



Manual do usuário

# AWS Push de mensagens para o usuário final



# AWS Push de mensagens para o usuário final: Manual do usuário

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

# Table of Contents

O que é AWS End User Messaging Push? .....	1
Você é usuário do AWS End User Messaging Push pela primeira vez? .....	1
Características do envio de mensagens push para o usuário AWS final .....	1
Acessando o envio de mensagens push para o usuário AWS final .....	2
Disponibilidade regional .....	3
Configurando um Conta da AWS .....	4
Inscreva-se para um Conta da AWS .....	4
Criar um usuário com acesso administrativo .....	4
Conceitos básicos .....	7
Criação de um aplicativo e ativação de canais push .....	8
Contextual .....	8
Pré-requisitos .....	9
Procedimento .....	9
Desativando canais push .....	11
Enviando uma mensagem push .....	12
Recursos adicionais do .....	25
Recebendo notificações push em seu aplicativo .....	26
Configurar notificações por push do Swift .....	26
Trabalhando com APNs tokens .....	26
Configurar as notificações por push em Android .....	27
Configurar notificações por push do Flutter .....	27
Configurar notificações por push do React Native .....	27
Criar uma aplicação do .....	27
Gerenciar notificações por push .....	28
Excluir um aplicativo .....	29
Contextual .....	29
Procedimento .....	29
Práticas recomendadas .....	30
Enviar um grande volume de notificações por push .....	30
Segurança .....	31
Proteção de dados .....	32
Criptografia de dados .....	33
Criptografia em trânsito .....	33
Gerenciamento de chaves .....	33

---

Privacidade do tráfego entre redes .....	33
Gerenciamento de identidade e acesso .....	34
Público .....	35
Autenticação com identidades .....	35
Gerenciar o acesso usando políticas .....	37
Como o AWS End User Messaging Push funciona com o IAM .....	38
Exemplos de políticas baseadas em identidade .....	44
Solução de problemas .....	48
Validação de conformidade .....	50
Resiliência .....	50
Segurança da infraestrutura .....	51
Análise de configuração e vulnerabilidade .....	51
Práticas recomendadas de segurança .....	51
Monitoramento .....	53
Monitoramento com CloudWatch .....	53
CloudTrail troncos .....	54
AWS Mensagens para o usuário final Envie informações para CloudTrail .....	54
Compreendendo as entradas do arquivo de log push de mensagens de usuário AWS final ...	55
AWS PrivateLink .....	56
Considerações .....	56
Como criar um endpoint de interface .....	57
Criar uma política de endpoint .....	57
Cotas .....	59
Histórico de documentos .....	60
.....	lxi

# O que é AWS End User Messaging Push?

## Note

Os recursos de notificação push do Amazon Pinpoint agora são chamados de AWS End User Messaging.

Com o AWS End User Messaging Push, você pode engajar os usuários de seus aplicativos enviando notificações push por meio de um canal de notificação push. Oferecemos suporte ao Apple Push Notification Service (APNs), Firebase Cloud Messaging (FCM), Amazon Device Messaging (ADM) e Baidu Push.

## Tópicos

- [Você é usuário do AWS End User Messaging Push pela primeira vez?](#)
- [Características do envio de mensagens push para o usuário AWS final](#)
- [Acessando o envio de mensagens push para o usuário AWS final](#)
- [Disponibilidade regional](#)

## Você é usuário do AWS End User Messaging Push pela primeira vez?


Se você é um usuário iniciante do AWS End User Messaging Push, recomendamos que comece lendo as seguintes seções:

- [Configurando um Conta da AWS](#)
- [Introdução ao AWS End User Messaging Push](#)
- [Criação de um aplicativo e ativação de canais push](#)

## Características do envio de mensagens push para o usuário AWS final

Você pode enviar notificações por push para aplicativos usando canais separados de notificação por push aos seguintes serviços:

- Firebase Cloud Messaging (FCM)
- Serviço de notificação push da Apple (APNs)

 Note

Você pode usar APNs para enviar mensagens para dispositivos iOS, como iPhones e iPads, bem como para o navegador Safari em dispositivos macOS, como laptops e desktops Mac.

- Baidu Cloud Push
- Amazon Device Messaging (ADM)

## Acessando o envio de mensagens push para o usuário AWS final

Explique resumidamente as diferentes formas de obter acesso ao serviço, seja por console, CLI ou API.

Você pode gerenciar o AWS End User Messaging Push usando as seguintes interfaces:

### AWS Console push de mensagens para o usuário final

A interface da web na qual você cria e gerencia recursos push de mensagens de usuário AWS final. Se você se inscreveu em um Conta da AWS, você pode acessar o console AWS End User Messaging Push a partir do Console de gerenciamento da AWS.

### AWS Command Line Interface

Interaja com AWS os serviços usando comandos em seu shell de linha de comando. O AWS Command Line Interface é compatível com Windows, macOS e Linux. Para obter mais informações sobre o AWS CLI, consulte o [Guia AWS Command Line Interface do usuário](#).

Você pode encontrar os comandos AWS End User Messaging Push na [Referência de AWS CLI Comandos](#).

### AWS SDKs

Se você é um desenvolvedor de software que prefere criar aplicativos usando uma linguagem específica APIs em vez de enviar uma solicitação por HTTP ou HTTPS, AWS fornece bibliotecas, exemplos de código, tutoriais e outros recursos. Essas bibliotecas fornecem funções básicas que automatizam tarefas, como assinar criptograficamente suas solicitações, repetir solicitações e

lidar com respostas de erro. Essas funções ajudam a tornar mais eficiente para você começar. Para obter mais informações, consulte [Ferramentas para criar na AWS](#).

## Disponibilidade regional

AWS O End User Messaging Push está disponível Regiões da AWS em vários países da América do Norte, Europa, Ásia e Oceania. Em cada região, AWS mantém várias zonas de disponibilidade. Essas zonas de disponibilidade são fisicamente isoladas umas das outras, mas são unidas por conexões de rede privadas, de baixa latência, de alta taxa de transferência e altamente redundantes. Essas zonas de disponibilidade são usadas para fornecer níveis muito altos de disponibilidade e redundância, além de minimizar a latência.

