Manual do usuário

AWS Push de mensagens para o usuário final

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

Table of Contents

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?

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:

- Mensagens na nuvem do Firebase () FCM
- Serviço de notificação push da Apple (APNs)

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

- Baidu Cloud Push
- Mensagens de dispositivos da Amazon (ADM)

Acessando o envio de mensagens push para o usuário AWS final

Explique resumidamente as diferentes maneiras de obter acesso ao serviço, seja por console ouAPI. CLI

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 AWS Management Console.

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 meio de HTTP ouHTTPS, 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 Amazon Pinpoint API 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 que você possa usar AWS Mensagens push para o usuário final Para enviar notificações push para seu aplicativo, primeiro você precisa obter um Conta da AWS com IAM permissões suficientes. Esse Conta da AWS também pode ser usado para outros serviços no AWS ecossistema.

Tópicos

Inscreva-se para um Conta da AWS

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

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 e inserir um código de verificação no teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário root tem acesso a todos 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, habilitar AWS IAM Identity Center 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 no AWS Management Consolecomo proprietário da conta, escolhendo o usuário root e inserindo seu Conta da AWS endereço de e-mail. Na próxima página, insira sua senha.

   Para obter ajuda para fazer login usando o usuário root, consulte Como fazer login como usuário root no Início de Sessão da AWS Guia do usuário.
2. Ative a autenticação multifator (MFA) para seu usuário root.

   Para obter instruções, consulte Habilitar um MFA dispositivo virtual para seu Conta da AWS usuário root (console) no Guia do IAM usuário.

Criar um usuário com acesso administrativo

1. Ative o IAM Identity Center.

   Para obter instruções, consulte Habilitando AWS IAM Identity Center no AWS IAM Identity Center Guia do usuário.
2. No IAM Identity Center, conceda acesso administrativo a um usuário.

   Para um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como sua fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no AWS IAM Identity Center Guia do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para entrar com seu usuário do IAM Identity Center, use o login URL que foi enviado ao seu endereço de e-mail quando você criou o usuário do IAM Identity Center.

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

Atribuir acesso a usuários adicionais

1. No IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

Para obter instruções, consulte Criar um conjunto de permissões no AWS IAM Identity Center Guia do usuário.

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

   Para obter instruções, consulte Adicionar grupos no AWS IAM Identity Center Guia do usuário.

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 para se autenticar APNs para cada notificação push que você enviar. 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 TLS certificado 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 TLS certificado 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, consulteIntrodução 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 Create application (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:
            - SSLcertificate — O arquivo.p12 do seu TLS certificado.
            - 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 JSON arquivo de serviço.
        ii. Em Credenciais chave, insira sua chave na APIchave.

6. (Opcional) Siga esta etapa opcional para ativar o Baidu Cloud Push.

    a. Para o Baidu Cloud Push, selecione Ativar.
    b. Para APIchave, insira sua API chave.
    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 Create application (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

O AWS End User Messaging Push API 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 do AWS End User Messaging Push API 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 o AWS End User Messaging PushAPI. O serviço Google Cloud Messaging (GCM) foi descontinuado pelo Google em 10 de abril de 2018. No entanto, o AWS End User Messaging Push API usa o nome do `GCM` serviço para as mensagens enviadas por meio do FCM serviço a fim de manter a compatibilidade com o API código que foi escrito antes da descontinuação do GCM serviço.

GCM (AWS CLI)

