aws

Guia do Desenvolvedor

Amazon Pinpoint

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Pinpoint: Guia do Desenvolvedor

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é o Amazon Pinpoint? 1
Atributos do Amazon Pinpoint 1
Definir segmentos de público 1
Envolver seu público com campanhas de mensagens 1
Enviar mensagens transacionais 2
Analisar o comportamento dos usuários 2
Disponibilidade regional 2
Tutoriais 3
Uso do Postman com o Amazon Pinpoint 3
Sobre este tutorial 3
Pré-requisitos 4
Etapa 1: criar políticas e perfis do IAM 5
Etapa 2: configurar o Postman 9
Etapa 3: enviar solicitações adicionais 17
Configuração de um sistema de registro de SMS 24
Sobre a ativação dupla 24
Sobre esta solução 25
Pré-requisitos 27
Etapa 1: configurar o Amazon Pinpoint 28
Etapa 2: criar políticas e perfis do IAM 34
Etapa 3: criar funções do Lambda 37
Etapa 4: configurar o Amazon API Gateway 49
Etapa 5: criar e implantar o formulário da Web 54
Próximas etapas 62
Integração ao seu aplicativo 66
Trabalhando com AWS SDKs 67
Conectar o aplicativo de front-end com o AWS Amplify 68
Próxima etapa 68
Registrar endpoints 68
Antes de começar 69
AWS mobile SDKs 69
AWS Amplify 69
Próximas etapas 69
Relatar eventos 70

Antes de começar ... 71
AWS mobile SDKs ... 71
Web e React Native ... 69
API de eventos do Amazon Pinpoint ... 72
Próxima etapa ... 72
Gerenciar notificações por push ... 72
Configurar notificações por push ... 72
Gerenciar notificações por push ... 75
Definir o público-alvo ... 76
Adicionar endpoints ... 77
Exemplos ... 77
Informações relacionadas ... 83
Associar usuários a endpoints ... 83
Exemplos ... 84
Informações relacionadas ... 89
Adicionar um lote de endpoints ... 89
Exemplos ... 89
Informações relacionadas ... 97
Importar endpoints ... 97
Antes de começar ... 98
Exemplos ... 98
Informações relacionadas ... 111
Excluir endpoints ... 111
Exemplos ... 111
Gerenciar endpoints de membros do público-alvo ... 114
Acessar os dados do público-alvo ... 116
Pesquisar endpoints ... 117
Exemplos ... 117
Informações relacionadas ... 122
Exportar endpoints ... 122
Antes de começar ... 123
Exemplos ... 123
Informações relacionadas ... 134
Listar IDs de endpoints ... 134
Criar segmentos ... 138
Criar segmentos ... 138

Criar segmentos com o AWS SDK for Java ... 138
Importar segmentos ... 142
Importar um segmento ... 142
Personalizar segmentos com o AWS Lambda ... 144
Dados de eventos ... 146
Criação de uma função do Lambda ... 147
Atribuir uma política de função do Lambda ... 149
Atribuição de uma função do Lambda a uma campanha ... 152
Criação de campanhas ... 153
Criar campanhas padrão ... 153
Criação de campanhas com o AWS SDK for Java ... 153
Criar campanhas de teste A/B ... 156
Criação de campanhas de teste A/B com o AWS SDK for Java ... 156
Usar a API de SMS e voz ... 159
Enviar e validar senhas de uso único ... 161
Envio de uma mensagem OTP ... 161
Resposta do `SendOtpMessage` ... 164
Validar uma mensagem OTP ... 165
Resposta do `VerifyOtpMessage` ... 165
Exemplos de código ... 166
Gerar um ID de referência ... 166
Envio de códigos OTP ... 166
Validar códigos OTP ... 168
Enviar e recuperar mensagens no aplicativo ... 170
Recuperação de mensagens no aplicativo para um endpoint ... 170
Entendimento das respostas da API `GetInAppMessages` ... 173
Objeto `InAppMessageCampaigns` ... 174
Objeto `InAppMessage` ... 176
Objeto `HeaderConfig` ... 177
Objeto `BodyConfig` ... 178
Objeto `InAppMessageContent` ... 178
Objeto `Schedule` ... 180
Objeto `InAppMessageButton` ... 180
Objeto `DefaultButtonConfig` ... 181
Objeto `OverrideButtonConfig` ... 182
Validar números de telefone ... 184

Casos de uso da validação de números de telefone ... 184
Usar o serviço de validação de números de telefone ... 185
Respostas da validação de números de telefone ... 186
Enviar mensagens ... 190
Enviar e-mail ... 190
Escolher um método de envio de e-mail ... 191
Escolher entre o Amazon Pinpoint e o Amazon Simple Email Service (SES) ... 191
Uso da API ... 191
Enviando um e-mail com cabeçalhos de cancelamento de inscrição ... 206
Enviar mensagens SMS ... 208
Enviar mensagens de voz ... 221
Enviar notificações por push ... 229
Criar canais personalizados ... 239
Criar uma campanha que envia mensagens por meio de um canal personalizado ... 239
Noções básicas sobre dados de eventos ... 240
Configurar webhooks ... 242
Configurar as funções do Lambda ... 242
Exemplo de função do Lambda ... 242
Formato de resposta da função do Lambda para o Amazon Pinpoint ... 246
Conceder permissão ao Amazon Pinpoint para invocar a função do Lambda ... 248
Transmissão de eventos ... 251
Configurar streaming de eventos ... 252
Pré-requisitos ... 252
AWS CLI ... 253
AWS SDK for Java ... 253
Desabilitar o streaming de eventos ... 254
Eventos de aplicativo ... 254
Exemplo ... 254
Atributos de eventos do aplicativo ... 256
Eventos de campanha ... 260
Evento de exemplo ... 260
Atributos de eventos de campanha ... 261
Eventos de jornada ... 268
Evento de exemplo ... 269
Atributos de eventos de jornadas ... 270
Eventos de e-mail ... 274

Eventos de exemplo ... 275
Atributos de eventos de e-mail ... 281
Eventos de SMS ... 288
Exemplo ... 288
Atributos de eventos de SMS ... 289
Consultar dados de análise ... 300
Métricas compatíveis ... 300
Noções básicas de consulta ... 302
Políticas do IAM ... 303
Métricas padrão ... 307
Métricas de aplicativo para campanhas ... 308
Métricas de aplicativo para mensagens de e-mail transacionais ... 315
Métricas de aplicativo para mensagens SMS transacionais ... 327
Métricas de campanha ... 335
Métricas de engajamento de jornada ... 347
Métricas de execução de jornada ... 355
Métricas de execução de atividades da jornada ... 357
Métricas de jornada e execução de campanhas ... 363
Consultar dados de campanha ... 365
Pré-requisitos ... 366
Consultar dados para uma campanha ... 367
Consultar dados para várias campanhas ... 372
Consultar dados de mensagens transacionais ... 379
Pré-requisitos ... 379
Consultar dados para mensagens de e-mail transacionais ... 380
Consultar dados para mensagens SMS transacionais ... 385
Usar resultados da consulta ... 391
Estrutura JSON ... 392
Objetos e campos JSON ... 397
Registrar em log chamadas de API ... 399
Informações do Amazon Pinpoint em CloudTrail ... 399
Ações da API Amazon Pinpoint que podem ser registradas por CloudTrail ... 401
Ações da API de e-mail do Amazon Pinpoint que podem ser registradas por CloudTrail ... 405
Ações da API de voz e SMS do Amazon Pinpoint versão 1 que podem ser registradas por CloudTrail ... 406
Exemplos: entradas de arquivo de log do Amazon Pinpoint ... 406

Marcar recursos ... 412
Gerenciar tags ... 412
Usar tags nas políticas do IAM ... 413
Adicionar tags a recursos ... 414
Adicionar tags usando a API ... 414
Adição de tags usando a AWS CLI ... 415
Exibir tags para recursos ... 416
Exibição de tags usando a API ... 417
Exibição de tags usando a AWS CLI ... 417
Atualizar tags para recursos ... 418
Remover tags de recursos ... 419
Remover tags usando a API ... 419
Remoção de tags usando a AWS CLI ... 420
Informações relacionadas ... 420
Personalizar recomendações com o AWS Lambda ... 421
Usar recomendações em mensagens ... 421
Criar a função do Lambda ... 424
Dados de evento de entrada ... 424
Dados e requisitos de resposta ... 426
Atribuir uma política de função do Lambda ... 431
Autorizar o Amazon Pinpoint a invocar a função ... 433
Configurar o modelo do recomendação ... 434
Excluir dados ... 435
Excluir endpoints ... 435
Excluir dados de segmentos e endpoints do Amazon S3 ... 436
Excluir todos os dados do projeto ... 436
Excluindo todos os dados AWS ... 437
Exemplos de código ... 439
Amazon Pinpoint ... 440
Ações ... 440
API de SMS e voz do Amazon Pinpoint ... 526
Ações ... 527
Segurança ... 536
Proteção de dados ... 537
Criptografia de dados ... 539
Privacidade do tráfego entre redes ... 540

Criar um endpoint da VPC de interface para o Amazon Pinpoint 541
Gerenciamento de identidade e acesso 542
Público 543
Autenticando com identidades 543
Gerenciando acesso usando políticas 547
Como o Amazon Pinpoint funciona com o IAM 550
Ações de política do Amazon Pinpoint 557
Exemplos de políticas baseadas em identidade 588
Perfis do IAM para tarefas comuns 602
Solução de problemas 619
Registrar e monitorar 621
Validação de conformidade 623
Resiliência 624
Segurança da infraestrutura 624
Análise de configuração e vulnerabilidade 625
Melhores práticas de segurança 626
Cotas 627
Cotas do projeto 627
Cotas de solicitação da API 628
Cotas de campanhas 630
Cotas de e-mail 632
Cotas de mensagens de e-mail 633
Cotas de remetentes e destinatários de e-mail 633
Cotas de envio de e-mail 634
Cotas de endpoint 636
Cotas de importação de endpoints 637
Cotas de ingestão de eventos 638
Cotas de jornadas 639
Cotas Lambda 640
Cotas de machine learning 641
Cotas de modelos de mensagem 642
Cotas de notificações por push 644
Cotas de mensagens no aplicativo 645
Cotas de segmentos 645
Cotas de SMS 646
Cotas 10DLC 648

Cotas de voz 649
Solicitar um aumento de cota 652
Histórico do documento 655
Atualizações anteriores 664
..... dclxix

O que é o Amazon Pinpoint?

O Amazon Pinpoint é um serviço da AWS que você pode usar para interagir com seus clientes em vários canais de mensagens. Você pode usar o Amazon Pinpoint para enviar notificações por push, e-mails, mensagens de texto SMS ou mensagens de voz.

As informações neste guia do desenvolvedor são destinadas a desenvolvedores de aplicativos. Este guia contém informações sobre como usar os atributos do Amazon Pinpoint de forma programática. Ele também contém informações de interesse específico para os desenvolvedores de aplicativos móveis, como os procedimentos para integrar os recursos de mensagens e análises ao aplicativo.

Há vários outros documentos que complementam esse documento. Os seguintes documentos fornecem informações de referência relacionadas às APIs do Amazon Pinpoint:

- Referência de API do Amazon Pinpoint
- API de SMS e voz do Amazon Pinpoint

Se você não estiver familiarizado com o Amazon Pinpoint, pode ser útil revisar o Guia do usuário do Amazon Pinpoint antes de prosseguir com este documento.

Atributos do Amazon Pinpoint

Esta seção descreve os principais atributos do Amazon Pinpoint e as tarefas que você pode realizar com eles.

Definir segmentos de público

Alcance o público correto para suas mensagens definindo segmentos de público. Um segmento define quais usuários receberão as mensagens enviadas por uma campanha. Você pode definir segmentos dinâmicos baseados em dados relatados pelo aplicativo, como sistema operacional ou tipo de dispositivo móvel. Você também pode importar segmentos estáticos definidos com o uso de outro serviço ou aplicativo.

Envolver seu público com campanhas de mensagens

Envolva seu público criando uma campanha de mensagens. Uma campanha envia mensagens personalizadas seguindo uma programação definida por você. Você pode criar campanhas que enviem push para dispositivos móveis, e-mail ou SMS.

Para experimentar estratégias alternativas de campanha, configure a sua como um teste A/B e analise os resultados com análise do Amazon Pinpoint.

Enviar mensagens transacionais

Mantenha seus clientes bem-informados enviando mensagens transacionais por push para dispositivos móveis e mensagens SMS, como mensagens de ativação de novas contas, confirmações de pedidos e notificações de redefinição de senha, diretamente a usuários específicos. Você pode enviar mensagens transacionais usando a API REST do Amazon Pinpoint.

Analisar o comportamento dos usuários

Obtenha informações sobre o seu público e a eficácia de suas campanhas usando as análises fornecidas pelo Amazon Pinpoint. Você pode visualizar as tendências sobre o grau de envolvimento de seus usuários, atividades de compra, dados demográficos, e muito mais. Você também pode monitorar seu tráfego de mensagens exibindo métricas como o número total de mensagens enviadas ou abertas para uma campanha ou aplicativo. Com a API do Amazon Pinpoint, seu aplicativo pode relatar dados personalizados que o Amazon Pinpoint disponibiliza para análise, e você pode consultar os dados de análise para determinadas métricas padrão.

Para analisar ou armazenar os dados de análise fora do Amazon Pinpoint, você pode configurá-lo para transmitir os dados ao Amazon Kinesis.

Disponibilidade regional

O Amazon Pinpoint está disponível em várias Regiões da AWS na América do Norte, Europa, Ásia e Oceania. Em cada região, a 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 permitem que forneçamos níveis muito altos de disponibilidade e redundância ao mesmo tempo que minimizamos a latência.

Para saber mais sobre Regiões da AWS, consulte Gerenciar Regiões da AWS na Referência geral da Amazon Web Services. Para obter uma lista de todas as Regiões em que o Amazon Pinpoint está disponível atualmente, consulte Endpoints e cotas do Amazon Pinpoint e Endpoints de serviço da AWS 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.

Tutoriais

Os tutoriais nesta seção são projetados para mostrar aos novos usuários do Amazon Pinpoint como executar várias tarefas importantes. Se o Amazon Pinpoint é novidade para você ou se não conhece determinados atributos, estes tutoriais serão um bom ponto de partida.

Os tutoriais neste guia incluem tarefas destinadas a um público de desenvolvedores ou administradores de sistemas. Estes tutoriais mostram como executar tarefas usando a API do Amazon Pinpoint, os AWS SDKs e a AWS CLI. Se você interage com o Amazon Pinpoint usando principalmente o console baseado na Web, consulte a seção Tutoriais do Guia do usuário do Amazon Pinpoint.

Tutoriais

- Tutorial: como usar o Postman com a API do Amazon Pinpoint
- Tutorial: configurar um sistema de registro de SMS

Tutorial: como usar o Postman com a API do Amazon Pinpoint

O Postman é uma ferramenta popular para testar APIs em um ambiente gráfico de fácil utilização. Você pode usar o Postman para enviar solicitações de API para qualquer API REST e para receber respostas para suas solicitações. Usar o Postman é uma maneira conveniente de testar e solucionar problemas em chamadas feitas para a API do Amazon Pinpoint. Este tutorial inclui procedimentos para configurar e usar o Postman com o Amazon Pinpoint.

> **Note**
>
> O Postman foi desenvolvido por uma empresa terceira. Não é desenvolvido nem suportado pela Amazon Web Services (AWS). Para saber mais sobre como usar o Postman, ou para obter assistência com problemas relacionados ao Postman, consulte o Centro de suporte no site do Postman.

Sobre este tutorial

Esta seção contém uma visão geral deste tutorial.

Público-alvo

Este tutorial é destinado a desenvolvedores e implementadores de sistemas. Você não precisa estar familiarizado com o Amazon Pinpoint nem com o Postman para concluir as etapas deste tutorial. É necessário saber gerenciar as políticas do IAM e modificar exemplos de código JSON.

Os procedimentos neste tutorial foram criados para impedir que novos usuários utilizem operações de API que possam excluir permanentemente recursos do Amazon Pinpoint. Os usuários avançados podem remover essa restrição modificando a política que está associada aos seus usuários do .

Recursos usados

Este tutorial inclui exemplos de uso para o seguinte atributo do Amazon Pinpoint:

- Interação com a API do Amazon Pinpoint usando o Postman

Tempo necessário

Ele levará aproximadamente 15 minutos para ser concluído.

Restrições regionais

Não há restrições regionais associadas ao uso desta solução.

Custos pelo uso de recursos

Não há cobrança para criar uma conta da AWS. No entanto, ao implementar esta solução, você pode ter custos de uso da AWS se você usar o Postman para executar uma das seguintes ações:

- Enviar e-mail, SMS ou mensagens de voz ou por push para dispositivos móveis
- Criar e enviar campanhas
- Usar o recurso de validação de número de telefone

Para obter mais informações sobre as cobranças associadas ao uso do Amazon Pinpoint, consulte Definição de preços do Amazon Pinpoint.

Pré-requisitos

Antes de começar este tutorial, conclua os seguintes pré-requisitos:

- Você deve ter uma conta da AWS. Para criar uma conta da AWS, acesse https://console.aws.amazon.com/ e escolha Criar uma nova conta da AWS.

- Certifique-se de que a conta que você usa para entrar no AWS Management Console possa criar novas políticas e perfis do IAM.
- Certifique-se de ter criado pelo menos um projeto de amostra que tenha o e-mail ativado e uma identidade de e-mail verificada. Consulte Criação de um projeto do Amazon Pinpoint com suporte a e-mail no Guia do usuário do Amazon Pinpoint.
- Verifique se você tem um ID de conta da AWS. O ID da conta da AWS pode ser encontrado no canto superior direito do console ou usando a interface de linha de comandos (CLI). Consulte Como encontrar o ID da conta da AWS.
- É preciso baixar e instalar o Postman no computador. Você pode baixar o Postman no site do Postman.
- Depois de instalar o Postman no computador, você deve criar uma conta nele. Quando você inicia o aplicativo do Postman pela primeira vez, é solicitado que você faça login ou crie uma conta. Siga as instruções fornecidas pelo Postman para fazer login em sua conta ou criar uma se ainda não tiver.

Etapa 1: criar políticas e perfis do IAM

Quando você usa o Postman para testar a API do Amazon Pinpoint, a primeira etapa é criar um usuário. Nesta seção, você cria uma política que permite que os usuários interajam com todos os recursos do Amazon Pinpoint. Depois, você cria um usuário e anexa a política diretamente a ele.

Criar uma política do IAM

Saiba como criar uma política do IAM. Usuários e funções que usam essa política podem interagir com todos os recursos na API do Amazon Pinpoint. Ele também fornece acesso a recursos associados à API de e-mail do Amazon Pinpoint, bem como à API de SMS e voz do Amazon Pinpoint.

Para criar a política

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas e, em seguida, Criar política.
3. No editor de políticas, selecione JSON. Exclua qualquer JSON atual no editor de políticas para que fique em branco. Copie e cole o seguinte JSON no editor de políticas e, em seguida, no editor de políticas, substitua todas as instâncias de *123456789012* pelo seu ID. Conta da AWS

Seu Conta da AWS ID pode ser encontrado no canto superior direito do console, ou você pode usar a CLI, consulte Encontrando o ID da sua AWS conta.

> **Note**
>
> Para proteger os dados na sua conta do Amazon Pinpoint, essa política inclui apenas permissões de leitura, criação e modificação de recursos. Ela não inclui permissões que permitem excluir recursos. Você pode modificar essa política usando o editor visual no console do IAM. Para obter mais informações, consulte Gerenciar políticas do IAM no Guia do usuário do IAM. Você também pode usar a CreatePolicyVersionoperação na API do IAM para atualizar essa política.
> Além disso, essa política inclui permissões para a interação com os serviços `ses` e `sms-voice`, além do serviço `mobiletargeting`. As permissões `ses` e `sms-voice` permitem que você interaja com a API de e-mail do Amazon Pinpoint e a API de SMS e voz do Amazon Pinpoint, respectivamente. As permissões `mobiletargeting` permitem que você interaja com a API do Amazon Pinpoint.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Update*",
                "mobiletargeting:Get*",
                "mobiletargeting:Send*",
                "mobiletargeting:Put*",
                "mobiletargeting:Create*"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:*:123456789012:apps/*",
                "arn:aws:mobiletargeting:*:123456789012:apps/*/campaigns/*",
                "arn:aws:mobiletargeting:*:123456789012:apps/*/segments/*"
            ]
        },
        {
            "Sid": "VisualEditor1",
```

