Identity Center Guía del usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con su usuario de IAM Identity Center, utilice el inicio de sesión URL que se envió a su dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario de IAM Identity Center, consulte Iniciar sesión en AWS acceda al portal en el AWS Sign-In Guía del usuario.

### Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos con privilegios mínimos.

   Para obtener instrucciones, consulte Crear un conjunto de permisos en AWS IAM Identity Center Guía del usuario.

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para obtener instrucciones, consulte Añadir grupos en AWS IAM Identity Center Guía del usuario.

# Cómo empezar con AWS End User Messaging Push

Para configurar AWS End User Messaging Push para que pueda enviar notificaciones push a sus aplicaciones, primero debe proporcionar las credenciales que autorizan a AWS End User Messaging Push a enviar mensajes a su aplicación. Las credenciales que se proporcionan dependen del sistema de notificaciones de inserción utilizado:

- Para obtener información sobre las credenciales del servicio de notificaciones push de Apple (APN), consulte Obtener una clave de cifrado y un identificador de clave de Apple y Obtener un certificado de proveedor de Apple en la documentación para desarrolladores de Apple.
- Para obtener las credenciales de Firebase Cloud Messaging (FCM), puedes obtenerlas a través de la consola de Firebase (consulta Firebase Cloud Messaging).
- Para ver las credenciales de Baidu, consulta Baidu.
- Para ver las credenciales de Amazon Device Messaging (ADM), consulte Obtener credenciales.

# Crear una aplicación y habilitar los canales push

Antes de poder utilizar AWS End User Messaging Push para enviar notificaciones push, primero tiene que crear una aplicación y habilitar el canal de notificaciones push.

## Contextual

Aplicación

Una aplicación es un contenedor de almacenamiento para todos sus ajustes de mensajería push para el usuario AWS final. La aplicación también almacena la configuración de los canales, campañas y viajes de Amazon Pinpoint.

Clave

Clave de firma privada utilizada por AWS End User Messaging Push para firmar criptográficamente los tokens de APNs autenticación. Puede obtener la clave de firma de su cuenta de desarrollador de Apple.

Si proporciona una clave de firma,AWS End User Messaging Push utiliza un token APNs para autenticarse en cada notificación push que envíe. Con tu clave de firma, puedes enviar notificaciones automáticas a entornos de APNs producción y entornos aislados.

A diferencia de los certificados, la clave de firma no vence. La clave solo se proporciona una vez y no necesita renovarla más adelante. Puede utilizar la misma clave de firma para varias aplicaciones. Para obtener más información, consulta Cómo comunicarse APNs mediante el uso de tokens de autenticación en la Ayuda de Xcode.

Certificate

Un TLS certificado que AWS End User Messaging Push utiliza para autenticarse APNs cuando envías notificaciones push. Un APNs certificado puede ser compatible con entornos de producción y de entorno aislado, o puede admitir solo el entorno de entorno aislado. Puede obtener el certificado de su cuenta de desarrollador de Apple.

Un certificado vence después de un año. Cuando esto suceda, debe crear un certificado nuevo y, a continuación, entregarlo a AWS End User Messaging Push para renovar las entregas de notificaciones push. Para obtener más información, consulta Cómo comunicarse APNs mediante un TLS certificado en la Ayuda de Xcode.

# Requisitos previos

Antes de poder utilizar cualquier canal push, necesita credenciales válidas para el servicio push. Para obtener más información sobre cómo obtener credenciales, consulteCómo empezar con AWS End User Messaging Push.

# Procedimiento

Siga estas instrucciones para crear una aplicación y habilitar cualquiera de los canales push. Para completar este procedimiento, solo tiene que introducir el nombre de la aplicación. Puede activar o desactivar cualquiera de los canales push más adelante.

1. Abra la consola push de mensajería para el usuario AWS final en https://console.aws.amazon.com/push-notifications/.
2. Elija Crear aplicación.
3. En Nombre de la aplicación, introduzca el nombre de la aplicación.
4. (Opcional) Siga este paso opcional para activar el servicio de notificaciones push de Apple (APNs).

   a. Para el servicio de notificaciones push de Apple (APNs), selecciona Activar.

   b. Para el tipo de autenticación predeterminado, elige una de las siguientes opciones:

      i. Si eliges Credenciales clave, proporciona la siguiente información de tu cuenta de desarrollador de Apple. AWS End User Messaging Push requiere esta información para crear los tokens de autenticación.

         - ID de clave: el ID asignado a la clave de firma.
         - Identificador de paquete: el ID que está asignado a la aplicación de iOS.
         - Identificador de equipo: el ID que está asignado al equipo de la cuenta de desarrollador de Apple.
         - Clave de autenticación: el archivo .p8 que descarga desde la cuenta de desarrollador de Apple al crear una clave de autenticación.

      ii. Si elige Certificate credentials (Credenciales de certificado), facilite la siguiente información:

         - SSLcertificado: el archivo.p12 de su TLS certificado.

- Contraseña de certificado: si ha asignado una contraseña al certificado, ingrésela aquí.
- Tipo de certificado: seleccione el tipo de certificado que se va a utilizar.

5. (Opcional) Sigue este paso opcional para habilitar Firebase Cloud Messaging (). FCM
   a. Para Firebase Cloud Messaging (FCM), selecciona Activar.
   b. Para el tipo de autenticación predeterminado, elige una de las siguientes opciones:
      i. Para las credenciales de token (recomendadas), selecciona Elegir archivos y, a continuación, elige tu JSON archivo de servicio.
      ii. En el caso de las credenciales clave, introduce tu clave en APIclave.
6. (Opcional) Sigue este paso opcional para activar Baidu Cloud Push.
   a. Para Baidu Cloud Push, selecciona Activar.
   b. Para APIclave, introduce tu API clave.
   c. En Clave secreta, introduzca su clave secreta.
7. (Opcional) Sigue este paso opcional para activar Amazon Device Messaging.
   a. Para Amazon Device Messaging, selecciona Activar.
   b. Para el ID de cliente, introduce tu ID de cliente.
   c. En Secreto de cliente, introduzca su secreto de cliente.
8. Elija Crear aplicación.

# Desactivación de los canales push

Siga estas instrucciones para desactivar cualquiera de los canales de inserción.

1. Abra la consola push de mensajería para el usuario AWS final en https://console.aws.amazon.com/push-notifications/.
2. Elija la aplicación que contiene sus credenciales push.
3. (Opcional) Para el servicio de notificaciones push de Apple (APNs), desactive Activar.
4. (Opcional) Para Firebase Cloud Messaging (FCM), desactive Activar.
5. (Opcional) Para Baidu Cloud Push, desactive Activar.
6. (Opcional) Para Amazon Device Messaging, desactive Activar.
7. Elija Guardar cambios.

# Envío de un mensaje

La mensajería push para el usuario AWS final API puede enviar notificaciones push transaccionales a identificadores de dispositivos específicos. Esta sección contiene ejemplos de códigos completos que puede utilizar para enviar notificaciones push a través de la función push de mensajería para el usuario AWS final API mediante un. AWS SDK

Puede utilizar estos ejemplos para enviar notificaciones push a través de cualquier servicio de notificaciones push compatible con AWS End User Messaging Push. Actualmente, AWS End User Messaging Push es compatible con los siguientes canales: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push y Amazon Device Messaging (ADM).