O exemplo a seguir usa mensagens de envio para enviar uma notificação GCM push com o. AWS CLI Substituir *`token`* com o token exclusivo do dispositivo e *`611e3e3cdd47474c9c1399a50example`* com o 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 GCM push, usando todas as chaves legadas, com o. AWS CLI Substituir *token* com o token exclusivo do dispositivo e *611e3e3cdd47474c9c1399a50example* com o 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 \"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 GCM push com carga de FCMv1 mensagem usando o. AWS CLI Substituir *token* com o token exclusivo do dispositivo e *611e3e3cdd47474c9c1399a50example* com o 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 \"lang\": \"string\",\n \"renotify\": false,\n \"requireInteraction\":
 true,\n \"silent\": false,\n \"tag\": \"tag\",\n \"timestamp\": 1707259524964,\n
 \"title\": \"hello\",\n \"vibrate\": [\n 100,\n 200,\n 300\n ]\n },\n \"data\": {\n
 \"data1\": \"priority message\",\n \"data2\": \"priority message\",\n \"data12\":
 \"priority message\",\n \"data3\": \"priority message\"\n }\n },\n \"data\": {\n
 \"data7\": \"priority message\",\n \"data5\": \"priority message\",\n \"data8\":
 \"priority message\",\n \"data9\": \"priority message\"\n }\n }\n \n}\n}",
      "TimeToLive" : 309744
    }
  },
  "Addresses": {
    token: {
      "ChannelType":"GCM"
    }
   }
}'
\ --region us-east-1
```

se estiver usando `ImageUrl` field forGCM, o pinpoint envia o campo como notificação de dados, com a chave sendo`pinpoint.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 a integração do 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 iPads e.

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 Substituir *token* com o token exclusivo do dispositivo e *611e3e3cdd47474c9c1399a50example* com o 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 APNS push com o. AWS CLI Substituir *token* com o token exclusivo do dispositivo, *611e3e3cdd47474c9c1399a50example* com o identificador do seu aplicativo e *GAME_INVITATION* com 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 for JavaScript em Node.js. Este exemplo pressupõe que você já tenha instalado e configurado o SDK for JavaScript no 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 Formulário no AWS SDKGuia do Desenvolvedor do Node.js. JavaScript

```
'use strict';

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

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

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

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

// The application ID that you want to use when you send this
```

```
// message. Make sure that the push channel is enabled for the project that
// you choose.
var applicationId = 'ce796be37f32f178af652b26eexample';

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

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

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

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

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

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

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

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

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

  return messageRequest
}

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

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

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

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

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

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

SendMessage()
```

Python

Use este exemplo para enviar notificações por push usando o AWS SDK for Python (Boto3). Este 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 AWS SDKpara Python (APIBoto3).

```
import json
import boto3
from botocore.exceptions import ClientError

# The AWS Region that you want to use to send the message. For a list of
# AWS Regions where the API is available
region = "us-east-1"
```

```
# The title that appears at the top of the push notification.
title = "Test message sent from End User Messaging Push."

# The content of the push notification.
message = ("This is a sample message sent from End User Messaging Push by using the
 "
           "AWS SDK for Python (Boto3).")

# The application ID to use when you send this message.
# Make sure that the push channel is enabled for the project or application
# that you choose.
application_id = "ce796be37f32f178af652b26eexample"

# A dictionary that contains the unique token of the device that you want to send
 the
# message to, and the push service that you want to use to send the message.
recipient = {
    "token": "a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0",
    "service": "GCM"
    }

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

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

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

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

# Boolean that specifies whether the notification is sent as a silent
# notification (a notification that doesn't display on the recipient's device).
```

```
silent = False

# Set the MessageType based on the values in the recipient variable.
def create_message_request():

    token = recipient["token"]
    service = recipient["service"]

    if service == "GCM":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'GCM'
                }
            },
            'MessageConfiguration': {
                'GCMMessage': {
                    'Action': action,
                    'Body': message,
                    'Priority' : priority,
                    'SilentPush': silent,
                    'Title': title,
                    'TimeToLive': ttl,
                    'Url': url
                }
            }
        }
    elif service == "APNS":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'APNS'
                }
            },
            'MessageConfiguration': {
                'APNSMessage': {
                    'Action': action,
                    'Body': message,
                    'Priority' : priority,
                    'SilentPush': silent,
                    'Title': title,
                    'TimeToLive': ttl,
                    'Url': url
                }
```

```
            }
        }
    elif service == "BAIDU":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'BAIDU'
                }
            },
            'MessageConfiguration': {
                'BaiduMessage': {
                    'Action': action,
                    'Body': message,
                    'SilentPush': silent,
                    'Title': title,
                    'TimeToLive': ttl,
                'Url': url
                }
            }
        }
    elif service == "ADM":
        message_request = {
            'Addresses': {
                token: {
                    'ChannelType': 'ADM'
                }
            },
            'MessageConfiguration': {
                'ADMMessage': {
                    'Action': action,
                    'Body': message,
                    'SilentPush': silent,
                    'Title': title,
                    'Url': url
                }
            }
        }
    else:
        message_request = None

    return message_request

# Show a success or failure message, and provide the response from the API.
def show_output(response):
```

```
    if response['MessageResponse']['Result'][recipient["token"]]['DeliveryStatus']
 == "SUCCESSFUL":
        status = "Message sent! Response information:\n"
    else:
        status = "The message wasn't sent. Response information:\n"
    print(status, json.dumps(response,indent=4))