```
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:TagResource",
                "mobiletargeting:PhoneNumberValidate",
                "mobiletargeting:ListTagsForResource",
                "mobiletargeting:CreateApp"
            ],
            "Resource": "arn:aws:mobiletargeting:*:123456789012:*"
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": [
                "ses:TagResource",
                "ses:Send*",
                "ses:Create*",
                "ses:Get*",
                "ses:List*",
                "ses:Put*",
                "ses:Update*",
                "sms-voice:SendVoiceMessage",
                "sms-voice:List*",
                "sms-voice:Create*",
                "sms-voice:Get*",
                "sms-voice:Update*"
            ],
            "Resource": "*"
        }
    ]
}
```

Selecione Next (Próximo).

4. Em Nome da política, insira um nome para a política, como**`PostmanAccessPolicy`**. Escolha Criar política.
5. (Opcional) Você pode adicionar tags à política selecionando Adicionar tag.
6. Selecione Next: Review (Próximo: revisar).

Criar um usuário do IAM

> **Warning**
>
> Os usuários do IAM têm credenciais de longo prazo, o que representa um risco de segurança. Para ajudar a reduzir esse risco, recomendamos que você forneça a esses usuários somente as permissões necessárias para realizar a tarefa e que você os remova quando não forem mais necessários.

Depois de criar a política, você pode criar um usuário e anexar a política a ele. Ao criar o usuário, o IAM fornece um conjunto de credenciais que permitem que o Postman execute operações da API do Amazon Pinpoint.

Para criar o usuário

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No console do IAM, no painel de navegação, escolha Usuários e, em seguida, escolha Criar usuários.
3. Em Detalhes do usuário, em Nome do usuário, insira um nome que identifique o usuário, como **`PostmanUser`**. Em seguida, escolha Próximo.
4. Em Definir permissões, em Opções de permissões, escolha Anexar políticas diretamente.
5. Em Políticas de permissões, escolha a política (**`PostmanAccessPolicy`**) que você criou em Criar uma política do IAM. Em seguida, escolha Próximo.
6. Na página Revisar e criar, você tem a opção de adicionar tags que ajudam a identificar o usuário. Para obter mais informações sobre o uso de tags, consulte Marcar recursos do IAM no Guia do usuário do IAM.
7. Quando você estiver pronto para criar o usuário, escolha Create user (Criar usuário).

Crie chaves de acesso

> **Warning**
>
> Este cenário precisa de usuários do IAM com acesso programático e credenciais de longo prazo, o que representa um risco de segurança. Para ajudar a reduzir esse risco, recomendamos que você forneça a esses usuários somente as permissões necessárias

para realizar a tarefa e que você os remova quando não forem mais necessários. As chaves de acesso podem ser atualizadas, se necessário. Para obter mais informações, consulte Atualização de chaves de acesso no Guia de usuário do IAM.

O IAM fornece um conjunto de credenciais que você pode usar para permitir que o Postman realize operações de API do Amazon Pinpoint.

Para criar o usuário

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação do console do IAM, selecione Usuários. Selecione o usuário (**`PostmanUser`**) criado em Criar um usuário do IAM e, em seguida, selecione a guia Credenciais de segurança.
3. Na seção Chaves de acesso, escolha Criar chave de acesso.
4. Na página de melhores práticas e alternativas da chave de acesso, selecione Aplicativo em execução externa AWS.

   Em seguida, escolha Próximo.
5. (Opcional) Você pode adicionar uma tag de descrição à política.
6. Selecione Create access key (Criar chave de acesso).
7. Na página Recuperar chaves de acesso, copie as credenciais mostradas nas colunas ID de chave de acesso e (Chave de acesso secreta).

   > **Note**
   >
   > Será necessário fornecer o ID de chave de acesso e a chave de acesso secreta posteriormente neste tutorial. Esta é a única vez que você poderá visualizar a chave de acesso secreta. Recomendamos que você copie e salve-a em um local seguro.
8. Depois de salvar as duas chaves, escolha Concluído.

Etapa 2: configurar o Postman

Agora que você criou um usuário que pode acessar a API do Amazon Pinpoint, é possível configurar o Postman. Nesta seção, você cria um ou mais ambientes no Postman. Em seguida, importe uma

coleção que contenha um modelo de solicitação para cada uma das operações na API do Amazon Pinpoint.

Crie o espaço de trabalho do Postman

No Postman, um espaço de trabalho é um contêiner organizacional para projetos e ambientes. Nesta seção, você criará pelo menos um espaço de trabalho para usar com o Amazon Pinpoint.

Para criar um workspace

No Postman, vá até o campo de mais ações, escolha Arquivo e, depois, Novo.

1. Na janela Criar novo, escolha Espaço de trabalho.
2. Insira um nome, um resumo e defina a visibilidade como pessoal. Em seguida, escolha Criar espaço de trabalho.

Crie ambientes Postman

No Postman, um ambiente é um conjunto de variáveis que são armazenadas como pares de chave-valor. Você pode usar ambientes para alterar a configuração das solicitações feitas por meio do Postman, sem precisar alterar as próprias solicitações da API.

Nesta seção, você criará pelo menos um ambiente para usar com o Amazon Pinpoint. Cada ambiente que você cria contém um conjunto de variáveis que são específicas para sua conta em uma única Região da AWS. Se você usar os procedimentos nesta seção para criar mais de um ambiente, poderá alternar facilmente entre Regiões, escolhendo um ambiente diferente no menu Ambiente no Postman.

Para criar um ambiente

1. No Postman, vá até o menu de mais ações, escolha Arquivo e, depois, Novo.
2. Na janela Create New (Criar novo), escolha Environment (Ambiente).
3. Na janela MANAGE ENVIRONMENTS (GERENCIAR AMBIENTES), de Environment Name (Nome do ambiente), insira **`Amazon Pinpoint - Region Name`**. Substitua *`Region Name (Nome da região)`* por um dos seguintes valores:

   - Leste dos EUA (Norte da Virgínia)
   - Oeste dos EUA (Oregon)
   - Ásia-Pacífico (Mumbai)

- Ásia-Pacífico (Sydney)
- Europa (Frankfurt)
- Europa (Irlanda)

> **Note**
>
> No mínimo, você só precisa criar um ambiente para um único Região da AWS, que Região da AWS deve conter um projeto. Se você não criou um projeto em um dos listados anteriormente Regiões da AWS, consulte Criação de um projeto do Amazon Pinpoint com suporte por e-mail no Guia do usuário do Amazon Pinpoint.

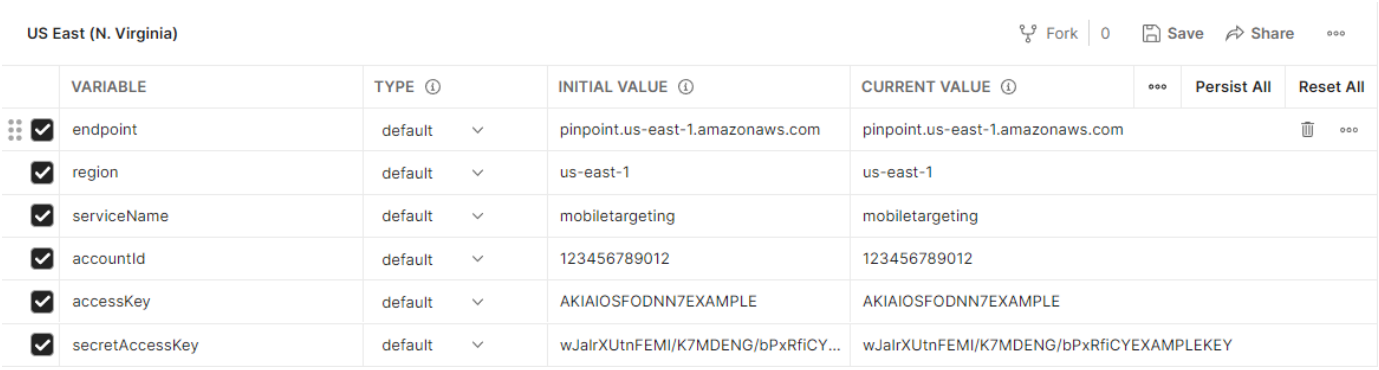
4. Crie seis novas variáveis: `endpoint`, `region`, `serviceName`, `accountId`, `accessKey` e `secretAccessKey`. Use a tabela a seguir para determinar qual valor inserir nas colunas Valor inicial e Valor atual para cada variável.

Região	Variável	Valor inicial e atual
Leste dos EUA (Norte da Virgínia)	`endpoint`	**`pinpoint.us-east-1.amazonaws.com`**
	`region`	**`us-east-1`**
	`serviceName`	**`mobiletargeting`**
	`accountId`	(ID AWS da sua conta)
	`accessKey`	(seu ID de chave de acesso do IAM)
	`secretAccessKey`	(sua chave de acesso secreta do IAM)
Oeste dos EUA (Oregon)	`endpoint`	**`pinpoint.us-west-2.amazonaws.com`**
	`region`	**`us-west-2`**

Região	Variável	Valor inicial e atual
	`serviceName`	**`mobiletargeting`**
	`accountId`	(ID AWS da sua conta)
	`accessKey`	(seu ID de chave de acesso do IAM)
	`secretAccessKey`	(sua chave de acesso secreta do IAM)
Ásia-Pacífico (Mumbai)	`endpoint`	**`pinpoint.ap-south-1.amazonaws.com`**
	`region`	**`ap-south-1`**
	`serviceName`	**`mobiletargeting`**
	`accountId`	(ID AWS da sua conta)
	`accessKey`	(seu ID de chave de acesso do IAM)
	`secretAccessKey`	(sua chave de acesso secreta do IAM)
Ásia-Pacífico (Sydney)	`endpoint`	**`pinpoint.ap-southeast-2.amazonaws.com`**
	`region`	**`ap-southeast-2`**
	`serviceName`	**`mobiletargeting`**
	`accountId`	(ID AWS da sua conta)

Região	Variável	Valor inicial e atual
	accessKey	(seu ID de chave de acesso do IAM)
	secretAccessKey	(sua chave de acesso secreta do IAM)
Europa (Frankfurt)	endpoint	**pinpoint.eu-central-1.amazonaws.com**
	region	**eu-central-1**
	serviceName	**mobiletargeting**
	accountId	(ID AWS da sua conta)
	accessKey	(seu ID de chave de acesso do IAM)
	secretAccessKey	(sua chave de acesso secreta do IAM)
Europa (Irlanda)	endpoint	**pinpoint.eu-west-1.amazonaws.com**
	region	**eu-west-1**
	serviceName	**mobiletargeting**
	accountId	(ID AWS da sua conta)
	accessKey	(seu ID de chave de acesso do IAM)
	secretAccessKey	(sua chave de acesso secreta do IAM)

Depois de criar essas variáveis, a janela MANAGE ENVIRONMENTS (GERENCIAR AMBIENTES) é semelhante ao exemplo mostrado na imagem a seguir.



Ao concluir, escolha Salvar.

> **Important**
>
> As chaves de acesso mostradas na imagem anterior são fictícias. Não compartilhe suas chaves de acesso do IAM com outras pessoas.
> O Postman inclui recursos que permitem compartilhar e exportar ambientes. Se você usar esses recursos, tenha o cuidado de não compartilhar seu ID de chave de acesso e chave de acesso secreta com ninguém que não tenha acesso a essas credenciais. Para obter mais informações, consulte Práticas recomendadas do IAM no Guia do usuário do IAM.

5. (Opcional) Repita as etapas de 1 a 4 para cada ambiente adicional que você deseja criar.

> **Tip**
>
> No Postman, você pode criar quantos ambientes forem necessários. Você pode usar ambientes das seguintes maneiras:
>
> - Crie um ambiente separado para cada Região onde você precisa testar a API do Amazon Pinpoint.
> - Crie ambientes associados a diferentes Contas da AWS.
> - Crie ambientes que usam credenciais associadas a outro usuário.

6. Ao terminar de criar ambientes, vá para a próxima seção.

Crie uma coleção Amazon Pinpoint no Postman

No Postman, uma coleção é um grupo de solicitações de API. Solicitações em uma coleção geralmente são unidas por um propósito comum. Nesta seção, você criará uma coleção que contém um modelo de solicitação para cada operação na API do Amazon Pinpoint.

Como criar a coleção do Amazon Pinpoint

1. No Postman, vá até o menu de mais ações, escolha Arquivo e, depois, Importar.
2. Na janela Importar, escolha Importar do link e, em seguida, insira o seguinte URL: https://raw.githubusercontent.com/awsdocs/ amazon-pinpoint-developer-guide /master/Amazon%20Pinpoint.postman_collection.json.

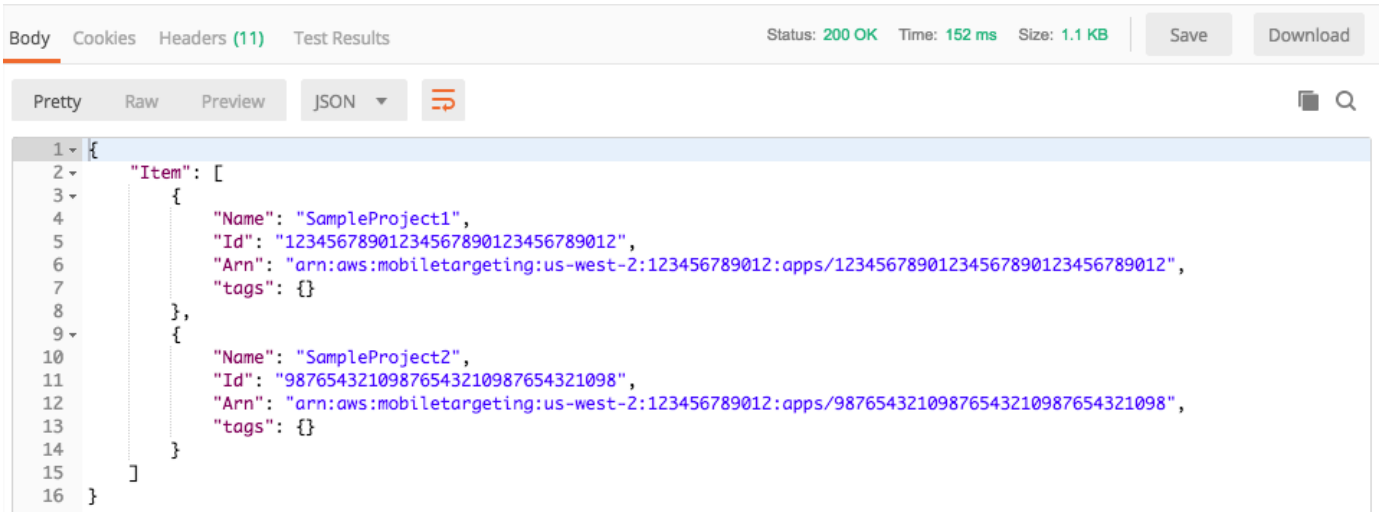
   Escolha Importar. O Postman importa a coleção do Amazon Pinpoint, que contém 120 solicitações de exemplo.

Testar a configuração do Postman

Depois de importar a coleção do Amazon Pinpoint, recomenda-se um teste rápido para garantir que todos os componentes estejam configurados corretamente. Você pode testar sua configuração enviando uma solicitação `GetApps`. Essa solicitação retorna uma lista de todos os projetos existentes na sua conta do Amazon Pinpoint na Região da AWS atual. Como essa solicitação não exige nenhuma configuração adicional, trata-se de uma boa maneira de testar a configuração.

Como testar a configuração da coleção do Amazon Pinpoint

1. No painel de navegação à esquerda, selecione Coleções, expanda a coleção do Amazon Pinpoint e expanda a pasta Aplicativos.
2. Na lista de solicitações, escolha GetApps.
3. Use o seletor de ambiente para escolher o ambiente que você criou em ambientes Create Postman.
4. Selecione Enviar. Se a solicitação for enviada com sucesso, o painel de resposta mostrará um status de `200 OK`. Você verá uma resposta que é semelhante ao exemplo na imagem a seguir.



> **Note**
>
> Se você não tiver nenhum projeto criado no Região da AWS , o Amazon Pinpoint retornará. `{ "Item": [] }`

Essa resposta mostra uma lista de todos os projetos do Amazon Pinpoint que existem na sua conta na Região escolhida na etapa 3.

Solução de problemas

Ao enviar sua solicitação, talvez você veja um erro. Consulte a lista a seguir para ver vários erros comuns que você pode encontrar e as etapas que você pode seguir para resolvê-los.

Mensagem de erro	Problema	Resolução
Could not get any response There was an error connectin g to https://%7B%7Bendpoint %7D%7D/v1/apps.	Não há um valor atual para a variável `{{endpoint}}` , que é definida quando você escolhe um ambiente.	Use o seletor de ambiente para escolher um ambiente.
O token de segurança incluído na solicitação é inválido.	O Postman não conseguiu encontrar o valor atual de seu ID de chave de acesso ou chave de acesso secreta.	Escolha o ícone de engrenage m perto do seletor de ambiente e escolha o ambiente atual. Certifiqu

Mensagem de erro	Problema	Resolução
		e-se de que os valores `accessKey` e `secretAccessKey` apareçam nas colunas VALOR INICIAL e VALOR ATUAL e que você tenha inserido as credenciais corretamente.
“Mensagem”: “Usuário: arn:aws:iam: :123456789012:user/ não está autorizado a realizar: mobiletargeting: on resource: arn:aws:mobiletargeting:us-west-2:123456789012PinpointPostmanUser: *” GetApps	A política do IAM associada ao seu usuário não incluem as permissões apropriadas.	Verifique se seu usuário tem as permissões descritas em Criar uma política do IAM e se você forneceu as credenciais corretas ao criar o ambiente no espaço de trabalho Create Postman.

Etapa 3: enviar solicitações adicionais

Ao terminar de configurar e testar o Postman, você poderá começar a enviar solicitações adicionais para a API do Amazon Pinpoint. Esta seção inclui informações que você precisa saber antes de iniciar o envio de solicitações. Isso também inclui dois exemplos de solicitações que descrevem como usar a coleção Amazon Pinpoint.

> **Important**
>
> Ao concluir os procedimentos desta seção, você enviará solicitações para a API do Amazon Pinpoint. Essas solicitações criam recursos em sua conta do Amazon Pinpoint, modificam recursos existentes, enviam mensagens, alteram a configuração de seus projetos do Amazon Pinpoint e usam outros recursos do Amazon Pinpoint. Tenha cuidado ao executar essas solicitações.

Sobre os exemplos na coleção do Amazon Pinpoint Postman

Você precisa configurar a maioria das operações na coleção do Amazon Pinpoint Postman para usá-las. Para as operações `GET` e `DELETE`, normalmente, você só precisa modificar as variáveis definidas na guia Pre-request Script (Script de pré-solicitação).

> **Note**
>
> Ao usar a política do IAM mostrada em Criar uma política do IAM, você não pode realizar nenhuma das `DELETE` solicitações incluídas nessa coleção.

Por exemplo, a operação `GetCampaign` exige que você especifique um `projectId` e um `campaignId`. Na guia Pre-request Script (Script de pré-solicitação), essas variáveis estão presentes e são preenchidas com valores de exemplo. Exclua os valores de exemplo e substitua-os pelos valores apropriados para a campanha e o projeto do Amazon Pinpoint.

Dessas variáveis, a mais comumente usada é a variável `projectId`. O valor dessa variável deve ser o identificador exclusivo para o projeto ao qual sua solicitação se aplica. Para obter uma lista desses identificadores para seus projetos, consulte a resposta à solicitação `GetApps` que você enviou na etapa anterior deste tutorial. Nessa resposta, o campo `Id` fornece o identificador exclusivo de um projeto. Para saber mais sobre a operação `GetApps` e o significado de cada campo na resposta, consulte Aplicativos na Referência de API do Amazon Pinpoint.

> **Note**
>
> No Amazon Pinpoint, um “projeto” é o mesmo que um “app” ou “aplicativo”.

Para as operações `POST` e `PUT`, você também precisa modificar o corpo da solicitação para incluir os valores que você deseja enviar para a API. Por exemplo, quando envia uma solicitação `CreateApp`, que é uma solicitação `POST`, é necessário especificar um nome para o projeto que você criar. É possível modificar a solicitação na guia Body (Corpo). Neste exemplo, substitua o valor ao lado de `"Name"` pelo nome do projeto. Se quiser adicionar tags ao projeto, especifique-os no objeto `tags`. Ou, se não desejar adicionar tags, exclua todo o objeto `tags`.

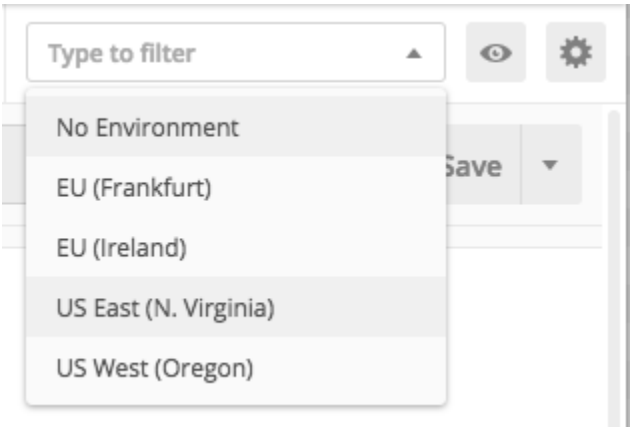
> **Note**
>
> A operação `UntagResource` também exige que você especifique parâmetros de URL. Você pode especificar esses parâmetros na guia Params (Parâmetros). Substitua os valores na coluna VALUE (Valor) pelas tags que você deseja excluir para o recurso especificado.

Exemplo de solicitação: criar um projeto com a operação **`CreateApp`**

Antes de criar segmentos e campanhas no Amazon Pinpoint, primeiro você precisa criar um projeto. No Amazon Pinpoint, um projeto consiste em segmentos, campanhas, configurações e dados unidos por um propósito comum. Por exemplo, você pode usar um projeto para conter todo o conteúdo relacionado a um determinado aplicativo ou a uma marca ou iniciativa de marketing específica. Quando você adiciona informações do cliente ao Amazon Pinpoint, essas informações são associadas a um projeto.

Para criar um projeto enviando uma solicitação de CreateApp API

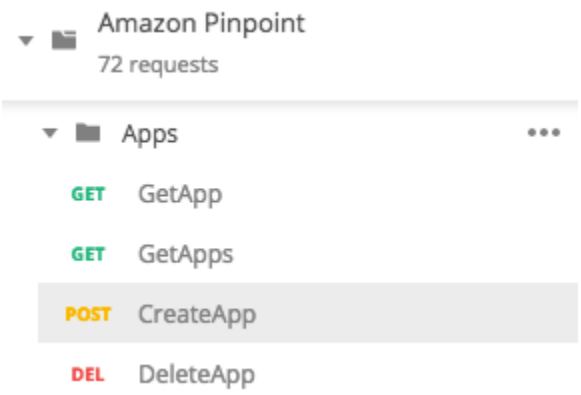
1. No menu Ambientes, escolha Região da AWS aquele em que você deseja criar o projeto.



   Neste exemplo, o Postman foi configurado para que o menu Environments (Ambientes) mostre as quatro opções a seguir:

   - Leste dos EUA (Norte da Virgínia)
   - Oeste dos EUA (Oregon)
   - Europa (Frankfurt)
   - Europa (Irlanda)

2. Na pasta Apps, escolha a CreateAppoperação>.



A pasta Aplicativos na coleção Amazon Pinpoint Postman foi expandida e mostra as seguintes solicitações:

- `GetApp`
- `GetApps`
- `CreateApp`
- `DeleteApp`

3. Na guia Body (Corpo), ao lado de `"Name"`, substitua o valor do espaço reservado (`"string"`) por um nome para a campanha, como **`"MySampleProject"`**.
4. Exclua a vírgula depois do nome da campanha e, depois, exclua o objeto `tags` inteiro nas linhas de 3 a 5. Ao terminar, sua solicitação deverá se parecer com o exemplo mostrado no trecho de código a seguir.

```
{
    "Name": "MySampleProject"
}
```

O Postman é configurado para enviar a solicitação como uma carga JSON bruta.

5. Selecione Enviar. Se a campanha for criada com êxito, o painel de resposta mostrará o status `201 Created`.

```
{
    "Name": "MySampleProject"
    "Id": "1234567890123456789012345678901 2",
    "Arn": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/1234567890123456789012345678901 2",
    "tags": {}
```

```
}
```

Exemplo: enviar um e-mail usando a operação **`SendMessages`**

É muito comum usar a API `SendMessages` do Amazon Pinpoint para enviar mensagens transacionais. Uma vantagem de enviar mensagens usando a API `SendMessages` em vez de criar campanhas é que você pode usar a API para enviar mensagens para qualquer endereço, como um endereço de e-mail, número de telefone ou token de dispositivo. O endereço para o qual você enviar mensagens não precisa existir em sua conta do Amazon Pinpoint. Compare esse método com o envio de mensagens por meio da criação de campanhas. Antes de enviar uma campanha no Amazon Pinpoint, você deve adicionar endpoints à sua conta do Amazon Pinpoint, criar segmentos, criar a campanha e realizar a campanha.

O exemplo nesta seção mostra como enviar uma mensagem de e-mail transacional diretamente para um determinado endereço de e-mail. Você pode modificar essa solicitação para enviar mensagens por meio de outros canais, como SMS, push ou voz para dispositivos móveis.

Para enviar uma mensagem de e-mail enviando uma SendMessages solicitação

1. Verifique se o canal de e-mail está habilitado para o projeto e se o endereço de e-mail ou domínio que você deseja usar para enviar e receber a mensagem está configurado. Para obter mais informações, consulte Ativação e desativação do canal de e-mail e Verificação de identidades de e-mail no Guia do usuário do Amazon Pinpoint.

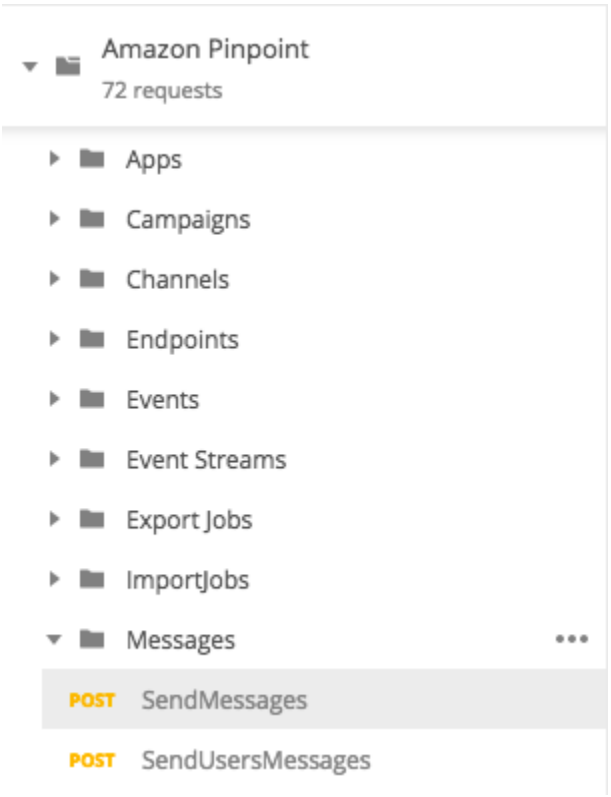
   > **Note**
   >
   > Para concluir o procedimento desta seção, você precisa verificar um endereço de e-mail

2. No menu Ambientes, escolha Região da AWS aquele do qual você deseja enviar a mensagem.

   Neste exemplo, o Postman foi configurado para que o menu Environments (Ambientes) mostre as quatro opções a seguir:

   - Leste dos EUA (Norte da Virgínia)
   - Oeste dos EUA (Oregon)
   - Europa (Frankfurt)
   - Europa (Irlanda)

3. Na pasta Mensagens, escolha a SendMessagesoperação.



4. Na guia Pre-request Script (Script de pré-solicitação), substitua o valor da variável `projectId` pelo ID de um projeto que já existe na região que você selecionou na etapa 2 desta seção.
5. Na guia Body (Corpo), exclua a solicitação de exemplo que é mostrada no editor de solicitações. Cole o seguinte código:

```
{
    "MessageConfiguration":{
        "EmailMessage":{
            "FromAddress":"sender@example.com",
            "SimpleEmail":{
                "Subject":{
                    "Data":"Sample Amazon Pinpoint message"
                },
                "HtmlPart":{
                    "Data":"<h1>Test message</h1><p>This is a sample message sent
 from <a href=\"https://aws.amazon.com/pinpoint\">Amazon Pinpoint</a> using the
 SendMessages API.</p>"
                },
                "TextPart":{
                    "Data":"This is a sample message sent from Amazon Pinpoint
 using the SendMessages API."
                }
```

```
            }
        }
    },
    "Addresses":{
        "recipient@example.com": {
            "ChannelType": "EMAIL"
        }
    }
}
```

6. No código anterior, substitua *sender@example.com* pelo endereço de e-mail verificado. Substitua *recipient@example.com* pelo endereço de e-mail verificado para o qual você deseja enviar a mensagem.

> **Note**
>
> Se sua conta ainda estiver na sandbox de e-mails do Amazon Pinpoint, você só poderá enviar e-mails de endereços ou domínios verificados na sua conta do Amazon Pinpoint. Para obter mais informações sobre como remover a conta da sandbox, consulte Solicitar acesso de produção para e-mail no no Guia do usuário do Amazon Pinpoint.

7. Selecione Enviar. Se a mensagem for enviada com êxito, o painel de respostas mostrará o status `200 OK`.