Para ver más ejemplos de código sobre puntos finales, segmentos y canales, consulta Ejemplos de código.

> **Note**
>
> Cuando envíes notificaciones push a través del servicio Firebase Cloud Messaging (FCM), usa el nombre del servicio `GCM` en la llamada al servicio push de mensajería para el usuario AWS final. API Google suspendió el servicio Google Cloud Messaging (GCM) el 10 de abril de 2018. Sin embargo, el servicio push de mensajería para el usuario AWS final API utiliza el nombre del `GCM` servicio para los mensajes que envía a través del FCM servicio a fin de mantener la compatibilidad con el API código que se escribió antes de la interrupción del GCM servicio.

GCM (AWS CLI)

En el siguiente ejemplo, se utilizan send-messages para enviar una notificación GCM push con el. AWS CLI Reemplazar *token* con el token único del dispositivo y *611e3e3cdd47474c9c1399a50example* con el identificador de su aplicación.

```
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"
      }
  }
}
```

En el siguiente ejemplo, se utilizan send-messages para enviar una notificación GCM push, utilizando todas las claves antiguas, con la. AWS CLI Reemplazar *token* con el token único del dispositivo y *611e3e3cdd47474c9c1399a50example* con el identificador de su aplicación.

```
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
```

En el siguiente ejemplo, se utilizan send-messages para enviar una notificación GCM push con una carga útil de FCMv1 mensajes mediante el. AWS CLI Reemplazar *token* con el token único del dispositivo y *611e3e3cdd47474c9c1399a50example* con el identificador de su aplicación.

```
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
```

si se utiliza el `ImageUrl` campo paraGCM, pinpoint envía el campo como notificación de datos, con la clave puesta`pinpoint.notification.imageUrl`, lo que puede impedir que la imagen se reproduzca de forma predeterminada. Utilice RawContent o añada el manejo de las claves de datos, por ejemplo, integrando su aplicación con AWS Amplify ellas.

Safari (AWS CLI)

Puedes usar AWS End User Messaging Push para enviar mensajes a ordenadores macOS que utilicen el navegador web Safari de Apple. Para enviar un mensaje al navegador Safari, debe especificar el contenido sin procesar del mensaje e incluir un atributo específico en la carga del mensaje. Para ello, puede crear una plantilla de notificaciones push con una carga útil de mensajes sin procesar o especificando el contenido del mensaje sin procesar directamente en un mensaje de campaña, en la Guía del usuario de Amazon Pinpoint.

> **Note**
>
> Este atributo especial es obligatorio para el envío a ordenadores portátiles y de sobremesa macOS que utilizan el navegador web Safari. No es obligatorio para enviar a dispositivos iOS como iPhones yiPads.

Para enviar un mensaje a los navegadores web Safari, debe especificar la carga del mensaje sin procesar. La carga del mensaje sin procesar debe incluir una matriz `url-args` dentro del objeto

`aps`. La matriz `url-args` es necesaria para enviar notificaciones de inserción al navegador web Safari. Sin embargo, es aceptable que la matriz contenga un único elemento vacío.

En el siguiente ejemplo, se utiliza el comando enviar mensajes para enviar una notificación al navegador web Safari con el. AWS CLI Reemplazar *token* con el token único del dispositivo y *611e3e3cdd47474c9c1399a50example* con el identificador de su aplicación.

```
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
```

Para obtener más información sobre las notificaciones de inserción de Safari, consulte Configuración de las notificaciones de inserción de Safari en el sitio web para desarrolladores de Apple.

APNS (AWS CLI)

En el siguiente ejemplo, se utilizan send-messages para enviar una notificación APNS push con el. AWS CLI Reemplazar *token* con el token único del dispositivo, *611e3e3cdd47474c9c1399a50example* con el identificador de su aplicación, y *GAME_INVITATION* con un identificador único.

```
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)

Utilice este ejemplo para enviar notificaciones push mediante el AWS SDK formulario JavaScript de Node.js. En este ejemplo se supone que ya ha instalado y configurado el SDK formulario JavaScript en Node.js.

En este ejemplo se supone que está utilizando un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario de existente. Para obtener más información, consulte Configurar las credenciales en el formulario JavaScript en la Guía AWS SDK para desarrolladores de Node.js.

```
'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

Utilice este ejemplo para enviar notificaciones push mediante el AWS SDK for Python (Boto3). En este ejemplo se supone que ya ha instalado y configurado SDK para Python (Boto3).

En este ejemplo se supone que está utilizando un archivo de credenciales compartidas para especificar la clave de acceso y la clave de acceso secreta para un usuario de existente. Para obtener más información, consulte Credenciales en la AWS SDKreferencia para Python (Boto3). API

```
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()
```

# Recursos adicionales de

- Para obtener más información sobre las plantillas de canales push, consulte Creación de plantillas de notificaciones push en la Guía del usuario de Amazon Pinpoint.

# Recibir notificaciones push en tu aplicación

Los siguientes temas describen cómo modificar tu aplicación Swift, Android, React Native o Flutter para que reciba notificaciones push.

Temas



## Configuración de notificaciones de inserción rápidas

Las notificaciones push para las aplicaciones iOS se envían mediante el servicio de notificaciones push de Apple (APNs). Para poder enviar notificaciones de inserción a dispositivos iOS, debe crear un ID de aplicación en el portal de Apple Developer y crear los certificados necesarios. Encontrarás más información sobre cómo completar estos pasos en Configurar los servicios de notificaciones push en la documentación de AWS Amplify.

### ¿Trabajando con fichas APNs

Como práctica recomendada, debe desarrollar la aplicación para que los tokens de dispositivo de los clientes se vuelvan a generar cuando se vuelva a instalar la aplicación.

Si un destinatario actualiza su dispositivo a una nueva versión principal de iOS (por ejemplo, de iOS 12 a iOS 13) y, posteriormente, vuelve a instalar la aplicación, la aplicación genera un nuevo token. Si la aplicación no actualiza el token, se utiliza el token más antiguo para enviar la notificación. Como resultado, el servicio de notificaciones push de Apple (APNs) rechaza la notificación porque el token ahora no es válido. Cuando intentes enviar la notificación, recibirás un mensaje de notificación de error de parte de élAPNs.

# Configuración de las notificaciones push de Android

Las notificaciones push para las aplicaciones de Android se envían mediante Firebase Cloud Messaging (FCM), que reemplaza a Google Cloud Messaging (GCM). Para poder enviar notificaciones push a dispositivos Android, debes obtener FCM las credenciales. Puede utilizar las credenciales para crear un proyecto de Android y lanzar una aplicación de muestra que pueda recibir notificaciones push. Puedes encontrar más información sobre cómo completar estos pasos en la sección de notificaciones push de la documentación de AWS Amplify.