Para saber mais sobre Regiões da AWS, consulte [Especificar qual Regiões da AWS sua conta pode usar](#) no Referência geral da Amazon Web Services. [Para obter uma lista de todas as regiões em que o AWS End User Messaging Push está disponível atualmente e o endpoint de cada região, consulte Endpoints e cotas para a API do Amazon Pinpoint e AWS endpoints de serviço no. Referência geral da Amazon Web Services](#) Para saber mais sobre quantas zonas de disponibilidade estão disponíveis em cada região, consulte [Infraestrutura global da AWS](#).

# Configurando um Conta da AWS

Antes de usar o AWS End User Messaging Push para enviar notificações push para seu aplicativo, primeiro você precisa obter um Conta da AWS com permissões suficientes do IAM. Isso também Conta da AWS pode ser usado para outros serviços no AWS ecossistema.

## Tópicos

- [Inscreva-se para um Conta da AWS](#)
- [Criar um usuário com acesso administrativo](#)

## Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

## Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS Centro de Identidade do AWS IAM, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

## Proteja seu Usuário raiz da conta da AWS

1. Faça login [Console de gerenciamento da AWS](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Fazer login como usuário-raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

## Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

Para obter instruções, consulte [Habilitar o Centro de Identidade do AWS IAM](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia Centro de Identidade do AWS IAM do usuário.

## Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

## Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Criar um conjunto de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Adicionar grupos](#) no Guia do usuário do Centro de Identidade do AWS IAM .

# Introdução ao AWS End User Messaging Push

Para configurar o AWS End User Messaging Push para que ele possa enviar notificações push para seus aplicativos, primeiro você precisa fornecer as credenciais que autorizam o AWS End User Messaging Push a enviar mensagens para seu aplicativo. As credenciais que você fornece dependem do sistema de notificação por push usado:

- Para obter as credenciais do serviço Apple Push Notification (APN), consulte [Obter uma chave de criptografia e um ID de chave da Apple](#) e [Obter um certificado de provedor da Apple na documentação do](#) desenvolvedor da Apple.
- [Para as credenciais do Firebase Cloud Messaging \(FCM\), elas podem ser obtidas por meio do console do Firebase, consulte Firebase Cloud Messaging.](#)
- [Para obter as credenciais do Baidu, consulte Baidu.](#)
- Para obter as credenciais do Amazon Device Messaging (ADM), consulte [Obter](#) credenciais.

# Criação de um aplicativo e ativação de canais push

Antes de usar o AWS End User Messaging Push para enviar notificações push, primeiro você precisa criar um aplicativo e ativar o canal de notificações push.

## Contextual

### Aplicativo

Um aplicativo é um contêiner de armazenamento para todas as suas configurações de envio de mensagens de usuário AWS final. O aplicativo também armazena suas configurações de canais, campanhas e viagens do Amazon Pinpoint.

### Chave

Uma chave de assinatura privada usada pelo AWS End User Messaging Push para assinar criptograficamente tokens de APNs autenticação. A chave de assinatura é obtida da sua conta de desenvolvedor da Apple.

Se você fornecer uma chave de assinatura, o AWS End User Messaging Push usará um token APNs para se autenticar para cada notificação push enviada. Com sua chave de assinatura, você pode enviar notificações push para ambientes APNs de produção e sandbox.

Ao contrário de certificados, sua chave de assinatura não expira. Você fornece sua chave apenas uma vez, e não é necessário renová-la posteriormente. Você pode usar a mesma chave de assinatura para vários aplicativos. Para obter mais informações, consulte [Comunique-se APNs usando tokens de autenticação](#) na Ajuda do Xcode.

### Certificado

Um certificado TLS que o AWS End User Messaging Push usa para se autenticar APNs quando você envia notificações push. Um APNs certificado pode oferecer suporte a ambientes de produção e sandbox, ou pode oferecer suporte somente ao ambiente sandbox. O certificado pode ser obtido da sua conta de desenvolvedor da Apple.

O certificado expira após um ano. Quando isso acontece, você deve criar um novo certificado, que você então fornece ao AWS End User Messaging Push para renovar as entregas de notificações push. Para obter mais informações, consulte [Comunique-se APNs usando um certificado TLS na Ajuda](#) do Xcode.

## Pré-requisitos

Antes de usar qualquer canal de push, você precisa de credenciais válidas para o serviço de push. Para obter mais informações sobre como obter credenciais, consulte [Introdução ao AWS End User Messaging Push](#).

## Procedimento

Siga estas instruções para criar um aplicativo e ativar qualquer um dos canais push. Para concluir esse procedimento, você só precisa inserir o nome do aplicativo. Você pode ativar ou desativar qualquer um dos canais de push posteriormente.

1. Abra o console AWS End User Messaging Push em <https://console.aws.amazon.com/push-notifications/>.
2. Selecione Criar aplicativo.
3. Em Nome do aplicativo, insira o nome do seu aplicativo.
4. (Opcional) Siga esta etapa opcional para ativar o serviço Apple Push Notification (APNs).
  - a. Para o serviço Apple Push Notification (APNs), selecione Ativar.
  - b. Para o tipo de autenticação padrão, escolha uma das seguintes opções:
    - i. Se você escolher Credenciais chave, forneça as seguintes informações da sua conta de desenvolvedor da Apple. AWS O End User Messaging Push requer essas informações para criar tokens de autenticação.
      - ID de chave: o ID atribuído à sua chave de assinatura.
      - Identificador do pacote: o ID atribuído ao seu aplicativo iOS.
      - Identificador da equipe: o ID atribuído à sua equipe de conta de Desenvolvedor da Apple.
      - Chave de autenticação: o arquivo .p8 que você baixa da sua conta de desenvolvedor da Apple ao criar uma chave de autenticação.
    - ii. Se você escolher Credenciais do certificado, forneça as seguintes informações:
      - Certificado SSL: o arquivo .p12 do certificado TLS.
      - Senha do certificado: se você atribuiu uma senha ao certificado, insira-a aqui.
      - Tipo de certificado: selecione o tipo de certificado a ser usado.