```
{
    "ApplicationId": "12345678901234567890123456789012",
    "RequestId": "<sampleValue>",
    "Result": {
        "recipient@example.com": {
            "DeliveryStatus": "SUCCESSFUL",
            "StatusCode": 200,
            "StatusMessage": "<sampleValue>",
            "MessageId": "<sampleValue>"
        }
    }
}
```

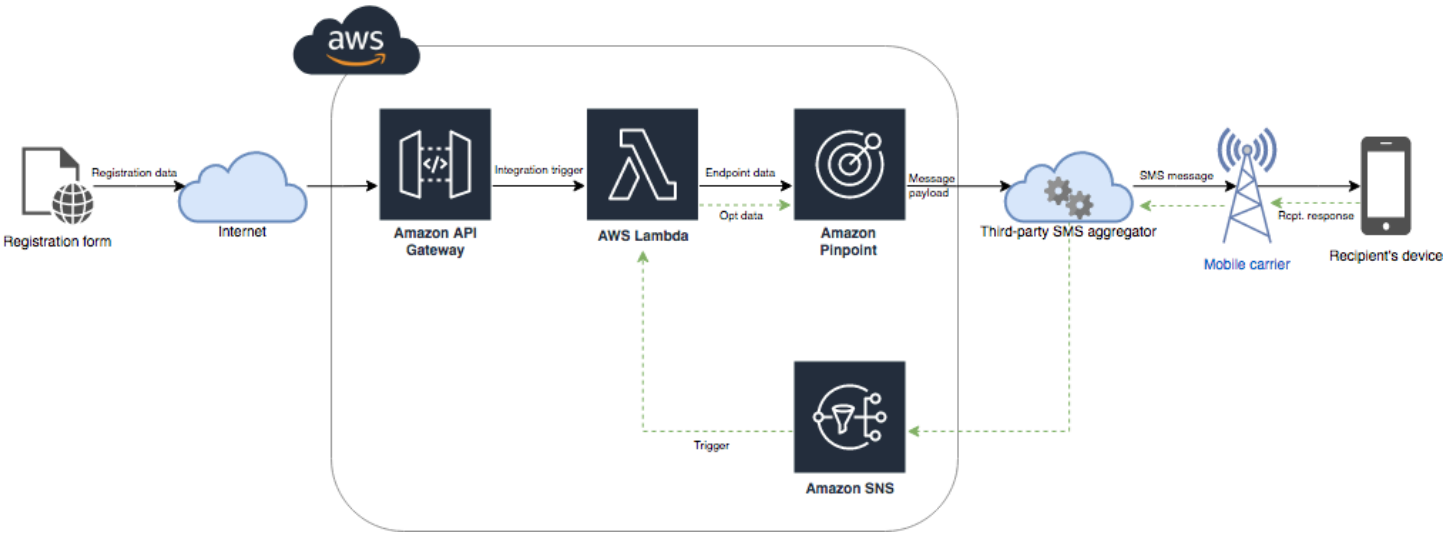
Tutorial: configurar um sistema de registro de SMS

Mensagens SMS (mensagens de texto) são uma ótima forma de enviar mensagens urgentes para seus clientes. Atualmente, muitas pessoas mantêm seus celulares por perto o tempo todo. Além disso, as mensagens SMS tendem a atrair mais a atenção das pessoas do que as notificações por push, e-mails ou telefonemas.

Uma maneira comum de capturar os números de celular de um cliente é usar um formulário na Web. Depois de verificar o número de telefone do cliente e confirmar sua assinatura, você pode começar a enviar mensagens SMS promocionais, transacionais e informativas para esse cliente.

Este tutorial mostra como configurar um formulário da Web para capturar informações de contrato de um cliente. O formulário da Web envia essas informações para o Amazon Pinpoint. Em seguida, o Amazon Pinpoint verifica se o número de telefone é válido e captura outros metadados relacionados ao número de telefone. Depois disso, o Amazon Pinpoint envia ao cliente uma mensagem solicitando a confirmação da assinatura. Depois que o cliente confirma sua assinatura, o Amazon Pinpoint opta por receber suas mensagens.

O diagrama de arquitetura a seguir mostra o fluxo de dados nessa solução.



Sobre a ativação dupla

Este tutorial mostra como configurar um sistema de inclusão dupla no Amazon Pinpoint que usa mensagens SMS bidirecionais.

Em um sistema de inclusão dupla por SMS, um cliente fornece seu número de telefone enviando-o em um formulário da Web ou em seu aplicativo. Quando você recebe a solicitação do cliente, cria

um endpoint no Amazon Pinpoint. O novo endpoint deve ser excluído de suas comunicações. Em seguida, você envia uma mensagem para esse número de telefone. Em sua mensagem, você solicita ao destinatário que confirme sua assinatura respondendo com uma palavra ou frase específica (como "Sim" ou "Confirmar"). Se o cliente responder à mensagem com a palavra ou frase que você especificou, você alterará o status do endpoint para incluído. Caso contrário, se o cliente não responder ou responder com uma palavra ou frase diferente, você poderá deixar o endpoint com o status de não incluído.

Sobre esta solução

Esta seção contém informações sobre a solução que você está criando neste tutorial.

Público-alvo

Este tutorial é destinado a públicos-alvo de desenvolvedores e implementadores de sistemas. Você não precisa estar familiarizado com o Amazon Pinpoint para concluir as etapas deste tutorial. No entanto, você deve se sentir à vontade para gerenciar políticas do IAM, criar funções do Lambda no Node.js e implantar conteúdo da Web.

Atributos usados

Este tutorial inclui exemplos de uso dos seguintes recursos do Amazon Pinpoint:

- Envio de mensagens SMS transacionais
- Obtenção de informações sobre números de telefone usando a validação do número de telefone
- Mensagens SMS recebidas usando mensagens SMS bidirecionais
- Criação de segmentos dinâmicos
- Criação de campanhas
- Interagindo com a API do Amazon Pinpoint usando AWS Lambda

Tempo necessário

Deve levar cerca de uma hora para concluir este tutorial. Depois de implementar essa solução, há etapas adicionais que você pode seguir para refinar a solução para se adequar ao seu caso de uso exclusivo.

Restrições regionais

Este tutorial exige que você arrende um código longo usando o console do Amazon Pinpoint. Você pode usar o console do Amazon Pinpoint para arrendar códigos longos dedicados com base em vários países. No entanto, somente os códigos longos baseados no Canadá podem ser usados para enviar mensagens SMS. (Você pode usar códigos longos baseados em outros países e regiões para enviar mensagens de voz.)

Desenvolvemos os exemplos de código neste tutorial com essa restrição em mente. Por exemplo, os exemplos de código supõem que o número de telefone do destinatário sempre tem 10 dígitos e um código de país de 1. Se você implementar essa solução nos países ou regiões dos Estados Unidos ou Canadá, será necessário modificar os exemplos de código adequadamente.

Custos pelo uso de recursos

Não há cobrança pela criação de uma AWS conta. No entanto, ao implementar essa solução, pode haver os seguintes custos:

- Custos de arrendamento de código longo: para concluir este tutorial, você precisa arrendar um código longo. Códigos longos baseados no Canadá custam USD 1,00 por mês.
- Uso de validação do número de telefone: a solução neste tutorial usa o recurso de validação do número de telefone do Amazon Pinpoint para verificar se cada número recebido é válido e formatado corretamente e para obter informações adicionais sobre o número de telefone. Você paga 0,006 USD para cada solicitação de validação de número de telefone.
- Custos de envio de mensagens: a solução neste tutorial envia mensagens SMS de saída. Você paga por cada mensagem enviada por meio do Amazon Pinpoint. O preço que você paga por cada mensagem depende do país ou região do destinatário. Se você enviar mensagens para destinatários nos Estados Unidos (excluindo os territórios dos EUA), pagará 0,00645 USD por mensagem. Se você enviar mensagens para destinatários no Canadá, pagará entre USD 0,001090 e USD 0,02, dependendo da operadora e do local do destinatário.
- Custos de recebimento de mensagens: esta solução também recebe e processa mensagens SMS recebidas. Você paga por cada mensagem recebida enviada para números de telefone associados à sua conta do Amazon Pinpoint. O preço que você paga depende do local em que o número de telefone de recebimento é baseado. Se o seu número de recebimento for baseado nos Estados Unidos (excluindo os territórios dos EUA), você pagará 0,0075 USD por mensagem recebida. Se o seu número é baseado no Canadá, você pagará 0,00155 USD por mensagem recebida.
- Uso do Lambda: esta solução usa duas funções do Lambda que interagem com a API do Amazon Pinpoint. Quando você chama uma função do Lambda, você é cobrado com base no número de solicitações de suas funções, pelo tempo para execução de seu código e pela quantidade de

memória usada por suas funções. As funções neste tutorial usam pouca memória e geralmente são executadas por 1 a 3 segundos. Alguns ou todos os seus usos desta solução podem se enquadrar no nível de uso gratuito do Lambda. Para obter mais informações, consulte Preço do Lambda.

- Uso do API Gateway: o formulário da web nesta solução chama uma API gerenciada pelo API Gateway. Para cada milhão de chamadas para o API Gateway, você paga de 3,50 a 3,70 USD, dependendo da região em que você AWS usa o Amazon Pinpoint. Para obter mais informações, consulte Preços do API Gateway.
- Custos de hospedagem na Web: esta solução inclui um formulário baseado na web que você precisa hospedar em seu site. O preço que você paga pela hospedagem desse conteúdo depende do provedor de hospedagem na Web.

> **Note**
>
> Todos os preços apresentados nesta lista são em dólares americanos (USD).

Próximo: pré-requisitos

Pré-requisitos

Antes de começar este tutorial, você precisa concluir os seguintes pré-requisitos:

- Você precisa ter uma conta da AWS. Para criar uma conta da AWS, acesse https://console.aws.amazon.com/ e escolha Criar uma nova conta da AWS.
- A conta que você usa para entrar no AWS Management Console deve ser capaz de executar as seguintes tarefas:
  - Criar políticas e perfis do IAM
  - Criar novos projetos do Amazon Pinpoint
  - Criar novas funções do Lambda
  - Criar novas APIs no API Gateway
- Você precisa ter um método de hospedagem de páginas da Web e deve saber como publicar páginas da Web. Embora você possa usar os serviços da AWS para hospedar as páginas da Web, não é necessário.

> **Tip**
>
> Para saber mais sobre como hospedar páginas da Web usando os serviços da AWS, consulte Hospedar um site estático.

Próximo: configurar o Amazon Pinpoint

Etapa 1: configurar o Amazon Pinpoint

A primeira etapa na implementação dessa solução é configurar o Amazon Pinpoint. Nesta seção, faça o seguinte:

- Crie um projeto do Amazon Pinpoint
- Ative o canal SMS e alugue um número de telefone
- Configure mensagens SMS bidirecionais

Antes de começar este tutorial, você deve revisar os pré-requisitos.

Crie um projeto do Amazon Pinpoint

Para começar, é preciso criar um projeto do Amazon Pinpoint. No Amazon Pinpoint, um projeto consiste em segmentos, campanhas, configurações e dados unidos por um propósito comum. Por exemplo, você pode usar um projeto para conter todo o conteúdo relacionado a um determinado aplicativo ou a uma marca ou iniciativa de marketing específica. Quando você adiciona informações do cliente ao Amazon Pinpoint, essas informações são associadas a um projeto.

As etapas envolvidas na criação de um projeto diferem dependendo se você criou um projeto no Amazon Pinpoint anteriormente.

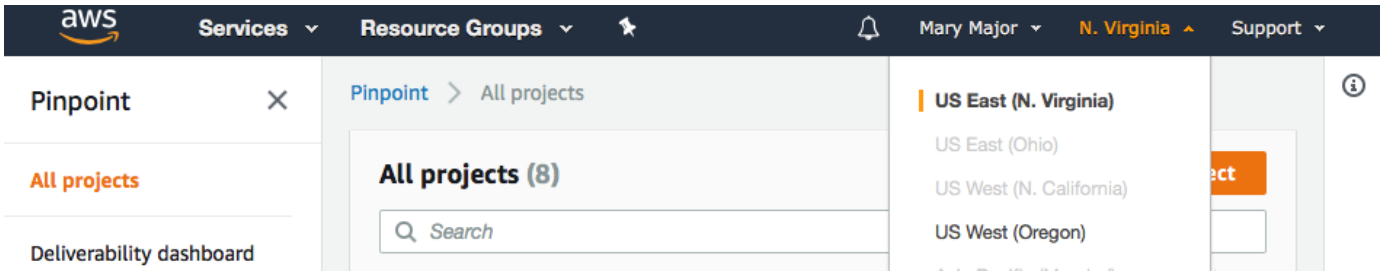
Criação de um projeto (novos usuários do Amazon Pinpoint)

Essas etapas descrevem o processo de criação de um novo projeto Amazon Pinpoint caso você nunca tenha criado um projeto na região atual AWS .

Para criar um projeto

1. Faça login AWS Management Console e abra o console do Amazon Pinpoint em https://console.aws.amazon.com/pinpoint/.

2. Use o seletor de região para escolher a AWS região que você deseja usar, conforme mostrado na imagem a seguir. Em caso de dúvida, escolha a região mais próxima de você.



3. Em Get started (Conceitos básicos), em Name (Nome), insira um nome para a campanha (como **`SMSRegistration`**) e escolha Create project (Criar projeto).
4. Na página Configurar atributos, escolha Ignorar esta etapa.
5. No painel de navegação, selecione All projects (Todos os projetos).
6. Na página All projects (Todos os projetos), ao lado do projeto recém-criado, copie o valor mostrado na coluna Project ID (ID do projeto).

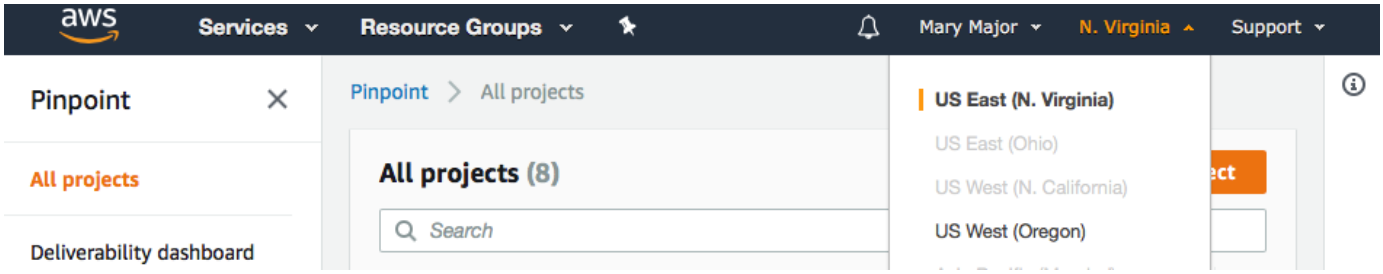
> **Tip**
>
> Você precisa usar esse ID em alguns lugares diferentes neste tutorial. Mantenha o ID do projeto em um local conveniente para que você possa copiá-lo mais tarde.

Criar um projeto (usuários existentes do Amazon Pinpoint)

Essas etapas descrevem o processo de criação de um novo projeto Amazon Pinpoint se você já tiver criado projetos na região atual AWS .

Para criar um projeto

1. Faça login AWS Management Console e abra o console do Amazon Pinpoint em https://console.aws.amazon.com/pinpoint/.
2. Use o seletor de região para escolher a AWS região que você deseja usar, conforme mostrado na imagem a seguir. Em caso de dúvida, escolha a região mais próxima de você.



3. Na página Todos os projetos, escolha Criar um novo projeto.
4. Na janela Create a project (Criar um projeto), em Project name (Nome do projeto), insira um nome para o projeto (como **`SMSRegistration`**). Selecione Create (Criar).
5. Na página Configurar atributos, escolha Ignorar esta etapa.
6. No painel de navegação, selecione All projects (Todos os projetos).
7. Na página All projects (Todos os projetos), ao lado do projeto recém-criado, copie o valor mostrado na coluna Project ID (ID do projeto).

> **Tip**
>
> Você precisa usar esse ID em alguns lugares diferentes neste tutorial. Mantenha o ID do projeto em um local conveniente para que você possa copiá-lo mais tarde.

Obter um número de telefone dedicado

> **Note**
>
> O Guia do usuário do Amazon Pinpoint foi atualizado. Para obter as informações mais recentes sobre como criar, configurar e gerenciar os recursos de voz e SMS do Amazon Pinpoint, consulte o novo Guia do usuário de SMS do Amazon Pinpoint.

Depois de criar um projeto, você pode começar a configurar recursos nesse projeto. Nesta seção, você habilita o canal de SMS e obtém um número de telefone dedicado para enviar mensagens SMS.

> **Note**
>
> Esta seção pressupõe que você esteja alugando um número de telefone 10DLC dos Estados Unidos após o registro da marca e da campanha, um número gratuito dos Estados Unidos ou

um código longo do Canadá. Se você seguir os procedimentos desta seção, mas escolher um país diferente dos Estados Unidos ou do Canadá, não poderá usar esse número para enviar mensagens SMS. Para saber mais sobre como arrendar códigos longos compatíveis com SMS em outros países além dos Estados Unidos ou do Canadá, consulte Países e regiões com suporte (canal SMS) no Guia do usuário de SMS do Amazon Pinpoint.

Para habilitar o canal de SMS usando o console do Amazon Pinpoint, siga estas etapas:

Ativar canal SMS

1. Faça login AWS Management Console e abra o console do Amazon Pinpoint em https://console.aws.amazon.com/pinpoint/.
2. No painel de navegação, em Configurações, selecione SMS e voz.
3. Ao lado de SMS settings (Configurações de SMS), escolha Edit (Editar).
4. Em General settings (Configurações gerais), selecione Enable the SMS channel for this project (Habilitar o canal de SMS para este projeto) e depois Save changes (Salvar alterações).

Para solicitar um número de telefone usando o console de SMS do Amazon Pinpoint, siga estas etapas:

Solicitar um número de telefone (console)

1. Abra o console de SMS do Amazon Pinpoint em https://console.aws.amazon.com/sms-voice/.

   > **Note**
   >
   > Certifique-se de solicitar um número de telefone no mesmo em Região da AWS que você criou seu projeto Amazon Pinpoint.

2. No painel de navegação, em Configurações, selecione Números de telefone e, depois, Solicitar originador.
3. Na página Selecionar país para País de destino da mensagem, escolha Estados Unidos ou Canadá. Selecione Next (Próximo).
4. Na seção Caso de uso de mensagens, faça o seguinte:

   - Em Recursos numéricos, selecione SMS.

> **Important**
>
> Os recursos de SMS e voz não podem ser alterados após a compra do número de telefone.

- Em Mensagens bidirecionais, selecione Sim.

5. Selecione Next (Próximo).
6. Em Selecionar tipo de originador, selecione Código longo ou 10DLC.

   Se você selecionar 10DLC e já tiver uma campanha registrada, poderá escolher a campanha em Associado a campanha registrada.
7. Selecione Next (Próximo).
8. Em Revisar e solicitar, é possível verificar e editar a solicitação antes de enviá-la. Escolha Solicitar.
9. A janela Registro obrigatório pode aparecer dependendo do tipo de número de telefone solicitado. O número de telefone está associado a esse registro e você não pode enviar mensagens até que ele seja aprovado. Para obter mais informações sobre os requisitos de registro, consulte Registros.

   a. Em Nome do formulário de inscrição, insira um nome amigável.

   b. Selecione Iniciar registro para concluir o registro do número de telefone ou Registrar mais tarde.

> **Important**
>
> O número de telefone não poderá enviar mensagens até o registro ser aprovado. Você ainda receberá a cobrança da taxa de locação mensal recorrente do número de telefone, independentemente do status do registro. Para obter mais informações sobre os requisitos de registro, consulte Registros.

Habilitar SMS bidirecional

Agora que você tem um número de telefone dedicado, poderá configurar o SMS bidirecional. A habilitação do SMS bidirecional possibilita que seus clientes respondam às mensagens SMS que

você envia a eles. Nesta solução, você usa SMS bidirecional para dar aos seus clientes uma maneira de confirmar que eles desejam se inscrever em seu programa de SMS.

Para habilitar o SMS bidirecional usando o console de SMS do Amazon Pinpoint, siga estas etapas:

Habilitar SMS bidirecional

1. Abra o console de SMS do Amazon Pinpoint em https://console.aws.amazon.com/sms-voice/.
2. No painel de navegação, em Configurações, selecione Números de telefone.
3. Na página Números de telefone, selecione um número.
4. Na guia SMS bidirecional, clique no botão Editar configurações.
5. Na página Editar configurações, selecione Habilitar mensagem bidirecional.
6. Em Destino, selecione Amazon SNS.
   - Novo tópico do Amazon SNS: o SMS do Amazon Pinpoint cria um tópico na conta. O tópico é criado automaticamente com todas as permissões necessárias. Para obter mais informações sobre os tópicos do Amazon SNS, consulte Configurar o Amazon SNS no Guia do desenvolvedor do Amazon SNS.
   - Em Destino da mensagem recebida, insira um nome de tópico, como **`SMSRegistrationFormTopic`**.
7. Em Perfil de canal bidirecional, selecione Usar políticas de tópicos do SNS.
8. Escolha Salvar alterações.

Use o console de SMS do Amazon Pinpoint para adicionar palavras-chave ao número de telefone que os clientes enviam para você para confirmar as assinaturas (como **`Yes`** ou **`Confirm`**).

Adicionar uma palavra-chave

1. Abra o console de SMS do Amazon Pinpoint em https://console.aws.amazon.com/sms-voice/.
2. No painel de navegação, em Configurações, selecione Número de telefone.
3. Na página Número de telefone, selecione o número de telefone ao qual adicionar uma palavra-chave.
4. Na guia Palavras-chave, selecione o botão Adicionar palavra-chave.
5. No painel Palavra-chave personalizada, adicione o seguinte:
   - Palavra-chave: a nova palavra-chave a ser adicionada (como **`Yes`** ou **`Confirm`**).

- Mensagem de resposta: a mensagem a ser enviada de volta ao destinatário.
- Ação de palavra-chave: a ação a ser executada quando a palavra-chave é recebida. Selecione Resposta automática.

6. Selecione Adicionar palavra-chave.

Próximo: criar políticas e perfis do IAM

Etapa 2: criar políticas e perfis do IAM

A próxima etapa na implementação da solução de registro de SMS é configurar uma política e uma função no AWS Identity and Access Management (IAM). No caso dessa solução, será preciso criar uma política que forneça acesso a determinados recursos relacionados ao Amazon Pinpoint. Crie uma função e anexe a política a ela. Posteriormente neste tutorial, você cria uma AWS Lambda função que usa essa função para chamar determinadas operações na API do Amazon Pinpoint.

Criar uma política do IAM

Esta seção mostra como criar uma política do IAM. Usuários e funções que usam essa política podem fazer o seguinte:

- Usar o recurso de validação de número de telefone
- Visualize, crie e atualize endpoints do Amazon Pinpoint
- Envie mensagens para endpoints do Amazon Pinpoint

Neste tutorial, você deseja dar ao Lambda a capacidade de executar essas tarefas. Contudo, para obter maior segurança, essa política usa o princípio de conceder o privilégio mínimo. Em outras palavras, ele concede apenas as permissões necessárias para concluir essa solução e não mais. Essa política é restrita das seguintes formas:

- Você só pode usá-lo para chamar a API de validação de número de telefone em uma região específica.
- Você só pode usá-la para exibir, criar ou atualizar endpoints associados a um projeto específico do Amazon Pinpoint.
- Você só pode usá-la para enviar mensagens para endpoints associados a um projeto específico do Amazon Pinpoint.

Para criar a política

1. Faça login AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Políticas e, em seguida, Criar política.
3. Na guia JSON, cole o seguinte código.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": "arn:aws:logs:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": "mobiletargeting:SendMessages",
            "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
        },
        {
            "Effect": "Allow",
            "Action": [
              "mobiletargeting:GetEndpoint",
              "mobiletargeting:UpdateEndpoint",
              "mobiletargeting:PutEvents"
            ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/
endpoints/*"
        },
        {
          "Effect": "Allow",
          "Action": "mobiletargeting:PhoneNumberValidate",
          "Resource": "arn:aws:mobiletargeting:region:accountId:phone/number/
validate"
        }
    ]
}
```

No exemplo anterior, faça o seguinte:

- *Substitua a AWS região pela região na qual você usa o Amazon Pinpoint, como us-east-1 ou.* `eu-central-1`

> **Tip**
>
> Para obter uma lista completa das AWS regiões em que o Amazon Pinpoint está disponível, consulte AWS regiões e endpoints no. Referência geral da AWS

- Substitua *`accountId`* pelo ID exclusivo da sua conta. AWS
- Substitua *`projectID`* pelo ID exclusivo do projeto que você criou em Criar um projeto Amazon Pinpoint deste tutorial.

> **Note**
>
> As `logs` ações permitem que o Lambda registre sua saída em CloudWatch Logs.

4. Selecione Next (Próximo).
5. Em Nome da política, insira um nome para a política, como**`RegistrationFormPolicy`**. Escolha Criar política.

Criar um perfil do IAM

Para criar o perfil

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação do console do IAM, selecione Perfis e, depois, Criar perfil.
3. Em Tipo de entidade confiável, escolha AWS serviço e, em Serviço ou caso de usuário, escolha Lambda na lista suspensa.
4. Selecione Next (Próximo).
5. Em Políticas de permissões, escolha ou pesquise a política que você criou na seção anterior e escolha Avançar.
6. Em Detalhes da função, em Nome da função, insira um nome para a função, como**`SMSRegistrationForm`**. Selecione Criar função.

Próximo: criar funções Lambda

Etapa 3: criar funções do Lambda

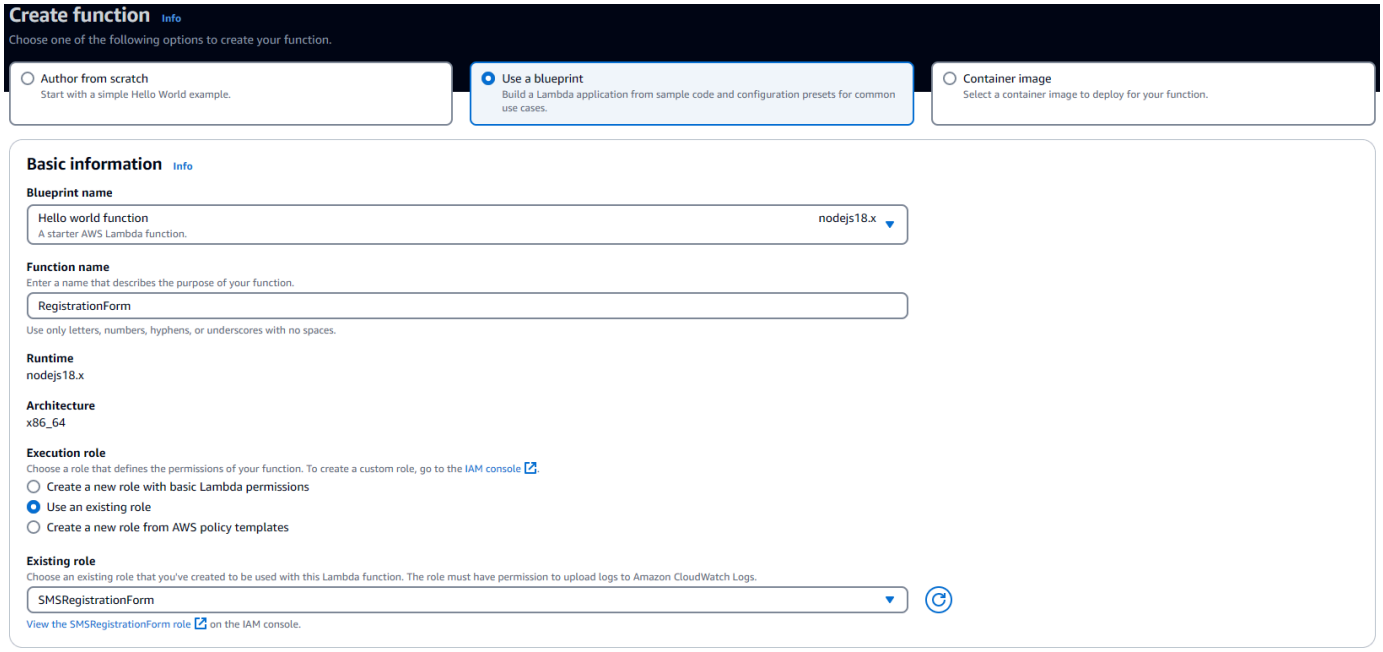
Esta solução usa duas funções do Lambda. Esta seção mostra como criar e configurar essas funções. Depois, você configurará o API Gateway e o Amazon Pinpoint para executar essas funções quando determinados eventos ocorrerem. Ambas as funções criam e atualizam endpoints no projeto do Amazon Pinpoint que você especifica. A primeira função também usa o recurso de validação do número de telefone.

Criar a função que valida as informações do cliente e cria endpoints

A primeira função recebe informações do seu formulário de registro, que recebe do Amazon API Gateway. Ela usa essas informações para obter informações sobre o número de telefone do cliente usando o recurso de validação de número de telefone do Amazon Pinpoint. Depois, a função usa os dados validados para criar um endpoint no projeto do Amazon Pinpoint que você especificar. Por padrão, o endpoint que a função cria é cancelado em suas futuras comunicações, mas esse status pode ser alterado pela segunda função. Por fim, essa função envia ao cliente uma mensagem solicitando que ele verifique se deseja receber suas comunicações por SMS.

Para criar a função do Lambda

1. Abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. Escolha a opção Criar função.
3. Em Criar uma função, escolha Usar um blueprint.
4. No campo de pesquisa, digite **`hello`** e pressione Enter. Na lista de resultados, escolha a função do Node.js `hello-world`, como mostra a imagem a seguir.



5. Em Basic information (Informações básicas), faça o seguinte:

   - Em Name (Nome), insira um nome para a função, como **`RegistrationForm`**.
   - Em Role (Função), selecione Choose an existing role (Escolher uma função existente).
   - Em Função existente, escolha a RegistrationForm função SMS que você criou em Criar uma função do IAM.

   Ao concluir, selecione Create function (Criar função).

6. Em Fonte de código, exclua a função de amostra no editor de código e cole o seguinte código:

```
import { PinpointClient, PhoneNumberValidateCommand, UpdateEndpointCommand,
 SendMessagesCommand } from "@aws-sdk/client-pinpoint"; // ES Modules import
const pinClient = new PinpointClient({region: process.env.region});

// Make sure the SMS channel is enabled for the projectId that you specify.
// See: https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
setup.html
var projectId = process.env.projectId;

// You need a dedicated long code in order to use two-way SMS.
// See: https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-voice-
manage.html#channels-voice-manage-request-phone-numbers
var originationNumber = process.env.originationNumber;
```

```
// This message is spread across multiple lines for improved readability.
var message = "ExampleCorp: Reply YES to confirm your subscription. 2 msgs per "
            + "month. No purchase req'd. Msg&data rates may apply. Terms: "
            + "example.com/terms-sms";

var messageType = "TRANSACTIONAL";

export const handler = async (event, context) => {
  console.log('Received event:', event);
  await validateNumber(event);
};

async function validateNumber (event) {
  var destinationNumber = event.destinationNumber;
  if (destinationNumber.length == 10) {
    destinationNumber = "+1" + destinationNumber;
  }
  var params = {
    NumberValidateRequest: {
      IsoCountryCode: 'US',
      PhoneNumber: destinationNumber
    }
  };
  try{
    const PhoneNumberValidateresponse = await pinClient.send( new
 PhoneNumberValidateCommand(params));
    console.log(PhoneNumberValidateresponse);
     if (PhoneNumberValidateresponse['NumberValidateResponse']['PhoneTypeCode'] ==
 0) {
        await createEndpoint(PhoneNumberValidateresponse, event.firstName,
 event.lastName, event.source);

     } else {
       console.log("Received a phone number that isn't capable of receiving "
                  +"SMS messages. No endpoint created.");
     }
  }catch(err){
    console.log(err);
  }
}

async function createEndpoint(data, firstName, lastName, source) {
  var destinationNumber = data['NumberValidateResponse']
['CleansedPhoneNumberE164'];
```

```
  var endpointId = data['NumberValidateResponse']
['CleansedPhoneNumberE164'].substring(1);

  var params = {
    ApplicationId: projectId,
    // The Endpoint ID is equal to the cleansed phone number minus the leading
    // plus sign. This makes it easier to easily update the endpoint later.
    EndpointId: endpointId,
    EndpointRequest: {
      ChannelType: 'SMS',
      Address: destinationNumber,
      // OptOut is set to ALL (that is, endpoint is opted out of all messages)
      // because the recipient hasn't confirmed their subscription at this
      // point. When they confirm, a different Lambda function changes this
      // value to NONE (not opted out).
      OptOut: 'ALL',
      Location: {
        PostalCode:data['NumberValidateResponse']['ZipCode'],
        City:data['NumberValidateResponse']['City'],
        Country:data['NumberValidateResponse']['CountryCodeIso2'],
      },
      Demographic: {
        Timezone:data['NumberValidateResponse']['Timezone']
      },
      Attributes: {
        Source: [
          source
        ]
      },
      User: {
        UserAttributes: {
          FirstName: [
            firstName
          ],
          LastName: [
            lastName
          ]
        }
      }
    }
  };
  try{
    const UpdateEndpointresponse = await pinClient.send(new
 UpdateEndpointCommand(params));
```

```
    console.log(UpdateEndpointresponse);
    await sendConfirmation(destinationNumber);
  }catch(err){
    console.log(err);
  }
}

async function sendConfirmation(destinationNumber) {
  var params = {
    ApplicationId: projectId,
    MessageRequest: {
      Addresses: {
        [destinationNumber]: {
          ChannelType: 'SMS'
        }
      },
      MessageConfiguration: {
        SMSMessage: {
          Body: message,
          MessageType: messageType,
          OriginationNumber: originationNumber
        }
      }
    }
  };
  try{
    const SendMessagesCommandresponse = await pinClient.send(new
 SendMessagesCommand(params));
    console.log("Message sent! "
          + SendMessagesCommandresponse['MessageResponse']['Result']
[destinationNumber]['StatusMessage']);
  }catch(err){
    console.log(err);
  }
}
```

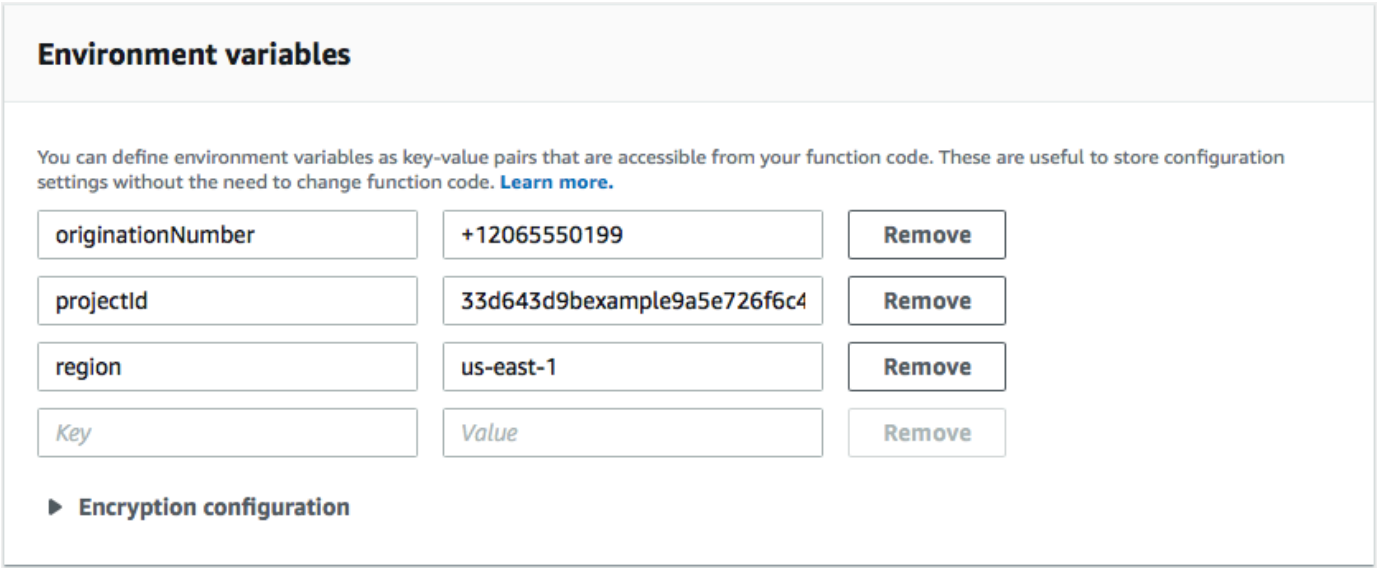
7. Na guia Configuração de variáveis de ambiente, escolha Editar e, em seguida, Adicionar variável de ambiente, faça o seguinte:

   - Na primeira linha, crie uma variável com uma chave de **`originationNumber`**. Em seguida, defina o valor para o número de telefone do código longo dedicado que você recebeu na Etapa 1.2.

> **Note**
>
> Não deixe de incluir o sinal de mais (+) e o código do país para o número de telefone. Não inclua outros caracteres especiais, como traços (-), pontos (.) ou parênteses.

- Na segunda linha, crie uma variável com uma chave de **`projectId`**. Em seguida, defina o valor para o ID exclusivo do projeto que você criou na Etapa 1.1.
- Na terceira linha, crie uma variável com uma chave de **`region`**. Em seguida, defina o valor para a região em que você usa o Amazon Pinpoint, como **`us-east-1`** ou **`us-west-2`**.

Quando você terminar, a seção Environment Variables (Variáveis de ambiente) deve ser semelhante ao exemplo mostrado na imagem a seguir.



8. Na parte superior da página, escolha Save (Salvar).

Testar a função

Depois de criar a função, você deve testá-la para verificar se está configurada corretamente. Além disso, verifique se o perfil do IAM que você criou tem as permissões apropriadas.