# Configuración de las notificaciones de inserción de Flutter

Las notificaciones push para las aplicaciones de Flutter se envían mediante Firebase Cloud Messaging (FCM) para Android y para APNs iOS. Puede encontrar más información acerca de cómo llevar a cabo estos pasos en la sección de notificaciones de inserción de la documentación de AWS Amplify Flutter.

# Configuración de las notificaciones de inserción de React Native

Las notificaciones push para las aplicaciones de React Native se envían mediante Firebase Cloud Messaging (FCM) para Android y APNs para iOS. Puedes encontrar más información sobre cómo completar estos pasos en la sección Notificaciones push de la documentación de AWS Amplify JavaScript.

# Cree una aplicación en AWS End User Messaging Push

Para empezar a enviar notificaciones push en AWS End User Messaging Push, debe crear una aplicación. A continuación, hay que proporcionar las credenciales adecuadas para habilitar los canales de notificaciones de inserción que se desea utilizar.

Puede crear nuevas aplicaciones y configurar canales de notificaciones push mediante la consola push de mensajería automática para el usuario AWS final. Para obtener más información, consulte Crear una aplicación y habilitar los canales push.

También puede crear y configurar una aplicación mediante las APIteclas AWS SDK, an o AWS Command Line Interface(AWS CLI). Para crear una aplicación, utilice el `Apps` recurso. Para configurar canales de notificaciones de inserción, utilice los siguientes recursos:

- APNscanal para enviar mensajes a los usuarios de dispositivos iOS mediante el servicio de notificaciones push de Apple.
- ADMcanal para enviar mensajes a los usuarios de los dispositivos Amazon Kindle Fire.
- Canal de Baidu para enviar mensajes a los usuarios de Baidu.
- GCMcanal para enviar mensajes a dispositivos Android mediante Firebase Cloud Messaging (FCM), que reemplaza a Google Cloud Messaging (GCM).

## Gestión de notificaciones push

Una vez que hayas obtenido las credenciales necesarias para enviar notificaciones push, puedes actualizar tu aplicación para que pueda recibirlas. Para obtener más información, consulta las notificaciones push: introducción en la documentación. AWS Amplify

# Eliminación de una aplicación

Este procedimiento elimina la aplicación de su cuenta y todos los recursos de la aplicación.

## Contextual

Aplicación

Una aplicación es un contenedor de almacenamiento para todos sus ajustes de mensajería push para el usuario AWS final. La aplicación también almacena la configuración de los canales, campañas y viajes de Amazon Pinpoint.

## Procedimiento

1. Abra la consola push de mensajería para el usuario AWS final en https://console.aws.amazon.com/push-notifications/.
2. Elija una aplicación y, a continuación, elija Eliminar.
3. En la ventana Eliminar aplicación, introduzca **`delete`** y, a continuación, seleccione Eliminar.

> **Important**
>
> También se eliminan todos los canales, campañas, viajes o segmentos de Amazon Pinpoint.

# Prácticas recomendadas

Incluso cuando tenga en cuenta el mayor interés para sus clientes, es posible que encuentre situaciones que afecten a la capacidad de entrega de sus mensajes. Las siguientes secciones contienen recomendaciones para ayudarle a garantizar que las comunicaciones de inserción lleguen al público deseado.

## Envío de un gran volumen de notificaciones de inserción

Antes de enviar un gran volumen de notificaciones push, asegúrate de que tu cuenta esté configurada para cumplir tus requisitos de rendimiento. De forma predeterminada, todas las cuentas están configuradas para enviar 25 000 mensajes por segundo. Si tiene la necesidad de poder enviar más de 25 000 mensajes en un segundo, solicite un aumento de cuota. Para obtener más información, consulte Cuotas para el envío de mensajes a los usuarios AWS finales.

Asegúrese de que su cuenta esté configurada correctamente con las credenciales de cada uno de los proveedores de notificaciones push que vaya a utilizar, como FCM oAPNs.

Por último, diseñe una forma de gestionar las excepciones. Cada servicio de notificaciones de inserción proporciona diferentes mensajes de excepción. En el caso de los envíos transaccionales, recibes un código de estado principal de 200 para la API llamada, y un código de estado de 400 por punto final (error permanente) si se determina que el token de plataforma correspondiente (por ejemploFCM) o el certificado (por ejemploAPN) no son válidos durante el envío de los mensajes.

# Seguridad en AWS Envío de mensajes para el usuario final

Seguridad en la nube en AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS y tú. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que se ejecuta AWS servicios en el Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte del AWS Programas de cumplimiento . Para obtener información sobre los programas de cumplimiento que se aplican a AWS Mensajes push para usuarios finales, consulte AWS Servicios incluidos en el ámbito de aplicación del programa de cumplimiento .
- Seguridad en la nube: su responsabilidad viene determinada por la AWS servicio que utiliza. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS Envío push de mensajes para el usuario final. En los temas siguientes se muestra cómo configurar AWS Envíe mensajes para el usuario final a fin de cumplir sus objetivos de seguridad y conformidad. También aprenderá a usar otros AWS servicios que le ayudan a monitorear y proteger sus AWS Recursos push de mensajería para usuarios finales.

Temas

# Protección de datos en AWS Envío push de mensajería para el usuario final

La AWS modelo de responsabilidad compartida modelo se aplica a la protección de datos en AWS Envío push de mensajes para el usuario final. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido que está alojado en esta infraestructura. También es responsable de las tareas de configuración y administración de la seguridad del Servicios de AWS que utilices. Para obtener más información sobre la privacidad de los datos, consulte la sección Privacidad de datos FAQ. Para obtener información sobre la protección de datos en Europa, consulte la AWS Modelo de responsabilidad compartida y entrada de GDPR blog sobre AWS Blog de seguridad.

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- UtiliceSSL/TLSpara comunicarse con AWS recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Trabajar con CloudTrail senderos en la AWS CloudTrail Guía del usuario.
- Uso AWS soluciones de cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder AWS a través de una interfaz de línea de comandos oAPI, utilice un FIPS punto final. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la Norma Federal de Procesamiento de Información (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS Mensajería Push para el usuario final u otra Servicios de AWS utilizando la consolaAPI, AWS CLI, o AWS SDKs. Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya información sobre las credenciales URL para validar su solicitud a ese servidor.

## Cifrado de datos

AWS Los datos push de mensajería de usuario final se cifran tanto en tránsito como en reposo. Cuando envía datos a AWS End User Messaging Push, cifra los datos a medida que los recibe y los almacena. Cuando se recuperan datos de AWS La mensajería push para el usuario final le transmite los datos mediante los protocolos de seguridad actuales.

### Cifrado en reposo

AWS End User Messaging Push cifra todos los datos que almacena para usted. Esto incluye los datos de configuración, los datos de usuario y punto final, los datos de análisis y cualquier dato que añada o importe AWS Envío push de mensajes para el usuario final. Para cifrar sus datos, AWS La mensajería Push para el usuario final utiliza mensajes internos AWS Key Management Service (AWS KMS) claves que el servicio posee y mantiene en su nombre. Rotamos estas claves periódicamente. Para obtener más información AWS KMS, consulte la AWS Key Management Service Guía para desarrolladores.