# Send the message through the appropriate channel.
def send_message():

    token = recipient["token"]
    service = recipient["service"]
    message_request = create_message_request()

    client = boto3.client('pinpoint',region_name=region)

    try:
        response = client.send_messages(
            ApplicationId=application_id,
            MessageRequest=message_request
        )
    except ClientError as e:
        print(e.response['Error']['Message'])
    else:
        show_output(response)

send_message()
```

Recursos adicionais

- 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

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 deAPNs.

Configurar as notificações por push em Android

As notificações push para aplicativos Android são enviadas usando o Firebase Cloud Messaging (FCM), que substitui o Google Cloud Messaging (GCM). Antes de enviar notificações push para

dispositivos Android, você precisa obter FCM as credenciais. 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 ter mais informações, consulte Criação de um aplicativo e ativação de canais push.

Você também pode criar e configurar o aplicativo usando o APIAWS SDK, an 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:

- APNscanal para enviar mensagens aos usuários de dispositivos iOS usando o serviço Apple Push Notification.
- ADMcanal para enviar mensagens para usuários de dispositivos Amazon Kindle Fire.
- Canal Baidu para enviar mensagens para usuários do Baidu.
- GCMcanal para enviar mensagens para 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 ter 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 ouAPNs.

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 API chamada, com um código de status por endpoint de 400 falha permanente se o token de plataforma correspondente (por exemplo,FCM) ou certificado (por exemplo,APN) for determinado como inválido durante o envio de mensagens.

Segurança em AWS Push de mensagens para o usuário final

Segurança na nuvem em AWS é a maior prioridade. Como um 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 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 funciona AWS 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 do AWS Programas de conformidade . Para saber mais sobre os programas de conformidade que se aplicam ao AWS Push de mensagens para o usuário final, consulte AWS Serviços no escopo do Programa de Conformidade .
- 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 AWS Push de mensagens para o usuário final. Os tópicos a seguir mostram como configurar AWS Push de mensagens para o usuário final para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam você a monitorar e proteger seu AWS Recursos push de mensagens para o usuário final.

Tópicos

Proteção de dados em AWS Push de mensagens para o usuário final

A ferramenta AWS modelo de responsabilidade compartilhada modelo se aplica à proteção de dados em AWS Push de mensagens para o usuário final. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todas as 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 do Serviços da AWS que você usa. Para obter mais informações sobre privacidade de dados, consulte Privacidade de dados FAQ. Para obter informações sobre proteção de dados na Europa, consulte AWS Modelo de responsabilidade compartilhada e postagem no GDPR blog sobre o AWS Blog de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS credenciais e configure usuários individuais com AWS IAM Identity Center 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 a autenticação multifator (MFA) com cada conta.
- UseSSL/TLSpara se comunicar com AWS recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Trabalhando com CloudTrail trilhas no AWS CloudTrail Guia do usuário.
- Use AWS soluções de criptografia, junto com todos os controles de segurança padrão dentro Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou umaAPI, use um FIPS endpoint. Para obter mais informações sobre os FIPS endpoints disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com AWS Mensagens push para o usuário final ou

outros Serviços da AWS usando o consoleAPI, AWS CLI, ou AWS SDKs. Quaisquer dados inseridos em tags ou 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 altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

Criptografia de dados

AWS Mensagens do usuário final Os dados push são criptografados em trânsito e em repouso. Quando você envia dados para AWS End User Messaging Push, ele criptografa os dados à medida que os recebe e os armazena. Quando você recupera dados de 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 de análise e quaisquer dados que você adicione ou importe para AWS Push de mensagens para o usuário final. Para criptografar seus dados, AWS O envio de mensagens push para o usuário final é interno AWS Key Management Service (AWS KMS) chaves que o serviço possui e mantém em seu nome. Nós mudamos essas chaves regularmente. Para obter mais informações sobre AWS KMS, veja o AWS Key Management Service Guia do 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, AWS Usos do Push de mensagens para o usuário final HTTPS e TLS 1.2. Além disso, quando você cria e gerencia AWS Mensagens para o usuário final Envie recursos 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 seu AWS Dados push de mensagens para o usuário final, AWS O envio de mensagens push para o usuário final é interno AWS KMS chaves que o serviço possui e mantém em seu nome. Nós mudamos essas chaves regularmente. Você não pode provisionar e usar o seu

próprio AWS KMS ou outras chaves para criptografar os dados que você armazena em AWS Push de mensagens para o usuário final.

Privacidade do tráfego entre redes

A privacidade do tráfego entre redes se refere à proteção de conexões e tráfego entre AWS End User Messaging Push e seus clientes e aplicativos locais, e entre AWS Mensagens para o usuário final, push e outros AWS recursos no mesmo AWS Região. Os recursos e práticas a seguir podem ajudá-lo a garantir a privacidade do tráfego entre redes para AWS Push de mensagens para o usuário final.