Para testar a função

1. Selecione a guia Testar.
2. Escolha Criar novo evento, faça o seguinte:

- Em Event name (Nome do evento), insira um nome para o evento de teste, como **`MyPhoneNumber`**.
- Apague o código de exemplo no editor de código. Cole o seguinte código:

```
{
  "destinationNumber": "+12065550142",
  "firstName": "Carlos",
  "lastName": "Salazar",
  "source": "Registration form test"
}
```

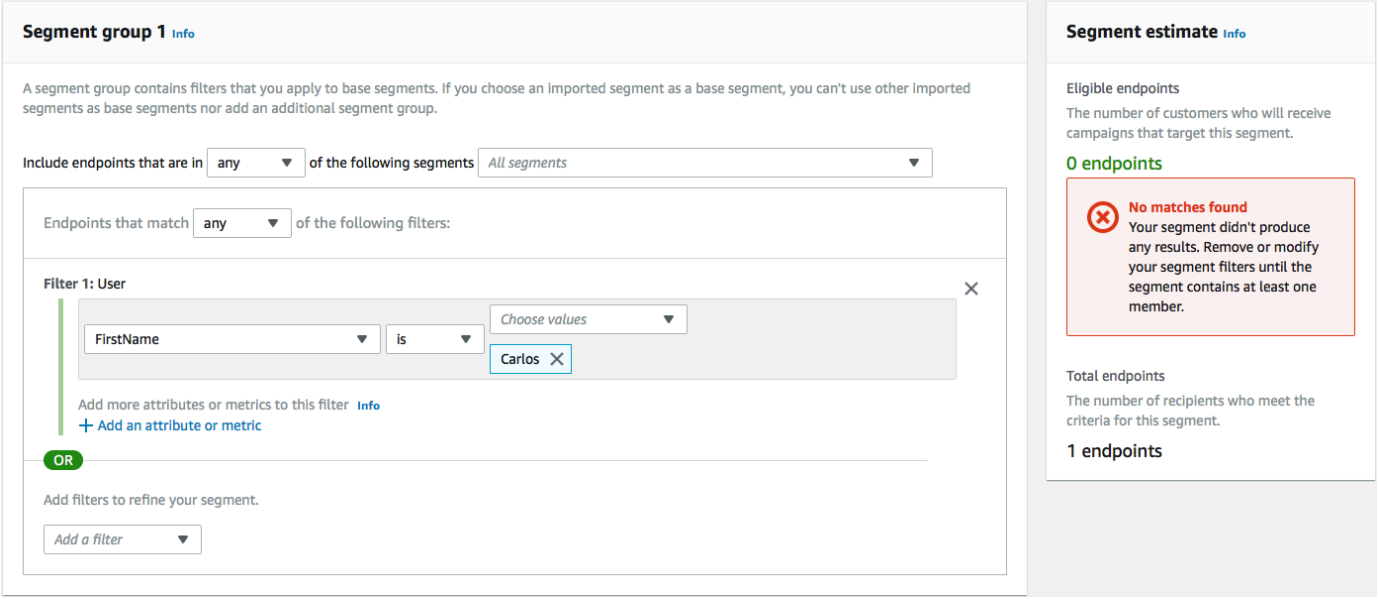
- No exemplo de código anterior, substitua os valores dos atributos `destinationNumber`, `firstName` e `lastName` pelos valores que você deseja usar para teste, como seus detalhes de contato pessoais. Ao testar essa função, ele envia uma mensagem SMS para o número de telefone especificado por você no atributo `destinationNumber`. Certifique-se de que o número de telefone especificado possa receber mensagens SMS.
- Selecione Create (Criar).

3. Escolha Testar.
4. Em Execution result: succeeded (Resultado de execução: bem-sucedido), selecione Details (Detalhes). Na seção Log output (Resultado de saída do log), analise a saída da função. Certifique-se de que a função foi executada sem erros.

   Verifique o dispositivo associado ao `destinationNumber` que você especificou para garantir que recebeu a mensagem de teste.
5. Faça login no console do Amazon Pinpoint em https://console.aws.amazon.com/pinpoint/.
6. Na página Todos os projetos, escolha o projeto que você criou em Criar um projeto do Amazon Pinpoint.
7. No painel de navegação, selecione Segmentos. Na página Segmentos, selecione Criar um segmento.
8. Em Segment group 1 (Grupo de segmentos 1), em Add filters to refine your segment (Adicionar filtros para refinar seu segmento), escolha Filter by user (Filtrar por usuário).
9. Em Escolher um atributo de usuário, escolha FirstName. Depois, para Choose values (Escolha valores), escolha o nome que você especificou no evento de teste.

   A seção Segment estimate (Estimativa de segmento) deve mostrar que há zero endpoints elegíveis e um endpoint total, conforme mostrado na imagem a seguir. Tal resultado é esperado.

Quando a função cria um endpoint, o endpoint é cancelado. Os segmentos no Amazon Pinpoint excluem automaticamente os endpoints excluídos.



Criar a função que habilita os clientes às suas comunicações

A segunda função é executada somente quando um cliente responde à mensagem enviada pela primeira função. Se a resposta do cliente incluir a palavra-chave especificada em Habilitar SMS bidirecional, a função atualizará o registro do endpoint para permitir que ele participe de futuras comunicações. O Amazon Pinpoint também responde automaticamente com a mensagem que você especificou em Habilitar SMS bidirecional.

Se o cliente não responder ou responder com algo diferente da palavra-chave designada, nada acontecerá. O endpoint do cliente permanece no Amazon Pinpoint, mas não pode ser direcionado por segmentos.

Para criar a função do Lambda

1. Abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. Escolha a opção Criar função.
3. Em Create function (Criar função), escolha Blueprints (Esquemas).
4. No campo de pesquisa, digite **`hello`** e pressione Enter. Na lista de resultados, escolha a função do Node.js `hello-world`, como mostra a imagem a seguir. Selecione Configurar.
5. Em Basic information (Informações básicas), faça o seguinte:

- Em Name (Nome), insira um nome para a função, como **`RegistrationForm_OptIn`**.
- Em Role (Função), selecione Choose an existing role (Escolher uma função existente).
- Em Função existente, escolha a RegistrationForm função SMS que você criou em Criar uma função do IAM.

Ao concluir, selecione Create function (Criar função).

6. Exclua a função de exemplo no editor de código e cole o código a seguir:

```
import { PinpointClient, UpdateEndpointCommand } from "@aws-sdk/client-
pinpoint"; // ES Modules import

// Create a new Pinpoint client instance with the region specified in the
 environment variables
const pinClient = new PinpointClient({ region: process.env.region });

// Get the Pinpoint project ID and the confirm keyword from environment variables
const projectId = process.env.projectId;
const confirmKeyword = process.env.confirmKeyword.toLowerCase();

// This is the main handler function that is invoked when the Lambda function is
 triggered
export const handler = async (event, context) => {
    console.log('Received event:', event);

    try {
        // Extract the timestamp, message, and origination number from the SNS
 event
        const timestamp = event.Records[0].Sns.Timestamp;
        const message = JSON.parse(event.Records[0].Sns.Message);
        const originationNumber = message.originationNumber;
        const response = message.messageBody.toLowerCase();

        // Check if the response message contains the confirm keyword
        if (response.includes(confirmKeyword)) {
            // If the confirm keyword is found, update the endpoint's opt-in status
            await updateEndpointOptIn(originationNumber, timestamp);
        }
    }catch (error) {
        console.error('An error occurred:', error);
        throw error; // Rethrow the error to handle it upstream
```

```
    }
};

// This function updates the opt-in status of a Pinpoint endpoint
async function updateEndpointOptIn(originationNumber, timestamp) {
    // Extract the endpoint ID from the origination number
    const endpointId = originationNumber.substring(1);

     // Prepare the parameters for the UpdateEndpointCommand
    const params = {
        ApplicationId: projectId,
        EndpointId: endpointId,
        EndpointRequest: {
            Address: originationNumber,
            ChannelType: 'SMS',
            OptOut: 'NONE',
            Attributes: {
                OptInTimestamp: [timestamp]
            },
        }
    };

    try {
        // Send the UpdateEndpointCommand to update the endpoint's opt-in status
        const updateEndpointResponse = await pinClient.send(new
 UpdateEndpointCommand(params));
        console.log(updateEndpointResponse);
        console.log(`Successfully changed the opt status of endpoint ID
 ${endpointId}`);
    } catch (error) {
        console.error('An error occurred while updating endpoint:', error);
        throw error; // Rethrow the error to handle it upstream
    }
}
```

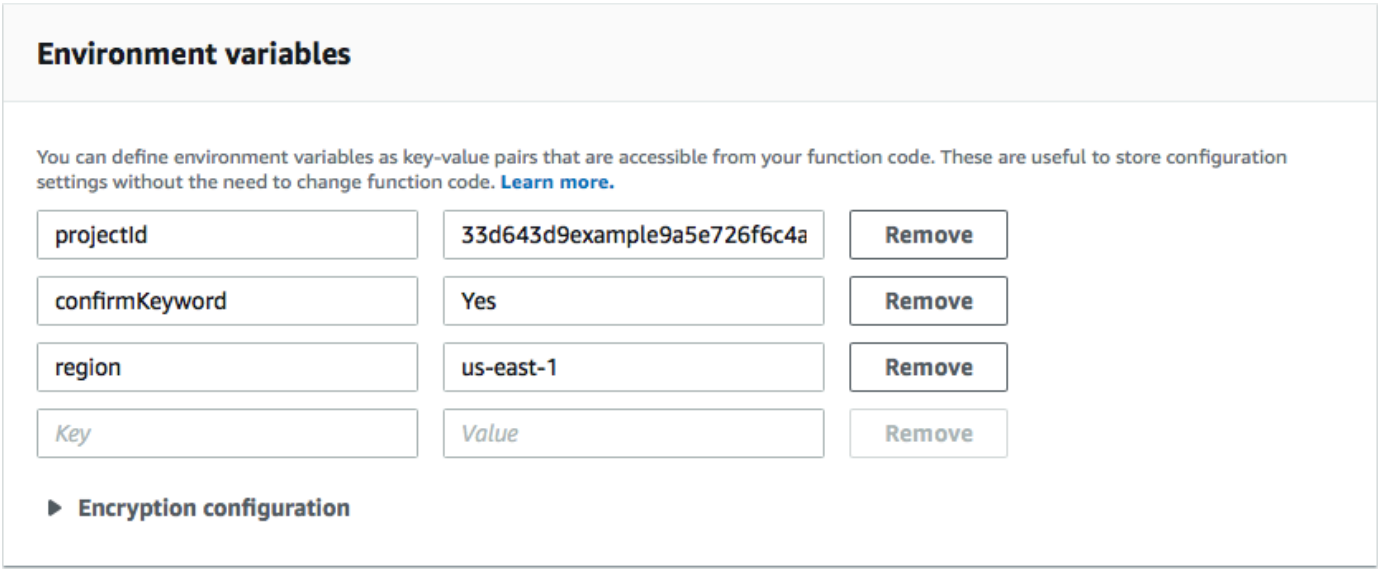
7. Em Environment variables (Variáveis de ambiente), faça o seguinte:

   - Na primeira linha, crie uma variável com uma chave de **`projectId`**. Em seguida, defina o valor como o ID exclusivo do projeto que você criou em Criar um projeto Amazon Pinpoint.
   - Na segunda linha, crie uma variável com uma chave de **`region`**. Em seguida, defina o valor para a região em que você usa o Amazon Pinpoint, como **`us-east-1`** ou **`us-west-2`**.

- Na terceira linha, crie uma variável com uma chave de **`confirmKeyword`**. Em seguida, defina o valor para a palavra-chave de confirmação que você criou em Habilitar SMS bidirecional.

> **Note**
>
> A palavra-chave não diferencia maiúsculas e minúsculas. Essa função converte a mensagem recebida em letras minúsculas.

Quando você terminar, a seção Environment Variables (Variáveis de ambiente) deve ser semelhante ao exemplo mostrado na imagem a seguir.



8. Na parte superior da página, escolha Save (Salvar).

Testar a função

Depois de criar a função, você deve testá-la para verificar se está configurada corretamente. Além disso, verifique se o perfil do IAM que você criou tem as permissões apropriadas.

Para testar a função

1. Escolha Testar.
2. Na janela Configure test event (Configurar evento de teste), faça o seguinte:
   a. Selecione Criar evento de teste.

b. Em Event name (Nome do evento), insira um nome para o evento de teste, como **`MyResponse`**.

c. Apague o código de exemplo no editor de código. Cole o seguinte código:

```
{
  "Records":[
    {
      "Sns":{
        "Message":"{\"originationNumber\":\"+12065550142\",\"messageBody\":
\"Yes\"}",
        "Timestamp":"2019-02-20T17:47:44.147Z"
      }
    }
  ]
}
```

No exemplo de código anterior, substitua os valores do atributo `originationNumber` pelo número de telefone que você usou quando você testou a função do Lambda anterior. Substitua o valor de pela `messageBody` palavra-chave SMS bidirecional que você especificou em Habilitar SMS bidirecional. Como opção, você pode substituir o valor de `Timestamp` pela data e hora atuais.

d. Selecione Create (Criar).

3. Escolha Test (Testar) novamente.
4. Em Execution result: succeeded (Resultado de execução: bem-sucedido), selecione Details (Detalhes). Na seção Log output (Resultado de saída do log), analise a saída da função. Certifique-se de que a função foi executada sem erros.
5. Faça login no console do Amazon Pinpoint em https://console.aws.amazon.com/pinpoint/.
6. Na página Todos os projetos, escolha o projeto que você criou em Criar um projeto do Amazon Pinpoint.
7. No painel de navegação, selecione Segmentos. Na página Segmentos, selecione Criar um segmento.
8. Em Segment group 1 (Grupo de segmentos 1), em Add filters to refine your segment (Adicionar filtros para refinar seu segmento), escolha Filter by user (Filtrar por usuário).
9. Em Escolher um atributo de usuário, escolha FirstName. Depois, para Choose values (Escolha valores), escolha o nome que você especificou no evento de teste.

A seção Segment estimate (Estimativa de segmento) deve mostrar que há um endpoint elegível e um endpoint total.

Próximo: configurar o Amazon API Gateway

Etapa 4: configurar o Amazon API Gateway

Nesta seção, você cria uma API usando o Amazon API Gateway. O formulário de registro que você implanta nessa solução chama essa API. Em seguida, o API Gateway passa as informações capturadas no formulário de registro para a função Lambda que você criou em Create Lambda functions.

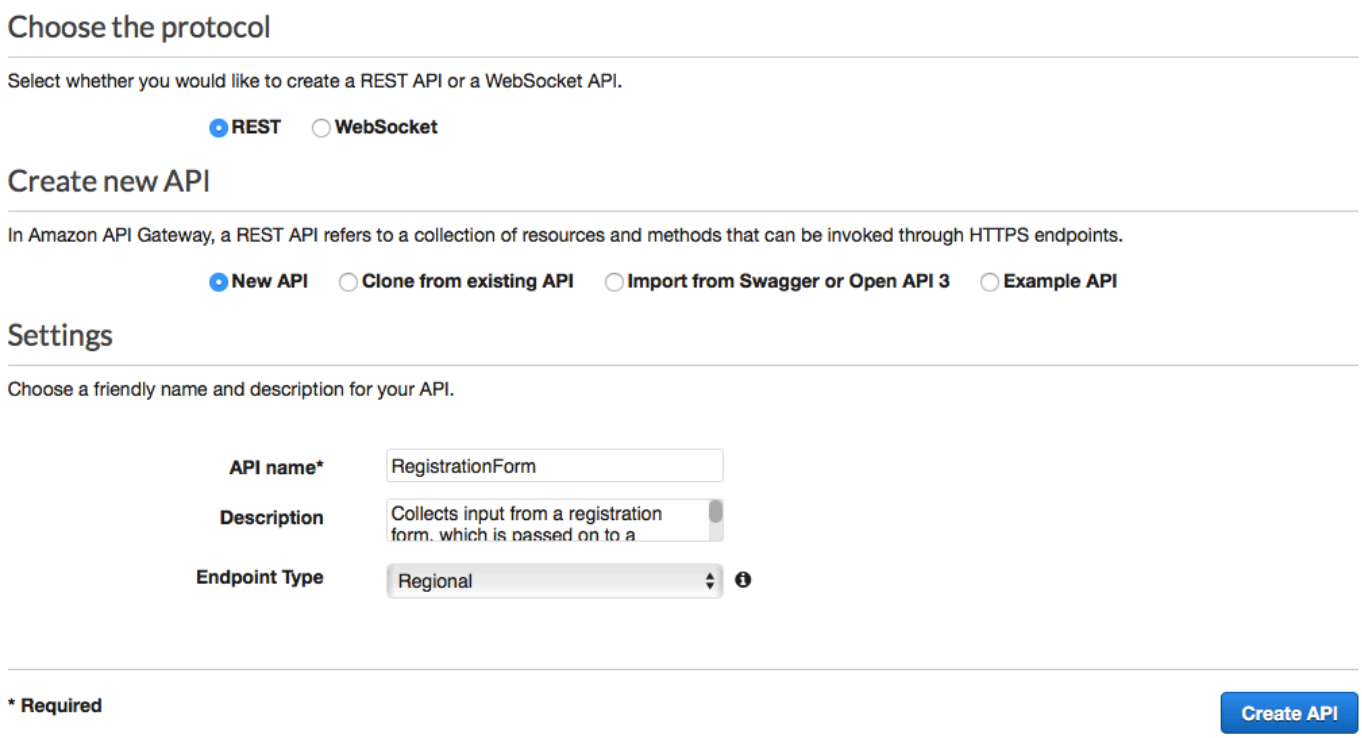
Criar a API

Primeiro, é preciso criar uma API no API Gateway. Os procedimentos a seguir mostram como criar uma API REST.

Para criar uma API

1. Abra o console do API Gateway em https://console.aws.amazon.com/apigateway/.
2. Selecione Criar API. Faça as seguintes seleções:
   - Em Choose the protocol (Escolher o protocolo), selecione REST.
   - Em Create new API (Criar nova API), selecione New API (Nova API).
   - Em Settings (Configurações), para Name (Nome), insira um nome, como **`RegistrationForm`**. Para Description (Descrição), insira, como opção, algum texto que descreva o objetivo da API. Em Endpoint Type (Tipo de endpoint), escolha Regional. Em seguida, selecione Create API (Criar API).

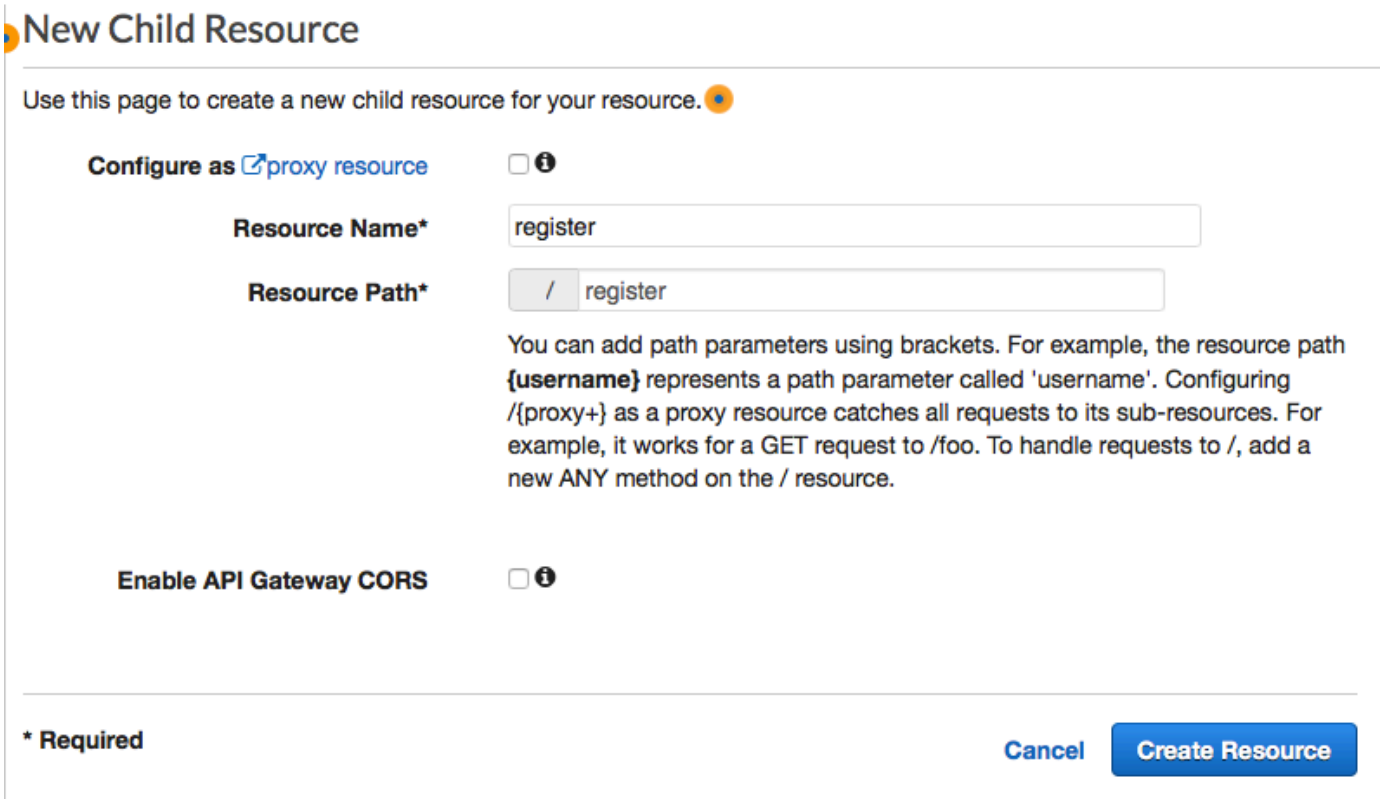
   Um exemplo dessas configurações é mostrado na imagem a seguir.



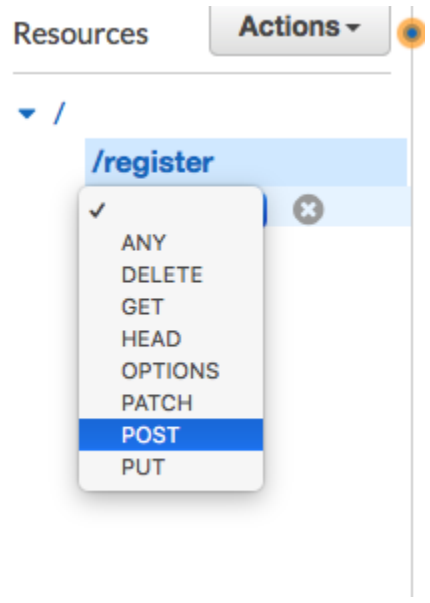
Criar um recurso

Agora que você criou uma API, é possível começar a adicionar recursos a ela. Depois disso, você adiciona um método POST ao recurso e informa ao API Gateway para transmitir os dados recebidos desse método para a função do Lambda.

1. No menu Actions (Ações), escolha Create Resource (Criar recurso). No painel New Child Resource (Novo recurso filho), para Resource Name (Nome do recurso), insira **`register`**, conforme mostrado na imagem a seguir. Escolha Create Resource (Criar recurso).



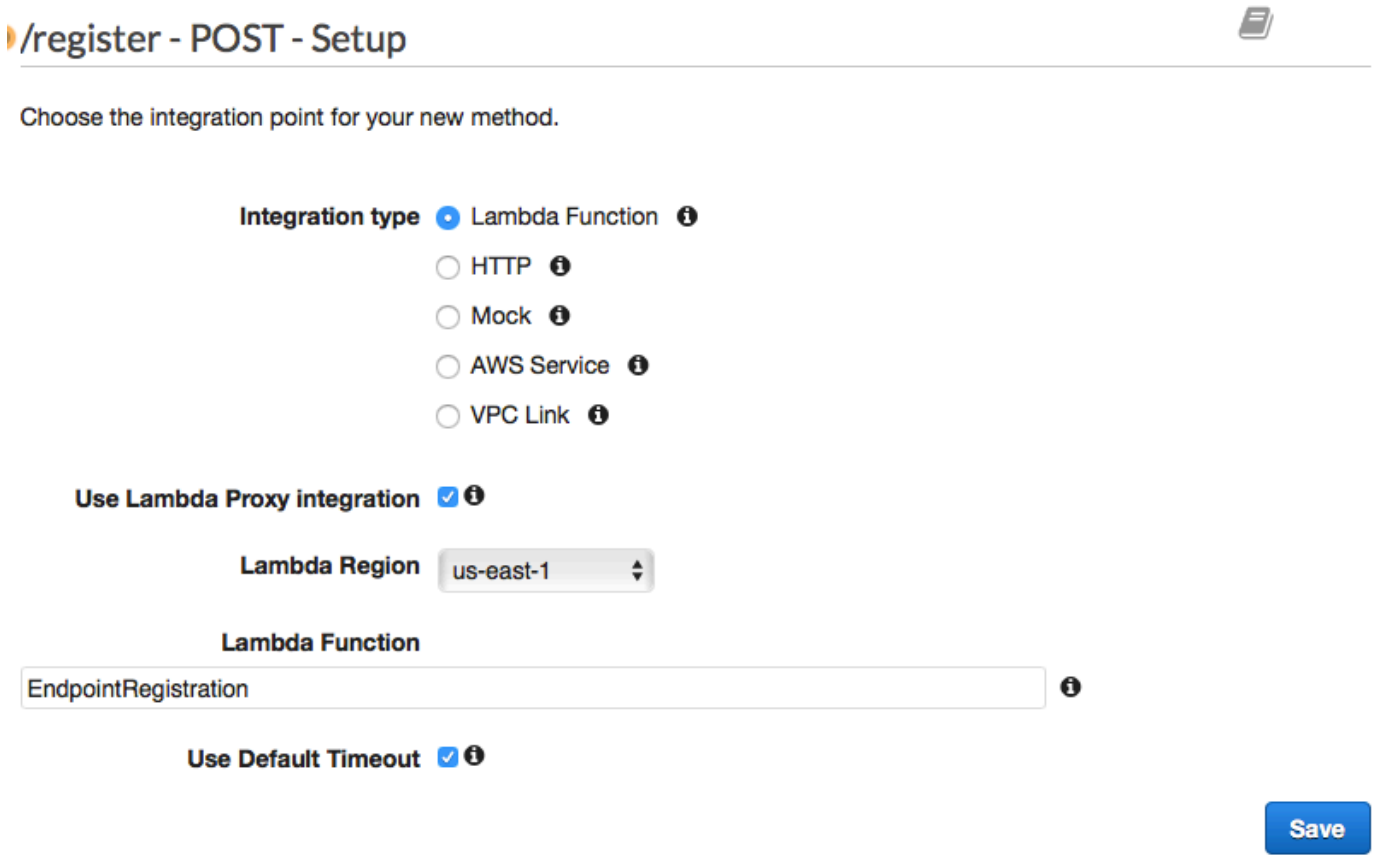
2. No menu Actions (Ações), selecione Create Method (Criar método). No menu exibido, escolha POST, conforme mostrado na imagem a seguir. Em seguida, escolha o botão de marca de seleção ( ).



3. No painel /register - POST - Setup, faça as seguintes seleções:

- Em Integration type (Tipo de integração), escolha Lambda Function (Função do Lambda).
- Selecione Usar a integração de proxy do Lambda.
- Em Região do Lambda, escolha a Região na qual você criou a função do Lambda.
- Para Função Lambda, escolha a RegisterEndpoint função que você criou em Criar funções Lambda.

Um exemplo dessas configurações é mostrado na imagem a seguir.



Selecione Save (Salvar). Na janela exibida, escolha OK para conceder permissão ao API Gateway para executar a função do Lambda.

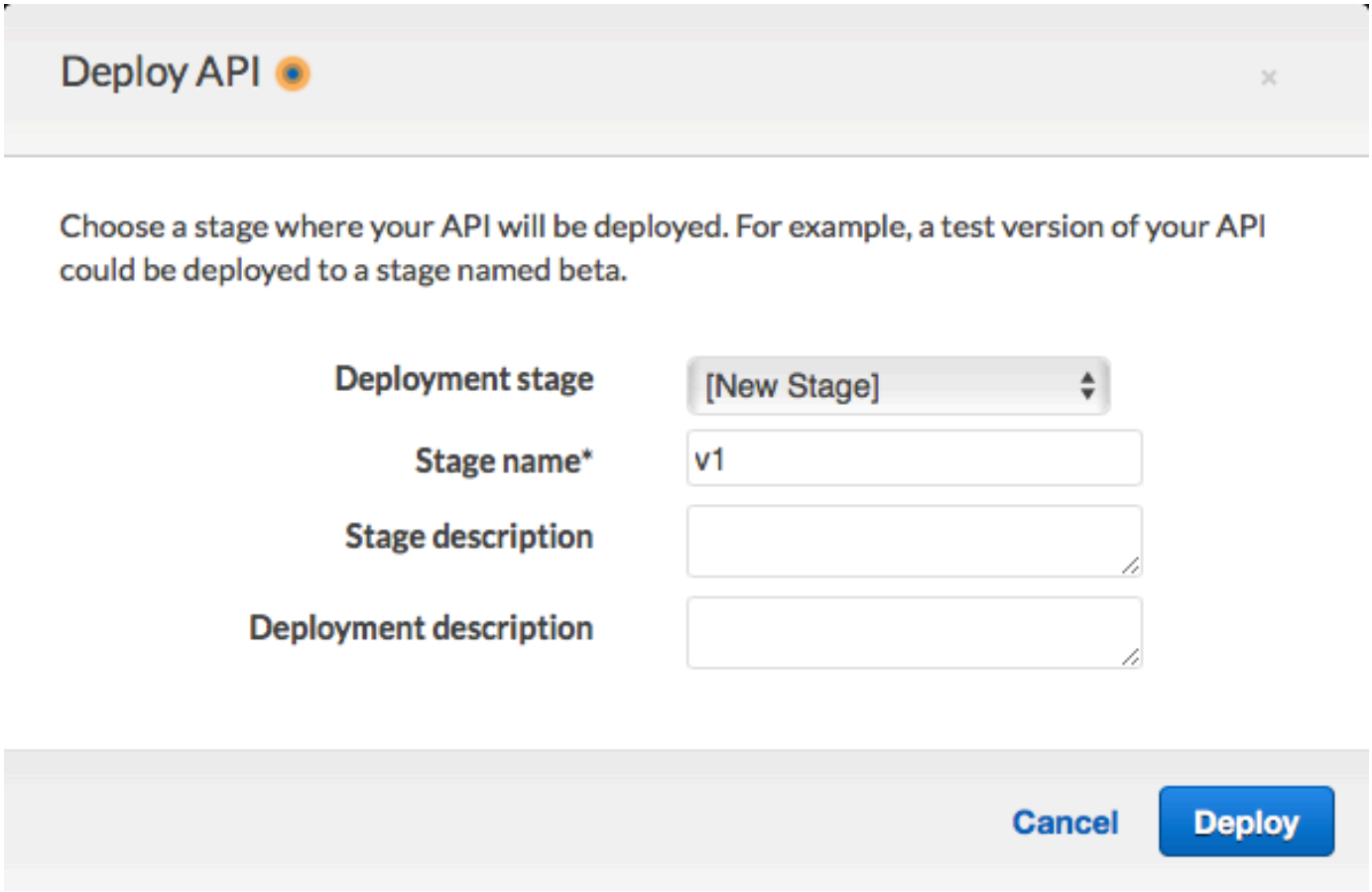
Implantar a API

Agora a API está pronta para uso. Nesse ponto, é preciso implantá-la para criar um endpoint publicamente acessível.