## Cifrado en tránsito

AWS End User Messaging Push utiliza HTTPS Transport Layer Security (TLS) 1.2 o una versión posterior para comunicarse con sus clientes y aplicaciones. Para comunicarse con otros AWS servicios, AWS Usos de End User Messaging Push HTTPS y TLS 1.2. Además, al crear y administrar AWS Recursos push de mensajería para el usuario final mediante la consola, un AWS SDK, o el AWS Command Line Interface, todas las comunicaciones se protegen mediante HTTPS y TLS 1.2.

## Administración de claves

Para cifrar su AWS Datos push de mensajería para el usuario final, AWS La mensajería push para el usuario final utiliza mensajes internos AWS KMS claves que el servicio posee y mantiene en su nombre. Rotamos estas claves periódicamente. No puedes aprovisionar ni usar las tuyas AWS KMS

u otras claves para cifrar los datos que almacenas AWS Envío push de mensajes para el usuario final.

## Privacidad del tráfico entre redes

La privacidad del tráfico entre redes se refiere a proteger las conexiones y el tráfico entre AWS La mensajería push para el usuario final y sus clientes y aplicaciones locales, y entre AWS Mensajería Push para el usuario final y otros AWS recursos en el mismo AWS Región. Las siguientes características y prácticas pueden ayudarle a garantizar la privacidad del tráfico entre redes para AWS Envío push de mensajes para el usuario final.

### Tráfico entre AWS Clientes y aplicaciones de mensajería push para usuarios finales y locales

Para establecer una conexión privada entre AWS La mensajería push para el usuario final y los clientes y aplicaciones de su red local, puede utilizar AWS Direct Connect. Esto le permite vincular su red a un AWS Direct Connect ubicación mediante un cable Ethernet de fibra óptica estándar. Un extremo del cable se conecta al enrutador. El otro extremo está conectado a un AWS Direct Connect router. Para obtener más información, consulte ¿Qué es AWS Direct Connect? en el AWS Direct Connect Guía del usuario.

Para ayudar a proteger el acceso a AWS Se ha publicado Push Through de mensajería para el usuario finalAPIs, le recomendamos que cumpla con AWS Requisitos de mensajería automática para el usuario final para API las llamadas. AWS La mensajería push para el usuario final requiere que los clientes utilicen Transport Layer Security (TLS) 1.2 o una versión posterior. Los clientes también deben admitir conjuntos de cifrado con total confidencialidad (PFS), como Ephemeral Diffie-Hellman () o Elliptic Curve Diffie-Hellman Ephemeral (DHE). ECDHE La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben firmarse con un identificador de clave de acceso y una clave de acceso secreta asociada a un AWS Identity and Access Management (IAM) principal para su AWS account. Como alternativa, puede utilizar el AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar las solicitudes.

### Tráfico entre AWS Mensajería Push para el usuario final y otros AWS resources

Para proteger las comunicaciones entre AWS Mensajería Push para el usuario final y otros AWS recursos en el mismo AWS Región, AWS End User Messaging Push utiliza HTTPS y TLS 1.2 de forma predeterminada.

# Administración de identidad y acceso para AWS Envío push de mensajería para el usuario final

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS recursos. IAMlos administradores controlan quién puede autenticarse (iniciar sesión) y quién está autorizado (tiene permisos) para usarlos AWS Recursos push de mensajería para usuarios finales. IAMes un Servicio de AWS que puede utilizar sin coste adicional.

Temas



## Público

¿Cómo se usa AWS Identity and Access Management (IAM) difiere según el trabajo que realices en AWS Envío push de mensajes para el usuario final.

Usuario del servicio: si utiliza el AWS El servicio push de mensajería para el usuario final le proporcionará las credenciales y los permisos que necesita para realizar su trabajo. A medida que utilice más AWS Para realizar su trabajo, es posible que necesite permisos adicionales gracias a las funciones push de mensajería para el usuario final. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una función en AWS Mensajes push para usuarios finales, consulteResolución de problemas AWS Identidad y acceso push de mensajería para el usuario final.

Administrador del servicio: si está a cargo de AWS Los recursos push de mensajería para usuarios finales de su empresa, probablemente tenga acceso total a AWS Envío push de mensajes para el usuario final. Es su trabajo determinar cuál AWS Funciones y recursos de End User Messaging Push a los que deben acceder los usuarios del servicio. A continuación, debe enviar solicitudes a su IAM

administrador para cambiar los permisos de los usuarios del servicio. Revise la información de esta página para comprender los conceptos básicos deIAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con AWS Mensajes push para usuarios finales, consulteCómo AWS End User Messaging Push funciona con IAM.

IAMadministrador: si es IAM administrador, puede que desee obtener más información sobre cómo redactar políticas para administrar el acceso a AWS Envío push de mensajes para el usuario final. Para ver un ejemplo AWS Políticas de mensajería push para usuarios finales basadas en la identidad que puede utilizarIAM, consulte. Ejemplos de políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

## Autenticación con identidades

La autenticación es la forma de iniciar sesión en AWS utilizando tus credenciales de identidad. Debe estar autenticado (haber iniciado sesión en AWS) como Usuario raíz de la cuenta de AWS, como IAM usuario o asumiendo un IAM rol.

Puede iniciar sesión en AWS como identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios de (IAMIdentity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, el administrador configuró previamente la federación de identidades mediante roles. IAM Cuando accedes AWS al usar la federación, está asumiendo un rol de manera indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en AWS Management Console o el AWS portal de acceso. Para obtener más información sobre cómo iniciar sesión en AWS, consulta Cómo iniciar sesión en tu Cuenta de AWS en la AWS Sign-In Guía del usuario.

Si accedes AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no usa AWS herramientas, debe firmar las solicitudes usted mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar AWS APIsolicitudes en la Guía IAM del usuario.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo: AWS recomienda que utilice la autenticación multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactorial en la AWS IAM Identity Center Guía del usuario y Uso de la autenticación multifactorial () MFA en AWS en la Guía del usuario de IAM.

## Cuenta de AWS usuario raíz

Al crear un Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todas Servicios de AWS y los recursos de la cuenta. Esta identidad se denomina Cuenta de AWS usuario root y se accede a él iniciando sesión con la dirección de correo electrónico y la contraseña que utilizó para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de tareas que requieren que inicie sesión como usuario root, consulte Tareas que requieren credenciales de usuario root en la Guía del IAM usuario.

## Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web, el AWS Directory Service, el directorio del Centro de identidades o cualquier usuario que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para la administración centralizada del acceso, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todos sus Cuentas de AWS y aplicaciones. Para obtener información sobre IAM Identity Center, consulte ¿Qué es IAM Identity Center? en el AWS IAM Identity Center Guía del usuario.

## Usuarios y grupos de IAM

Un IAMusuario es una identidad dentro de tu Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte Rotar las claves de acceso con regularidad para los casos de uso que requieran credenciales de larga duración en la Guía del IAM usuario.

Un IAMgrupo es una identidad que especifica un conjunto de IAM usuarios. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar IAM los recursos.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Cuándo crear un IAM usuario (en lugar de un rol) en la Guía del IAM usuario.