5. (Opcional) Siga esta etapa opcional para ativar o Firebase Cloud Messaging (FCM).
  - a. Para Firebase Cloud Messaging (FCM), selecione Ativar.
  - b. Para o tipo de autenticação padrão, escolha uma das seguintes opções:
    - i. Em Credenciais de token (recomendado), escolha Escolher arquivos e, em seguida, escolha seu arquivo JSON de serviço.
    - ii. Em Credenciais chave, insira sua chave na chave de API.
6. (Opcional) Siga esta etapa opcional para ativar o Baidu Cloud Push.
  - a. Para o Baidu Cloud Push, selecione Ativar.
  - b. Em Chave de API, insira sua chave de API.
  - c. Em Chave secreta, insira sua chave secreta.
7. (Opcional) Siga esta etapa opcional para ativar o Amazon Device Messaging.
  - a. Para Amazon Device Messaging, selecione Ativar.
  - b. Em Client ID, insira seu ID de cliente.
  - c. Em Segredo do cliente, insira o segredo do seu cliente.
8. Selecione Criar aplicativo.

## Desativando canais push

Siga estas instruções para desativar qualquer um dos canais de push.

1. Abra o console AWS End User Messaging Push em <https://console.aws.amazon.com/push-notifications/>.
2. Escolha o aplicativo que contém suas credenciais de push.
3. (Opcional) Para o serviço Apple Push Notification (APNs), desmarque Ativar.
4. (Opcional) Para Firebase Cloud Messaging (FCM), desmarque Ativar.
5. (Opcional) Para o Baidu Cloud Push, desmarque a opção Ativar.
6. (Opcional) Para Amazon Device Messaging, desmarque Ativar.
7. Escolha Salvar alterações.

# Enviar uma mensagem

A API AWS End User Messaging Push pode enviar notificações push transacionais para identificadores de dispositivos específicos. Esta seção contém exemplos de código completos que você pode usar para enviar notificações push por meio da API AWS End User Messaging Push usando um AWS SDK.

Você pode usar esses exemplos para enviar notificações push por meio de qualquer serviço de notificação push compatível com AWS End User Messaging Push. Atualmente, o AWS End User Messaging Push é compatível com os seguintes canais: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push e Amazon Device Messaging (ADM).

Para obter mais exemplos de código sobre endpoints, segmentos e canais, consulte [Exemplos de código](#).

## Note

Ao enviar notificações push por meio do serviço Firebase Cloud Messaging (FCM), use o nome do serviço GCM em sua chamada para a API Push de mensagens de usuário AWS final. O serviço Google Cloud Messaging (GCM) foi descontinuado pelo Google em 10 de abril de 2018. No entanto, a AWS End User Messaging Push API usa o nome do GCM serviço para as mensagens enviadas por meio do serviço FCM para manter a compatibilidade com o código da API que foi escrito antes da descontinuação do serviço GCM.

## GCM (AWS CLI)

O exemplo a seguir usa [mensagens de envio](#) para enviar uma notificação push do GCM com o. AWS CLI *token* Substitua pelo token exclusivo do dispositivo e *611e3e3cdd47474c9c1399a50example* pelo identificador do seu aplicativo.

```
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"
  }
}
}

```

O exemplo a seguir usa [mensagens de envio](#) para enviar uma notificação push do GCM, usando todas as chaves legadas, com o. AWS CLI `token` substitua pelo token exclusivo do dispositivo e `611e3e3cdd47474c9c1399a50example` pelo identificador do seu aplicativo.

```

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 ], \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
```

O exemplo a seguir usa [send-messages](#) para enviar uma notificação push do GCM com carga útil de FCMv1 mensagem usando o. AWS CLI *token* Substitua pelo token exclusivo do dispositivo e *611e3e3cdd47474c9c1399a50example* pelo identificador do seu aplicativo.

```
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 \n \n "lang": \n "string",\n \n "renotify": false,\n \n "requireInteraction":
true,\n \n "silent": false,\n \n "tag": \n "tag",\n \n "timestamp": 1707259524964,\n
\n "title": \n "hello",\n \n "vibrate": [\n 100,\n 200,\n 300\n ]\n },\n \n "data": {\n
\n "data1": \n "priority message",\n \n "data2": \n "priority message",\n \n "data12":
\n "priority message",\n \n "data3": \n "priority message"\n }\n },\n \n "data": {\n
\n "data7": \n "priority message",\n \n "data5": \n "priority message",\n \n "data8":
\n "priority message",\n \n "data9": \n "priority message"\n }\n }\n \n }\n \n }",
  "TimeToLive" : 309744
}
},
"Addresses": {
  token: {
    "ChannelType": "GCM"
  }
}
}'
\ --region us-east-1

```

se estiver usando o `ImageUrl` campo para GCM, o pinpoint envia o campo como notificação de dados, com a chave `sendpinpoint.notification.imageUrl`, o que pode impedir que a imagem seja renderizada fora da caixa. Use `RawContent` ou adicione o tratamento das chaves de dados, como integrar seu aplicativo com AWS Amplify.

## Safari (AWS CLI)

Você pode usar o AWS End User Messaging Push para enviar mensagens para computadores macOS que usam o navegador Safari da Apple. Para enviar uma mensagem para o navegador Safari, você deve especificar o conteúdo bruto da mensagem e incluir um atributo específico na carga da mensagem. Você pode fazer isso [criando um modelo de notificação push com uma carga de mensagem bruta](#) ou especificando o conteúdo bruto da mensagem diretamente em uma mensagem de [campanha](#), no Guia do usuário do Amazon Pinpoint.

### Note

Esse atributo especial é necessário para enviar para laptops e computadores desktop macOS que usam o navegador Safari. Não é necessário para enviar para dispositivos iOS, como iPhones e iPads.

Para enviar uma mensagem para os navegadores Safari, você deve especificar a carga da mensagem bruta. A carga da mensagem bruta deve incluir uma matriz `url-args` dentro do

objeto aps. A matriz `url-args` é necessária para enviar notificações por push para o navegador Safari. No entanto, é aceitável que a matriz contenha um único elemento vazio.