Tráfego entre AWS Clientes e aplicativos push de mensagens de usuário final e locais

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

Para ajudar a proteger o acesso ao AWS Mensagens de usuário final publicadas por push throughAPIs, recomendamos que você cumpra com AWS Requisitos de envio de mensagens de usuário final para API chamadas. 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 direto perfeito (PFS), como Ephemeral Diffie-Hellman () ou Elliptic Curve Diffie-Hellman Ephemeral (). DHE ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um AWS Identity and Access Management (IAM) principal para o seu 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 Mensagens para o usuário final, push e outros AWS recursos

Para proteger as comunicações entre AWS Mensagens para o usuário final, push e outros AWS recursos no mesmo AWS Região, AWS O End User Messaging Push usa HTTPS e TLS 1.2 por padrão.

Gerenciamento de identidade e acesso para AWS Push de mensagens para o usuário final

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

Tópicos

Público

Como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz em AWS Push de mensagens para o usuário final.

Usuário do serviço — Se você usar o AWS Serviço push de mensagens de usuário final para fazer seu trabalho e, em seguida, seu administrador fornece as credenciais e permissões de que você precisa. À medida que você usa mais AWS Recursos de envio de mensagens para o usuário final Para fazer seu trabalho, você pode precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um recurso no AWS Push de mensagens para o usuário final, consulteSolução de problemas AWS Mensagens para o usuário final Envie identidade e acesso.

Administrador de serviços — Se você é responsável por AWS Mensagens para o usuário final Envie recursos em sua empresa, você provavelmente tem acesso total aos AWS Push de mensagens para o usuário final. É seu trabalho determinar quais AWS Recursos e recursos do End User Messaging Push que seus usuários do serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações

nesta página para entender os conceitos básicos doIAM. Para saber mais sobre como sua empresa pode usar IAM com AWS Push de mensagens para o usuário final, consulteComo AWS O End User Messaging Push funciona com IAM.

IAMadministrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso a AWS Push de mensagens para o usuário final. Para ver um exemplo AWS Mensagens para o usuário final Envie políticas baseadas em identidade que você pode usar emIAM, consulte. Exemplos de políticas baseadas em identidade para AWS Push de mensagens para o usuário final

Autenticando com identidades

A autenticação é como você faz login no AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado em AWS) como o Usuário raiz da conta da AWS, como IAM usuário ou assumindo uma IAM função.

Você pode fazer login em AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. Quando você acessa AWS ao usar a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou o AWS portal de acesso. Para obter mais informações sobre como fazer login no AWS, veja Como fazer login no seu Conta da AWS no Início de Sessão da AWS Guia do usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte Assinatura AWS APIsolicitações no Guia do IAM usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no AWS IAM Identity Center Guia do usuário e uso da autenticação multifator (MFA) em AWS no IAM Guia do usuário.

Conta da AWS usuário raiz

Quando você cria um Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS e recursos na conta. Essa identidade é chamada de Conta da AWS usuário root e é acessado fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para ver a lista completa de tarefas que exigem que você faça login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do IAM usuário.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, 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 de usuários corporativo, um provedor de identidade da web, o AWS Directory Service, o diretório do Identity Center ou qualquer usuário que acesse Serviços da AWS usando credenciais fornecidas por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, eles assumem funções, e as funções fornecem credenciais temporárias.

Para gerenciamento de acesso centralizado, recomendamos que você use AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todas as suas Contas da AWS e aplicativos. Para obter informações sobre o IAM Identity Center, consulte O que é o IAM Identity Center? no AWS IAM Identity Center Guia do usuário.

Grupos e usuários do IAM

Um IAMusuário é uma identidade dentro do seu Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo no Guia do IAMusuário.

Um IAMgrupo é uma identidade que especifica uma coleção de IAM usuários. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de

uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um IAM usuário (em vez de uma função) no Guia do IAM usuário.

IAMfunções

Um IAMpapel é uma identidade dentro de você Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console trocando de papéis. Você pode assumir uma função chamando um AWS CLI ou AWS APIoperação ou usando um personalizadoURL. Para obter mais informações sobre métodos de uso de funções, consulte Usando IAM funções no Guia IAM do usuário.

IAMfunções com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter informações sobre funções para federação, consulte Criação de uma função para um provedor de identidade terceirizado no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em. IAM Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões no AWS IAM Identity Center Guia do usuário.
- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

