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? 1
¿Es la primera vez que utiliza mensajería push para usuarios AWS finales? 1
Características de la mensajería AWS push para usuarios finales 1
Acceder a la mensajería push para AWS usuarios finales 2
Disponibilidad regional 3
Configuración de un Cuenta de AWS 4
Inscríbase en un Cuenta de AWS 4
Creación de un usuario con acceso administrativo 4
Introducción 7
Crear una aplicación y habilitar los canales push 8
Contextual 8
Requisitos previos 9
Procedimiento 9
Desactivar los canales push 11
Envío de un mensaje push 12
Recursos adicionales 25
Recibir notificaciones push en tu aplicación 26
Configuración de notificaciones de inserción rápidas 26
¿Trabajando con fichas APNs 26
Configuración de las notificaciones push de Android 27
Configuración de notificaciones push para Flutter 27
Configuración de las notificaciones de inserción de React Native 27
Creación de una aplicación de 27
Gestión de notificaciones push 28
Eliminación de una aplicación de 29
Contextual 29
Procedimiento 29
Prácticas recomendadas 30
Envío de un gran volumen de notificaciones de inserción 30
Seguridad 31
Protección de los datos 32
Cifrado de datos 33
Cifrado en tránsito 33
Administración de claves 33

Privacidad del tráfico entre redes 34
Identity and Access Management 35
Público 35
Autenticación con identidades 36
Administración de acceso mediante políticas 40
Cómo funciona la mensajería push para usuarios AWS finales con IAM 42
Ejemplos de políticas basadas en identidades 50
Solución de problemas 54
Validación de conformidad 56
Resiliencia 57
Seguridad de infraestructuras 57
Configuración y análisis de vulnerabilidades 58
Prácticas recomendadas de seguridad 58
Monitorización 59
Monitorización con CloudWatch 60
CloudTrail registros 60
AWS Mensajería para el usuario final Inserte información en CloudTrail 60
Descripción de las entradas del archivo de registro push de mensajería para el usuario AWS final 62
AWS PrivateLink 63
Consideraciones 63
Creación de un punto de conexión de interfaz 64
Creación de una política de punto de conexión 64
Cuotas 66
Historial de documentos 68
..... lxix

¿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 mensajería push para usuarios AWS finales?
- Características de la mensajería AWS push para usuarios finales
- Acceder a la mensajería push para AWS usuarios finales
- Disponibilidad regional

¿Es la primera vez que utiliza mensajería push para usuarios AWS finales?

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 y iPads, así como al navegador Safari en dispositivos macOS, como ordenadores portátiles y de sobremesa Mac.

- Baidu Cloud Push
- Amazon Device Messaging (ADM)

Acceder a la mensajería push para AWS usuarios finales

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

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 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 eres un desarrollador de software que prefiere crear aplicaciones con un lenguaje específico APIs en lugar de enviar una solicitud a través de HTTP o HTTPS, 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 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 América del Norte, 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 Puntos de enlace y cuotas de la API AWS y los puntos de enlace de servicio de Amazon Pinpoint 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 poder utilizar AWS End User Messaging Push para enviar notificaciones push a tu aplicación, primero tienes que obtener una Cuenta de AWS con los permisos de IAM suficientes. Esto también se Cuenta de AWS puede usar para otros servicios del AWS ecosistema.

Temas

- Inscríbase en un Cuenta de AWS
- Creación de un usuario con acceso administrativo

Inscríbase en un Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse 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 implica recibir una llamada telefónica o un mensaje de texto e introducir un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa 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 crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

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

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

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

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso 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 Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

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

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In 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 de privilegios mínimos.

Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

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

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

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 (APN) de Apple, 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 certificado TLS 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 certificado TLS 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 la obtención de 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 Creación de 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:

            - SSL certificate (Certificado SSL): archivo .p12 del certificado TLS.

- 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 el archivo JSON de tu servicio.
      ii. En el caso de las credenciales clave, introduce tu clave en la clave de la API.