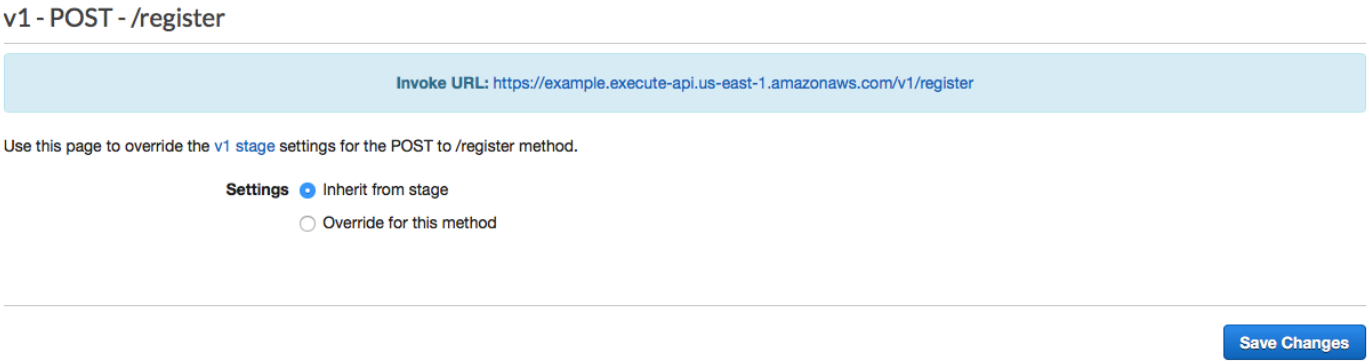
1. No menu Actions (Ações), escolha Deploy API (Implantar API). Na janela Deploy API (Implantar API), faça as seguintes seleções:

   - Em Deployment stage (Estágio de implantação), escolha [New Stage] ([Novo estágio]).
   - Em Stage name (Nome do estágio), insira **v1**.
   - Escolha Deploy (Implantar).

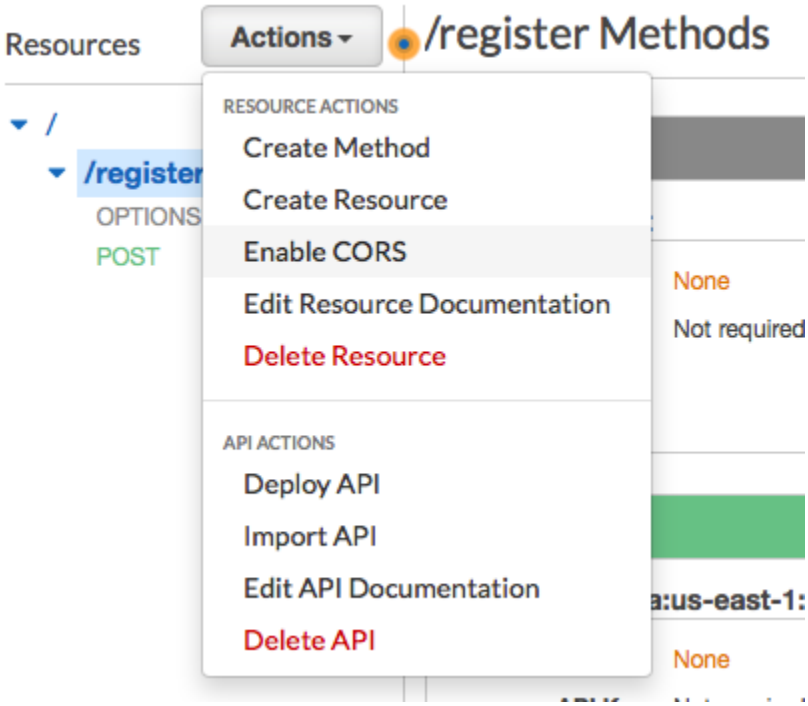
   Um exemplo dessas seleções é mostrado na imagem a seguir.



2. No painel v1 Stage Editor (Editor do estágio v1), escolha o recurso /register e depois o método POST. Copie o endereço exibido ao lado de Invoke URL Invocar URL), conforme mostrado na imagem a seguir.



3. No painel de navegação, escolha Resources (Recursos). Na lista de recursos, escolha o recurso /register. Por fim, no menu Actions (Ações), escolha Enable CORS (Habilitar o CORS), conforme mostrado na imagem a seguir.



4. Na painel Enable CORS (Habilitar o CORS), escolha Enable CORS and replace existing CORS headers (Habilitar o CORS e substituir os cabeçalhos do CORS existentes).

Próximo: Criar e implantar o formulário da Web

Etapa 5: criar e implantar o formulário da Web

Todos os componentes dessa solução que usam AWS serviços já estão prontos. A última etapa é criar e implantar o formulário da Web que captura os dados do cliente.

Crie o manipulador JavaScript de formulários

Nesta seção, você cria uma JavaScript função que analisa o conteúdo do formulário web criado na próxima seção. Depois de analisar o conteúdo, essa função envia os dados para a API que você criou em Configurar o Amazon API Gateway.

Para criar o manipulador de formulários

1. Em um editor de texto, crie um novo arquivo.
2. No editor, cole o código a seguir.

```
$(document).ready(function() {

  // Handle form submission.
  $("#submit").click(function(e) {

    var firstName = $("#firstName").val(),
        lastName = $("#lastName").val(),
        source = window.location.pathname,
        optTimestamp = undefined,
        utcSeconds = Date.now() / 1000,
        timestamp = new Date(0),
        phone = $("#areaCode").val()
              + $("#phone1").val()
              + $("#phone2").val();

    e.preventDefault();

    if (firstName == "") {
      $('#form-response').html('<div class="mt-3 alert alert-info"
 role="alert">Please enter your first name.</div>');
    } else if (lastName == "") {
      $('#form-response').html('<div class="mt-3 alert alert-info"
 role="alert">Please enter your last name.</div>');
    } else if (phone.match(/[^0-9]/gi)) {
      $('#form-response').html('<div class="mt-3 alert alert-info"
 role="alert">Your phone number contains invalid characters. Please check the phone
 number that you supplied.</div>');
    } else if (phone.length < 10) {
      $('#form-response').html('<div class="mt-3 alert alert-info"
 role="alert">Please enter your phone number.</div>');
    } else if (phone.length > 10) {
```

```
     $('#form-response').html('<div class="mt-3 alert alert-info"
role="alert">Your phone number contains too many digits. Please check the phone
number that you supplied.</div>');
   } else {
     $('#submit').prop('disabled', true);
     $('#submit').html('<span class="spinner-border spinner-border-sm"
role="status" aria-hidden="true"></span>  Saving your preferences</button>');

     timestamp.setUTCSeconds(utcSeconds);

     var data = JSON.stringify({
       'destinationNumber': phone,
       'firstName': firstName,
       'lastName': lastName,
       'source': source,
       'optTimestamp': timestamp.toString()
     });

     $.ajax({
       type: 'POST',
       url: 'https://example.execute-api.us-east-1.amazonaws.com/v1/register',
       contentType: 'application/json',
       data: data,
       success: function(res) {
         $('#form-response').html('<div class="mt-3 alert alert-success"
role="alert"><p>Congratulations! You&apos;ve successfully registered for SMS
Alerts from ExampleCorp.</p><p>We just sent you a message. Follow the instructions
in the message to confirm your subscription. We won&apos;t send any additional
messages until we receive your confirmation.</p><p>If you decide you don&apos;t
want to receive any additional messages from us, just reply to one of our messages
with the keyword STOP.</p></div>');
         $('#submit').prop('hidden', true);
         $('#unsubAll').prop('hidden', true);
         $('#submit').text('Preferences saved!');
       },
       error: function(jqxhr, status, exception) {
         $('#form-response').html('<div class="mt-3 alert alert-danger"
role="alert">An error occurred. Please try again later.</div>');
         $('#submit').text('Save preferences');
         $('#submit').prop('disabled', false);
       }
     });
   }
 });
```

```
});
```

3. No exemplo anterior, substitua *https://example.execute-api.us-east-1.amazonaws.com/v1/register* pelo URL de invocação que você obteve em Implantar a API.
4. Salve o arquivo.

Crie o arquivo do formulário

Nesta seção, você cria um arquivo HTML que contém o formulário que os clientes usam para se inscrever no seu programa de SMS. Esse arquivo usa o manipulador de JavaScript formulários que você criou na seção anterior para transmitir os dados do formulário para sua função Lambda.

> **Important**
>
> Quando um usuário envia esse formulário, ele aciona uma função do Lambda que chama várias operações da API do Amazon Pinpoint. Usuários mal-intencionados podem lançar um ataque ao seu formulário que poderia provocar um grande número de solicitações. Caso você planeje usar essa solução para produção, deverá protegê-la usando um sistema como o reCAPTCHA do Google.

Para criar o formulário

1. Em um editor de texto, crie um novo arquivo.
2. No editor, cole o código a seguir.

```
<!doctype html>
<html lang="en">

<head>
  <!-- Meta tags required by Bootstrap -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

```
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
 crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/
popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
 crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/
bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/
nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>

  <script type="text/javascript" src="SMSFormHandler.js"></script>
  <title>SMS Registration Form</title>
</head>

<body>
  <div class="container">
    <div class="row justify-content-center mt-3">
      <div class="col-md-6">
        <h1>Register for SMS Alerts</h1>
        <p>Enter your phone number below to sign up for PromotionName messages from
 ExampleCorp.</p>
        <p>We don't share your contact information with anyone else. For more
 information, see our <a href="http://example.com/privacy">Privacy Policy</a>.</p>
        <p>ExampleCorp alerts are only available to recipients in the United
 States.</p>
      </div>
    </div>
    <div class="row justify-content-center">
      <div class="col-md-6">
        <form>
          <div class="form-group">
            <label for="firstName" class="font-weight-bold">First name</label>
            <input type="text" class="form-control" id="firstName"
 placeholder="Your first name" required>
          </div>
          <div class="form-group">
            <label for="lastName" class="font-weight-bold">Last name</label>
            <input type="text" class="form-control" id="lastName" placeholder="Your
 last name" required>
          </div>
          <label for="areaCode" class="font-weight-bold">Phone number</label>
```

```
          <div class="input-group">
            <span class="h3">( </span>
            <input type="tel" class="form-control" id="areaCode" placeholder="Area
 code" required>
            <span class="h3"> ) </span>
            <input type="tel" class="form-control" id="phone1" placeholder="555"
 required>
            <span class="h3"> - </span>
            <input type="tel" class="form-control" id="phone2" placeholder="0199"
 required>
          </div>
          <div id="form-response"></div>
          <button id="submit" type="submit" class="btn btn-primary btn-block
 mt-3">Submit</button>
        </form>
      </div>
    </div>
    <div class="row mt-3">
      <div class="col-md-12 text-center">
        <small class="text-muted">Copyright © 2019, ExampleCorp or its
 affiliates.</small>
      </div>
    </div>
  </div>
</body>

</html>
```

3. No exemplo anterior, substitua *SMS FormHandler .js* pelo caminho completo para o JavaScript arquivo do manipulador de formulários que você criou na seção anterior.
4. Salve o arquivo.

Carregue os arquivos do formulário

Agora que você criou o formulário HTML e o manipulador de JavaScript formulários, a última etapa é publicar esses arquivos na Internet. Esta seção pressupõe que você tenha um provedor de hospedagem da Web existente. Se você não tiver um provedor de hospedagem existente, poderá lançar um site usando o Amazon Route 53, o Amazon Simple Storage Service (Amazon S3) e a Amazon. CloudFront Para obter mais informações, consulte Hospedar um site estático.

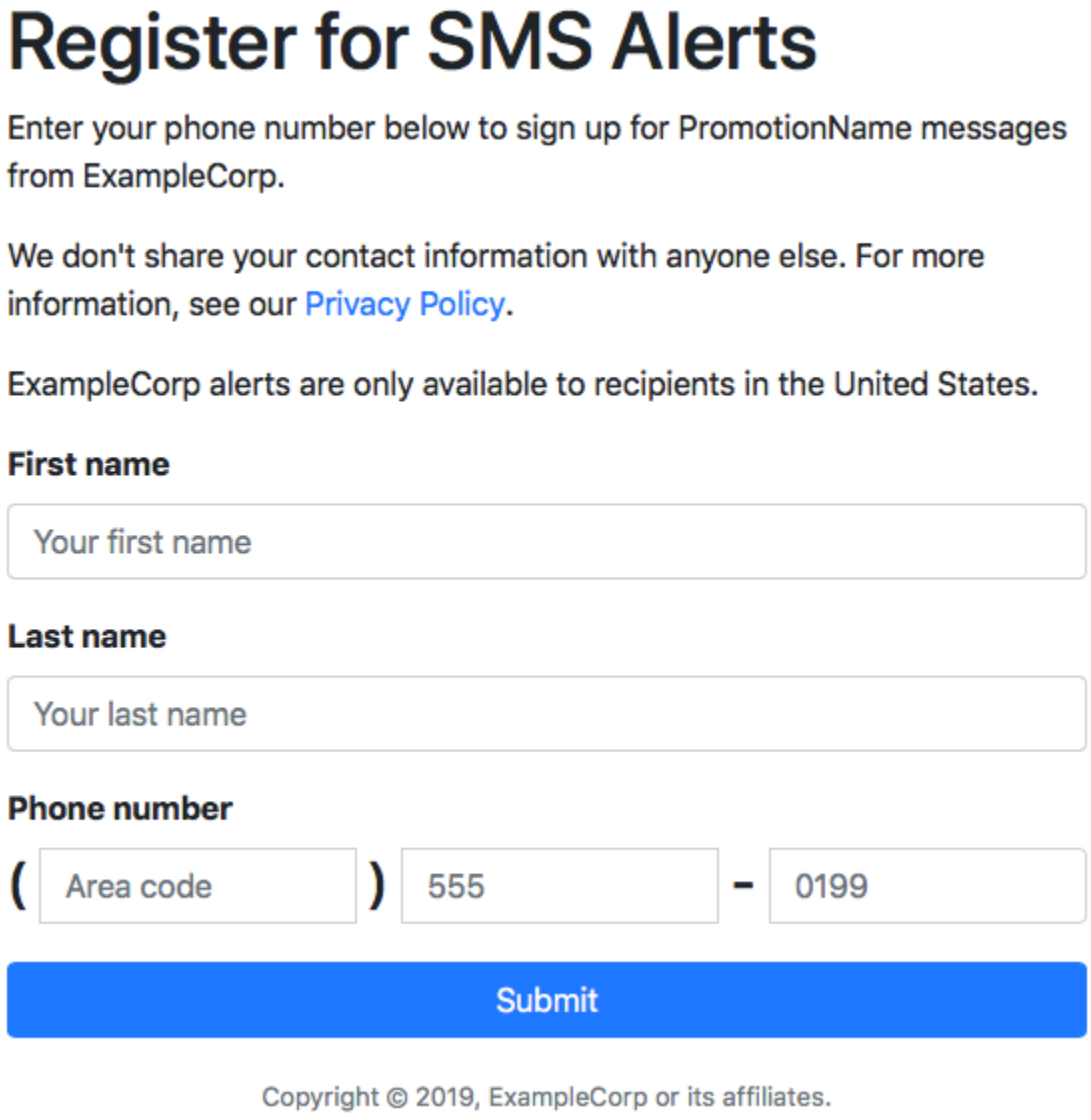
Se você usar outro provedor de hospedagem da Web, consulte a documentação do provedor para obter mais informações sobre a publicação de páginas da Web.

Testar o formulário

Depois de publicar o formulário, você deve enviar alguns eventos de teste para garantir que ele funcione conforme o esperado.

Para testar o formulário de registro

1. Em um navegador da Web, acesse o local em que carregou o formulário de registro. Se você usou o exemplo de código de Criar o manipulador de JavaScript formulários, verá um formulário semelhante ao exemplo na imagem a seguir.

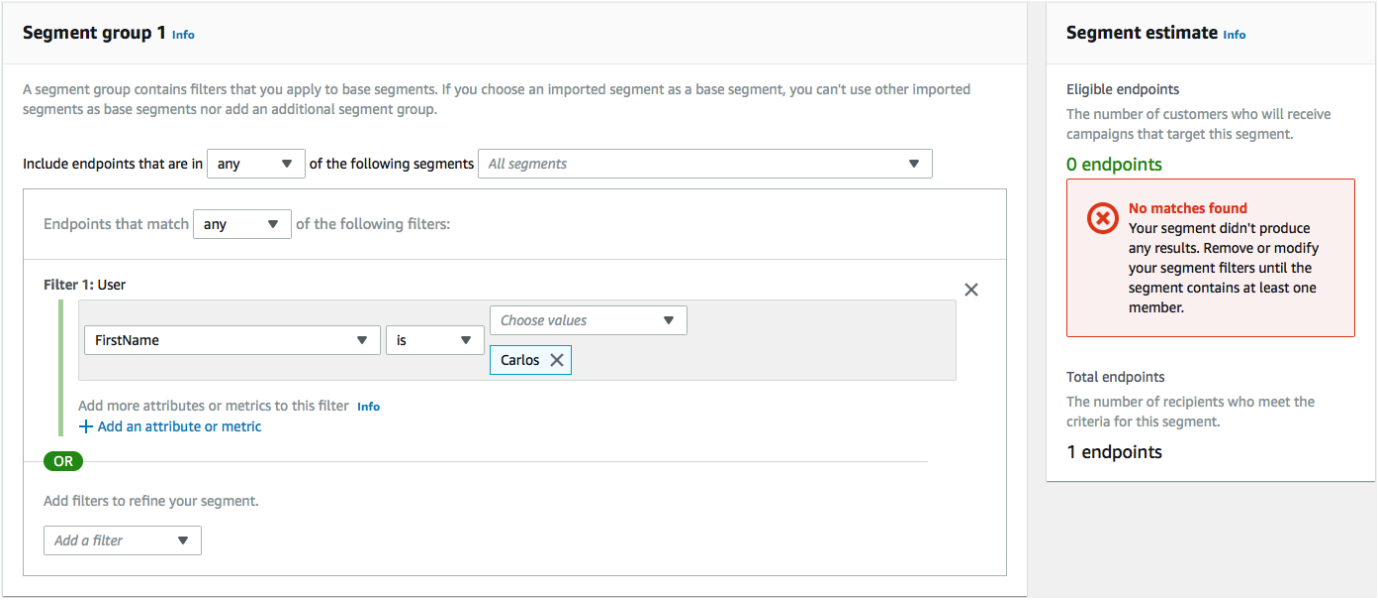


2. Insira suas informações de contato nos campos First name (Nome), Last name (Sobrenome) e Phone number (Número de telefone).

> **Note**
>
> Quando você envia o formulário, o Amazon Pinpoint tenta enviar uma mensagem para o número de telefone que você especificou. Por causa dessa funcionalidade, você deve usar um número de telefone real para testar a solução do início ao fim.
> Se você testou a função Lambda em Criar funções Lambda, seu projeto Amazon Pinpoint já contém pelo menos um endpoint. Ao testar esse formulário, você deve enviar um número de telefone diferente no formulário ou excluir o endpoint existente usando a operação da DeleteEndpointAPI.

3. Verifique o dispositivo associado ao número de telefone que você especificou para garantir que recebeu a mensagem.
4. Faça login no console do Amazon Pinpoint em https://console.aws.amazon.com/pinpoint/.
5. Na página Todos os projetos, escolha o projeto que você criou em Criar um projeto Amazon Pinpoint.
6. No painel de navegação, selecione Segmentos. Na página Segmentos, selecione Criar um segmento.
7. Em Segment group 1 (Grupo de segmentos 1), em Add filters to refine your segment (Adicionar filtros para refinar seu segmento), escolha Filter by user (Filtrar por usuário).
8. Em Escolher um atributo de usuário, escolha FirstName. Em seguida, em Choose values (Escolher valores), escolha o primeiro nome que você especificou quando enviou o formulário.

   A seção Segment estimate (Estimativa de segmento) deve mostrar que há zero endpoints elegíveis e um endpoint (em Total de endpoints), conforme mostrado no exemplo a seguir. Tal resultado é esperado. Quando a função do Lambda cria um endpoint, o endpoint é cancelado por padrão.



9. No dispositivo que recebeu a mensagem, responda à mensagem com a palavra-chave SMS bidirecional que você especificou em Habilitar SMS bidirecional. O Amazon Pinpoint envia uma mensagem de resposta imediatamente.
10. No console do Amazon Pinpoint, repita as etapas de 4 a 8. Desta vez, ao criar o segmento, você verá um endpoint elegível e um endpoint total. Esse resultado é esperado, porque o endpoint agora está ativado.

Próximas etapas

Ao concluir este tutorial, você fez o seguinte:

- Criou um projeto do Amazon Pinpoint, configurou o canal do SMS e obteve um código longo dedicado.
- Criou uma política do IAM que usa o princípio do privilégio mínimo para conceder direitos de acesso e associou essa política a uma função.
- Criou duas funções do Lambda que usam as operações PhoneNumberValidate, UpdateEndpoint e SendMessages na API do Amazon Pinpoint.
- Criou uma API REST usando o API Gateway.
- Criou e implantou um formulário baseado na Web que coleta informações de contato dos clientes.
- Realizou testes na solução para ter certeza de que funciona.

Esta seção discute algumas maneiras pelas quais você pode usar as informações do cliente coletadas por meio dessa solução. Ela também inclui algumas sugestões de formas de personalizar essa solução para se adequar ao seu caso de uso exclusivo.

Criar segmentos de clientes

Todos os detalhes do cliente coletados por meio desse formulário são armazenados como endpoints. Essa solução cria endpoints que contêm vários atributos que você pode usar para fins de segmentação.

Por exemplo, essa solução captura um atributo de endpoint chamado `Source`. Esse atributo contém o caminho completo para o local onde o formulário foi hospedado. Ao criar um segmento, você pode filtrar o segmento por endpoint e refinar ainda mais o filtro, escolhendo um atributo `Source`.

Criar segmentos com base no atributo `Source` pode ser útil de várias maneiras. Primeiro, isso permite que você crie rapidamente um segmento de clientes que se cadastraram para receber mensagens SMS da sua parte. Além disso, a ferramenta de segmentação no Amazon Pinpoint exclui automaticamente os endpoints que não optaram por receber mensagens.

O atributo `Source` também é útil se você decidir hospedar o formulário de registro em vários locais diferentes. Por exemplo, seu material de marketing pode se referir a um formulário hospedado em um único local, embora os clientes que encontrarem o formulário ao navegar em seu site possam ver uma versão hospedada em outro lugar. Ao fazer isso, os atributos de origem dos clientes que preenchem o formulário depois de ver seus materiais de marketing são diferentes daqueles que preenchem o formulário depois de encontrá-lo em seu site. Você pode usar essa diferença para criar segmentos distintos e enviar comunicações personalizadas para cada um desses públicos.

Enviar mensagens de campanha personalizadas

Depois de criar segmentos, você pode começar a enviar campanhas para esses segmentos. Ao criar mensagens de campanha, você pode personalizá-las especificando quais atributos de endpoint deseja incluir na mensagem. Por exemplo, o formulário da Web usado nesta solução exige que o cliente insira seu nome e sobrenome. Esses valores são armazenados no registro do usuário associado ao endpoint.

Por exemplo, se você usar a operação da API `GetEndpoint` para recuperar informações sobre um endpoint que foi criado usando essa solução, verá uma seção semelhante ao seguinte exemplo:

```
...
```

```
  "User": {
    "UserAttributes": {
      "FirstName": [
        "Carlos"
      ],
      "LastName": [
        "Salazar"
      ]
    }
  }
  ...
```

Se você quiser incluir os valores desses atributos em sua mensagem de campanha, use a notação de ponto para se referir ao atributo. Depois, coloque a referência inteira entre chaves duplas. Por exemplo, para incluir o primeiro nome de cada destinatário em uma mensagem de campanha, inclua a seguinte string na mensagem: `{{User.UserAttributes.FirstName}}`. Quando o Amazon Pinpoint envia a mensagem, ela substitui a cadeia de caracteres pelo valor do atributo `FirstName`.

Usar o formulário para coletar informações adicionais

Você pode modificar essa solução para coletar informações adicionais no formulário de registro. Por exemplo, você poderia pedir ao cliente para fornecer seu endereço e usar os dados de endereço para preencher os campos `Location.City`, `Location.Country`, `Location.Region` e `Location.PostalCode` no recurso de `Endpoint`. A coleta de informações de endereço no formulário de registro pode resultar no endpoint contendo informações mais precisas. Para fazer essa alteração, você precisa adicionar os campos apropriados ao formulário da Web. Você também precisa modificar o código JavaScript do formulário para passar os novos valores. Por fim, você precisa modificar a função do Lambda que cria o endpoint para manipular as novas informações de entrada.

Você também pode modificar o formulário para que ele colete informações de contato em outros canais. Por exemplo, você pode usar o formulário para coletar o endereço de e-mail do cliente, além do número de telefone dele. Para fazer essa alteração, você precisa modificar o HTML e o JavaScript do formulário da Web. Também é necessário modificar a função do Lambda que cria o endpoint para que ele crie dois endpoints separados (um para o endpoint de e-mail e outro para o endpoint do SMS). Você também deve modificar a função do Lambda para que ela gere um valor exclusivo para o atributo `User.UserId` e associe esse valor a ambos os endpoints.

Registrar atributos adicionais para fins de auditoria

Essa solução registra dois atributos valiosos quando cria e atualiza endpoints. Primeiro, quando a primeira função do Lambda cria o endpoint, ela registra o URL do próprio formulário no atributo `Attributes.Source`. Se o cliente responder à mensagem, a segunda função do Lambda criará um atributo `Attributes.OptInTimestamp`. Esse atributo contém a data e a hora exatas em que o cliente forneceu seu consentimento para receber mensagens da sua parte.

Esses dois campos podem ser úteis se você for solicitado por uma operadora de celular ou uma agência reguladora para fornecer evidências do consentimento de um cliente. Você pode recuperar essas informações a qualquer momento usando a operação da API GetEndpoint.

Você também pode modificar as funções do Lambda para registrar dados adicionais que podem ser úteis para fins de auditoria, como o endereço IP do qual a solicitação de registro foi enviada.

Integrar o Amazon Pinpoint ao aplicativo

Integre o Amazon Pinpoint ao seu código de cliente para compreender e envolver os usuários.

Após a integração, quando os usuários executam seu aplicativo, ele se conecta ao serviço do Amazon Pinpoint para adicionar ou atualizar os endpoints. Os endpoints representam os destinos para os quais você pode enviar mensagens, como dispositivos do usuário, endereços de e-mail ou números de telefone.

Além disso, seu aplicativo fornece dados de uso ou eventos. Visualize os dados de eventos no console do Amazon Pinpoint para saber quantos usuários você tem, a frequência com que eles usam o aplicativo, quando eles o utilizam e muito mais.

Depois que seu aplicativo fornece endpoints e eventos, você pode usar essas informações para personalizar campanhas de mensagens para audiências ou segmentos específicos. (Você também pode enviar mensagens usando listas simples de destinatários, sem criar campanhas.)

Use os tópicos desta seção para integrar o Amazon Pinpoint ao cliente móvel ou web. Esses tópicos incluem exemplos de código e procedimentos para integração com um aplicativo Android JavaScript, Swift ou Flutter. Para iniciar a integração dos aplicativos, consulte the section called “Conectar o aplicativo de front-end com o AWS Amplify”.

Fora do seu cliente, você pode usar os AWS SDKs compatíveis ou a API do Amazon Pinpoint para importar endpoints, exportar dados de eventos, definir segmentos de clientes, criar e executar campanhas e muito mais.

Tópicos

- Usando o Amazon Pinpoint com um SDK AWS
- Conectar o aplicativo de front-end ao Amazon Pinpoint com o AWS Amplify
- Registrar endpoints no aplicativo
- Relatar eventos no aplicativo
- Gerenciar notificações por push

Usando o Amazon Pinpoint com um SDK AWS

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que facilitam a criação de aplicações em seu idioma preferido pelos desenvolvedores.

Documentação do SDK	Exemplos de código
AWS SDK for C++	AWS SDK for C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK for Go	AWS SDK for Go exemplos de código
AWS SDK for Java	AWS SDK for Java exemplos de código
AWS SDK for JavaScript	AWS SDK for JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK for .NET	AWS SDK for .NET exemplos de código
AWS SDK for PHP	AWS SDK for PHP exemplos de código
AWS Tools for PowerShell	Ferramentas para exemplos PowerShell de código
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemplos de código
AWS SDK for Ruby	AWS SDK for Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
SDK da AWS para SAP ABAP	SDK da AWS para SAP ABAP exemplos de código
AWS SDK for Swift	AWS SDK for Swift exemplos de código

Para obter exemplos específicos do Amazon Pinpoint, consulte Exemplos de código para o Amazon Pinpoint usando AWS SDKs.

> **Exemplo de disponibilidade**
>
> Você não consegue encontrar o que precisa? Solicite um código de exemplo no link Fornecer feedback na parte inferior desta página.

Conectar o aplicativo de front-end ao Amazon Pinpoint com o AWS Amplify

Use o AWS Amplify para integrar seu aplicativo ao AWS. Para aplicativos Swift, consulte Conceitos básicos na documentação do Amplify para Swift. Para aplicativos Android, consulte a documentação sobre Conceitos básicos de SDK do Amplify para Android. Para o aplicativo React Native, consulte Conceitos básicos na documentação do Amplify JavaScript. Para aplicativos Flutter, consulte a documentação sobre Conceitos básicos do SDK do Flutter. Esses tópicos ajudam você a:

- Configuração dos recursos de back-end.
- Conecte o aplicativo aos recursos de back-end usando as bibliotecas do Amplify.

Para saber mais sobre como conectar o aplicativo de front-end ao Amazon Pinpoint para análises, mensagens no aplicativo e notificações por push, consulte AWS Amplify.

Próxima etapa

Você integrou o AWS Amplify ao aplicativo. Em seguida, atualize o código para registrar os dispositivos dos usuários como endpoints. Consulte Registrar endpoints no aplicativo.

Registrar endpoints no aplicativo

Quando um usuário inicia uma sessão (por exemplo, ao executar um aplicativo móvel), o aplicativo móvel ou web pode registrar automaticamente (ou atualizar) um endpoint com o Amazon Pinpoint. O endpoint representa o dispositivo com o qual o usuário inicia a sessão. Ele inclui os atributos que descrevem o dispositivo e também pode incluir atributos personalizados que você define. Os endpoints também podem representar outros métodos de comunicação com os clientes, como endereços de e-mail ou números de celular.

Depois que o aplicativo registra os endpoints, você pode segmentar seu público com base nos atributos do endpoint. Em seguida, você pode atrair esses segmentos com campanhas de mensagens personalizadas. Você pode também usar a página Analytics no console do Amazon Pinpoint para exibir gráficos sobre o registro de endpoints e de atividade, como novos endpoints e endpoints ativos diariamente.

Você pode atribuir um único ID de usuário a vários endpoints. Um ID de usuário representa um único usuário, ao passo que cada endpoint que recebe o ID de usuário representa um dos dispositivos do usuário. Assim que atribuir IDs de usuário aos endpoints, poderá exibir gráficos sobre a atividade dos usuários no console, como Daily active users e Monthly active users.

Antes de começar

Se você ainda não tiver feito isso, integre o AWS Mobile SDK para Android ou iOS ou a biblioteca JavaScript do AWS Amplify ao seu aplicativo. Consulte Conectar o aplicativo de front-end ao Amazon Pinpoint com o AWS Amplify.

Registrar endpoints com os AWS Mobile SDKs para Android ou iOS

Você pode usar os AWS Mobile SDKs para Android ou iOS para registrar e personalizar endpoints. Para obter mais informações e ver exemplos de código, consulte estes documentos:

- Registrar endpoints no seu aplicativo na documentação do SDK do Android.
- Registrar endpoints no aplicativo na documentação do SDK do iOS.

Registrar endpoints com a biblioteca JavaScript do AWS Amplify

Você pode usar a biblioteca AWS Amplify JavaScript para registrar e atualizar endpoints em seus aplicativos. Para obter mais informações e ver exemplos de código, consulte Atualizar endpoint na documentação do AWS Amplify JavaScript.

Próximas etapas

Você atualizou seu aplicativo para registrar os endpoints. Agora, quando os usuários executam seu aplicativo, as informações do dispositivo e os atributos personalizados são fornecidos ao Amazon Pinpoint. Você pode usar essas informações para definir segmentos de público. No console, você pode ver métricas sobre endpoints e, se aplicável, os usuários que receberam IDs de usuário.

Em seguida, execute as etapas em Relatar eventos no aplicativo para atualizar o aplicativo para relatar dados de uso.

Relatar eventos no aplicativo

Em seu aplicativo móvel ou Web, você pode usar os AWS SDKs Mobile ou a API de eventos do Amazon Pinpoint para relatar dados de uso ou eventos ao Amazon Pinpoint. Você pode relatar eventos para capturar informações, como os tempos das sessões, o comportamento de compra dos usuários, as tentativas de login ou qualquer tipo de evento personalizado que você precisar.

Depois que o aplicativo relata os eventos, você pode visualizar uma análise no console do Amazon Pinpoint. Os gráficos na página Análises fornecem métricas para muitos aspectos do comportamento dos usuários. Para obter mais informações, consulte Referência em gráfico para análises do Amazon Pinpoint no Guia do usuário do Amazon Pinpoint.

Para analisar e armazenar os dados do evento fora do Amazon Pinpoint, configure-o para transmitir os dados ao Amazon Kinesis. Para obter mais informações, consulte Transmissão de eventos do Amazon Pinpoint para o Kinesis.

Usando os AWS Mobile SDKs e as bibliotecas AWS Amplify JavaScript, você pode chamar a API do Amazon Pinpoint para relatar os seguintes tipos de eventos:

Eventos de sessões

Indicam quando e com que frequência os usuários abrem e fecham o aplicativo.

Após o aplicativo relatar eventos de sessão, use a página Analytics no console do Amazon Pinpoint para visualizar os gráficos de Sessões, endpoints ativos diariamente, Taxa de retenção de 7 dias, entre outros.

Eventos personalizados

São eventos fora do padrão que você define ao atribuir um tipo de evento personalizado. Você pode adicionar atributos e métricas personalizados a um evento personalizado.

Na página Análises do console, a guia Eventos exibe as métricas de todos os eventos personalizados relatados pelo aplicativo.

Eventos de monetização

Relate a receita que é gerada pelo seu aplicativo e o número de itens comprados pelos usuários.

Na página Análises, a guia Receita exibe os gráficos de Receita, Usuários pagantes, Unidades vendidas, entre outros.

Eventos de autenticação

Indica a frequência com que os usuários são autenticados no seu aplicativo.

Na página Análises, a guia Usuários exibe os gráficos de Logins, Cadastros e Falhas de autenticação.

Antes de começar

Se você ainda não tiver feito isso, faça o seguinte:

- Integre seu aplicativo ao AWS Amplify. Consulte Conectar o aplicativo de front-end ao Amazon Pinpoint com o AWS Amplify.
- Atualize o aplicativo para registrar os endpoints. Consulte Registrar endpoints no aplicativo.

Relatar eventos com os AWS Mobile SDKs para Android ou iOS

Você pode habilitar um aplicativo móvel para relatar eventos ao Amazon Pinpoint usando os AWS Mobile SDKs para iOS e Android.

Para obter mais informações sobre como atualizar o aplicativo para gravar e enviar eventos ao Amazon Pinpoint, consulte estas páginas na documentação do AWS Amplify:

- Análises na documentação do SDK do iOS
- Análises na documentação do SDK do Android

Relatar eventos com a biblioteca JavaScript do AWS Amplify

Você pode habilitar os aplicativos JavaScript e React Native para relatar eventos de uso do aplicativo para o Amazon Pinpoint usando a biblioteca AWS Amplify JavaScript. Para obter mais informações sobre como atualizar o aplicativo para enviar eventos ao Amazon Pinpoint, consulte Analytics na documentação do AWS Amplify JavaScript.

Relatório de eventos usando a API do Amazon Pinpoint

Você pode usar a API do Amazon Pinpoint ou um AWS SDK para enviar eventos em massa ao Amazon Pinpoint. Para obter mais informações, consulte Eventos na Referência de API do Amazon Pinpoint.

Próxima etapa

Você atualizou seu aplicativo para relatar os eventos. Agora, quando os usuários interagem com o aplicativo, ele envia dados de uso para o Amazon Pinpoint. Você pode visualizar esses dados no console e transmiti-los para o Amazon Kinesis.

Em seguida, atualize o aplicativo para tratar das notificações por push que você envia pelo Amazon Pinpoint. Consulte Gerenciar notificações por push.

Gerenciar notificações por push

Os tópicos a seguir descrevem como modificar o aplicativo Swift, Android, React Native ou Flutter para receber notificações por push que você envia usando o Amazon Pinpoint.

Tópicos

- Configurar notificações por push para Amazon Pinpoint
- Gerenciar notificações por push

Configurar notificações por push para Amazon Pinpoint

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

- Para aplicativos iOS, você fornece um certificado SSL, que obtém do portal do desenvolvedor da Apple. O certificado autoriza o Amazon Pinpoint a enviar mensagens para seu aplicativo por meio do Apple Push Notification Service.
- Para aplicativos Android, forneça uma chave de API da Web, obtida do console do Firebase. Essas credenciais autorizam o Amazon Pinpoint a enviar mensagens para seu aplicativo pelo Firebase Cloud Messaging.

Depois de obter as credenciais para um canal de notificação por push, você precisa criar um projeto no Amazon Pinpoint e fornecer as credenciais para o serviço de notificações por push.

Tópicos

- Configurar notificações por push do Swift
- Configurar as notificações por push em Android
- Configurar notificações por push do Flutter
- Configurar notificações por push do React Native
- Criar um projeto no Amazon Pinpoint

Configurar notificações por push do Swift

Notificações por push para aplicativos iOS e feito por meio do Apple Push Notification Service (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 notificações por push na documentação do AWS Amplify.

Trabalhar com tokens de APNs

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 Apple Push Notification Service (APNs) rejeitará a notificação, pois o token agora é inválido. Ao tentar enviar a notificação, você receberá uma notificação de falha de mensagem do APNs.

Configurar as notificações por push em Android

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

Configurar notificações por push do Flutter

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

Configurar notificações por push do React Native

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

Criar um projeto no Amazon Pinpoint

No Amazon Pinpoint, um projeto é um conjunto de configurações, dados, campanhas e segmentos que possuem uma finalidade comum. Na API do Amazon Pinpoint, os projetos também são chamados de aplicativos. Esta seção usa a palavra "projeto" exclusivamente ao fazer referência a esse conceito.

Para iniciar o envio de notificações por push no Amazon Pinpoint, você precisa criar um projeto. Em seguida, você precisa habilitar os canais de notificação por push que você deseja usar, fornecendo as credenciais apropriadas.

Você pode criar novos projetos e configurar canais de notificação por push usando o console do Amazon Pinpoint. Para obter mais informações, consulte Configuração dos canais de notificação push do Amazon Pinpoint no Guia do usuário do Amazon Pinpoint.

Você também pode criar e configurar projetos usando a API do Amazon Pinpoint, um AWS SDK ou a AWS Command Line Interface (AWS CLI). Para criar um projeto, use o recurso `Apps`. Para configurar canais de notificação por push, use os seguintes recursos:

- Canal APNs para enviar mensagens para usuários de dispositivos iOS usando o Apple Push Notification service.
- Canal ADM para enviar mensagens para usuários de dispositivos Amazon Kindle Fire.
- Canal Baidu para enviar mensagens para usuários do Baidu.
- Canal GCM para enviar mensagens a dispositivos Android usando o Firebase Cloud Messaging (FCM), que substitui o Google Cloud Messaging (GCM).

Gerenciar notificações por push

Depois de obter as credenciais necessárias para enviar notificações por push, você pode atualizar seus aplicativos para que eles possam receber notificações por push. Para obter mais informações, consulte Notificações por push — Conceitos básicos na documentação do AWS Amplify.

Definir o público-alvo para o Amazon Pinpoint

No Amazon Pinpoint, cada membro de seu público-alvo é representado por um ou mais endpoints. Quando usa o Amazon Pinpoint para enviar uma mensagem, você direciona a mensagem para os endpoints que representam os membros de seu público-alvo. Cada definição de endpoint inclui um destino de mensagens, como um token de dispositivo, um endereço de e-mail ou um número de telefone. Também inclui dados sobre seus usuários e seus dispositivos. Antes de analisar, segmentar ou envolver seu público-alvo, a primeira etapa é adicionar endpoints a seu projeto do Amazon Pinpoint.

Para adicionar endpoints, você pode:

- Integrar o Amazon Pinpoint com o cliente Android, iOS ou JavaScript para que os endpoints sejam adicionados automaticamente quando os usuários visitarem seu aplicativo.
- Use a API do Amazon Pinpoint para adicionar endpoints individualmente ou em lotes.
- Importar definições de endpoint que são armazenadas fora do Amazon Pinpoint.

Depois de adicionar endpoints, você pode:

- Visualizar análises sobre seu público-alvo no console do Amazon Pinpoint.
- Saber mais sobre seu público-alvo examinando ou exportando dados do endpoint.
- Definir segmentos do público-alvo com base em atributos do endpoint, como dados demográficos ou interesses dos usuários.
- Envolver seu público-alvo com campanhas de mensagens personalizadas.
- Enviar mensagens diretamente para listas de endpoints.

Usar os tópicos desta seção para adicionar, atualizar e excluir endpoints usando a API do Amazon Pinpoint. Para adicionar endpoints automaticamente de seu cliente Android, iOS ou JavaScript, consulte Registrar endpoints no aplicativo.

Tópicos

- Adicionar endpoints ao Amazon Pinpoint
- Associar usuários aos endpoints do Amazon Pinpoint
- Adição de um lote de endpoints ao Amazon Pinpoint
- Importar endpoints para o Amazon Pinpoint

- Excluir endpoints do Amazon Pinpoint
- Gerenciar o número máximo de endpoints de um público-alvo

Adicionar endpoints ao Amazon Pinpoint

Um endpoint representa um destino para o qual é possível enviar mensagens, como um dispositivo móvel, um número de telefone ou um endereço de e-mail. Para poder enviar mensagens a um membro de seu público-alvo, você deve definir um ou mais endpoints para esse indivíduo.

Ao definir um endpoint, você especifica o canal e o endereço. O canal é o tipo de plataforma que você usa para enviar mensagens para o endpoint. Os exemplos de canais incluem um serviço de notificação por push, SMS ou e-mail. O endereço especifica para onde enviar mensagem ao endpoint, como um token de dispositivo, um número de telefone ou um endereço de e-mail.

Para adicionar mais detalhes sobre o seu público-alvo, você pode enriquecer seus endpoints com atributos padrão e personalizados. Esses atributos podem incluir dados sobre seus usuários, suas preferências, seus dispositivos, as versões do cliente que usam ou seus locais. Quando adiciona esse tipo de dados a seus endpoints, você pode:

- Visualizar gráficos sobre seu público-alvo no console do Amazon Pinpoint.
- Segmentar seu público-alvo com base em atributos de endpoint para que possa enviar suas mensagens ao público-alvo correto.
- Personalizar suas mensagens incorporando variáveis na mensagem que são substituídas pelos valores dos atributos do endpoint.

Um aplicativo cliente móvel ou JavaScript registra endpoints automaticamente se você integrar o Amazon Pinpoint usando os SDKs do AWS Mobile ou a biblioteca Amplify JavaScript da AWS. O cliente registra um endpoint para cada novo usuário e atualiza endpoints para usuários que retornam. Para registrar endpoints de um cliente móvel ou JavaScript, consulte Registrar endpoints no aplicativo.

Exemplos

Os exemplos a seguir mostram como adicionar um endpoint a um projeto do Amazon Pinpoint. O endpoint representa um membro do público-alvo que vive em Seattle e usa um iPhone. Essa pessoa pode receber mensagens por meio do Apple Push Notification service (APNs). O endereço do endpoint é o token do dispositivo que é fornecido pelo APNs.

AWS CLI

Você pode usar o Amazon Pinpoint executando comandos com a AWS CLI.

Example Comando Update endpoint

Para adicionar ou atualizar um endpoint, use o comando update-endpoint:

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
> --endpoint-request file://endpoint-request-file.json
```

Em que:

- application-id é o ID do projeto do Amazon Pinpoint no qual você está adicionando ou atualizando um endpoint.
- example-endpoint é o ID que você está atribuindo a um novo endpoint ou o ID de um endpoint existente que você está atualizando.
- endpoint-request-file.json é o caminho para um arquivo JSON local que contém a entrada para o parâmetro `--endpoint-request`.

Example Arquivo de solicitação de endpoint

O comando `update-endpoint` de exemplo usa um arquivo JSON como o argumento para o parâmetro `--endpoint-request`. Esse arquivo contém uma definição de endpoint, como a seguinte:

```
{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
```

```
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

Para os atributos que você pode usar para definir um endpoint, consulte o esquema EndpointRequest na Referência da API do Amazon Pinpoint.

AWS SDK for Java

Você pode usar a API do Amazon Pinpoint em seus aplicativos Java usando o cliente fornecido pelo AWS SDK for Java.

Example Código

Para adicionar um endpoint, inicialize um objeto `EndpointRequest` e passe-o para o método `updateEndpoint` do cliente do `AmazonPinpoint`:

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.*;
import java.util.Arrays;

public class AddExampleEndpoint {

 public static void main(String[] args) {
```

```
 final String USAGE = "\n" +
   "AddExampleEndpoint - Adds an example endpoint to an Amazon Pinpoint
application." +
   "Usage: AddExampleEndpoint <applicationId>" +
   "Where:\n" +
   "  applicationId - The ID of the Amazon Pinpoint application to add the example
" +
   "endpoint to.";

 if (args.length < 1) {
  System.out.println(USAGE);
  System.exit(1);
 }

 String applicationId = args[0];

 // The device token assigned to the user's device by Apple Push Notification
 // service (APNs).
 String deviceToken =
"1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f";

 // Initializes an endpoint definition with channel type and address.
 EndpointRequest wangXiulansIphoneEndpoint = new EndpointRequest()
   .withChannelType(ChannelType.APNS)
   .withAddress(deviceToken);

 // Adds custom attributes to the endpoint.
 wangXiulansIphoneEndpoint.addAttributesEntry("interests", Arrays.asList(
   "technology",
   "music",
   "travel"));

 // Adds custom metrics to the endpoint.
 wangXiulansIphoneEndpoint.addMetricsEntry("technology_interest_level", 9.0);
 wangXiulansIphoneEndpoint.addMetricsEntry("music_interest_level", 6.0);
 wangXiulansIphoneEndpoint.addMetricsEntry("travel_interest_level", 4.0);

 // Adds standard demographic attributes.
 wangXiulansIphoneEndpoint.setDemographic(new EndpointDemographic()
   .withAppVersion("1.0")
   .withMake("apple")
   .withModel("iPhone")
   .withModelVersion("8")
   .withPlatform("ios")
```

```
    .withPlatformVersion("11.3.1")
    .withTimezone("America/Los_Angeles"));

  // Adds standard location attributes.
  wangXiulansIphoneEndpoint.setLocation(new EndpointLocation()
    .withCountry("US")
    .withCity("Seattle")
    .withPostalCode("98121")
    .withLatitude(47.61)
    .withLongitude(-122.33));

  // Initializes the Amazon Pinpoint client.
  AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
    .withRegion(Regions.US_EAST_1).build();

  // Updates or creates the endpoint with Amazon Pinpoint.
  UpdateEndpointResult result = pinpointClient.updateEndpoint(new
 UpdateEndpointRequest()
    .withApplicationId(applicationId)
    .withEndpointId("example_endpoint")
    .withEndpointRequest(wangXiulansIphoneEndpoint));

  System.out.format("Update endpoint result: %s\n",
 result.getMessageBody().getMessage());

 }
}
```

HTTP

Você pode usar o Amazon Pinpoint fazendo solicitações HTTP diretamente para a API REST.

Example Solicitação PUT endpoint

Para adicionar um endpoint, envie uma solicitação PUT para o recurso Endpoint no seguinte URI:

`/v1/apps/`*`application-id`*`/endpoints/`*`endpoint-id`*

Em que:

- application-id é o ID do projeto do Amazon Pinpoint no qual você está adicionando ou atualizando um endpoint.
- endpoint-id é o ID que você está atribuindo a um novo endpoint ou o ID de um endpoint existente que você está atualizando.

Em sua solicitação, inclua os cabeçalhos necessários e forneça o JSON EndpointRequest como o corpo:

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180428T004705Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180428/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
```

```
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

Se sua solicitação for bem-sucedida, você receberá uma resposta como a seguinte:

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

Informações relacionadas

Para obter mais informações sobre o recurso Endpoint na API do Amazon Pinpoint, incluindo os métodos HTTP e os parâmetros de solicitação compatíveis, consulte Endpoint na Referência da API do Amazon Pinpoint.

Para obter mais informações sobre como personalizar mensagens com variáveis, consulte Variáveis de mensagens no Guia do usuário do Amazon Pinpoint.

Para obter informações sobre as cotas que se aplicam a endpoints, como o número de atributos que você pode atribuir, consulte the section called “Cotas de endpoint”.

Associar usuários aos endpoints do Amazon Pinpoint

Um endpoint pode incluir atributos que definem um usuário, que representa uma pessoa em seu público-alvo. Por exemplo, um usuário pode representar alguém que instalou seu aplicativo móvel, ou alguém que tem uma conta em seu site.

Você define um usuário especificando um ID de usuário exclusivo e, opcionalmente, atributos de usuário personalizados. Se alguém usar seu aplicativo em vários dispositivos ou se essa pessoa puder receber mensagens em vários endereços, você poderá atribuir o mesmo ID de usuário a vários endpoints. Nesse caso, o Amazon Pinpoint sincroniza os atributos do usuário em todos os endpoints. Portanto, se você adicionar um atributo de usuário a um endpoint, o Amazon Pinpoint adicionará esse atributo a cada endpoint que inclui o mesmo ID de usuário.

Você pode adicionar atributos de usuário para rastrear dados que se aplicam a um indivíduo e não variam com base no dispositivo que a pessoa está usando. Por exemplo, você pode adicionar atributos ao nome, idade ou status da conta de uma pessoa.

> **Tip**
>
> Se seu aplicativo usar grupos de usuários do Amazon Cognito para lidar com a autenticação de usuários, o Amazon Cognito poderá adicionar IDs e atributos de usuários a seus endpoints automaticamente. Para o valor do ID do usuário do endpoint, o Amazon Cognito atribui o valor `sub` que é atribuído ao usuário no grupo de usuários. Para saber mais sobre como adicionar usuários ao Amazon Cognito, consulte Usar análise do Amazon Pinpoint com grupos de usuários do Amazon Cognito no Guia do desenvolvedor do Amazon Cognito..

Depois de adicionar definições de usuários a seus endpoints, você tem mais opções de como segmentar seu público-alvo. Você pode definir um segmento com base nos atributos dos usuários ou definir um segmento importando uma lista de IDs de usuários. Quando você envia uma mensagem a um segmento baseado em usuários, os destinos potenciais incluem cada endpoint que está associado a cada usuário no segmento.

Você também tem mais opções de como enviar mensagens a seu público-alvo. Você pode usar uma campanha para enviar mensagens para um segmento de usuários ou enviar uma mensagem diretamente para uma lista de IDs de usuários. Para personalizar sua mensagem, você pode incluir variáveis na mensagem que são substituídas pelos valores dos atributos dos usuários.

Exemplos

Os exemplos a seguir mostram como adicionar uma definição de usuário a um endpoint.

AWS CLI

Você pode usar o Amazon Pinpoint executando comandos com a AWS CLI.

Example Comando Update endpoint

Para adicionar um usuário a um endpoint, use o comando update-endpoint. Para o parâmetro `--endpoint-request`, você pode definir um novo endpoint, que pode incluir um usuário. Ou, para atualizar um endpoint existente, você pode fornecer apenas os atributos que deseja alterar. O exemplo a seguir adiciona um usuário a um endpoint existente fornecendo apenas os atributos do usuário:

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
> --endpoint-request file://endpoint-request-file.json
```

Em que:

- *application-id* é o ID do projeto do Amazon Pinpoint no qual você está adicionando ou atualizando um endpoint.
- *endpoint-id* é o ID que você está atribuindo a um novo endpoint ou o ID de um endpoint existente que você está atualizando.
- *endpoint-request-file.json* é o caminho para um arquivo JSON local que contém a entrada para o parâmetro `--endpoint-request`.

Example Arquivo de solicitação de endpoint

O comando `update-endpoint` de exemplo usa um arquivo JSON como o argumento para o parâmetro `--endpoint-request`. Esse arquivo contém uma definição de usuário, como a seguinte:

```
{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":["Wang"],
            "LastName":["Xiulan"],
            "Gender":["Female"],
            "Age":["39"]
        }
    }
}
```

Para obter os atributos que podem ser usados para definir um usuário, consulte o objeto `User` no esquema EndpointRequest na Referência da API do Amazon Pinpoint.

AWS SDK for Java

Você pode usar a API do Amazon Pinpoint em seus aplicativos Java usando o cliente fornecido pelo AWS SDK for Java.

Example Código

Para adicionar um usuário a um endpoint, inicialize um objeto EndpointRequest e passe-o para o método `updateEndpoint` do cliente do `AmazonPinpoint`. Você pode usar esse objeto para definir um novo endpoint, que pode incluir um usuário. Ou, para atualizar um endpoint existente, você pode atualizar apenas as propriedade que deseja alterar. O exemplo a seguir adiciona um usuário a um endpoint existente adicionando um objeto EndpointUser ao objeto EndpointRequest:

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
   public static void updatePinpointEndpoint(PinpointClient pinpoint, String
applicationId, String endPointId) {
       try {
           List<String> wangXiList = new ArrayList<>();
           wangXiList.add("cooking");
           wangXiList.add("running");
           wangXiList.add("swimming");

           Map myMapWang = new HashMap<>();
           myMapWang.put("interests", wangXiList);

           List<String> myNameWang = new ArrayList<>();
           myNameWang.add("Wang ");
           myNameWang.add("Xiulan");

           Map wangName = new HashMap<>();
           wangName.put("name", myNameWang);

           EndpointUser wangMajor = EndpointUser.builder()
                   .userId("example_user_10")
                   .userAttributes(wangName)
```

```
                .build();

        // Create an EndpointBatchItem object for Mary Major.
        EndpointRequest wangXiulanEndpoint = EndpointRequest.builder()
                .channelType(ChannelType.EMAIL)
                .address("wang_xiulan@example.com")
                .attributes(myMapWang)
                .user(wangMajor)
                .build();

        // Adds multiple endpoint definitions to a single request object.
        UpdateEndpointRequest endpointList = UpdateEndpointRequest.builder()
                .applicationId(applicationId)
                .endpointRequest(wangXiulanEndpoint)
                .endpointId(endPointId)
                .build();

        UpdateEndpointResponse result = pinpoint.updateEndpoint(endpointList);
        System.out.format("Update endpoint result: %s\n",
result.messageBody().message());

    } catch (PinpointException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
  }
```

Para ver o exemplo completo do SDK, consulte AddExampleEndpoints.java no GitHub.

HTTP

Você pode usar o Amazon Pinpoint fazendo solicitações HTTP diretamente para a API REST.

Example Solicitação Put endpoint com definição de usuário

Para adicionar um usuário a um endpoint, envie uma solicitação PUT para o recurso Endpoint no seguinte URI:

`/v1/apps/`*`application-id`*`/endpoints/`*`endpoint-id`*

Em que:

- *`application-id`* é o ID do projeto do Amazon Pinpoint no qual você está adicionando ou atualizando um endpoint.

- *endpoint-id* é o ID que você está atribuindo a um novo endpoint ou o ID de um endpoint existente que você está atualizando.