- Acesso entre serviços — Alguns Serviços da AWS use recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um IAM usuário ou uma função para realizar ações no AWS, você é considerado diretor. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FASusa as permissões do diretor chamando um AWS service (Serviço da AWS), combinado com a solicitação AWS service (Serviço da AWS) para fazer solicitações para serviços posteriores. FASas solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou recursos para concluir. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte Encaminhar sessões de acesso.
  - Função de serviço — Uma função de serviço é uma IAMfunção que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamenteIAM. Para obter mais informações, consulte Criação de uma função para delegar permissões a um AWS service (Serviço da AWS) no IAM Guia do usuário.
  - Função vinculada a serviços — Uma função vinculada a serviços é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir a função de realizar uma ação em seu nome. As funções vinculadas ao serviço aparecem em seu Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.
- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI ou AWS APIsolicitações. Isso é preferível ao armazenamento de chaves de acesso na EC2 instância. Para atribuir um AWS Ao atribuir a uma EC2 instância e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância que é anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon no Guia IAM do usuário.

Para saber se usar IAM funções ou IAM usuários, consulte Quando criar uma IAM função (em vez de um usuário) no Guia do IAM usuário.

Gerenciando acesso usando políticas

Você controla o acesso em AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto em AWS que, quando associados a uma identidade ou recurso, definem suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada em AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte Visão geral das JSON políticas no Guia IAM do usuário.

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

Por padrão, usuários e funções não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console, o AWS CLI, ou o AWS API.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. 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 Criação de IAM políticas no Guia do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas gerenciadas incluem AWS políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolha entre políticas gerenciadas e políticas em linha no Guia do IAMusuário.

Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS.

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

Listas de controle de acesso (ACLs)

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

Amazon S3, AWS WAF, e a Amazon VPC são exemplos de serviços que oferecem suporteACLs. Para saber maisACLs, consulte a visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAMusuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para IAM entidades no Guia IAM do usuário.

- Políticas de controle de serviço (SCPs) — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades em contas de membros, incluindo cada Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations eSCPs, consulte Políticas de controle de serviços no AWS Organizations Guia do usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia IAM do usuário.

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 determina se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte Lógica de avaliação de políticas no Guia IAM do usuário.

Como AWS O End User Messaging Push funciona com IAM

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

IAMrecursos que você pode usar com AWS Push de mensagens para o usuário final

IAMrecurso	AWS Suporte push para mensagens de usuário final
Políticas baseadas em identidade	Sim
Políticas baseadas em atributos	Sim

IAMrecurso	AWS Suporte push para mensagens de usuário final
Ações das políticas	Sim
Atributos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC(tags nas políticas)	Parcial
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados ao serviço	Não

Para obter uma visão de alto nível de como AWS Mensagens para o usuário final, push e outros AWS os serviços funcionam com a maioria dos IAM recursos, consulte AWS serviços que funcionam com IAM o Guia IAM do Usuário.

Políticas baseadas em identidade para AWS Push de mensagens para o usuário final

Compatível com políticas baseadas em identidade: Sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. 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 Criação de IAM políticas no Guia do IAMusuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode

usar em uma JSON política, consulte a referência IAM JSON de elementos de política no Guia IAM do usuário.

Exemplos de políticas baseadas em identidade para AWS Push de mensagens para o usuário final

Para ver exemplos de AWS Mensagens de usuário final Push políticas baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS Push de mensagens para o usuário final

Políticas baseadas em recursos dentro AWS Push de mensagens para o usuário final

Compatível com políticas baseadas em recursos: sim

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS.

Para habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso estão em condições diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

Ações políticas para AWS Push de mensagens para o usuário final

Compatível com ações de políticas: Sim

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

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que as associadas AWS APIoperação. Há algumas exceções, como ações somente de permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de AWS Ações push de mensagens para o usuário final, consulte Ações definidas por AWS Envio de mensagens para o usuário final na referência de autorização de serviço.

Ações políticas em AWS O End User Messaging Push usa o seguinte prefixo antes da ação:

```
mobiletargeting
```

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

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

Para ver exemplos de AWS Mensagens de usuário final Push políticas baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS Push de mensagens para o usuário final

Recursos políticos para AWS Push de mensagens para o usuário final

Compatível com recursos de políticas: Sim

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

O elemento `Resource` JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu Amazon Resource Name (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista de AWS Tipos de recursos push de mensagens de usuário final e seusARNs, consulte Recursos definidos por AWS Envio de mensagens para o usuário final na referência de autorização de serviço. Para saber com quais ações você pode especificar cada recurso, consulte Ações definidas por ARN AWS Push de mensagens para o usuário final.

Para ver exemplos de AWS Mensagens de usuário final Push políticas baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS Push de mensagens para o usuário final

Chaves de condição de política para AWS Push de mensagens para o usuário final

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

Os administradores podem usar AWS JSONpolíticas 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` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. 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.

Se você especificar vários `Condition` elementos em uma instrução ou várias chaves em um único `Condition` elemento, AWS os avalia usando uma `AND` operação lógica. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder permissão a um IAM usuário para acessar um recurso somente se ele estiver marcado com o nome de IAM usuário. Para obter mais informações, consulte elementos de IAM política: variáveis e tags no Guia IAM do usuário.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver tudo AWS chaves de condição globais, consulte AWS chaves de contexto de condição global no Guia IAM do usuário.

Para ver uma lista de AWS Chaves de condição push de mensagens para o usuário final, consulte Chaves de condição para AWS Envio de mensagens para o usuário final na referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Ações definidas por AWS Push de mensagens para o usuário final.

Para ver exemplos de AWS Mensagens de usuário final Push políticas baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS Push de mensagens para o usuário final

ACLsem AWS Push de mensagens para o usuário final

SuportesACLs: Não

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

ABACcom AWS Push de mensagens para o usuário final

Suportes ABAC (tags nas políticas): Parciais

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitas AWS recursos. Marcar entidades e recursos é a primeira etapa doABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABACé útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

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 oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobreABAC, consulte O que éABAC? no Guia do IAM usuário. Para ver um tutorial com etapas de configuraçãoABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do IAMusuário.

Usando credenciais temporárias com AWS Push de mensagens para o usuário final

Compatível com credenciais temporárias: Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS trabalhar com credenciais temporárias, consulte Serviços da AWS que funcionam com IAM o Guia IAM do Usuário.

Você está usando credenciais temporárias se fizer login no AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre a troca de funções, consulte Alternando para uma função (console) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Você pode então usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias emIAM.

Permissões principais entre serviços para AWS Push de mensagens para o usuário final

Suporta sessões de acesso direto (FAS): Sim

Quando você usa um IAM usuário ou uma função para realizar ações no AWS, você é considerado diretor. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FASusa as permissões do diretor chamando um AWS service (Serviço da AWS), combinado com a solicitação AWS service (Serviço da AWS) para fazer solicitações para serviços posteriores. FASas solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou recursos para concluir. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte Encaminhar sessões de acesso.

Funções de serviço para AWS Push de mensagens para o usuário final

Compatível com perfis de serviço: Sim

Uma função de serviço é uma IAMfunção que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamenteIAM. Para

obter mais informações, consulte Criação de uma função para delegar permissões a um AWS service (Serviço da AWS) no IAM Guia do usuário.

> **Warning**
>
> A alteração das permissões de uma função de serviço pode falhar AWS Funcionalidade push de mensagens para o usuário final. Edite funções de serviço somente quando AWS O End User Messaging Push fornece orientação para fazer isso.

Funções vinculadas a serviços para AWS Push de mensagens para o usuário final

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 a função de realizar uma ação em seu nome. As funções vinculadas ao serviço aparecem em seu Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte AWS serviços que funcionam com IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Função vinculada 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 Push de mensagens para o usuário final

Por padrão, usuários e funções não têm permissão para criar ou modificar AWS Recursos push de mensagens para o usuário final. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI), ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte Criação de IAM políticas no Guia do IAMusuário.

Para obter detalhes sobre ações e tipos de recursos definidos pelo AWS Push de mensagens para o usuário final, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte Ações,

recursos e chaves de condição para AWS Envio de mensagens para o usuário final na referência de autorização de serviço.

Tópicos

- Melhores práticas de política
- Usar o AWS Console push de mensagens para o usuário final
- Permitir que usuários visualizem suas próprias permissões

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir AWS Mensagens para o usuário final Envie recursos para sua conta. Essas ações podem incorrer em custos para o seu Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com AWS políticas gerenciadas e migrar para permissões com privilégios mínimos — Para começar a conceder permissões para seus usuários e cargas de trabalho, use o AWS políticas gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis em seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo AWS políticas gerenciadas pelo cliente que são específicas para seus casos de uso. Para ter mais informações, consulte AWS políticas gerenciadas ou AWS políticas gerenciadas para funções de trabalho no Guia IAM do usuário.
- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre IAM como usar para aplicar permissões, consulte Políticas e permissões IAM no Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usandoSSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de um determinado AWS service (Serviço da AWS), por exemplo, AWS CloudFormation. Para obter mais informações, consulte Elementos IAM JSON da política: Condição no Guia IAM do usuário.
- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas

sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAMO Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação da política do IAM Access Analyzer no Guia do IAM Usuário.

- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root em seu Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte Configurando o API acesso MFA protegido no Guia do IAMusuário.

Para obter mais informações sobre as melhores práticas emIAM, consulte as melhores práticas de segurança IAM no Guia IAM do usuário.

Usar o AWS Console push de mensagens para o usuário final

Para acessar o AWS No console End User Messaging Push, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre o AWS Mensagens para o usuário final Envie recursos para o seu Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas exigidas, o console não funcionará conforme planejado para entidades (usuários ou funções) com essa política.

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

Para garantir que usuários e funções ainda possam usar o AWS Console Push de mensagens para o usuário final, também conecte o `AWSEndUserMessaging` AWS política gerenciada para as entidades. Para obter mais informações, consulte Adicionar permissões a um usuário no Guia do IAM usuário.