O exemplo a seguir usa [mensagens de envio](#) para enviar uma notificação ao navegador Safari com o. AWS CLI `token` Substitua pelo token exclusivo do dispositivo e `611e3e3cdd47474c9c1399a50example` pelo identificador do seu aplicativo.

```
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 obter mais informações sobre as notificações por push do Safari, consulte [Configurar notificações por push do Safari](#) no site para desenvolvedores da Apple.

## APNS (AWS CLI)

O exemplo a seguir usa [mensagens de envio](#) para enviar uma notificação push do APNS com o. AWS CLI `token` Substitua pelo token exclusivo do dispositivo, `611e3e3cdd47474c9c1399a50example` pelo identificador do aplicativo e `GAME_INVITATION` por um identificador exclusivo.

```
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)

Use este exemplo para enviar notificações push usando o AWS SDK para JavaScript em Node.js. Este exemplo pressupõe que você já tenha instalado e configurado o SDK JavaScript em Node.js.

Esse exemplo também pressupõe que você esteja usando um arquivo de credenciais compartilhadas para especificar a chave de acesso e a chave de acesso secreta de um usuário existente do . Para obter mais informações, consulte [Configuração de credenciais](#) no AWS SDK para JavaScript o Guia do Desenvolvedor do Node.js.

```
'use strict';

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

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

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

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

// The application ID that you want to use when you send this
// message. Make sure that the push channel is enabled for the project that
```

```
// you choose.
var applicationId = 'ce796be37f32f178af652b26eexample';

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

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

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

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

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

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

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

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

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

return messageRequest
}

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

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