## IAMroles

Un IAMrol es una identidad dentro de tu Cuenta de AWS que tiene permisos específicos. Es similar a un IAM usuario, pero no está asociado a una persona específica. Puede asumir temporalmente un IAM rol en el AWS Management Console cambiando de rol. Puede asumir un rol llamando a un AWS CLI o AWS APIoperación o mediante una operación personalizadaURL. Para obtener más información sobre los métodos de uso de roles, consulte Uso de IAM roles en la Guía del IAM usuario.

IAMlos roles con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información sobre los roles para la federación, consulte Creación de un rol para un proveedor de identidad externo en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en. IAM Para obtener información sobre los conjuntos de permisos, consulte los conjuntos de permisos en la AWS IAM Identity Center Guía del usuario.
- Permisos de IAM usuario temporales: un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- Acceso multicuenta: puedes usar un IAM rol para permitir que alguien (un responsable de confianza) de una cuenta diferente acceda a los recursos de tu cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puede adjuntar una política directamente a un recurso (en lugar de utilizar un rol como proxy). Para saber

la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los recursos entre cuentas IAM en la Guía del IAM usuario.

- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un IAM usuario o un rol para realizar acciones en AWS, se le considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del director que llama a un Servicio de AWS, combinado con la solicitud Servicio de AWS para realizar solicitudes a los servicios intermedios. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completar. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar las sesiones de acceso.
  - Función de servicio: una función de servicio es una IAMfunción que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
  - Función vinculada a un servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un Servicio de AWS. El servicio puede asumir la función de realizar una acción en su nombre. Los roles vinculados al servicio aparecen en su Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver, pero no editar, los permisos de las funciones vinculadas al servicio.
- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y se están creando AWS CLI o AWS APIsolicitudes. Esto es preferible a almacenar las claves de acceso dentro de la EC2 instancia. Para asignar un AWS Un rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia que se adjunte a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Uso de un IAM rol para conceder permisos a aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del IAM usuario.

Para saber si se deben usar IAM roles o IAM usuarios, consulte Cuándo crear un IAM rol (en lugar de un usuario) en la Guía del IAM usuario.

## Administración de acceso mediante políticas

Usted controla el acceso en AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto en AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan en AWS como JSON documentos. Para obtener más información sobre la estructura y el contenido de los documentos de JSON políticas, consulte Descripción general de JSON las políticas en la Guía del IAM usuario.

Los administradores pueden utilizar AWS JSONpolíticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

IAMlas políticas definen los permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre su función en AWS Management Console, el AWS CLI, o el AWS API.

### Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario. IAM

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y funciones de su Cuenta de AWS. Las políticas gestionadas incluyen AWS las políticas gestionadas y las políticas gestionadas por el cliente. Para saber cómo elegir entre una política

gestionada o una política en línea, consulte Elegir entre políticas gestionadas y políticas integradas en la Guía del IAMusuario.

### Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar AWS políticas gestionadas desde una política basada IAM en recursos.

### Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3, AWS WAF, y Amazon VPC son ejemplos de servicios que admitenACLs. Para obtener más informaciónACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

### Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAMusuario o rol). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites

de los permisos, consulte los límites de los permisos para IAM las entidades en la Guía del IAMusuario.

- Políticas de control de servicios (SCPs): SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU) en AWS Organizations. AWS Organizations es un servicio para agrupar y administrar de forma centralizada múltiples Cuentas de AWS que es propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar las políticas de control de servicios (SCPs) a cualquiera de tus cuentas o a todas ellas. SCPLimita los permisos de las entidades en las cuentas de los miembros, incluidas todas Usuario raíz de la cuenta de AWS. Para obtener más información acerca de OrganizationsSCPs, consulte Políticas de control de servicios en AWS Organizations Guía del usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte las políticas de sesión en la Guía del IAM usuario.

### Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información sobre cómo AWS determina si se permite una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del IAM usuario.

## Cómo AWS End User Messaging Push funciona con IAM

Antes de usarlo IAM para administrar el acceso a AWS End User Messaging Push, conozca qué IAM funciones están disponibles para su uso con AWS Envío push de mensajes para el usuario final.

IAMfunciones que puede utilizar con AWS Envío push de mensajería para el usuario final

| IAMfunción | AWS Soporte push de mensajería para el usuario final |
| --- | --- |
| Políticas basadas en identidades | Sí |

| IAMfunción | AWS Soporte push de mensajería para el usuario final |
| --- | --- |
| Políticas basadas en recursos | Sí |
| Acciones de políticas | Sí |
| Recursos de políticas | Sí |
| Claves de condición de política | Sí |
| ACLs | No |
| ABAC(etiquetas en las políticas) | Parcial |
| Credenciales temporales | Sí |
| Permisos de entidades principales | Sí |
| Roles de servicio | Sí |
| Roles vinculados al servicio | No |

Para obtener una visión general de cómo AWS Mensajería para el usuario final, Push y otros AWS los servicios funcionan con la mayoría de IAM las funciones, consulte AWS servicios con los que funcionan IAM en la Guía IAM del usuario.

## Políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario. IAM

Con las políticas IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones.

No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la referencia sobre los elementos de la IAM JSON política en la Guía del IAMusuario.

Ejemplos de políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

Para ver ejemplos de AWS Políticas de mensajería push para usuarios finales basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

## Políticas basadas en recursos dentro AWS Envío push de mensajería para el usuario final

Compatibilidad con las políticas basadas en recursos: sí

Las políticas basadas en recursos son documentos JSON de políticas que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Para habilitar el acceso entre cuentas, puede especificar una cuenta completa o IAM entidades de otra cuenta como principales en una política basada en recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el IAM administrador de la cuenta de confianza también debe conceder permiso a la entidad principal (usuario o rol) para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte el tema Acceso a recursos entre cuentas IAM en la Guía del IAM usuario.

## Acciones políticas para AWS Envío de mensajes para el usuario final

Compatibilidad con las acciones de política: sí

Los administradores pueden usar AWS JSONpolíticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que las asociadas AWS APIoperación. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de AWS Acciones push de mensajería para el usuario final, consulte Acciones definidas por AWS Inserte la referencia de autorización del servicio para enviar mensajes al usuario final.