```
{
 "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 usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
```

```
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Solução de problemas AWS Mensagens para o usuário final Envie 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 Push de mensagens para o usuário final IAM e.

Tópicos

- Não estou autorizado a realizar uma ação em AWS Push de mensagens para o usuário final
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha Conta da AWS para acessar meu AWS Recursos push de mensagens para o usuário final

Não estou autorizado a realizar uma ação em AWS Push de mensagens para o usuário final

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, é preciso atualizar suas políticas para permitir que você realize a ação.

O exemplo de erro a seguir ocorre quando o `mateojackson` IAM usuário tenta usar o console para ver detalhes sobre um *`my-example-widget`* recurso fictício, mas não tem as permissões fictíciasmobiletargeting:*`GetWidget`*.

```
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 AWS Push de mensagens para o usuário final.

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 IAM usuário chamado `marymajor` tenta usar o console para realizar uma ação no AWS Push de mensagens para o usuário final. 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 Conta da AWS para acessar meu AWS Recursos push de mensagens para o usuário final

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode 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 AWS O End User Messaging Push oferece suporte a esses recursos, consulteComo AWS O End User Messaging Push funciona com IAM.
- Para saber como fornecer acesso aos seus recursos em Contas da AWS que você possui, consulte Fornecendo acesso a um IAM usuário em outro Conta da AWS que você possui no Guia do IAM Usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Fornecendo acesso a Contas da AWS propriedade de terceiros no Guia do IAM Usuário.
- Para saber como fornecer acesso por meio da federação de identidades, consulte Fornecendo acesso a usuários autenticados externamente (federação de identidades) no Guia do IAMusuário.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

Validação de conformidade para AWS Push de mensagens para o usuário final

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS no escopo do Programa de Conformidade e escolha o programa de conformidade no qual você está interessado. Para obter informações gerais, consulte AWS Programas de conformidade .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixando relatórios em AWS Artifact.

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinado pela confidencialidade de seus dados, pelos objetivos de conformidade da sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos em AWS focados em segurança e conformidade.
- Arquitetura para HIPAA segurança e conformidade na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos elegíveis.

> **Note**
>
> Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a Referência de serviços HIPAA elegíveis.

- AWS Recursos de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeie a orientação para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- Avaliando recursos com regras no AWS Config Guia do desenvolvedor — O AWS Config O serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes do setor e os regulamentos.
- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança em AWS. O Security Hub usa controles de segurança para avaliar sua AWS recursos e para verificar sua conformidade com os padrões e as melhores práticas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.
- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças ao seu Contas da AWS, cargas de trabalho, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência em AWS Push de mensagens para o usuário final

A ferramenta AWS a infraestrutura global é construída em torno de Regiões da AWS e 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 aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre Regiões da AWS e zonas de disponibilidade, consulte AWS Infraestrutura global.

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

Segurança de infraestrutura em AWS Push de mensagens para o usuário final