```

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

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

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

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

SendMessage()

```

## Python

Use este exemplo para enviar notificações por push usando o AWS SDK para Python (Boto3). Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para Python (Boto3).

Esse exemplo também pressupõe que você esteja usando um arquivo de credenciais compartilhadas para especificar a chave de acesso e a chave de acesso secreta de um usuário existente do . Para obter mais informações, consulte [Credenciais](#) na Referência da API do 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 adicionais do

- Para obter mais informações sobre modelos de canais push, consulte [Criação de modelos de notificação push](#) no Guia do usuário do Amazon Pinpoint.

# Recebendo notificações push em seu aplicativo

Os tópicos a seguir descrevem como modificar seu aplicativo Swift, Android, React Native ou Flutter para que ele receba notificações push.

## Tópicos

- [Configurar notificações por push do Swift](#)
- [Configurar notificações por push no Android](#)
- [Configurar notificações por push do Flutter](#)
- [Configurar notificações por push do React Native](#)
- [Crie um aplicativo no AWS End User Messaging Push](#)
- [Gerenciar notificações por push](#)

## Configurar notificações por push do Swift

As notificações push para aplicativos iOS são enviadas usando o serviço Apple Push Notification (APNs). Para enviar notificações por push para dispositivos iOS, crie um ID de aplicativo no portal do desenvolvedor da Apple e os certificados necessários. Você pode encontrar mais informações sobre como concluir essas etapas em [Configurar serviços de notificação push](#) na documentação do AWS Amplify.

## Trabalhando com APNs tokens

Como melhor prática, você deve desenvolver seu aplicativo para que os tokens de dispositivo dos clientes sejam gerados novamente quando o aplicativo for reinstalado.

Se um destinatário atualizar o dispositivo para uma nova versão principal do iOS (por exemplo, do iOS 12 para o iOS 13) e, posteriormente, reinstalar o aplicativo, o aplicativo gerará um novo token. Se o aplicativo não atualizar o token, o token mais antigo será usado para enviar a notificação. Como resultado, o serviço Apple Push Notification (APNs) rejeita a notificação, porque o token agora é inválido. Ao tentar enviar a notificação, você recebe uma mensagem de notificação de falha de APNs.

## Configurar notificações por push no Android

As notificações por push para aplicativos Android são enviadas usando o Firebase Cloud Messaging (FCM), que substitui o Google Cloud Messaging (GCM). Para poder enviar notificações por push para dispositivos Android, você deve obter credenciais do FCM. Você pode usar essas credenciais para criar um projeto Android e iniciar um aplicativo de exemplo que possa receber notificações por push. Você pode encontrar mais informações sobre como concluir essas etapas na seção [Notificações push](#) na documentação do AWS Amplify.

## Configurar notificações por push do Flutter

As notificações push para aplicativos Flutter são enviadas usando o Firebase Cloud Messaging (FCM) para Android e iOS. APNs Você pode encontrar mais informações sobre como concluir essas etapas na seção de notificações por push da [documentação do AWS Amplify Flutter](#).

## Configurar notificações por push do React Native

As notificações push para aplicativos React Native são enviadas usando o Firebase Cloud Messaging (FCM) para Android e APNs iOS. Você pode encontrar mais informações sobre como concluir essas etapas na seção Notificações push da documentação do [AWS Amplify JavaScript](#).

## Crie um aplicativo no AWS End User Messaging Push

Para começar a enviar notificações push no AWS End User Messaging Push, você precisa criar um aplicativo. Em seguida, você precisa habilitar os canais de notificação por push que você deseja usar, fornecendo as credenciais apropriadas.

Você pode criar novos aplicativos e configurar canais de notificação push usando o console AWS End User Messaging Push. Para obter mais informações, consulte [Criação de um aplicativo e ativação de canais push](#).

Você também pode criar e configurar o aplicativo usando a [API](#), um [AWS SDK](#) ou o [AWS Command Line Interface](#) (AWS CLI). Para criar um aplicativo, use o Apps recurso. Para configurar canais de notificação por push, use os seguintes recursos:

- [APNs canal](#) para enviar mensagens aos usuários de dispositivos iOS usando o serviço Apple Push Notification.

- [Canal ADM](#) para enviar mensagens para usuários de dispositivos Amazon Kindle Fire.
- [Canal Baidu](#) para enviar mensagens para usuários do Baidu.
- [Canal GCM](#) para enviar mensagens a dispositivos Android usando o Firebase Cloud Messaging (FCM), que substitui o Google Cloud Messaging (GCM).

## Gerenciar notificações por push

Depois de obter as credenciais necessárias para enviar notificações push, você pode atualizar seu aplicativo para que eles possam receber notificações push. Para obter mais informações, consulte [Notificações push — Introdução na documentação](#). AWS Amplify

# Excluir um aplicativo

Esse procedimento remove o aplicativo da sua conta e de todos os recursos do aplicativo.

## Contextual

### Aplicativo

Um aplicativo é um contêiner de armazenamento para todas as suas configurações de envio de mensagens de usuário AWS final. O aplicativo também armazena suas configurações de canais, campanhas e viagens do Amazon Pinpoint.

## Procedimento

1. Abra o console AWS End User Messaging Push em <https://console.aws.amazon.com/push-notifications/>.
2. Escolha um aplicativo e, em seguida, escolha Excluir.
3. Na janela Excluir aplicativo, digite **delete** e escolha Excluir.

### Important

Todos os canais, campanhas, viagens ou segmentos do Amazon Pinpoint também são excluídos.

## Práticas recomendadas

Mesmo tendo os melhores interesses dos seus clientes em mente, você ainda pode encontrar situações que afetam a capacidade de entrega das suas mensagens. As seções a seguir contêm recomendações para ajudar a garantir que as suas comunicações por push atinjam seu público-alvo.

### Enviar um grande volume de notificações por push

Antes de enviar um grande volume de notificações push, certifique-se de que sua conta esteja configurada para atender aos seus requisitos de taxa de transferência. Por padrão, todas as contas são configuradas para enviar 25.000 mensagens por segundo. Se precisar enviar mais de 25.000 mensagens em um segundo, solicite um aumento de cota. Para obter mais informações, consulte [Cotas para envio de mensagens push para usuários AWS finais](#).

Certifique-se de que sua conta esteja configurada corretamente com as credenciais de cada um dos provedores de notificação push que você planeja usar, como FCM ou APNs

Por fim, crie uma maneira de lidar com exceções. Cada serviço de notificação por push fornece mensagens de exceção diferentes. Para envios transacionais, você recebe um código de status principal de 200 para a chamada de API, com um código de status por endpoint de falha permanente 400 se o token de plataforma correspondente (por exemplo, FCM) ou certificado (por exemplo, APN) for determinado como inválido durante o envio de mensagens.

# Segurança no envio de mensagens push para o usuário AWS final

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam ao AWS End User Messaging Push, consulte [AWS Serviços no escopo do programa de conformidade AWS](#) .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AWS End User Messaging Push. Os tópicos a seguir mostram como configurar o AWS End User Messaging Push para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos de envio de mensagens de usuário AWS final.

## Tópicos

- [Proteção de dados no AWS End User Messaging Push](#)
- [Gerenciamento de identidade e acesso para AWS End User Messaging Push](#)
- [Validação de conformidade para envio de mensagens push para usuários AWS finais](#)
- [Resiliência no envio de mensagens para o usuário AWS final](#)
- [Segurança da infraestrutura no envio de mensagens push para o usuário AWS final](#)
- [Análise de configuração e vulnerabilidade](#)
- [Práticas recomendadas de segurança](#)

## Proteção de dados no AWS End User Messaging Push

O modelo de [responsabilidade AWS compartilhada O modelo](#) se aplica à proteção de dados no AWS End User Messaging Push. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para saber mais sobre a privacidade de dados, consulte as [Data Privacy FAQ](#). Para saber mais sobre a proteção de dados na Europa, consulte a postagem do blog [AWS Shared Responsibility Model and RGPD](#) no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para saber mais sobre os endpoints FIPS disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com AWS End User Messaging Push ou outro Serviços da AWS usando o console AWS CLI, a API ou AWS SDKs. Quaisquer dados inseridos em tags ou em

campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

## Criptografia de dados

**AWS Mensagens do usuário final** Os dados push são criptografados em trânsito e em repouso. Quando você envia dados para o AWS End User Messaging Push, ele criptografa os dados à medida que os recebe e os armazena. Quando você recupera dados do AWS End User Messaging Push, ele transmite os dados para você usando os protocolos de segurança atuais.

### Criptografia em repouso

**AWS O End User Messaging Push** criptografa todos os dados que ele armazena para você. Isso inclui dados de configuração, dados do usuário e do endpoint, dados analíticos e quaisquer dados que você adicione ou importe para o AWS End User Messaging Push. Para criptografar seus dados, o AWS End User Messaging Push usa chaves internas AWS Key Management Service (AWS KMS) que o serviço possui e mantém em seu nome. Nós mudamos essas chaves regularmente. Para obter informações sobre AWS KMS, consulte o [Guia do AWS Key Management Service desenvolvedor](#).

### Criptografia em trânsito

**AWS O End User Messaging Push** usa HTTPS e Transport Layer Security (TLS) 1.2 ou posterior para se comunicar com seus clientes e aplicativos. Para se comunicar com outros AWS serviços, o AWS End User Messaging Push usa HTTPS e TLS 1.2. Além disso, quando você cria e gerencia recursos push de mensagens de usuário AWS final usando o console, um AWS SDK ou o AWS Command Line Interface, todas as comunicações são protegidas usando HTTPS e TLS 1.2.

## Gerenciamento de chaves

Para criptografar seus dados do AWS End User Messaging Push, o AWS End User Messaging Push usa AWS KMS chaves internas que o serviço possui e mantém em seu nome. Nós mudamos essas chaves regularmente. Você não pode provisionar e usar suas próprias chaves AWS KMS ou outras chaves para criptografar dados que você armazena no AWS End User Messaging Push.

## Privacidade do tráfego entre redes

A privacidade do tráfego entre redes se refere à proteção de conexões e tráfego entre o AWS End User Messaging Push e seus clientes e aplicativos locais, e entre o AWS End User Messaging Push

e outros AWS recursos na mesma região. AWS Os recursos e práticas a seguir podem ajudá-lo a garantir a privacidade do tráfego entre redes para o AWS End User Messaging Push.

## Tráfego entre AWS o End User Messaging Push e clientes e aplicativos locais

Para estabelecer uma conexão privada entre AWS End User Messaging Push e clientes e aplicativos em sua rede local, você pode usar Direct Connect. Isso permite vincular a rede a um local do AWS Direct Connect usando um cabo Ethernet de fibra ótica padrão. Uma extremidade do cabo é conectada ao roteador. A outra extremidade está conectada a um Direct Connect roteador. Para obter mais informações, consulte [O que é o Direct Connect?](#) no Guia do usuário do Direct Connect .

Para ajudar a proteger o acesso ao AWS End User Messaging Push por meio do Published APIs, recomendamos que você cumpra os requisitos do AWS End User Messaging Push para chamadas de API. AWS O End User Messaging Push exige que os clientes usem o Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS), como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID de chave de acesso e uma chave de acesso secreta associada a um principal AWS Identity and Access Management (IAM) da sua AWS conta. Como alternativa, você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

## Tráfego entre AWS o End User Messaging Push e outros AWS recursos

Para proteger as comunicações entre o AWS End User Messaging Push e outros AWS recursos na mesma AWS região, o AWS End User Messaging Push usa HTTPS e TLS 1.2 por padrão.

## Gerenciamento de identidade e acesso para AWS End User Messaging Push

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos push de mensagens de usuário AWS final. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

## Tópicos

- [Público](#)
- [Autenticação com identidades](#)
- [Gerenciar o acesso usando políticas](#)
- [Como o AWS End User Messaging Push funciona com o IAM](#)
- [Exemplos de políticas baseadas em identidade para AWS End User Messaging Push](#)
- [Solução de problemas de mensagens de usuário AWS final Push: identidade e acesso](#)

## Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte [Solução de problemas de mensagens de usuário AWS final Push: identidade e acesso](#)).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte [Como o AWS End User Messaging Push funciona com o IAM](#))
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte [Exemplos de políticas baseadas em identidade para AWS End User Messaging Push](#))

## Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte [Como fazer login em sua Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte [AWS Signature Version 4 para solicitações de API](#) no Guia do usuário do IAM.

## Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

## Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte [O que é o IAM Identity Center?](#) no Guia do usuário do Centro de Identidade do AWS IAM .

## Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte [Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte [Casos de uso de usuários do IAM](#) no Guia do usuário do IAM.

## Perfis do IAM

Uma [perfil do IAM](#) é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função [mudando de um usuário para uma função do IAM \(console\)](#) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte [Métodos para assumir um perfil](#) no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

## Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

### Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

### Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte [Políticas de controle de serviço](#) no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

## Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

## Como o AWS End User Messaging Push funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao AWS End User Messaging Push, saiba quais recursos do IAM estão disponíveis para uso com o AWS End User Messaging Push.

## Recursos do IAM que você pode usar com o AWS End User Messaging Push

Recurso do IAM	AWS Suporte push para mensagens de usuário final
<a href="#">Políticas baseadas em identidade</a>	Sim
<a href="#">Políticas baseadas em atributos</a>	Sim
<a href="#">Ações de políticas</a>	Sim
<a href="#">Recursos de políticas</a>	Sim
<a href="#">Chaves de condição de políticas</a>	Sim
<a href="#">ACLs</a>	Não
<a href="#">ABAC (tags em políticas)</a>	Parcial
<a href="#">Credenciais temporárias</a>	Sim
<a href="#">Permissões de entidade principal</a>	Sim
<a href="#">Perfis de serviço</a>	Sim
<a href="#">Perfis vinculados a serviço</a>	Não

Para ter uma visão de alto nível de como o AWS End User Messaging Push e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte [AWS os serviços que funcionam com o IAM](#) no Guia do usuário do IAM.

### Políticas baseadas em identidade para envio de mensagens push para usuários AWS finais

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elemento de política JSON do IAM](#) no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para AWS End User Messaging Push

Para ver exemplos de políticas baseadas em identidade do AWS End User Messaging Push, consulte. [Exemplos de políticas baseadas em identidade para AWS End User Messaging Push](#)

## Políticas baseadas em recursos no AWS End User Messaging Push

Compatível com políticas baseadas em recursos: sim

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

## Ações de política para envio de mensagens push para usuários AWS finais

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações push de mensagens de usuário AWS final, consulte [Ações definidas por push de mensagens de usuário AWS final](#) na Referência de autorização de serviço.

As ações de política no AWS End User Messaging Push usam o seguinte prefixo antes da ação:

```
mobiletargeting
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

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

Para ver exemplos de políticas baseadas em identidade do AWS End User Messaging Push, consulte [Exemplos de políticas baseadas em identidade para AWS End User Messaging Push](#)

## Recursos de política para envio de mensagens push para usuários AWS finais

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (\*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de recursos do AWS End User Messaging Push e seus ARNs, consulte [Recursos definidos pelo AWS End User Messaging Push](#) na Referência de Autorização do Serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Ações definidas pelo AWS End User Messaging Push](#).

Para ver exemplos de políticas baseadas em identidade do AWS End User Messaging Push, consulte [Exemplos de políticas baseadas em identidade para AWS End User Messaging Push](#)

## Chaves de condição de política para AWS End User Messaging Push

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Para ver uma lista das chaves de condição do AWS End User Messaging Push, consulte [Chaves de condição para AWS End User Messaging Push](#) na Referência de Autorização do Serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas pelo envio de mensagens push do usuário AWS final](#).

Para ver exemplos de políticas baseadas em identidade do AWS End User Messaging Push, consulte [Exemplos de políticas baseadas em identidade para AWS End User Messaging Push](#)

## ACLs no envio de mensagens push para o usuário AWS final

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

## ABAC com envio de mensagens para o usuário AWS final

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte [Definir permissões com autorização do ABAC](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso por atributo \(ABAC\)](#) no Guia do usuário do IAM.

## Usando credenciais temporárias com AWS End User Messaging Push

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte [Credenciais de segurança temporárias no IAM](#) e [Serviços da Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

## Permissões principais entre serviços para AWS End User Messaging Push

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).

## Funções de serviço para AWS End User Messaging Push

Compatível com perfis de serviço: sim

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.

### Warning

Alterar as permissões de uma função de serviço pode interromper a funcionalidade AWS End User Messaging Push. Edite as funções de serviço somente quando AWS o End User Messaging Push fornecer orientação para fazer isso.

## Funções vinculadas ao serviço para AWS End User Messaging Push

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte [Serviços da AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

## Exemplos de políticas baseadas em identidade para AWS End User Messaging Push

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos de envio de mensagens para usuários AWS finais. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte [Criar políticas do IAM \(console\)](#) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo AWS End User Messaging Push, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição para envio de mensagens de usuário AWS final](#) na Referência de Autorização de Serviço.