6. (Opcional) Sigue este paso opcional para activar Baidu Cloud Push.

   a. Para Baidu Cloud Push, selecciona Activar.
   b. Para la clave de API, introduce tu clave de API.
   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 Creación de aplicación.

Desactivación de los canales push

Siga estas instrucciones para desactivar cualquiera de los canales push.

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

Puedes usar 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 obtener más ejemplos de código sobre puntos de conexión, segmentos y canales, consulte 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 a la API push de mensajería para el AWS usuario final. Google dejó de utilizar el servicio Google Cloud Messaging (GCM) el 10 de abril de 2018. Sin embargo, la API push de mensajería para el usuario AWS final usa el nombre del `GCM` servicio para los mensajes que envía a través del servicio de FCM, a fin de mantener la compatibilidad con el código de la API que se escribió antes de la interrupción del servicio de GCM.

GCM (AWS CLI)

En el siguiente ejemplo, se utilizan send-messages para enviar una notificación push de GCM con el. AWS CLI*token*Sustitúyalo por el token único del dispositivo y por el identificador de la *611e3e3cdd47474c9c1399a50example* 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 push de GCM, utilizando todas las claves antiguas, con el. AWS CLI*token*Sustitúyalo por el token único del dispositivo y por el identificador de la *611e3e3cdd47474c9c1399a50example* 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 push de GCM con una carga útil de FCMv1 mensajes mediante el. AWS CLI*token*Sustitúyalo por el token único del dispositivo y *611e3e3cdd47474c9c1399a50example* por el identificador de la 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 utilizas `ImageUrl` el campo para GCM, pinpoint envía el campo como notificación de datos, con la clave`pinpoint.notification.imageUrl`, lo que puede impedir que la imagen se reproduzca de forma predeterminada. Utiliza RawContent o añade el manejo de las claves de datos, por ejemplo, integrando tu aplicación con ellas. AWS Amplify

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 y iPads.

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 utilizan send-messages para enviar una notificación al navegador web Safari con el. AWS CLI*token*Sustitúyalo por el token único del dispositivo y por el identificador *611e3e3cdd47474c9c1399a50example* de la 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 push de APNS con el. AWS CLI*token*Sustitúyalo por el token único del dispositivo, *611e3e3cdd47474c9c1399a50example* por el identificador de la aplicación y *GAME_INVITATION* por 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 JavaScript de Node.js. En este ejemplo se supone que ya has instalado y configurado el SDK para JavaScript 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, consulta la Guía para desarrolladores de Node.js sobre cómo configurar las credenciales JavaScript en el AWS SDK.

```
'use strict';

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

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

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

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

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

// An object that contains the unique token of the device that you want to send
// the message to, and the push service that you want to use to send the message.
var recipient = {
  'token': 'a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0',
  'service': 'GCM'
  };

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

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

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

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

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

function CreateMessageRequest() {
  var token = recipient['token'];
  var service = recipient['service'];
  if (service == 'GCM') {
    var messageRequest = {
      'Addresses': {
        [token]: {
```

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

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

  return messageRequest
}

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

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

```
  var messageRequest = CreateMessageRequest();

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

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

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

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

SendMessage()
```