Em sua solicitação, inclua os cabeçalhos necessários e forneça o JSON EndpointRequest como o corpo. O corpo da solicitação pode definir um novo endpoint, que pode incluir um usuário. Ou, para atualizar um endpoint existente, você pode fornecer apenas os atributos que deseja alterar. O exemplo a seguir adiciona um usuário a um endpoint existente fornecendo apenas os atributos do usuário:

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":"Wang",
            "LastName":"Xiulan",
            "Gender":"Female",
            "Age":"39"
        }
    }
}
```

Se a solicitação for bem-sucedida, você receberá uma resposta como a seguinte:

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

Informações relacionadas

Para obter mais informações sobre o recurso Endpoint na API do Amazon Pinpoint, incluindo os métodos HTTP e os parâmetros de solicitação compatíveis, consulte Endpoint na Referência da API do Amazon Pinpoint.

Para obter mais informações sobre como personalizar mensagens com variáveis, consulte Variáveis de mensagens no Guia do usuário do Amazon Pinpoint.

Para saber como definir um segmento importando uma lista de IDs de usuários, consulte Importar segmentos no Guia do usuário do Amazon Pinpoint.

Para obter informações sobre como enviar uma mensagem direta para até 100 IDs de usuários, consulte Mensagens de usuários na Referência da API do Amazon Pinpoint.

Para obter informações sobre as cotas que se aplicam aos endpoints, incluindo o número de atributos de usuário que você pode atribuir, consulte the section called “Cotas de endpoint”.

Adição de um lote de endpoints ao Amazon Pinpoint

Você pode adicionar ou atualizar vários endpoints em uma única operação fornecendo os endpoints em lotes. Cada solicitação em lotes pode incluir até 100 definições de endpoints.

Para adicionar ou atualizar mais de 100 endpoints em uma única operação, consulte Importar endpoints para o Amazon Pinpoint.

Exemplos

Os exemplos a seguir mostram como adicionar dois endpoints de uma vez, incluindo os endpoints em uma solicitação em lotes.

AWS CLI

Você pode usar o Amazon Pinpoint executando comandos com a AWS CLI.

Example Comando Update endpoints batch

Para enviar uma solicitação de endpoints em lotes, use o comando `update-endpoints-batch`:

```
$ aws pinpoint update-endpoints-batch \
```

```
> --application-id application-id \
> --endpoint-batch-request file://endpoint_batch_request_file.json
```

Em que:

- *application-id* é o ID do projeto do Amazon Pinpoint no qual você está adicionando ou atualizando os endpoints.
- *endpoint_batch_request_file.json* é o caminho para um arquivo JSON local que contém a entrada para o parâmetro `--endpoint-batch-request`.

Example Arquivo de solicitação de lote de endpoints

O comando `update-endpoints-batch` de exemplo usa um arquivo JSON como o argumento para o parâmetro `--endpoint-request`. Esse arquivo contém um lote de definições de endpoints, como o seguinte:

```
{
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
```

```
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
        }
    ]
}
```

Para os atributos que você pode usar para definir um lote de endpoints, consulte o esquema EndpointBatchRequest na API do Amazon Pinpoint.

AWS SDK for Java

Você pode usar a API do Amazon Pinpoint em seus aplicativos Java usando o cliente fornecido pelo AWS SDK for Java.

Example Código

Para enviar uma solicitação de endpoints em lotes, inicialize um objeto `EndpointBatchRequest` e passe-o para o método `updateEndpointsBatch` do cliente do `AmazonPinpoint`. O exemplo a seguir preenche um objeto `EndpointBatchRequest` com dois objetos `EndpointBatchItem`:

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class AddExampleEndpoints {

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <appId>

                                Where:
```

```
                                appId - The ID of the application.

                            """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String applicationId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                updateEndpointsViaBatch(pinpoint, applicationId);
                pinpoint.close();
        }

        public static void updateEndpointsViaBatch(PinpointClient pinpoint, String
applicationId) {
                try {
                        List<String> myList = new ArrayList<>();
                        myList.add("music");
                        myList.add("books");

                        Map myMap = new HashMap<String, List>();
                        myMap.put("attributes", myList);

                        List<String> myNames = new ArrayList<String>();
                        myList.add("Richard");
                        myList.add("Roe");

                        Map myMap2 = new HashMap<String, List>();
                        myMap2.put("name", myNames);

                        EndpointUser richardRoe = EndpointUser.builder()
                                        .userId("example_user_1")
                                        .userAttributes(myMap2)
                                        .build();

                        // Create an EndpointBatchItem object for Richard Roe.
                        EndpointBatchItem richardRoesEmailEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.EMAIL)
```

```
                                        .address("richard_roe@example.com")
                                        .id("example_endpoint_1")
                                        .attributes(myMap)
                                        .user(richardRoe)
                                        .build();

                        List<String> myListMary = new ArrayList<String>();
                        myListMary.add("cooking");
                        myListMary.add("politics");
                        myListMary.add("finance");

                        Map myMapMary = new HashMap<String, List>();
                        myMapMary.put("interests", myListMary);

                        List<String> myNameMary = new ArrayList<String>();
                        myNameMary.add("Mary ");
                        myNameMary.add("Major");

                        Map maryName = new HashMap<String, List>();
                        myMapMary.put("name", myNameMary);

                        EndpointUser maryMajor = EndpointUser.builder()
                                        .userId("example_user_2")
                                        .userAttributes(maryName)
                                        .build();

                        // Create an EndpointBatchItem object for Mary Major.
                        EndpointBatchItem maryMajorsSmsEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.SMS)
                                        .address("+16145550100")
                                        .id("example_endpoint_2")
                                        .attributes(myMapMary)
                                        .user(maryMajor)
                                        .build();

                        // Adds multiple endpoint definitions to a single request
object.
                        EndpointBatchRequest endpointList =
EndpointBatchRequest.builder()
                                        .item(richardRoesEmailEndpoint)
                                        .item(maryMajorsSmsEndpoint)
                                        .build();
```

```
                        // Create the UpdateEndpointsBatchRequest.
                        UpdateEndpointsBatchRequest batchRequest =
 UpdateEndpointsBatchRequest.builder()
                                            .applicationId(applicationId)
                                            .endpointBatchRequest(endpointList)
                                            .build();

                        // Updates the endpoints with Amazon Pinpoint.
                        UpdateEndpointsBatchResponse result =
 pinpoint.updateEndpointsBatch(batchRequest);
                        System.out.format("Update endpoints batch result: %s\n",
 result.messageBody().message());

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Para ver o exemplo completo do SDK, consulte AddExampleEndpoints.java no GitHub.

HTTP

Você pode usar o Amazon Pinpoint fazendo solicitações HTTP diretamente para a API REST.

Example Solicitação Put endpoints

Para enviar uma solicitação de endpoints em lotes, envie uma solicitação PUT para o recurso Endpoints no seguinte URI:

`/v1/apps/`*`application-id`*`/endpoints`

Em que *`application-id`* é o ID do projeto do Amazon Pinpoint no qual você está adicionando ou atualizando os endpoints.

Em sua solicitação, inclua os cabeçalhos necessários e forneça o JSON EndpointBatchRequest como o corpo:

```
PUT /v1/apps/application_id/endpoints HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180501T184948Z
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
```

```
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
        }
    ]
}
```

Se sua solicitação for bem-sucedida, você receberá uma resposta como a seguinte:

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

Informações relacionadas

Para obter mais informações sobre o recurso Endpoint na API do Amazon Pinpoint, incluindo os métodos HTTP e os parâmetros de solicitação compatíveis, consulte Endpoint na .Referência da API do Amazon Pinpoint.

Importar endpoints para o Amazon Pinpoint

Você pode adicionar ou atualizar um grande número de endpoints importando-os de um bucket do Amazon S3. A importação de endpoints é útil se você tiver registros sobre seu público-alvo fora do Amazon Pinpoint e desejar adicionar essas informações para um projeto do Amazon Pinpoint. Nesse caso, você deve:

1. Criar definições de endpoints com base nos dados de seu próprio público-alvo.
2. Salvar essas definições de endpoints em um ou mais arquivos e fazer upload dos arquivos em um bucket do Amazon S3.
3. Adicionar os endpoints a seu projeto do Amazon Pinpoint importando-os do bucket.

Cada tarefa de importação pode transferir até 1 GB de dados. Em uma tarefa típica, onde cada endpoint tem 4 KB ou menos, você pode importar cerca de 250.000 endpoints. Você pode executar até duas tarefas de importação simultâneas por conta da AWS. Se mais largura de banda for necessária para seus trabalhos de importação, você poderá enviar uma solicitação de aumento de cota de serviço ao AWS Support. Para obter mais informações, consulte Solicitar um aumento de cota.

Antes de começar

Para poder importar endpoints, você precisa dos seguintes recursos em sua conta da AWS:

- Um bucket do Amazon S3. Para criar um bucket, veja Criação de um bucket no Guia do usuário do Amazon Simple Storage Service.
- Um perfil do AWS Identity and Access Management (IAM) que conceda permissões de leitura ao Amazon Pinpoint em seu bucket do S3. Para criar a função, consulte Perfil do IAM para importação de endpoints ou segmentos.

Exemplos

Os exemplos a seguir demonstram como adicionar definições de endpoints a seu bucket do Amazon S3 e, depois, importar esses endpoints em um projeto do Amazon Pinpoint.

Arquivos com definições de endpoints

Os arquivos que você adiciona ao bucket do Amazon S3 podem conter definições de endpoint em formato CSV ou JSON delimitado por novas linhas. Para os atributos que você pode usar para definir os endpoints, consulte o esquema JSON EndpointRequest na Referência da API do Amazon Pinpoint.

CSV

Você pode importar endpoints definidos em um arquivo CSV, como no exemplo a seguir:

```
ChannelType,Address,Location.Country,Demographic.Platform,Demographic.Make,User.UserId
SMS,12065550182,CN,Android,LG,example-user-id-1
APNS,1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f,US,iOS,Apple,example-user-id-2
EMAIL,john.stiles@example.com,US,iOS,Apple,example-user-id-2
```

A primeira linha é o cabeçalho, que contém os atributos de endpoint. Especifique atributos aninhados usando a notação de ponto, como em `Location.Country`.

As linhas seguintes definem os endpoints fornecendo valores para cada um dos atributos no cabeçalho.

Para incluir uma vírgula ou aspas em um valor, coloque o valor desejado entre aspas, como em `"aaa,bbb"`.

Não há suporte para quebras de linha em um valor no CSV.

JSON

Você pode importar endpoints definidos em um arquivo JSON delimitado por nova linha, como no exemplo a seguir:

```
{"ChannelType":"SMS","Address":"12065550182","Location":
{"Country":"CN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

Nesse formato, cada linha é um objeto JSON completo que contém uma definição de endpoint individual.

Solicitações de trabalhos de importação

Os exemplos a seguir mostram como adicionar definições de endpoint ao Amazon S3 fazendo upload de um arquivo local em um bucket. Em seguida, os exemplos importam as definições de endpoint em um projeto do Amazon Pinpoint.

AWS CLI

Você pode usar o Amazon Pinpoint executando comandos com a AWS CLI.

Example Comando CP do S3

Para fazer upload de um arquivo local em um bucket do Amazon S3, use o comando `cp` do Amazon S3:

```
$ aws s3 cp ./endpoints-file s3://bucket-name/prefix/
```

Em que:

- ./endpoints-file é o caminho do arquivo para um arquivo local que contém as definições de endpoints.
- bucket-name/prefix/ é o nome de seu bucket do Amazon S3 e, opcionalmente, um prefixo que ajuda a organizar os objetos em seu bucket hierarquicamente. Por exemplo, um prefixo útil pode ser `pinpoint/imports/endpoints/`.

Example Comando Create import job

Para importar definições de endpoint de um bucket do Amazon S3, use o comando `create-import-job`:

```
$ aws pinpoint create-import-job \
> --application-id application-id \
> --import-job-request \
> S3Url=s3://bucket-name/prefix/key,\
> RoleArn=iam-import-role-arn,\
> Format=format,\
> RegisterEndpoints=true
```

Em que:

- application-id é o ID do projeto do Amazon Pinpoint para o qual você está importando endpoints.
- bucket-name/prefix/key é o local no Amazon S3 que contém um ou mais objetos para importação. A localização pode terminar com a chave de um objeto individual ou terminar com um prefixo que qualifica vários objetos.
- iam-import-role-arn é o nome do recurso da Amazon (ARN) de um perfil do IAM que concede acesso de leitura do Amazon Pinpoint ao bucket.

- O formato poder ser `JSON` ou `CSV`, dependendo do formato que você usou para definir os endpoints. Se o local do Amazon S3 incluir vários objetos de formatos mistos, o Amazon Pinpoint importará apenas os objetos que corresponderem ao formato especificado.
- RegisterEndpoints pode ser `true` ou `false`. Quando definido como verdadeiro, o trabalho de importação registra os endpoints no Amazon Pinpoint, quando as definições do endpoint são importadas.

  Combinações RegisterEndpoints e DefineSegments

RegisterEndpoints	DefineSegments	Descrição
verdadeiro	verdadeiro	O Amazon Pinpoint importará os endpoints e criará um segmento que contém os endpoints.
verdadeiro	false	O Amazon Pinpoint importará os endpoints e não criará um segmento.
false	verdadeiro	O Amazon Pinpoint importará os endpoints e criará um segmento que contém os endpoints . Os endpoints não serão salvos e não substituirão os endpoints existentes.
false	false	O Amazon Pinpoint rejeitará essa solicitação.

A resposta inclui detalhes sobre o trabalho de importação:

```
{
    "ImportJobResponse": {
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
```

```
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://bucket-name/prefix/key"
        },
        "Id": "d5ecad8e417d498389e1d5b9454d4e0c",
        "JobStatus": "CREATED",
        "Type": "IMPORT"
    }
}
```

A resposta fornece o ID do trabalho com o atributo `Id`. Você pode usar esse ID para verificar o status atual do trabalho de importação.

Example Comando Get import job

Para verificar o status atual do trabalho de importação, use o comando `get-import-job`:

```
$ aws pinpoint get-import-job \
> --application-id application-id \
> --job-id job-id
```

Em que:

- application-id é o ID do projeto do Amazon Pinpoint para o qual o trabalho de importação foi iniciado.
- job-id é o ID do trabalho de importação que você está verificando.

A resposta a esse comando fornece o estado atual do trabalho de importação:

```
{
    "ImportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-24T21:26:45.308Z",
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
```

```
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://s3-bucket-name/prefix/endpoint-definitions.json"
        },
        "FailedPieces": 0,
        "Id": "job-id",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 3,
        "Type": "IMPORT"
    }
}
```

A resposta fornece o status do trabalho com o atributo `JobStatus`.

AWS SDK for Java

Você pode usar a API do Amazon Pinpoint em seus aplicativos Java usando o cliente fornecido pelo AWS SDK for Java.

Example Código

Para fazer upload de um arquivo com definições de endpoint para o Amazon S3, use o método `putObject` do cliente `AmazonS3`.

Para importar os endpoints em um projeto do Amazon Pinpoint, inicialize um objeto `CreateImportJobRequest`. Em seguida, passe esse objeto para o método `createImportJob` do cliente do `AmazonPinpoint`.

```
package com.amazonaws.examples.pinpoint;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.CreateImportJobRequest;
import com.amazonaws.services.pinpoint.model.CreateImportJobResult;
import com.amazonaws.services.pinpoint.model.Format;
import com.amazonaws.services.pinpoint.model.GetImportJobRequest;
import com.amazonaws.services.pinpoint.model.GetImportJobResult;
import com.amazonaws.services.pinpoint.model.ImportJobRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
```

```
import com.amazonaws.services.s3.model.AmazonS3Exception;
import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.List;
import java.util.concurrent.TimeUnit;

public class ImportEndpoints {

    public static void main(String[] args) {

        final String USAGE = "\n" +
                "ImportEndpoints - Adds endpoints to an Amazon Pinpoint application
 by: \n" +
                "1.) Uploading the endpoint definitions to an Amazon S3 bucket. \n"
 +
                "2.) Importing the endpoint definitions from the bucket to an Amazon
 Pinpoint " +
                "application.\n\n" +
                "Usage: ImportEndpoints <endpointsFileLocation> <s3BucketName>
 <iamImportRoleArn> " +
                "<applicationId>\n\n" +
                "Where:\n" +
                "  endpointsFileLocation - The relative location of the JSON file
 that contains the " +
                "endpoint definitions.\n" +
                "  s3BucketName - The name of the Amazon S3 bucket to upload the
 JSON file to. If the " +
                "bucket doesn't exist, a new bucket is created.\n" +
                "  iamImportRoleArn - The ARN of an IAM role that grants Amazon
 Pinpoint read " +
                "permissions to the S3 bucket.\n" +
                "  applicationId - The ID of the Amazon Pinpoint application to add
 the endpoints to.";

        if (args.length < 1) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String endpointsFileLocation = args[0];
        String s3BucketName = args[1];
        String iamImportRoleArn = args[2];
        String applicationId = args[3];
```

```
        Path endpointsFilePath = Paths.get(endpointsFileLocation);
        File endpointsFile = new
File(endpointsFilePath.toAbsolutePath().toString());
        uploadToS3(endpointsFile, s3BucketName);

        importToPinpoint(endpointsFile.getName(), s3BucketName, iamImportRoleArn,
applicationId);

    }

    private static void uploadToS3(File endpointsFile, String s3BucketName) {

        // Initializes Amazon S3 client.
        final AmazonS3 s3 = AmazonS3ClientBuilder.defaultClient();

        // Checks whether the specified bucket exists. If not, attempts to create
one.
        if (!s3.doesBucketExistV2(s3BucketName)) {
            try {
                s3.createBucket(s3BucketName);
                System.out.format("Created S3 bucket %s.\n", s3BucketName);
            } catch (AmazonS3Exception e) {
                System.err.println(e.getErrorMessage());
                System.exit(1);
            }
        }

        // Uploads the endpoints file to the bucket.
        String endpointsFileName = endpointsFile.getName();
        System.out.format("Uploading %s to S3 bucket %s . . .\n", endpointsFileName,
s3BucketName);
        try {
            s3.putObject(s3BucketName, "imports/" + endpointsFileName,
endpointsFile);
            System.out.println("Finished uploading to S3.");
        } catch (AmazonServiceException e) {
            System.err.println(e.getErrorMessage());
            System.exit(1);
        }
    }

    private static void importToPinpoint(String endpointsFileName, String
s3BucketName,
```

```
            String iamImportRoleArn, String applicationId) {

        // The S3 URL that Amazon Pinpoint requires to find the endpoints file.
        String s3Url = "s3://" + s3BucketName + "/imports/" + endpointsFileName;

        // Defines the import job that Amazon Pinpoint runs.
        ImportJobRequest importJobRequest = new ImportJobRequest()
                .withS3Url(s3Url)
                .withRegisterEndpoints(true)
                .withRoleArn(iamImportRoleArn)
                .withFormat(Format.JSON);
        CreateImportJobRequest createImportJobRequest = new CreateImportJobRequest()
                .withApplicationId(applicationId)
                .withImportJobRequest(importJobRequest);

        // Initializes the Amazon Pinpoint client.
        AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
                .withRegion(Regions.US_EAST_1).build();

        System.out.format("Importing endpoints in %s to Amazon Pinpoint application
%s . . .\n",
                endpointsFileName, applicationId);

        try {

            // Runs the import job with Amazon Pinpoint.
            CreateImportJobResult importResult =
pinpointClient.createImportJob(createImportJobRequest);

            String jobId = importResult.getImportJobResponse().getId();
            GetImportJobResult getImportJobResult = null;
            String jobStatus = null;

            // Checks the job status until the job completes or fails.
            do {
                getImportJobResult = pinpointClient.getImportJob(new
GetImportJobRequest()
                        .withJobId(jobId)
                        .withApplicationId(applicationId));
                jobStatus =
getImportJobResult.getImportJobResponse().getJobStatus();
                System.out.format("Import job %s . . .\n", jobStatus.toLowerCase());
                TimeUnit.SECONDS.sleep(3);
            } while (!jobStatus.equals("COMPLETED") && !jobStatus.equals("FAILED"));
```

```
            if (jobStatus.equals("COMPLETED")) {
                System.out.println("Finished importing endpoints.");
            } else {
                System.err.println("Failed to import endpoints.");
                System.exit(1);
            }

            // Checks for entries that failed to import.
            // getFailures provides up to 100 of the first failed entries for the
 job, if
            // any exist.
            List<String> failedEndpoints =
 getImportJobResult.getImportJobResponse().getFailures();
            if (failedEndpoints != null) {
                System.out.println("Failed to import the following entries:");
                for (String failedEndpoint : failedEndpoints) {
                    System.out.println(failedEndpoint);
                }
            }

        } catch (AmazonServiceException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }

    }

}
```

HTTP

Você pode usar o Amazon Pinpoint fazendo solicitações HTTP diretamente para a API REST.

Example Solicitação de objeto PUT do S3

Para adicionar suas definições de endpoint a um bucket, use a operação PUT object do Amazon S3 e forneça as definições de endpoint como o corpo:

```
PUT /prefix/key HTTP/1.1
Content-Type: text/plain
Accept: application/json
```

```
Host: bucket-name.s3.amazonaws.com
X-Amz-Content-Sha256:
 c430dc094b0cec2905bc88d96314914d058534b14e2bc6107faa9daa12fdff2d
X-Amz-Date: 20180605T184132Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/s3/aws4_request, SignedHeaders=accept;cache-control;content-
length;content-type;host;postman-token;x-amz-content-sha256;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{"ChannelType":"SMS","Address":"2065550182","Location":
{"Country":"CAN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

Em que:

- /prefix/key é o prefixo e o nome da chave do objeto que conterá as definições de endpoints após o upload. Você pode usar o prefixo para organizar seus objetos hierarquicamente. Por exemplo, um prefixo útil pode ser `pinpoint/imports/endpoints/`.
- bucket-name é o nome do bucket do Amazon S3 ao qual você está adicionando as definições de endpoint.

Example Solicitação POST de trabalho de importação

Para importar definições de endpoint de um bucket do Amazon S3, envie uma solicitação POST para o recurso Tarefas de importação. Em sua solicitação, inclua os cabeçalhos necessários e forneça o JSON ImportJobRequest como o corpo:

```
POST /v1/apps/application_id/jobs/import HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T214912Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
```

```
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3Url": "s3://bucket-name/prefix/key",
  "RoleArn": "iam-import-role-arn",
  "Format": "format",
  "RegisterEndpoints": true
}
```

Em que:

- application-id é o ID do projeto do Amazon Pinpoint para o qual você está importando endpoints.
- bucket-name/prefix/key é o local no Amazon S3 que contém um ou mais objetos para importação. A localização pode terminar com a chave de um objeto individual ou terminar com um prefixo que qualifica vários objetos.
- iam-import-role-arn é o nome do recurso da Amazon (ARN) de um perfil do IAM que concede acesso de leitura do Amazon Pinpoint ao bucket.
- O formato poder ser `JSON` ou `CSV`, dependendo do formato que você usou para definir os endpoints. Se o local do Amazon S3 incluir vários arquivos de formatos mistos, o Amazon Pinpoint importará apenas os arquivos que corresponderem ao formato especificado.

Se sua solicitação for bem-sucedida, você receberá uma resposta como a seguinte:

```
{
    "Id": "a995ce5d70fa44adb563b7d0e3f6c6f5",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-05T21:49:15.288Z",
    "Type": "IMPORT",
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key",
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

A resposta fornece o ID do trabalho com o atributo `Id`. Você pode usar esse ID para verificar o status atual do trabalho de importação.

Example Solicitação GET de trabalho de importação

Para verificar o status atual de uma tarefa de importação, envie uma solicitação `GET` para o recurso Tarefas de importação:

```
GET /v1/apps/application_id/jobs/import/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T220744Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

Em que:

- application_id é o ID do projeto do Amazon Pinpoint para o qual o trabalho de importação foi iniciado.
- job_id é o ID do trabalho de importação que você está verificando.

Se sua solicitação for bem-sucedida, você receberá uma resposta como a seguinte:

```
{
    "ApplicationId": "application_id",
    "Id": "70a51b2cf442447492d2c8e50336a9e8",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-05T22:04:49.213Z",
    "CompletionDate": "2018-06-05T22:04:58.034Z",
    "Type": "IMPORT",
    "TotalFailures": 0,
    "TotalProcessed": 3,
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key.json",
```

```
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

A resposta fornece o status do trabalho com o atributo `JobStatus`.

Informações relacionadas

Para obter mais informações sobre o recurso de trabalhos de exportação na API do Amazon Pinpoint, incluindo os métodos HTTP e os parâmetros de solicitação compatíveis, consulte Trabalhos de importação na Referência da API do Amazon Pinpoint.

Excluir endpoints do Amazon Pinpoint

Você pode excluir endpoints quando não desejar mais enviar mensagens a um determinado destino, como quando o destino se tornar inacessível ou quando um cliente fechar uma conta.

Exemplos

Os exemplos a seguir mostram como excluir um endpoint.

AWS CLI

Você pode usar o Amazon Pinpoint executando comandos com a AWS CLI.

Example Comando Delete endpoint

Para excluir um endpoint, use o comando `delete-endpoint`:

```
$ aws pinpoint delete-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id
```

Em que:

- application-id é o ID do projeto do Amazon Pinpoint que contém o endpoint.

- endpoint-id é o ID do endpoint que você está excluindo.

A resposta a esse comando é a definição JSON do endpoint excluído.

AWS SDK for Java

Você pode usar a API do Amazon Pinpoint em seus aplicativos Java usando o cliente fornecido pelo AWS SDK for Java.

Example Código

Para excluir um endpoint, use o método `deleteEndpoint` do cliente do `AmazonPinpoint`. Forneça um objeto `DeleteEndpointRequest` como o argumento do método:

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appName> <endpointId >

                Where:
                  appId - The id of the application to delete.
```

```
                  endpointId - The id of the endpoint to delete.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
 String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

Para ver o exemplo completo do SDK, consulte DeleteEndpoint.java no GitHub.

HTTP

Você pode usar o Amazon Pinpoint fazendo solicitações HTTP diretamente para a API REST.

Example Solicitação DELETE endpoint

Para excluir um endpoint, envie uma solicitação DELETE para o recurso Endpoint:

```
DELETE /v1/apps/application-id/endpoints/endpoint-id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

Em que:

- application-id é o ID do projeto do Amazon Pinpoint que contém o endpoint.
- endpoint-id é o ID do endpoint que você está excluindo.

A resposta a essa solicitação é a definição JSON do endpoint excluído.

Gerenciar o número máximo de endpoints de um público-alvo

Cada membro do seu público-alvo pode ter no máximo 15 endpoints associados aos seus respectivos UserId. Consulte Cotas de endpoint. Se você tentar adicionar um 16º endpoint, dependendo do ChannelType, você obterá BadRequestException ou será bem-sucedido removendo o endpoint com a EffectiveDate mais antiga.

Adicionar um 16º endpoint

- Se o novo tipo de canal para o endpoint for SMS, PUSH, VOICE, EMAIL, CUSTOM ou IN_APP, o retorno será BadRequestException, pois o membro do público-alvo está no número máximo de endpoints. Você precisa remover um endpoint associado ao membro do público-alvo e tentar novamente. ConsulteExcluir endpoints do Amazon Pinpoint.
- Se o novo tipo de canal para o endpoint for ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX ou BAIDU:
  - Verifique se pelo menos um endpoint atualmente associado ao membro do público-alvo tem um ChannelType de ADM, GCM, APNS, APNS_VOICE, APNS_VOIP_SANDBOX ou BAIDU. Se não houver, o retorno será BadRequestException e um endpoint precisará ser removido antes de você tentar novamente. Consulte Excluir endpoints do Amazon Pinpoint.

- Caso contrário, o endpoint com a EffectiveDate mais antiga é definido para `INACTIVE` onde o ChannelType seja ADM, GCM, APNS, APNS_VOIP, APNS_VOIP_SANDBOX ou BAIDU.
  - O UserID do endpoint antigo é removido.
  - O novo endpoint está associado ao membro do público-alvo e ainda tem o número máximo de endpoints.

O endpoint pode ser reativado definindo o Status como `ACTIVE` e adicionando o UserID de volta ao endpoint.

Acessar os dados do público-alvo no Amazon Pinpoint

Conforme você adiciona endpoints ao Amazon Pinpoint, ele cresce como um repositório de dados do público-alvo. Esses dados consistem em:

- Os endpoints que você adiciona ou atualiza usando a API do Amazon Pinpoint.
- Endpoints que seu código cliente adiciona ou atualiza conforme os usuários entram em seu aplicativo.

Conforme seu público-alvo aumenta e muda, os dados de seu endpoint também aumentam e mudam. Para visualizar as informações mais recentes contidas no Amazon Pinpoint sobre seu público-alvo, você pode pesquisar endpoints individualmente ou exportar todos os endpoints para um projeto do Amazon Pinpoint. Visualizando os dados de seus endpoints, você pode aprender sobre as características do público-alvo que você registra em seus endpoints, como:

- Os dispositivos e plataformas de seus usuários.
- Os fusos horários dos usuários.
- As versões de seu aplicativo que estão instaladas nos dispositivos dos usuários.
- As localizações dos usuários, como suas cidades ou países.
- Todos os atributos ou métricas personalizados que você registrar.

O console do Amazon Pinpoint também fornece análise dos dados demográficos e atributos personalizados que são capturados em seus endpoints.

Antes de pesquisar endpoints, você deve adicioná-los a seu projeto do Amazon Pinpoint. Para adicionar endpoints, consulte Definir o público-alvo para o Amazon Pinpoint.

Use os tópicos desta seção para pesquisar ou exportar endpoints usando a API do Amazon Pinpoint.

Tópicos

- Pesquisar endpoints com o Amazon Pinpoint
- Exportar endpoints do Amazon Pinpoint
- Listar IDs de endpoint com o Amazon Pinpoint

Pesquisar endpoints com o Amazon Pinpoint

Você pode pesquisar os detalhes de qualquer endpoint individual adicionado a um projeto do Amazon Pinpoint. Esses detalhes podem incluir o endereço de destino para suas mensagens, o canal de mensagens, os dados sobre o dispositivo do usuário, dados sobre a localização do usuário e todos os atributos personalizados que você registrar em seus endpoints.

Para pesquisar um endpoint, você precisa do ID do endpoint. Se você não souber o ID, você poderá obter os dados do endpoint por meio da exportação. Para exportar endpoints, consulte the section called “Exportar endpoints”.

Exemplos

Os exemplos a seguir mostram como pesquisar um endpoint individual especificando seu ID.

AWS CLI

Você pode usar o Amazon Pinpoint executando comandos com a AWS CLI.

Example Comando Get endpoint

Para pesquisar um endpoint, use o comando `get-endpoint`:

```
$ aws pinpoint get-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id
```

Em que:

- *`application-id`* é o ID do projeto do Amazon Pinpoint que contém o endpoint.
- *`endpoint-id`* é o ID do endpoint que você está pesquisando.

A resposta a esse comando é a definição JSON do endpoint, como no exemplo a seguir:

```
{
    "EndpointResponse": {
        "Address":
 "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
        "ApplicationId": "application-id",
        "Attributes": {
            "Interests": [
```

```
                "Technology",
                "Music",
                "Travel"
            ]
        },
        "ChannelType": "APNS",
        "CohortId": "63",
        "CreationDate": "2018-05-01T17:31:01.046Z",
        "Demographic": {
            "AppVersion": "1.0",
            "Make": "apple",
            "Model": "iPhone",
            "ModelVersion": "8",
            "Platform": "ios",
            "PlatformVersion": "11.3.1",
            "Timezone": "America/Los_Angeles"
        },
        "EffectiveDate": "2018-05-07T19:03:29.963Z",
        "EndpointStatus": "ACTIVE",
        "Id": "example_endpoint",
        "Location": {
            "City": "Seattle",
            "Country": "US",
            "Latitude": 47.6,
            "Longitude": -122.3,
            "PostalCode": "98121"
        },
        "Metrics": {
            "music_interest_level": 6.0,
            "travel_interest_level": 4.0,
            "technology_interest_level": 9.0
        },
        "OptOut": "ALL",
        "RequestId": "7f546cac-6858-11e8-adcd-2b5a07aab338",
        "User": {
            "UserAttributes": {
                "Gender": "Female",
                "FirstName": "Wang",
                "LastName": "Xiulan",
                "Age": "39"
            },
            "UserId": "example_user"
        }
    }
```

```
}
```

AWS SDK for Java

Você pode usar a API do Amazon Pinpoint em seus aplicativos Java usando o cliente fornecido pelo AWS SDK for Java.

Example Código

Para pesquisar um endpoint, inicialize um objeto `GetEndpointRequest`. Em seguida, passe esse objeto para o método `getEndpoint` do cliente do `AmazonPinpoint`:

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
```

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:   <appId> <endpoint>

                Where:
                  appId - The ID of the application to delete.
                  endpoint - The ID of the endpoint.\s
                  """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String appId,
String endpoint) {
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
            System.out.println(endpointJson);
```

```
        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

Para imprimir os dados do endpoint em um formato legível, este exemplo usa a biblioteca GSON do Google para converter o objeto `EndpointResponse` em uma string JSON.