Acciones políticas en AWS En End User Messaging Push, utilice el siguiente prefijo antes de la acción:

```
mobiletargeting
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de AWS Políticas de mensajería push para usuarios finales basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

## Recursos de políticas para AWS Envío de mensajes para el usuario final

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar AWS JSONpolíticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso mediante su nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de AWS Los tipos de recursos push de mensajería para usuarios finales y sus tipos de recursosARNs, consulte los recursos definidos por AWS Inserte la referencia de autorización del servicio para la mensajería del usuario final. Para saber con qué acciones puede especificar cada recurso, consulte Acciones definidas por ARN AWS Envío push de mensajes para el usuario final.

Para ver ejemplos de AWS Políticas de mensajería push para usuarios finales basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

## Claves de condición de la política para AWS Envío push de mensajería para el usuario final

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar AWS JSONpolíticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios `Condition` elementos en una declaración o varias claves en un solo `Condition` elemento, AWS los evalúa mediante una `AND` operación lógica. Si especifica varios

valores para una sola clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder a un IAM usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de IAM usuario. Para obtener más información, consulte los elementos IAM de la política: variables y etiquetas en la Guía del IAM usuario.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas AWS claves de condición globales, consulte AWS claves de contexto de condiciones globales en la Guía IAM del usuario.

Para ver una lista de AWS Teclas de condición push de mensajería para el usuario final, consulte Claves de condición para AWS Introduzca la referencia de autorización del servicio en los mensajes del usuario final. Para saber con qué acciones y recursos puede utilizar una clave condicionada, consulte Acciones definidas por AWS Envío push de mensajes para el usuario final.

Para ver ejemplos de AWS Políticas de mensajería push para usuarios finales basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

## ACLsen AWS Envío push de mensajería para el usuario final

SoportesACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

## ABACcon AWS Envío push de mensajería para el usuario final

Soportes ABAC (etiquetas en las políticas): parciales

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchas AWS recursos. Etiquetar entidades y recursos es el primer paso deABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABACes útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información al respectoABAC, consulte ¿Qué es? ABAC en la Guía IAM del usuario. Para ver un tutorial con los pasos de configuraciónABAC, consulte Usar el control de acceso basado en atributos (ABAC) en la Guía del IAMusuario.

## Uso de credenciales temporales con AWS Envío push de mensajería para el usuario final

Compatibilidad con credenciales temporales: sí

Alguno Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluyendo qué Servicios de AWS trabaje con credenciales temporales, consulte Servicios de AWS que funcionan IAM en la Guía IAM del usuario.

Está utilizando credenciales temporales si inicia sesión en el AWS Management Console utilizando cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accede AWS mediante el enlace de inicio de sesión único (SSO) de su empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de rol, consulte Cambiar a un rol (consola) en la Guía del IAMusuario.

Puede crear credenciales temporales manualmente mediante el AWS CLI o AWS API. A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

## Permisos principales entre servicios para AWS Envío push de mensajería para el usuario final

Admite sesiones de acceso directo (FAS): Sí

Cuando utiliza un IAM usuario o un rol para realizar acciones en AWS, se le considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del director que llama a un Servicio de AWS, combinado con la solicitud Servicio de AWS para realizar solicitudes a los servicios intermedios. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completar. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar las sesiones de acceso.

## Funciones de servicio para AWS Envío push de mensajería para el usuario final

Compatibilidad con roles de servicio: sí

Una función de servicio es una IAMfunción que asume un servicio para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Es posible que el cambio de los permisos de un rol de servicio no funcione AWS Función push de mensajería para el usuario final. Edite las funciones de servicio solo cuando AWS End User Messaging Push proporciona instrucciones para hacerlo.

## Funciones vinculadas al servicio para AWS Envío push de mensajería para el usuario final

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir la función de realizar una acción en su nombre. Los roles vinculados al servicio aparecen en su Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver, pero no editar, los permisos de las funciones vinculadas al servicio.

Para obtener más información sobre la creación o la administración de funciones vinculadas a un servicio, consulte AWS servicios con los que funcionan. IAM Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

# Ejemplos de políticas basadas en la identidad para AWS Envío push de mensajería para el usuario final

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar AWS Recursos de mensajería push para usuarios finales. Tampoco pueden realizar tareas mediante el AWS Management Console, AWS Command Line Interface (AWS CLI), o AWS API. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte Creación de IAM políticas en la Guía del IAMusuario.

Para obtener información detallada sobre las acciones y los tipos de recursos definidos por AWS La mensajería push ARNs para el usuario final, incluido el formato de cada uno de los tipos de recursos, consulte las claves de acciones, recursos y condición para AWS Inserte la mensajería del usuario final en la referencia de autorización del servicio.

Temas

- Prácticas recomendadas sobre las políticas
- Uso de AWS Consola push de mensajería para el usuario final
- Cómo permitir a los usuarios consultar sus propios permisos

## Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear, acceder o eliminar AWS Recursos push de mensajería para usuarios finales en su cuenta. Estas acciones pueden suponer costes para su Cuenta de AWS. Al crear o editar políticas basadas en la identidad, siga estas directrices y recomendaciones:

- Comience con AWS políticas gestionadas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice la AWS políticas gestionadas que conceden permisos para muchos casos de uso habituales. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo AWS políticas gestionadas por el cliente que sean específicas para sus casos de uso. Para obtener más

información, consulte AWS políticas gestionadas o AWS políticas gestionadas para las funciones laborales en la Guía IAM del usuario.

- Aplique permisos con privilegios mínimos: cuando establezca permisos con IAM políticas, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Para obtener más información sobre cómo IAM aplicar permisos, consulte Políticas y permisos IAM en la IAM Guía del usuario.
- Utilice las condiciones en IAM las políticas para restringir aún más el acceso: puede añadir una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse medianteSSL. También puede utilizar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de un procedimiento específico Servicio de AWS, como, por ejemplo, AWS CloudFormation. Para obtener más información, consulte los elementos IAM JSON de la política: Condición en la Guía del IAM usuario.
- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAMAccess Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarle a crear políticas seguras y funcionales. Para obtener más información, consulte la validación de políticas de IAM Access Analyzer en la Guía del IAM usuario.
- Requerir autenticación multifactorial (MFA): si tiene un escenario que requiere IAM usuarios o un usuario raíz en su Cuenta de AWS, actívala MFA para mayor seguridad. Para solicitarlo MFA cuando se cancelen API las operaciones, añada MFA condiciones a sus políticas. Para obtener más información, consulte Configuración del API acceso MFA protegido en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadasIAM, consulte las prácticas recomendadas de seguridad IAM en la Guía del IAM usuario.

## Uso de AWS Consola push de mensajería para el usuario final

Para acceder a la AWS En la consola push de mensajería para el usuario final, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre AWS Recursos push de mensajería para usuarios finales en su Cuenta de AWS. Si creas una política basada en la identidad que sea más restrictiva que los permisos mínimos requeridos, la consola no funcionará según lo previsto para las entidades (usuarios o roles) que cuenten con esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realizan llamadas al AWS CLI o el AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la AWS Consola push de mensajería para el usuario final, conecte también el `AWSEndUserMessaging` AWS política gestionada a las entidades. Para obtener más información, consulte Añadir permisos a un usuario en la Guía del IAM usuario.

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

## Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo se muestra cómo se puede crear una política que permita a IAM los usuarios ver las políticas integradas y administradas asociadas a su identidad de usuario. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante el AWS CLI o 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": "*"
        }
    ]
}
```

# Resolución de problemas AWS Identidad y acceso push de mensajería para el usuario final

Utilice la siguiente información para ayudarle a diagnosticar y solucionar problemas comunes que puedan surgir al trabajar con AWS Mensajes push para el usuario final yIAM.

Temas



## No estoy autorizado a realizar ninguna acción en AWS Envío push de mensajería para el usuario final

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

El siguiente ejemplo de error se produce cuando el `mateojackson` IAM usuario intenta usar la consola para ver detalles sobre un *`my-example-widget`* recurso ficticio pero no tiene los `mobiletargeting:`*`GetWidget`* permisos ficticios.