Python

Utilice este ejemplo para enviar notificaciones push mediante el AWS SDK para Python (Boto3). En este ejemplo se presupone que ya ha instalado y configurado el 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 Referencia de la API del AWS SDK para Python (Boto3).

```
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 para 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

- 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 push rápidas
- Configuración de notificaciones push para Android
- Configuración de notificaciones push para Flutter
- Configuración de las notificaciones de inserción de React Native
- Cree una aplicación en AWS End User Messaging Push
- Gestión de notificaciones push

Configuración de notificaciones push 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 él APNs.

Configuración de notificaciones push para Android

Las notificaciones de inserción de las aplicaciones de Android se envían mediante Firebase Cloud Messaging (FCM), que sustituye a Google Cloud Messaging (GCM). Antes de poder enviar notificaciones de inserción a dispositivos Android, debe obtener credenciales de FCM. 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 notificaciones push para 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 e APNs 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 la API, un AWS SDK o el AWS Command Line Interface(AWS CLI). Para crear una aplicación, usa el `Apps` recurso. Para configurar canales de notificaciones de inserción, utilice los siguientes recursos:

- APNs canal para enviar mensajes a los usuarios de dispositivos iOS mediante el servicio de notificaciones push de Apple.
- Canal de ADM para enviar mensajes a los usuarios de dispositivos Amazon Kindle Fire.
- Canal de Baidu para enviar mensajes a los usuarios de Baidu.
- Canal de GCM para enviar mensajes a dispositivos Android mediante Firebase Cloud Messaging (FCM), que sustituye 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 con 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.

Asegúrate de que tu cuenta esté configurada correctamente con las credenciales de cada uno de los proveedores de notificaciones push que vayas a utilizar, como FCM o APNs.

Por último, diseñe una forma de gestionar las excepciones. Cada servicio de notificaciones de inserción proporciona diferentes mensajes de excepción. Para envíos transaccionales, puede recibir un código de estado principal de 200 para la llamada a la API, con un código de estado por punto de conexión de error permanente de 400 si se determina que el token de plataforma (por ejemplo, FCM) o el certificado (por ejemplo, APN) correspondientes no son válidos durante el envío de los mensajes.

Seguridad en la mensajería push para el usuario AWS final

La seguridad en la nube 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 usted y usted. 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 ejecuta AWS los servicios en la 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 de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento que se aplican a la mensajería push para usuarios AWS finales, consulte AWS Servicios incluidos en el ámbito de aplicación del programa AWS .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida al utilizar AWS End User Messaging Push. En los siguientes temas, se muestra cómo configurar la mensajería push para el usuario AWS final para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de mensajería push para el usuario AWS final.

Temas

- Protección de datos en AWS End User Messaging Push
- Administración de identidad y acceso para la mensajería AWS push de usuario final
- Validación del cumplimiento de la mensajería AWS push para usuarios finales
- Resiliencia en la AWS transmisión de mensajes a los usuarios finales
- La seguridad de la infraestructura en la mensajería AWS push para usuarios finales
- Configuración y análisis de vulnerabilidades
- Prácticas recomendadas de seguridad

Protección de datos en AWS End User Messaging Push

El modelo de se aplica a protección de datos en AWS End User Messaging Push. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulta las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las 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:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta Estándar de procesamiento de la información federal (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 End User Messaging Push u otro tipo de mensajería automática que Servicios de AWS utilice la consola, la API o AWS CLI AWS

SDKs Cualquier dato que ingrese en etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Cifrado de datos

AWS Los datos de mensajería push para el usuario final se cifran tanto en tránsito como en reposo. Cuando envías datos a AWS End User Messaging Push, este los cifra a medida que los recibe y los almacena. Cuando recupera datos de AWS End User Messaging Push, éste 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 a AWS End User Messaging Push. Para cifrar sus datos, AWS End User Messaging Push utiliza claves internas AWS Key Management Service (AWS KMS) que el servicio posee y mantiene en su nombre. Rotamos estas claves periódicamente. Para obtener más información al respecto AWS KMS, consulte la Guía para AWS Key Management Service desarrolladores.

Cifrado en tránsito

AWS End User Messaging Push utiliza HTTPS y Transport Layer Security (TLS) 1.2 o una versión posterior para comunicarse con sus clientes y aplicaciones. Para comunicarse con otros AWS servicios, AWS End User Messaging Push utiliza HTTPS y TLS 1.2. Además, al crear y administrar los recursos de mensajería push para el usuario AWS 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 los datos de AWS End User Messaging Push, AWS End User Messaging Push utiliza AWS KMS claves internas que el servicio posee y mantiene en tu nombre. Rotamos estas claves periódicamente. No puede aprovisionar ni utilizar claves propias AWS KMS ni de otro tipo para cifrar los datos que almacene en AWS End User Messaging Push.

Privacidad del tráfico entre redes

La privacidad del tráfico entre redes se refiere a proteger las conexiones y el tráfico entre AWS End User Messaging Push y sus clientes y aplicaciones locales, y entre AWS End User Messaging Push y otros AWS recursos de la misma región. AWS Las siguientes funciones y prácticas pueden ayudarle a garantizar la privacidad del tráfico entre redes para la mensajería push de usuario AWS final.