Como um serviço gerenciado, AWS O envio de mensagens push para o usuário final é protegido pelo AWS procedimentos globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa AWS APIchamadas publicadas para acessar AWS Mensagens para o usuário final Push pela rede. Os clientes devem oferecer suporte ao Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com sigilo direto perfeito (), como (Ephemeral Diffie-HellmanPFS) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. Ou você pode 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, AWS O envio de mensagens push para o usuário final é protegido pelo AWS procedimentos 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.

Melhores práticas de segurança

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

- Crie um usuário individual do IAM para cada pessoa que gerencia recursos do , incluindo você mesmo. Não use AWS credenciais raiz para gerenciar recursos.
- Conceda a cada usuário o conjunto mínimo de permissões necessárias para realizar suas funções.
- Use IAM grupos para gerenciar com eficiência as permissões de vários usuários.
- Mude suas credenciais do IAM regularmente.

Para obter mais informações sobre a segurança, consulte Segurança em AWS Push de mensagens para o usuário final. Para obter mais informações sobreIAM, consulte AWS Identity and Access Management. Para obter informações sobre as IAM melhores práticas, consulte as IAM melhores práticas.

Monitoramento push de mensagens do usuário AWS final

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho do AWS End User Messaging Push e de suas outras AWS soluções. AWSfornece 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. É possível 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 CPU uso 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 indicar 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 CloudTrailcaptura API chamadas 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 usuários finais 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 API chamadas push de mensagens do 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 API chamadas 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 API operações 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 as informações a seguir.

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações suportados
- Configurando SNS notificações da Amazon 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 AWS End User Messaging Push API Reference. 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:

- 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 um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o CloudTrail userIdentityelemento.

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 API chamadas públicas, 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 o seu VPC e o AWS End User Messaging Push. Você pode acessar o AWS End User Messaging Push como se estivesse no seuVPC, sem o uso de um gateway de internet, NAT dispositivo, VPN conexão ou AWS Direct Connect conexão. Suas instâncias VPC não precisam de endereços IP públicos para acessar o AWS End User Messaging Push.

Você estabelece essa conectividade privada criando um endpoint de interface, desenvolvido 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 API ações por meio do endpoint da interface.

VPCas políticas de endpoint não são suportadas pelo 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 VPC console da Amazon ou o AWS Command Line Interface (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 privado DNS para o endpoint da interface, poderá fazer API solicitações ao AWS End User Messaging Push usando seu DNS nome 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 IAM recurso que você pode anexar a um 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 a partir do seuVPC, anexe uma política de endpoint personalizada ao endpoint da interface.

Uma política de endpoint especifica as seguintes informações:

- Os diretores que podem realizar ações (Contas da AWS, IAM usuários e IAM funções).
- As ações que podem ser executadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte Controlar o Acesso a Serviços Usando Políticas de Endpoint no AWS PrivateLink Guia.

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

O exemplo a seguir refere-se a uma política de endpoint personalizada. 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 AWSserviços e selecione Amazon Pinpoint.

Sua AWS conta 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 Service Quotas
Tamanho da carga útil da mensagem 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
Tamanho da carga da mensagem na sandbox do APNs	4 KB por mensagem	Não
Tamanho da carga da mensagem no Baidu Cloud Push	4 KB por mensagem	Não

Recurso	Cota padrão	Qualificada para aumento
Tamanho da carga útil 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
Lançamento inicial	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.