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

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `mobiletargeting:`*`GetWidget`*.

Si necesitas ayuda, ponte en contacto con tu AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

## No estoy autorizado a realizar tareas como: PassRole

Si recibes un mensaje de error que indica que no estás autorizado a realizar la `iam:PassRole` acción, debes actualizar tus políticas para que puedas transferir una función a AWS Envío push de mensajes para el usuario final.

Alguno Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un IAM usuario denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS Envío push de mensajes para el usuario final. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

## Quiero permitir que personas ajenas a mi Cuenta de AWS para acceder a mi AWS Recursos push de mensajería para usuarios finales

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puede usar esas políticas para permitir que las personas accedan a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si AWS End User Messaging Push admite estas funciones, consulteCómo AWS End User Messaging Push funciona con IAM.
- Para obtener información sobre cómo proporcionar acceso a sus recursos en Cuentas de AWS que te pertenezca, consulta Proporcionar acceso a un IAM usuario en otro Cuenta de AWS que le pertenezca en la Guía IAM del usuario.
- Para obtener información sobre cómo proporcionar acceso a sus recursos a terceros Cuentas de AWS, consulte Proporcionar acceso a Cuentas de AWS propiedad de terceros en la Guía IAM del usuario.

- Para obtener información sobre cómo proporcionar acceso mediante la federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (federación de identidades) en la Guía del IAM usuario.
- Para saber la diferencia entre el uso de roles y políticas basadas en recursos para el acceso entre cuentas, consulte el acceso a recursos entre cuentas IAM en la Guía del usuario. IAM

# Validación de conformidad para AWS Envío push de mensajería para el usuario final

Para saber si un Servicio de AWS está dentro del ámbito de los programas de cumplimiento específicos, consulte Servicios de AWS dentro del ámbito de aplicación por programa de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte AWS Programas de cumplimiento .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Su responsabilidad de cumplimiento al usar Servicios de AWS viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos en AWS que se centran en la seguridad y el cumplimiento.
- Diseño de arquitectura para garantizar HIPAA la seguridad y el cumplimiento en Amazon Web Services: en este documento técnico se describe cómo las empresas pueden utilizar AWS para crear aplicaciones aptasHIPAA.

> **Note**
>
> No todos Servicios de AWS son HIPAA elegibles. Para obtener más información, consulta la Referencia de servicios HIPAA aptos.

- AWS Recursos de cumplimiento : esta colección de libros de trabajo y guías puede aplicarse a su sector y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar

Servicios de AWS y mapear la guía con los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad de la Industria de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).

- Evaluación de los recursos con las reglas del AWS Config Guía para desarrolladores: la AWS Config El servicio evalúa en qué medida las configuraciones de sus recursos cumplen con las prácticas internas, las pautas y las regulaciones del sector.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa de su estado de seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar su AWS recursos y para comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty — Esto Servicio de AWS detecta posibles amenazas para su Cuentas de AWS, cargas de trabajo, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas o maliciosas. GuardDuty puede ayudarle a cumplir diversos requisitos de conformidad, por ejemplo PCIDSS, cumpliendo con los requisitos de detección de intrusiones exigidos por determinados marcos de conformidad.
- AWS Audit Manager— Esto Servicio de AWS le ayuda a auditar continuamente su AWS uso para simplificar la forma en que gestiona el riesgo y el cumplimiento de las normas y los estándares del sector.

# Resiliencia en AWS Push de mensajería para el usuario final

La AWS La infraestructura global se basa en Regiones de AWS y zonas de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas a redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información acerca de Regiones de AWS y zonas de disponibilidad, consulte AWS Infraestructura global.

Además de AWS infraestructura global, AWS End User Messaging Push ofrece varias funciones para ayudarlo a satisfacer sus necesidades de respaldo y resiliencia de datos.

# Seguridad de la infraestructura en AWS Envío push de mensajería para el usuario final

Como servicio gestionado, AWS End User Messaging Push está protegido por el AWS procedimientos de seguridad de redes globales que se describen en el documento técnico Amazon Web Services: descripción general de los procesos de seguridad.

Usas AWS APIllamadas publicadas para acceder AWS Mensajes para el usuario final que se transmiten a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Los clientes también deben admitir conjuntos de cifrado con total confidencialidad (PFS), como (Ephemeral Diffie-Hellman) o DHE ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben firmarse con un identificador de clave de acceso y una clave de acceso secreta que esté asociada a un director. IAM O bien, puede utilizar la AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar las solicitudes.

## Configuración y análisis de vulnerabilidades

Como servicio gestionado, AWS End User Messaging Push está protegido por el AWS procedimientos de seguridad de redes globales que se describen en el documento técnico Amazon Web Services: descripción general de los procesos de seguridad. Esto significa que AWS administra y lleva a cabo tareas y procedimientos de seguridad básicos para reforzar, parchear, actualizar y mantener la infraestructura subyacente de su cuenta y sus recursos. Estos procedimientos han sido revisados y certificados por los terceros pertinentes.

## Prácticas recomendadas de seguridad

Uso AWS Cuentas Identity and Access Management (IAM) para controlar el acceso a API las operaciones, especialmente a las operaciones que crean, modifican o eliminan recursos. Para ellosAPI, estos recursos incluyen proyectos, campañas y viajes.

- Cree un usuario individual para cada persona que administre recursos de , incluido usted mismo. No utilice AWS credenciales raíz para administrar los recursos.
- Asigne a cada usuario el conjunto mínimo de permisos requerido para realizar sus tareas.
- Utilice IAM grupos para gestionar eficazmente los permisos de varios usuarios.

- Rote con regularidad sus credenciales de IAM.

Para obtener más información acerca de la seguridad, consulte Seguridad en AWS Envío de mensajes para el usuario final. Para obtener más información al respectoIAM, consulte AWS Identity and Access Management. Para obtener información sobre las IAM mejores prácticas, consulte las IAM mejores prácticas.

# Supervisión del envío de mensajes de usuario AWS final

El monitoreo es una parte importante para mantener la confiabilidad, la disponibilidad y el rendimiento de AWS End User Messaging Push y sus demás AWS soluciones. AWSproporciona las siguientes herramientas de supervisión para controlar la mensajería push de los usuarios AWS finales, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario:

- Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puedes CloudWatch hacer un seguimiento CPU del uso u otras métricas de tus EC2 instancias de Amazon y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde EC2 instancias de Amazon y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.
- Amazon se EventBridge puede utilizar para automatizar sus AWS servicios y responder automáticamente a los eventos del sistema, como los problemas de disponibilidad de las aplicaciones o los cambios de recursos. Los eventos de AWS los servicios se envían casi EventBridge en tiempo real. Puede crear reglas sencillas para indicar qué eventos le resultan de interés, así como qué acciones automatizadas se van a realizar cuando un evento cumple una de las reglas. Para obtener más información, consulta la Guía EventBridge del usuario de Amazon.
- AWS CloudTrailcaptura API las llamadas y los eventos relacionados realizados por su AWS cuenta o en su nombre y envía los archivos de registro a un bucket de Amazon S3 que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron. Para obtener más información, consulte la Guía del usuario de AWS CloudTrail.