Tráfico entre la mensajería push de usuario AWS final y los clientes y aplicaciones locales

Para establecer una conexión privada entre AWS End User Messaging Push y los clientes y aplicaciones de su red local, puede utilizar. AWS Direct Connect Esto le permite vincular su red a una ubicación de AWS Direct Connect mediante un cable de 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 la Guía del usuario de AWS Direct Connect .

Para ayudar a proteger el acceso a la mensajería automática para el usuario AWS final publicada APIs, le recomendamos que cumpla con los requisitos de mensajería automática para el usuario AWS final en relación con las llamadas a la API. AWS La función Push de mensajería 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 confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (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 que estén asociadas al principal AWS Identity and Access Management (IAM) de su AWS cuenta. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Tráfico entre AWS End User Messaging Push y otros recursos AWS

Para proteger las comunicaciones entre AWS End User Messaging Push y otros AWS recursos de la misma AWS región, AWS End User Messaging Push utiliza HTTPS y TLS 1.2 de forma predeterminada.

Administración de identidad y acceso para la mensajería AWS push de usuario final

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos push de mensajería automática para el usuario AWS final. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración de acceso mediante políticas
- Cómo funciona la mensajería push para usuarios AWS finales con IAM
- Ejemplos de políticas basadas en la identidad para AWS la mensajería automática de usuarios finales
- Solución de problemas de identidad y acceso a la mensajería push para el usuario AWS final

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que se realice en AWS End User Messaging Push.

Usuario del servicio: si utiliza el servicio push de mensajería para el usuario AWS final para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más funciones de mensajería push para el usuario AWS final para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una función de AWS End User Messaging Push, consulteSolución de problemas de identidad y acceso a la mensajería push para el usuario AWS final.

Administrador del servicio: si está a cargo de los recursos de mensajería push para el usuario AWS final en su empresa, probablemente tenga pleno acceso a la mensajería push para el usuario AWS final. Es su trabajo determinar a qué funciones y recursos de mensajería push para el usuario AWS final deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su gestionador de

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

Administrador de IAM: si es administrador de IAM, puede que le interese obtener más información sobre cómo redactar políticas para administrar el acceso a AWS End User Messaging Push. Para ver ejemplos de políticas push de mensajería de usuario AWS final basadas en la identidad que puede utilizar en IAM, consulte. Ejemplos de políticas basadas en la identidad para AWS la mensajería automática de usuarios finales

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), 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, su gestionador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

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

Si accede 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 utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes API en la Guía del usuario de IAM.

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

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza 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 las tareas que requieren que inicie sesión como usuario raíz, consulta Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

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 AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al 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 una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulta ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puedes iniciar sesión como grupo. Puedes 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 los recursos de IAM.

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 Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM 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 acerca de roles de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener

información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

- 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 usuario o un rol de IAM para realizar acciones en AWS ellas, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
  - Función vinculada al 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 el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.
- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto 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 Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto 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 AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON 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. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utiliza 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 el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, 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 más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de 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 roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM 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 políticas AWS gestionadas de IAM en una política basada 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. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, 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 característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puedes 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 Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.

- Políticas de control de servicios (SCPs): SCPs son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations 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 más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

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

Cómo funciona la mensajería push para usuarios AWS finales con IAM