### Tópicos

- [Práticas recomendadas de política](#)
- [Usando o console Push de mensagens para usuários AWS finais](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)

### Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do AWS End User Messaging Push em sua conta. Essas ações podem incorrer em custos para

sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte [Elementos da política JSON do IAM: condição](#) no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

## Usando o console Push de mensagens para usuários AWS finais

Para acessar o console AWS End User Messaging Push, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do AWS End User Messaging Push em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o console AWS End User Messaging Push, anexe também a política `AWSEndUserMessaging` AWS gerenciada às entidades. Para obter informações, consulte [Adicionar permissões a um usuário](#) no Guia do usuário do IAM.

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSEndUserMessaging",
      "Effect": "Allow",
      "Action": [
        "mobiletargeting:CreateApp",
        "mobiletargeting:GetApp",
        "mobiletargeting:GetApps",
        "mobiletargeting>DeleteApp",
        "mobiletargeting:GetChannels",
        "mobiletargeting:GetApnsChannel",
        "mobiletargeting:GetApnsVoipChannel",
        "mobiletargeting:GetApnsVoipSandboxChannel",
        "mobiletargeting:GetApnsSandboxChannel",
        "mobiletargeting:GetAdmChannel",
        "mobiletargeting:GetBaiduChannel",
        "mobiletargeting:GetGcmChannel",
        "mobiletargeting:UpdateApnsChannel",
        "mobiletargeting:UpdateApnsVoipChannel",
        "mobiletargeting:UpdateApnsVoipSandboxChannel",
        "mobiletargeting:UpdateBaiduChannel",
```

```

        "mobiletargeting:UpdateGcmChannel",
        "mobiletargeting:UpdateAdmChannel"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

## Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
}
]
```

## Solução de problemas de mensagens de usuário AWS final Push: identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS End User Messaging Push e IAM.

### Tópicos

- [Não estou autorizado a realizar uma ação no AWS End User Messaging Push](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos de envio de mensagens de usuário AWS final](#)

### Não estou autorizado a realizar uma ação no AWS End User Messaging Push

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM mateojackson tenta usar o console para visualizar detalhes sobre um atributo *my-example-widget* fictício, mas não tem as permissões `mobiletargeting:GetWidget` fictícias.

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

Nesse caso, a política do usuário mateojackson deve ser atualizada para permitir o acesso ao recurso *my-example-widget* usando a ação `mobiletargeting:GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS End User Messaging Push.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para realizar uma ação no AWS End User Messaging Push. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Quero permitir que pessoas fora da minha acessem meus Conta da AWS recursos de envio de mensagens de usuário AWS final

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o AWS End User Messaging Push oferece suporte a esses recursos, consulte [Como o AWS End User Messaging Push funciona com o IAM](#).
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte [Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui](#) no Guia do usuário do IAM.

- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Como fornecer acesso Contas da AWS a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

## Validação de conformidade para envio de mensagens push para usuários AWS finais

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a [documentação AWS de segurança](#).

## Resiliência no envio de mensagens para o usuário AWS final

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Além da infraestrutura AWS global, o AWS End User Messaging Push oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

## Segurança da infraestrutura no envio de mensagens push para o usuário AWS final

Como um serviço gerenciado, o AWS End User Messaging Push é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper [Amazon Web Services: Overview of Security Processes](#).

Você usa chamadas de API AWS publicadas para acessar o AWS End User Messaging Push pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

## Análise de configuração e vulnerabilidade

Como um serviço gerenciado, o AWS End User Messaging Push é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper [Amazon Web Services: Visão geral dos processos de segurança](#). Isso significa que AWS gerencia e executa tarefas e procedimentos básicos de segurança para fortalecer, corrigir, atualizar e, de outra forma, manter a infraestrutura subjacente de sua conta e recursos. Esses procedimentos foram revisados e certificados por terceiros certificados.

## Práticas recomendadas de segurança

Use contas de AWS Identity and Access Management (IAM) para controlar o acesso às operações da API, especialmente operações que criam, modificam ou excluem recursos. Para a API do , esses recursos incluem projetos, campanhas e jornadas.

- Crie um usuário individual do IAM para cada pessoa que gerencia recursos do , incluindo você mesmo. Não use credenciais AWS raiz para gerenciar recursos.

- Conceda a cada usuário o conjunto mínimo de permissões necessárias para realizar suas funções.
- Use grupos do IAM para gerenciar efetivamente permissões para vários usuários.
- Mude suas credenciais do IAM regularmente.

Para obter mais informações sobre a segurança, consulte [Segurança no envio de mensagens push para o usuário AWS final](#). Para obter mais informações sobre o IAM, consulte [AWS Identity and Access Management](#). Para obter informações sobre as práticas recomendadas do IAM, acesse [Melhores práticas do IAM](#).

# Monitorando AWS o push de mensagens do usuário final

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho do AWS End User Messaging Push e de suas outras soluções da AWS. A AWS fornece as seguintes ferramentas de monitoramento para monitorar o envio de mensagens do usuário AWS final, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos em que você executa AWS em tempo real. Você pode coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode CloudWatch rastrear o uso da CPU ou outras métricas de suas EC2 instâncias da Amazon e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus arquivos de log de EC2 instâncias da Amazon e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).
- A Amazon EventBridge pode ser usada para automatizar seus AWS serviços e responder automaticamente a eventos do sistema, como problemas de disponibilidade de aplicativos ou alterações de recursos. Os eventos dos AWS serviços são entregues quase EventBridge em tempo real. Você pode escrever regras simples para determinar quais eventos são do seu interesse, e as ações automatizadas a serem tomadas quando um evento corresponder à regra. Para obter mais informações, consulte o [Guia EventBridge do usuário da Amazon](#).
- AWS CloudTrail captura chamadas de API e eventos relacionados feitos por ou em nome de sua AWS conta e entrega os arquivos de log para um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#).

## Monitorando AWS o envio de mensagens push para o usuário final com a Amazon CloudWatch

Você pode monitorar o AWS End User Messaging Push usando CloudWatch, que coleta dados brutos e os processa em métricas legíveis, quase em tempo real. Essas estatísticas são mantidas

por 15 meses, de maneira que você possa acessar informações históricas e ter uma perspectiva melhor de como o aplicativo web ou o serviço está se saindo. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).

Para obter uma lista de métricas e dimensões, consulte [Monitoramento do Amazon Pinpoint com o Guia do CloudWatch usuário do Amazon Pinpoint](#).

## Registrando chamadas de Push API de mensagens de usuário AWS final usando AWS CloudTrail

AWS O End User Messaging Push é integrado ao End User Messaging Push AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um AWS serviço no AWS End User Messaging Push. CloudTrail captura todas as chamadas de API para AWS End User Messaging Push como eventos. As chamadas capturadas incluem chamadas do console AWS End User Messaging Push e chamadas de código para as operações da API AWS End User Messaging Push. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para AWS End User Messaging Push. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao AWS End User Messaging Push, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

## AWS Mensagens para o usuário final Envie informações para CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando a atividade ocorre no AWS End User Messaging Push, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes no seu Conta da AWS. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo dos eventos em seu Conta da AWS, incluindo eventos do AWS End User Messaging Push, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar

outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [CloudTrail serviços e integrações suportados](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [Recebendo arquivos de CloudTrail log de várias contas](#)

Todas as ações do AWS End User Messaging Push são registradas CloudTrail e documentadas na [Referência da API AWS End User Messaging Push](#). Por exemplo, chamadas para o `GetAdmChannel` `UpdateApnsChannel` e `GetApnsVoipChannel` as ações geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário root ou AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte [Elemento userIdentity do CloudTrail](#).

## Compreendendo as entradas do arquivo de log push de mensagens de usuário AWS final

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

## Acesse AWS o End User Messaging Push usando um endpoint de interface ()AWS PrivateLink

Você pode usar AWS PrivateLink para criar uma conexão privada entre sua VPC e o AWS End User Messaging Push. Você pode acessar o AWS End User Messaging Push como se estivesse em sua VPC, sem o uso de um gateway de internet, dispositivo NAT, conexão VPN ou conexão. Direct Connect As instâncias em sua VPC não precisam de endereços IP públicos para acessar o AWS End User Messaging Push.

Estabeleça essa conectividade privada criando um endpoint de interface, habilitado pelo AWS PrivateLink. Criaremos um endpoint de interface de rede em cada sub-rede que você habilitar para o endpoint de interface. Essas são interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao AWS End User Messaging Push.

Para obter mais informações, consulte [Acesso Serviços da AWS por meio AWS PrivateLink](#) do AWS PrivateLink Guia.

## Considerações sobre o envio de mensagens AWS push para o usuário final

Antes de configurar um endpoint de interface para AWS End User Messaging Push, analise [as Considerações](#) no AWS PrivateLink Guia.

AWS O End User Messaging Push suporta a realização de chamadas para todas as suas ações de API por meio do endpoint da interface.

As políticas de endpoint de VPC não são compatíveis com o AWS End User Messaging Push. Por padrão, o acesso total ao AWS End User Messaging Push é permitido por meio do endpoint da interface. Como alternativa, você pode associar um grupo de segurança às interfaces de rede do endpoint para controlar o tráfego para o AWS End User Messaging Push por meio do endpoint da interface.

# Crie um endpoint de interface para AWS End User Messaging Push

Você pode criar um endpoint de interface para AWS End User Messaging Push usando o console Amazon VPC ou AWS Command Line Interface o AWS CLI(). Para obter mais informações, consulte [Criar um endpoint de interface](#) no Guia do usuário do AWS PrivateLink .

Crie um endpoint de interface para AWS End User Messaging Push usando o seguinte nome de serviço:

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

Se você habilitar o DNS privado para o endpoint da interface, poderá fazer solicitações de API ao AWS End User Messaging Push usando seu nome DNS regional padrão. Por exemplo, `com.amazonaws.us-east-1.pinpoint`.

## Crie uma política de endpoint para seu endpoint de interface.

Uma política de endpoint é um recurso do IAM que você pode anexar ao endpoint de interface. A política de endpoint padrão permite acesso total ao AWS End User Messaging Push por meio do endpoint da interface. Para controlar o acesso permitido ao AWS End User Messaging Push de sua VPC, anexe uma política de endpoint personalizada ao endpoint da interface.

Uma política de endpoint especifica as seguintes informações:

- As entidades principais que podem realizar ações (Contas da AWS, usuários do IAM e perfis do IAM).
- As ações que podem ser realizadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte [Controlar o acesso aos serviços usando políticas de endpoint](#) no Guia do AWS PrivateLink .

Exemplo: política de VPC endpoint para ações push de mensagens de usuário AWS final

Veja a seguir um exemplo de uma política de endpoint personalizado. Quando você anexa essa política ao seu endpoint de interface, ela concede acesso às ações push de mensagens do usuário AWS final listadas para todos os diretores em todos os recursos.

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

## Cotas para envio de mensagens push para usuários AWS finais

Você Conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada AWS serviço. A menos que especificado de outra forma, cada cota é específica da região . É possível solicitar aumentos para algumas cotas e outras cotas não podem ser aumentadas.

Para ver as cotas do AWS End User Messaging Push, abra o console [Service Quotas](#). No painel de navegação, escolha os serviços da AWS e selecione Amazon Pinpoint.

Sua conta da AWS tem as seguintes cotas relacionadas ao AWS End User Messaging Push.

Recurso	Cota padrão	Qualificada para aumento
O número máximo de notificações por push que podem ser enviadas por segundo em uma campanha	25.000 notificações por segundo	Sim, use o console <a href="#">Service Quotas</a>
Tamanho da carga da mensagem no Amazon Device Messaging (ADM)	6 KB por mensagem	Não
Tamanho da carga útil da mensagem do serviço Apple Push Notification (APNs)	4 KB por mensagem	Não
APNs tamanho da carga útil da mensagem do sandbox	4 KB por mensagem	Não
Tamanho da carga da mensagem no Baidu Cloud Push	4 KB por mensagem	Não
Tamanho da carga da mensagem do Firebase Cloud Messaging (FCM)	4 KB por mensagem	Não

# Histórico de documentos do Guia do usuário do AWS End User Messaging Push

A tabela a seguir descreve as versões da documentação do AWS End User Messaging Push.

Alteração	Descrição	Data
<a href="#">Lançamento inicial</a>	Versão inicial do Guia do usuário do AWS End User Messaging Push	24 de julho de 2024

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.