# Supervisión de la mensajería push de los usuarios AWS finales con Amazon CloudWatch

Puede monitorear la mensajería push para el usuario AWS final CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles y casi en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

Para obtener una lista de métricas y dimensiones, consulte Monitorización de Amazon Pinpoint con CloudWatch en la Guía del usuario de Amazon Pinpoint.

# Registro de API llamadas push de mensajería de usuario AWS final mediante AWS CloudTrail

AWS End User Messaging Push está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en AWS End User Messaging Push. CloudTrail captura todas las API llamadas de mensajería automática para el usuario AWS final como eventos. Las llamadas capturadas incluyen las llamadas desde la consola Push de mensajería para el usuario AWS final y las llamadas en código a las API operaciones Push de mensajería para el usuario AWS final. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para AWS End User Messaging Push. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada CloudTrail, puede determinar la solicitud que se realizó a AWS End User Messaging Push, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía AWS CloudTrail del usuario.

## AWS Mensajería para el usuario final Inserte información en CloudTrail

CloudTrail está habilitada en tu cuenta Cuenta de AWS al crear la cuenta. Cuando se produce una actividad en AWS End User Messaging Push, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar eventos

recientes en su Cuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de tus eventos Cuenta de AWS, incluidos los eventos relacionados con AWS End User Messaging Push, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configuración de SNS las notificaciones de Amazon para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas las acciones push de mensajería de usuario AWS final se registran CloudTrail y se documentan en la APIreferencia de envío de mensajes de usuario AWS final. Por ejemplo, las llamadas a `UpdateApnsChannel` y `GetApnsVoipChannel` las acciones generan entradas en los archivos de CloudTrail registro. `GetAdmChannel`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el CloudTrail userIdentityelemento.

# Descripción de las entradas del archivo de registro push de mensajería para el usuario AWS final

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las API llamadas públicas, por lo que no aparecen en ningún orden específico.

# Acceda a la mensajería push para el usuario AWS final mediante un punto final de interfaz (AWS PrivateLink)

Puede usarlo AWS PrivateLink para crear una conexión privada entre su mensajería push VPC y la del usuario AWS final. Puede acceder a AWS End User Messaging Push como si estuviera en el suyoVPC, sin necesidad de utilizar una pasarela de Internet, NAT dispositivo, VPN conexión o AWS Direct Connect conexión. Las instancias VPC que tenga no necesitan direcciones IP públicas para acceder a AWS End User Messaging Push.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a la mensajería push del usuario AWS final.

Para obtener más información, consulte Acceso directo AWS PrivateLink en la Servicios de AWS guía.AWS PrivateLink

## Consideraciones sobre la mensajería push para el usuario AWS final

Antes de configurar un punto final de interfaz para la mensajería push de usuario AWS final, consulte las consideraciones de la AWS PrivateLink guía.

AWS End User Messaging Push permite realizar llamadas a todas sus API acciones a través del punto final de la interfaz.

VPCLas políticas de punto final no son compatibles con la mensajería push para el usuario AWS final. De forma predeterminada, se permite el acceso total a la mensajería push de usuario AWS final a través del punto final de la interfaz. Como alternativa, puede asociar un grupo de seguridad a las interfaces de red de los terminales para controlar el tráfico que se envía a la mensajería push del usuario AWS final a través del punto final de la interfaz.

# Cree un punto final de interfaz para la mensajería push de usuario AWS final

Puede crear un punto AWS final de interfaz para End User Messaging Push mediante la VPC consola de Amazon o el AWS Command Line Interface (AWS CLI). Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

Cree un punto final de interfaz para AWS End User Messaging Push con el siguiente nombre de servicio:

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

Si habilita la opción privada DNS para el punto final de la interfaz, puede realizar API solicitudes a AWS End User Messaging Push utilizando su DNS nombre regional predeterminado. Por ejemplo, `com.amazonaws.us-east-1.pinpoint`.

# Creación de una política de puntos de conexión para el punto de conexión de interfaz

Una política de punto final es un IAM recurso que se puede adjuntar a un punto final de interfaz. La política de punto final predeterminada permite el acceso total a la mensajería push del usuario AWS final a través del punto final de la interfaz. Para controlar el acceso permitido a la mensajería push de usuario AWS final desde su dispositivoVPC, adjunte una política de punto final personalizada al punto final de la interfaz.

Una política de punto de conexión especifica la siguiente información:

- Los principales que pueden realizar acciones (Cuentas de AWS IAMusuarios y IAM funciones).
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

Para obtener más información, consulte Control del acceso a los servicios con políticas de punto de conexión en la Guía del usuario de AWS PrivateLink .

Ejemplo: política de VPC punto AWS final para las acciones push de mensajería de los usuarios finales

El siguiente es un ejemplo de una política de un punto de conexión personalizado. Al adjuntar esta política al punto final de la interfaz, todos los directores de todos los recursos pueden acceder a las acciones push de mensajería para el usuario AWS final que figuran en la lista.

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

# Cuotas para el envío de mensajes a los usuarios AWS finales

Cuenta de AWS Tiene cuotas predeterminadas, anteriormente denominadas límites, para cada AWS servicio. A menos que se indique lo contrario, cada cuota es específica de la región de . Puede solicitar el aumento de algunas cuotas, pero otras no se pueden aumentar.

Para ver las cuotas de AWS End User Messaging Push, abra la consola Service Quotas. En el panel de navegación, elija AWSservicios y seleccione Amazon Pinpoint.

Su AWS cuenta tiene las siguientes cuotas relacionadas con la mensajería push para usuarios AWS finales.

| Recurso | Cuota predeterminada | Puede optar a un aumento de la cuota |
|---|---|---|
| Número máximo de notificaciones de inserción que se pueden enviar por segundo en una campaña | 25 000 notificaciones por segundo | Sí, utilice la consola Service Quotas |
| Tamaño de carga útil de mensajes de Amazon Device Messaging (ADM) | 6 KB por mensaje | No |
| Tamaño de la carga útil de los mensajes del servicio de notificaciones push de Apple (APNs) | 4 KB por mensaje | No |
| Tamaño de carga de mensajes de entorno de pruebas de APNs | 4 KB por mensaje | No |
| Tamaño de carga de mensajes de Baidu Cloud Push | 4 KB por mensaje | No |

| Recurso | Cuota predeterminada | Puede optar a un aumento de la cuota |
| --- | --- | --- |
| Tamaño de la carga útil de los mensajes de Firebase Cloud Messaging (FCM) | 4 KB por mensaje | No |

# Historial de documentos de la Guía de usuario de AWS End User Messaging Push

En la siguiente tabla se describen las versiones de la documentación de AWS End User Messaging Push.

| Cambio | Descripción | Fecha |
| --- | --- | --- |
| Versión inicial | Versión inicial de la Guía de usuario de AWS End User Messaging Push | 24 de julio de 2024 |

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.