Antes de usar IAM para administrar el acceso a AWS End User Messaging Push, averigüe qué funciones de IAM están disponibles para usar con AWS End User Messaging Push.

Funciones de IAM que puede utilizar con AWS End User Messaging Push

Característica de IAM	AWS Soporte de mensajería push para el usuario final
Políticas basadas en identidades	Sí
Políticas basadas en recursos	Sí
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACLs	No
ABAC (etiquetas en 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 funcionan la mensajería push para el usuario AWS final y otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

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

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, 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 más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, 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 asociada. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para AWS la mensajería automática de usuarios finales

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

Políticas basadas en los recursos de End User Messaging Push AWS

Compatibilidad con las políticas basadas en recursos: sí

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM 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 toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de 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 administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Acciones políticas para la mensajería AWS push de usuarios finales

Compatibilidad con las acciones de políticas: sí

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

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. 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 las acciones push de mensajería del usuario AWS final, consulte las acciones definidas por el envío de mensajes del usuario AWS final en la referencia de autorización del servicio.

Las acciones políticas de AWS End User Messaging Push utilizan 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 políticas de mensajería push para usuarios AWS finales basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS la mensajería automática de usuarios finales

Recursos de políticas para la mensajería push para usuarios AWS finales

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes 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, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de recursos de mensajería push para el usuario AWS final y sus tipos de recursos ARNs, consulte los recursos definidos por el envío de mensajes push para el usuario AWS final en la referencia de autorización del servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS End User Messaging Push.

Para ver ejemplos de políticas de mensajería push para usuarios AWS finales basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS la mensajería automática de usuarios finales

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

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

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes 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. Puedes crear expresiones condicionales que utilizan 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 elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios

valores para una única 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 puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

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

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

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

ACLs en AWS End User Messaging Push

Soportes ACLs: 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. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con mensajería push para el AWS usuario final

Compatibilidad con ABAC (etiquetas en las políticas): parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta 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 sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con AWS End User Messaging Push

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu 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 roles, consulte Cambio de un usuario a un rol de IAM (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS 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 la mensajería AWS push de usuario final

Admite sesiones de acceso directo (FAS): sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Funciones de servicio para la mensajería push de usuario AWS final

Compatibilidad con roles de servicio: sí

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

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de mensajería push para el usuario AWS final. Edite las funciones de servicio únicamente cuando AWS End User Messaging Push le indique cómo hacerlo.

Funciones vinculadas al servicio para la mensajería push de usuario AWS 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 el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con 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 la mensajería automática de usuarios finales

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar los recursos de mensajería push para el usuario AWS final. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

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

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la consola push de mensajería para el AWS 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, eliminar o acceder a los recursos de AWS End User Messaging Push de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y opte por los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con

el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, 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. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

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

Uso de la consola push de mensajería para el AWS usuario final

Para acceder a la consola push de mensajería para el usuario AWS final, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de mensajería push para el usuario AWS final que tiene en su cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la

consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

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

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

```
{
 "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 podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

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

Solución de problemas de identidad y acceso a la mensajería push para el usuario AWS final

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con AWS End User Messaging Push e IAM.

Temas

- No estoy autorizado a realizar ninguna acción en AWS End User Messaging Push
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de mensajería automática para el usuario AWS final

No estoy autorizado a realizar ninguna acción en AWS End User Messaging Push

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

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `mobiletargeting:`*`GetWidget`*.

```
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 necesita ayuda, póngase en contacto con su AWS administrador. El gestionador 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 End User Messaging Push.

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

El siguiente ejemplo de error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS End User Messaging Push. 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 gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de mensajería automática para el usuario AWS final

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 utilizar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si AWS End User Messaging Push admite estas funciones, consulte. Cómo funciona la mensajería push para usuarios AWS finales con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.

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

Validación del cumplimiento de la mensajería AWS push para usuarios finales

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

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 Servicios de AWS utilizarlos 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:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la HIPAA.
- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria 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 la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.

- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas 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: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en la AWS transmisión de mensajes a los usuarios finales

La infraestructura AWS global se basa en zonas de disponibilidad Regiones de AWS y zonas de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante 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 sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS

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

La seguridad de la infraestructura en la mensajería AWS push para usuarios finales

Como servicio gestionado, AWS End User Messaging Push está protegido por los procedimientos de seguridad de red AWS global que se describen en el documento técnico Amazon Web Services: Overview of Security Processes.

Utiliza las llamadas a la API AWS publicadas para acceder a AWS End User Messaging Push 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 ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como DHE (Ephemeral Diffie-Hellman) o 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 estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Configuración y análisis de vulnerabilidades

Como servicio gestionado, AWS End User Messaging Push está protegido por los procedimientos de seguridad de la red AWS global 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, de otro modo, 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

Utilice las cuentas de AWS Identity and Access Management (IAM) para controlar el acceso a las operaciones de la API, especialmente a las operaciones que crean, modifican o eliminan recursos. En el caso de la API de , estos recursos incluyen proyectos, campañas y recorridos.

- Cree un usuario individual para cada persona que administre recursos de , incluido usted mismo. No utilices credenciales AWS raíz para administrar los recursos.
- Asigne a cada usuario el conjunto mínimo de permisos requerido para realizar sus tareas.
- Use los grupos de IAM para administrar con eficacia los permisos para varios usuarios.
- Rote con regularidad sus credenciales de IAM.

Para obtener más información acerca de la seguridad, consulte Seguridad en la mensajería push para el usuario AWS final. Para obtener más información acerca de IAM, consulte AWS Identity and Access Management. Para obtener información acerca de las prácticas recomendadas de IAM, consulte Prácticas recomendadas de IAM.

Supervisión del envío de mensajes por parte del usuario AWS final

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS End User Messaging Push y del resto de soluciones de AWS. AWS proporciona las siguientes herramientas de supervisión para controlar el envío de mensajes 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 del uso de la CPU 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 entregan EventBridge prácticamente 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 las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre y entrega 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 AWS CloudTrail Guía del usuario de .

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 llamadas a la API 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 llamadas a la API para enviar mensajes de 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 operaciones de la API 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 los

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 las notificaciones de Amazon SNS 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 para el usuario AWS final se registran CloudTrail y se documentan en la referencia de la API push de mensajería para el 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 le 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 elemento userIdentity de CloudTrail .

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 llamadas a la API pública, 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 utilizarla AWS PrivateLink para crear una conexión privada entre su VPC y AWS End User Messaging Push. Puede acceder a AWS End User Messaging Push como si estuviera en su VPC, sin utilizar una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o AWS Direct Connect una conexión. Las instancias de su VPC 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 acciones de API a través del punto final de la interfaz.

Las políticas de punto final de VPC no son compatibles con la mensajería push de 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 consola Amazon VPC 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 el DNS privado para el punto final de la interfaz, puede realizar solicitudes de API a AWS End User Messaging Push utilizando su nombre de DNS 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 de conexión es un recurso de IAM que puede adjuntar al punto de conexión de su interfaz. La política de punto final predeterminada permite el acceso total a la mensajería push de 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 VPC, adjunte una política de punto final personalizada al punto final de la interfaz.

Una política de punto de conexión especifica la siguiente información:

- Las entidades principales que pueden llevar a cabo acciones (Cuentas de AWS, usuarios de IAM y roles de IAM).
- 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 punto final de VPC para acciones push de mensajería de usuario AWS final

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 principales usuarios de todos los recursos pueden acceder a las acciones push de mensajería de usuario AWS final que se muestran 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 los servicios de AWS y seleccione Amazon Pinpoint.

Su cuenta de AWS tiene las siguientes cuotas relacionadas con AWS End User Messaging Push.

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 de mensajes de Amazon Device Messaging (ADM)	6 KB por mensaje	No
Tamaño de carga útil de los mensajes del servicio de notificaciones push de Apple (APNs)	4 KB por mensaje	No
APNs tamaño de carga útil de los mensajes de entorno aislado	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 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.