HTTP

Você pode usar o Amazon Pinpoint fazendo solicitações HTTP diretamente para a API REST.

Example Solicitação GET endpoint

Para pesquisar um endpoint, envie uma solicitação `GET` para o recurso Endpoint:

```
GET /v1/apps/application_id/endpoints/endpoint_id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

Em que:

- *`application-id`* é o ID do projeto do Amazon Pinpoint que contém o endpoint.
- *`endpoint-id`* é o ID do endpoint que você está pesquisando.

A resposta a essa solicitação é a definição JSON do endpoint, como no exemplo a seguir:

```
{
    "ChannelType": "APNS",
    "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "RequestId": "b720cfa8-6924-11e8-aeda-0b22e0b0fa59",
    "Location": {
        "Latitude": 47.6,
        "Longitude": -122.3,
        "PostalCode": "98121",
```

```
        "City": "Seattle",
        "Country": "US"
    },
    "Demographic": {
        "Make": "apple",
        "Model": "iPhone",
        "ModelVersion": "8",
        "Timezone": "America/Los_Angeles",
        "AppVersion": "1.0",
        "Platform": "ios",
        "PlatformVersion": "11.3.1"
    },
    "EffectiveDate": "2018-06-06T00:58:19.865Z",
    "Attributes": {
        "Interests": [
            "Technology",
            "Music",
            "Travel"
        ]
    },
    "Metrics": {
        "music_interest_level": 6,
        "travel_interest_level": 4,
        "technology_interest_level": 9
    },
    "User": {},
    "ApplicationId": "application_id",
    "Id": "example_endpoint",
    "CohortId": "39",
    "CreationDate": "2018-06-06T00:58:19.865Z"
}
```

Informações relacionadas

Para obter mais informações sobre o recurso Endpoint na API do Amazon Pinpoint, consulte Endpoint na Referência da API do Amazon Pinpoint.

Exportar endpoints do Amazon Pinpoint

Para obter todas as informações de seu público-alvo contidas no Amazon Pinpoint, você pode exportar as definições do endpoint que pertencem a um projeto. Quando você exporta, o Amazon

Pinpoint coloca as definições do endpoint em um bucket do Amazon S3 especificado por você. A exportação de endpoints é útil quando você deseja:

- Visualizar os dados mais recentes sobre endpoints novos e existentes que seu aplicativo cliente registrou no Amazon Pinpoint.
- Sincronizar os dados do endpoint no Amazon Pinpoint com seu próprio sistema Customer Relationship Management (CRM - Gerenciamento de relação com o cliente).
- Criar relatórios sobre os dados de seus clientes ou analisá-los.

Antes de começar

Para poder exportar endpoints, você precisa dos seguintes recursos em sua conta da AWS:

- Um bucket do Amazon S3. Para criar um bucket, consulte Criar um bucket no Guia do usuário do Amazon Simple Storage Service.
- Um perfil do AWS Identity and Access Management (IAM) que conceda permissões de gravação ao Amazon Pinpoint em seu bucket do Amazon S3. Para criar a função, consulte Perfil do IAM para exportação de endpoints ou de segmentos.

Exemplos

Os exemplos a seguir demonstram como exportar endpoints de um projeto do Amazon Pinpoint e fazer download desses endpoints de seu bucket do Amazon S3.

AWS CLI

Você pode usar o Amazon Pinpoint executando comandos com a AWS CLI.

Example Comando Create de trabalho de exportação

Para exportar os endpoints em seu projeto do Amazon Pinpoint, use o comando `create-export-job`:

```
$ aws pinpoint create-export-job \
> --application-id application-id \
> --export-job-request \
> S3UrlPrefix=s3://bucket-name/prefix/,\
> RoleArn=iam-export-role-arn
```

Em que:

- *application-id* é o ID do projeto do Amazon Pinpoint que contém os endpoints.
- *bucket-name/prefix/* é o nome do bucket do Amazon S3 e, opcionalmente, um prefixo que ajuda a organizar hierarquicamente os objetos no bucket. Por exemplo, um prefixo útil pode ser `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* é o nome do recurso da Amazon (ARN) de um perfil do IAM que concede ao Amazon Pinpoint acesso de gravação ao bucket.

A resposta a esse comando fornece detalhes sobre o trabalho de exportação:

```
{
    "ExportJobResponse": {
        "CreationDate": "2018-06-04T22:04:20.585Z",
        "Definition": {
            "RoleArn": "iam-export-role-arn",
            "S3UrlPrefix": "s3://s3-bucket-name/prefix/"
        },
        "Id": "7390e0de8e0b462380603c5a4df90bc4",
        "JobStatus": "CREATED",
        "Type": "EXPORT"
    }
}
```

A resposta fornece o ID do trabalho com o atributo `Id`. Você pode usar esse ID para verificar o status atual do trabalho de exportação.

Example Comando Get de trabalho de exportação

Para verificar o status atual do trabalho de exportação, use o comando `get-export-job`:

```
$ aws pinpoint get-export-job \
> --application-id application-id \
> --job-id job-id
```

Em que:

- *application-id* é o ID do projeto do Amazon Pinpoint do qual você exportou os endpoints.
- *job-id* é o ID do trabalho que você está verificando.

A resposta a esse comando fornece o estado atual do trabalho de exportação:

```
{
    "ExportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-08T22:16:48.228Z",
        "CreationDate": "2018-05-08T22:16:44.812Z",
        "Definition": {},
        "FailedPieces": 0,
        "Id": "6c99c463f14f49caa87fa27a5798bef9",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 215,
        "Type": "EXPORT"
    }
}
```

A resposta fornece o status do trabalho com o atributo `JobStatus`. Quando o valor do status do trabalho for `COMPLETED`, você obterá os endpoints exportados em seu bucket do Amazon S3.

Example Comando CP do S3

Para fazer download dos endpoints exportados, use o comando `cp` do Amazon S3:

```
$ aws s3 cp s3://bucket-name/prefix/key.gz /local/directory/
```

Em que:

- *`bucket-name/prefix/key`* é o local do arquivo .gz que o Amazon Pinpoint adicionou ao bucket quando você exportou os endpoints. Esse arquivo contém as definições dos endpoints exportados. Por exemplo, no URL `https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv`, `PINPOINT-EXAMPLE-BUCKET` é o nome do bucket e `Exports/example.csv` é a chave. Para obter mais informações sobre Chaves, consulte Chaves no Guia do usuário do Amazon S3.
- */local/directory/* é o caminho do arquivo para o diretório local no qual você deseja fazer download dos endpoints.

AWS SDK for Java

Você pode usar a API do Amazon Pinpoint em seus aplicativos Java usando o cliente fornecido pelo AWS SDK for Java.

Example Código

Para exportar endpoints de um projeto do Amazon Pinpoint, inicialize um objeto `CreateExportJobRequest`. Em seguida, passe esse objeto para o método `createExportJob` do cliente do `AmazonPinpoint`.

Para fazer download dos endpoints exportados do Amazon Pinpoint, use o método `getObject` do cliente `AmazonS3`.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;
```

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:
```

```
                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and prints
them.
                Usage: ExportEndpoints <applicationId> <s3BucketName>
<iamExportRoleArn> <path>

                Where:
                  applicationId - The ID of the Amazon Pinpoint application that has
the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the JSON
file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint write permissions to the S3 bucket.  path - The path where the files
downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
                """;

       if (args.length != 4) {
           System.out.println(usage);
           System.exit(1);
       }

       String applicationId = args[0];
       String s3BucketName = args[1];
       String iamExportRoleArn = args[2];
       String path = args[3];
       System.out.println("Deleting an application with ID: " + applicationId);

       Region region = Region.US_EAST_1;
       PinpointClient pinpoint = PinpointClient.builder()
               .region(region)
               .build();

       S3Client s3Client = S3Client.builder()
               .region(region)
               .build();

       exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
       pinpoint.close();
       s3Client.close();
   }
```

```
    public static void exportAllEndpoints(PinpointClient pinpoint,
            S3Client s3Client,
            String applicationId,
            String s3BucketName,
            String path,
            String iamExportRoleArn) {

        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
 s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
 o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint, S3Client
 s3Client, String s3BucketName,
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
```

```
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
                    .build();

            System.out.format("Exporting endpoints from Amazon Pinpoint application
%s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);

            CreateExportJobResponse exportResult =
pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
```

```
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                    .jobId(jobId)
                    .applicationId(applicationId)
                    .build();

            do {
                getExportJobResult = pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path location.
    public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();
```

```
                // Write the data to a local file.
                String fileSuffix = new
 SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

Para ver o exemplo completo do SDK, consulte ExportEndpoints.java no GitHub.

HTTP

Você pode usar o Amazon Pinpoint fazendo solicitações HTTP diretamente para a API REST.

Example Solicitação POST de trabalho de exportação

Para exportar os endpoints em seu projeto do Amazon Pinpoint, envie uma solicitação POST para o recurso Trabalhos de exportação:

```
POST /v1/apps/application_id/jobs/export HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T001238Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3UrlPrefix": "s3://bucket-name/prefix",
  "RoleArn": "iam-export-role-arn"
}
```

Em que:

- *application-id* é o ID do projeto do Amazon Pinpoint que contém os endpoints.
- *bucket-name/prefix* é o nome do bucket do Amazon S3 e, opcionalmente, um prefixo que ajuda a organizar hierarquicamente os objetos no bucket. Por exemplo, um prefixo útil pode ser `pinpoint/exports/endpoints/`.
- *iam-export-role-arn* é o nome do recurso da Amazon (ARN) de um perfil do IAM que concede ao Amazon Pinpoint acesso de gravação ao bucket.

A resposta a essa solicitação fornece detalhes sobre o trabalho de exportação:

```
{
    "Id": "611bdc54c75244bfa51fe7001ddb2e36",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "Type": "EXPORT",
    "Definition": {
        "S3UrlPrefix": "s3://bucket-name/prefix",
        "RoleArn": "iam-export-role-arn"
    }
}
```

A resposta fornece o ID do trabalho com o atributo `Id`. Você pode usar esse ID para verificar o status atual do trabalho de exportação.

Example Solicitação GET de trabalho de exportação

Para verificar o status atual de um trabalho de exportação, envie uma solicitação `GET` para o recurso de Trabalho de exportação:

```
GET /v1/apps/application_id/jobs/export/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T002443Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

Em que:

- *application-id* é o ID do projeto do Amazon Pinpoint do qual você exportou os endpoints.
- *job-id* é o ID do trabalho que você está verificando.

A resposta a essa solicitação fornece o estado atual do trabalho de exportação:

```
{
    "ApplicationId": "application_id",
    "Id": "job_id",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "CompletionDate": "2018-06-06T00:13:01.141Z",
    "Type": "EXPORT",
    "TotalFailures": 0,
    "TotalProcessed": 217,
    "Definition": {}
}
```

A resposta fornece o status do trabalho com o atributo `JobStatus`. Quando o valor do status do trabalho for `COMPLETED`, você obterá os endpoints exportados em seu bucket do Amazon S3.

Informações relacionadas

Para obter mais informações sobre o recurso de trabalhos de exportação na API do Amazon Pinpoint, incluindo os métodos HTTP e os parâmetros de solicitação compatíveis consulte Trabalhos de exportação na Referência da API do Amazon Pinpoint.

Listar IDs de endpoint com o Amazon Pinpoint

Para atualizar ou excluir um endpoint, você precisa do ID do endpoint. Portanto, para executar essas operações em todos os endpoints em um projeto do Amazon Pinpoint, a primeira etapa é listar todos os IDs de endpoints que pertencem a esse projeto. Em seguida, você pode repetir esses IDs para, por exemplo, adicionar um atributo globalmente ou excluir todos os endpoints de seu projeto.

O exemplo a seguir usa o AWS SDK for Java e faz o seguinte:

1. Chama o método `exportEndpointsToS3` de exemplo do código de exemplo em Exportar endpoints do Amazon Pinpoint. Esse método exporta as definições de endpoints de um projeto do Amazon Pinpoint. As definições de endpoints são adicionadas como arquivos gzip a um bucket do Amazon S3.
2. Faz download dos arquivos gzip exportados.
3. Lê os arquivos gzip e obtém o ID do endpoint de cada definição JSON do endpoint.
4. Imprime os IDs de endpoints no console.
5. Limpa por meio da exclusão dos arquivos que o Amazon Pinpoint adicionou ao Amazon S3.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <applicationId> <userId>

                Where:
```

```
                    applicationId - The ID of the Amazon Pinpoint application that has
the endpoint.
                    userId - The user id applicable to the endpoints""";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String userId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllEndpoints(pinpoint, applicationId, userId);
        pinpoint.close();
    }

    public static void listAllEndpoints(PinpointClient pinpoint,
            String applicationId,
            String userId) {

        try {
            GetUserEndpointsRequest endpointsRequest =
GetUserEndpointsRequest.builder()
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

            GetUserEndpointsResponse response =
pinpoint.getUserEndpoints(endpointsRequest);
            List<EndpointResponse> endpoints = response.endpointsResponse().item();

            // Display the results.
            for (EndpointResponse endpoint : endpoints) {
                System.out.println("The channel type is: " + endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
```

```
    }
}
```

Para ver o exemplo completo do SDK, consulte ListEndpointIs.java no GitHub.

Criar segmentos

Um segmento de usuário representa um subconjunto de seus usuários com base em características compartilhadas, tais como os usuários que acessaram o seu aplicativo ou que plataforma de dispositivo usam. Um segmento designa que usuários receberão as mensagens entregues por uma campanha. Defina segmentos para que você possa alcançar o público correto quando desejar convidar usuários de volta para seu aplicativo, fazer ofertas especiais, ou aumentar o envolvimento dos usuários e as aquisições.

Depois de criar um segmento, você pode usá-lo em uma ou mais campanhas. A campanha oferece mensagens personalizadas para os usuários no segmento.

Para mais informações, leia Segmentos.

Tópicos

- Criar segmentos
- Importar segmentos
- Personalizar segmentos com o AWS Lambda

Criar segmentos

Para chegar ao público-alvo de uma campanha, crie um segmento baseado nos dados relatados pelo aplicativo. Por exemplo, para entrar em contato com usuários que não usaram o aplicativo recentemente, você pode definir um segmento para aqueles que não o acessaram nos últimos 30 dias.

Criar segmentos com o AWS SDK for Java

O exemplo a seguir demonstra como criar um segmento com o AWS SDK for Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
```

```
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
                final String usage = """

                                Usage:   <appId>

                                Where:
                                  appId - The application ID to create a segment for.
```

```
                        """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();
                        segmentAttributes.put("Team", AttributeDimension.builder()
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());

                        RecencyDimension recencyDimension = RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors = SegmentBehaviors.builder()
                                        .recency(recencyDimension)
                                        .build();

                        SegmentDemographics segmentDemographics = SegmentDemographics
                                        .builder()
                                        .build();

                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
```

```
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
 WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
 CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();

                        CreateSegmentResponse createSegmentResult =
 client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
 createSegmentResult.segmentResponse().id());
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
        }
}
```

Quando você executa este exemplo, o conteúdo abaixo é impresso na janela do console do seu IDE:

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

Para ver o exemplo completo do SDK, consulte CreateSegment.java no GitHub.

Importar segmentos

Com o Amazon Pinpoint, você pode definir um segmento de usuário importando informações sobre os endpoints que pertencem ao segmento. Um endpoint é um destino de mensagens único, como um token de push para dispositivos móveis, um número de celular ou um endereço de e-mail.

A importação de segmentos é útil se você já criou segmentos de usuários fora do Amazon Pinpoint, mas deseja envolver os usuários em campanhas do Amazon Pinpoint.

Quando você importa um segmento, o Amazon Pinpoint obtém os endpoints do segmento do Amazon Simple Storage Service (Amazon S3). Antes de importar, adicione os endpoints ao Amazon S3 e crie um perfil do IAM que conceda ao Amazon Pinpoint acesso ao Amazon S3. Em seguida, você informa a ao Amazon Pinpoint o local de armazenamento dos endpoints do Amazon S3, e o Amazon Pinpoint adiciona cada endpoint ao segmento.

Para criar o perfil do IAM, consulte Perfil do IAM para importação de endpoints ou segmentos. Para obter informações sobre a importação de um segmento usando o console do Amazon Pinpoint, consulte Importar segmentos no Guia do usuário do Amazon Pinpoint.

Importar um segmento

O exemplo a seguir demonstra como importar um segmento usando o AWS SDK for Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the segment
 definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon Pinpoint to access S3.
 You need to set trust management for this to work. See https://docs.aws.amazon.com/
IAM/latest/UserGuide/reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket, key,
 roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for import
 job status.");
```

```
        System.out.println("See application " + response.jobStatus() + " for import job
 status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse = client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

Para ver o exemplo completo do SDK, consulte ImportingSegments.java no GitHub.

Personalizar segmentos com o AWS Lambda

Essa documentação é de pré-lançamento para um atributo em versão beta pública. Está sujeita a alteração.

Você pode usar uma função do AWS Lambda para ajustar como uma campanha do Amazon Pinpoint envolve o seu público-alvo. Com o AWS Lambda, você pode modificar o segmento da campanha no momento em que o Amazon Pinpoint enviar a mensagem da campanha.

O AWS Lambda é um serviço de computação que pode ser usado para executar código sem provisionamento ou gerenciamento de servidores. Você empacota seu código e faz upload no Lambda como funções do Lambda. O Lambda executa uma função quando ela é invocada, o que pode ser feito manualmente por você ou automaticamente em resposta a eventos. Para obter mais informações, consulte o Guia do desenvolvedor do AWS Lambda.

Para atribuir uma função do Lambda a uma campanha, defina as configurações `CampaignHook` da campanha usando o recurso Campanha na API do Amazon Pinpoint. Essas configurações incluem o nome da função do Lambda. Elas também incluem o modo `CampaignHook` que define se o Amazon Pinpoint recebe um valor de retorno da função.

Uma função do Lambda que você atribui a uma campanha é referida como uma extensão do Amazon Pinpoint.

Com as configurações de `CampaignHook` definidas, o Amazon Pinpoint invoca automaticamente a função do Lambda ao executar a campanha, antes de enviar a mensagem da campanha. Quando o Amazon Pinpoint invoca a função, ele fornece dados de eventos sobre a entrega da mensagem. Esses dados incluem o segmento da campanha, que é a lista de endpoints aos quais o Amazon Pinpoint envia a mensagem.

Se o modo `CampaignHook` estiver definido como `FILTER`, o Amazon Pinpoint permite que a função modifique e retorne o segmento antes de enviar a mensagem. Por exemplo, a função pode atualizar as definições de endpoint com atributos que contenham dados de uma origem externa ao Amazon Pinpoint. Ou a função pode filtrar o segmento removendo determinados endpoints, com base em condições no código da função. Depois de o Amazon Pinpoint receber o segmento modificado da função, ele envia a mensagem para cada um dos endpoints do segmento usando o canal de entrega da campanha.

Ao processar os segmentos com o AWS Lambda, você tem mais controle sobre os destinatários e o conteúdo das mensagens que envia. Você pode personalizar suas campanhas em tempo real, no momento em que as mensagens da campanha são enviadas. A filtragem de segmentos permite que você envolva subconjuntos de segmentos definidos de maneira mais restrita. Adicionar ou atualizar os atributos do endpoint permite que você disponibilize novos dados para variáveis de mensagens.

> **Note**
>
> Você também pode usar as configurações de `CampaignHook` para atribuir uma função do Lambda que lide com a entrega de mensagens. Esse tipo de função é útil para a entrega de mensagens por meio de canais personalizados aos quais o Amazon Pinpoint não oferece suporte, como plataformas de mídia social. Para obter mais informações, consulte Criação de canais personalizados no Amazon Pinpoint.
>
> Ao invocar um hook do Lambda usando o Amazon Pinpoint, a função do Lambda também deve estar na mesma região do projeto do Amazon Pinpoint.

Para modificar segmentos de campanha com o AWS Lambda, primeiro crie uma função que processe os dados de eventos enviados pelo Amazon Pinpoint e retorne um segmento modificado. Em seguida, autorize o Amazon Pinpoint a invocar a função atribuindo uma política de função do Lambda. Por fim, atribua a função a uma ou mais campanhas, definindo as configurações `CampaignHook`.

Dados de eventos

Quando o Amazon Pinpoint invoca a função do Lambda, ele fornece a carga a seguir como dados de eventos:

```
{
  "MessageConfiguration": {Message configuration}
  "ApplicationId": ApplicationId,
  "CampaignId": CampaignId,
  "TreatmentId": TreatmentId,
  "ActivityId": ActivityId,
  "ScheduledTime": Scheduled Time,
  "Endpoints": {
    EndpointId: {Endpoint definition}
    . . .
  }
}
```

O AWS Lambda transmite os dados de evento para o código da função. Os dados de eventos fornecem os seguintes atributos:

- `MessageConfiguration`: tem a mesma estrutura do objeto `DirectMessageConfiguration` do recurso Mensagens na API do Amazon Pinpoint.

- `ApplicationId`: o ID do projeto do Amazon Pinpoint ao qual a campanha pertence.
- `CampaignId`: o ID da campanha do Amazon Pinpoint para o qual a função é invocada.
- `TreatmentId`: o ID de uma variação de campanha usada para testes A/B.
- `ActivityId`: o ID da atividade sendo executada pela campanha.
- `ScheduledTime`: a data e o horário, em formato ISO 8601, em que as mensagens da campanha serão entregues.
- `Endpoints`: um mapa que associa IDs de endpoint a definições de endpoint. Cada carga de dados de eventos contém até 50 endpoints. Se o segmento da campanha contiver mais de 50 endpoints, o Amazon Pinpoint invocará a função repetidamente, com até 50 endpoints por vez, até que todos os endpoints sejam processados.

Criação de uma função do Lambda

Para saber como criar uma função do Lambda, consulte Conceitos básicos no Guia do desenvolvedor do AWS Lambda. Ao criar a função, lembre-se de que a entrega de mensagens falha nas seguintes condições:

- A função do Lambda leva mais de 15 segundos para retornar o segmento modificado.
- O Amazon Pinpoint não consegue decodificar o valor de retorno da função.
- A função requer mais de 3 tentativas do Amazon Pinpoint para invocá-lo com êxito.

O Amazon Pinpoint só aceita definições de endpoint no valor de retorno da função. A função não consegue modificar outros elementos nos dados de eventos.

Exemplo de função do Lambda

Sua função do Lambda processa os dados de eventos enviados pelo Amazon Pinpoint e retorna os endpoints modificados, conforme mostrado pelo seguinte handler de exemplo, escrito em Node.js:

```
'use strict';

exports.handler = (event, context, callback) => {
    for (var key in event.Endpoints) {
        if (event.Endpoints.hasOwnProperty(key)) {
            var endpoint = event.Endpoints[key];
            var attr = endpoint.Attributes;
            if (!attr) {
```

```
                attr = {};
                endpoint.Attributes = attr;
            }
            attr["CreditScore"] = [ Math.floor(Math.random() * 200) + 650];
        }
    }
    console.log("Received event:", JSON.stringify(event, null, 2));
    callback(null, event.Endpoints);
};
```

O Lambda passa os dados de eventos para o handler como o parâmetro `event`.

Neste exemplo, o handler percorre cada endpoint no objeto `event.Endpoints` e adiciona um novo atributo, `CreditScore`, ao endpoint. O valor do atributo `CreditScore` é simplesmente um número aleatório.

A instrução `console.log()` registra o evento no CloudWatch Logs.

A instrução `callback()` retorna os endpoints modificados ao Amazon Pinpoint. Normalmente, o parâmetro `callback` é opcional em funções do Lambda Node.js, mas é necessário neste contexto porque a função deve retornar os endpoints atualizados ao Amazon Pinpoint.

Sua função deve retornar endpoints no mesmo formato fornecido pelos dados do evento, que é um mapa que associa IDs de endpoint a definições de endpoint, como no exemplo a seguir:

```
{
    "eqmj8wpxszeqy/b3vch04sn41yw": {
        "ChannelType": "GCM",
        "Address": "4d5e6f1a2b3c4d5e6f7g8h9i0j1a2b3c",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "android"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    },
    "idrexqqtn8sbwfex0ouscod0yto": {
        "ChannelType": "APNS",
        "Address": "1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
```

```
            "Make": "apple"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    }
}
```

A função de exemplo modifica e retorna o objeto `event.Endpoints` que recebeu nos dados do evento.

Opcionalmente, você pode incluir os atributos `BodyOverride` e `TitleOverride` nas definições de endpoint que retorna.

> **Note**
>
> Quando você usa essa solução para enviar mensagens, o Amazon Pinpoint respeita os atributos `TitleOverride` e `BodyOverride` apenas para endpoints em que o valor do atributo `ChannelType` é um dos seguintes: `ADM`, `APNS`, `APNS_SANDBOX`, `APNS_VOIP`, `APNS_VOIP_SANDBOX`, `BAIDU`, `GCM` ou `SMS`.
> O Amazon Pinpoint não respeita esses atributos para endpoints em que o valor do atributo `ChannelType` é `EMAIL`.

Atribuir uma política de função do Lambda

Para usar a função do Lambda para processar endpoints, você deve primeiro autorizar o Amazon Pinpoint a invocar essa função. Para conceder permissão de invocação, atribua uma política de função do Lambda à função. A política de função do Lambda é uma política de permissões com base em recursos que designa as entidades que podem usar sua função e que ações essas entidades podem executar.

Para mais informações, consulte Uso de políticas com base em recursos para o AWS Lambda no Guia do usuário do AWS Lambda.

Exemplo de política de função

A política a seguir concede permissão à entidade principal de serviço do Amazon Pinpoint para usar a ação `lambda:InvokeFunction` para uma campanha específica (*`campaign-id`*):

```
{
```

```
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:account-id:function:function-name}",
  "Condition": {
    "StringEquals": {
      "AWS:SourceAccount": "111122223333"
    },
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-
id/campaigns/campaign-id"
    }
  }
}
```

Sua política de função requer um bloco `Condition` que inclua uma chave `AWS:SourceArn`. Esse código informa que a campanha do Amazon Pinpoint tem permissão para invocar a função. Neste exemplo, a política concede permissão apenas a uma única campanha. O bloco `Condition` também deve incluir uma chave `AWS:SourceAccount`, que controla qual conta AWS pode invocar a ação.

Para escrever uma política mais genérica, use um curinga que corresponda a vários caracteres (*). Por exemplo, você pode usar o seguinte bloco `Condition` para permitir que qualquer campanha em um projeto específico do Amazon Pinpoint (*application-id*) chame a função:

```
...
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "111122223333"
  },
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/*"
  }
}
...
```

Para que a função do Lambda seja a função padrão usada por todas as campanhas de um projeto, recomendamos configurar o bloco `Condition` para a política da maneira anterior. Para obter

informações sobre como definir uma função do Lambda como padrão para todas as campanhas em um projeto, consulte Atribuição de uma função do Lambda a uma campanha.

Conceder permissão de invocação do Amazon Pinpoint

Você pode usar a AWS Command Line Interface (AWS CLI) para adicionar permissões à política de função do Lambda atribuída à função do Lambda. Para permitir que o Amazon Pinpoint invoque uma função para uma campanha específica, use o comando do Lambda `add-permission`, conforme mostrado no exemplo a seguir:

```
$ aws lambda add-permission \
> --function-name function-name \
> --statement-id sid \
> --action lambda:InvokeFunction \
> --principal pinpoint.us-east-1.amazonaws.com \
> --source-account 111122223333
> --source-arn arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/campaign-id
```

Você pode pesquisar seus IDs de campanha usando o comando get-campaigns no. AWS CLI. Você também pode pesquisar o ID do seu aplicativo usando o comando get-apps.

Ao executar o comando `add-permission` do Lambda, o Lambda retornará o seguinte resultado:

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/application-id/
campaigns/campaign-id\"}}
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
}
```

O valor de `Statement` é uma versão da cadeia de caracteres JSON da instrução adicionada à política da função do Lambda.

Atribuição de uma função do Lambda a uma campanha

Você pode atribuir uma função do Lambda a uma campanha do Amazon Pinpoint específica. Ou é possível definir a função do Lambda como o padrão usado por todas as campanhas de um projeto, exceto nas campanhas para as quais você atribui uma função individualmente.

Para atribuir uma função do Lambda a uma campanha específica, use a API do Amazon Pinpoint para criar ou atualizar um objeto `Campaign` e definir seu atributo `CampaignHook`. Para definir uma função do Lambda como padrão para todas as campanhas em um projeto, crie ou atualize o recurso `Settings` para esse projeto e defina seu objeto `CampaignHook`.

Em ambos os casos, defina os seguintes atributos `CampaignHook`:

- `LambdaFunctionName`: o nome ou ARN da função do Lambda que o Amazon Pinpoint invoca antes de enviar mensagens para a campanha.
- `Mode`: defina como `FILTER`. Com esse modo, o Amazon Pinpoint invoca a função e aguarda até que ela retorne os endpoints modificados. Após recebê-los, o Amazon Pinpoint envia a mensagem. O Amazon Pinpoint aguarda até 15 segundos antes de falhar na entrega da mensagem.

Com as configurações `CampaignHook` definidas para uma campanha, o Amazon Pinpoint invoca a função do Lambda especificada antes de enviar as mensagens da campanha. O Amazon Pinpoint aguarda o recebimento dos endpoints modificados da função. Se o Amazon Pinpoint recebe os endpoints atualizados, ele prossegue com a entrega da mensagem, usando os dados dos endpoints atualizados.

Criação de campanhas

Para ajudar a aumentar o envolvimento entre seu aplicativo e os usuários, use o Amazon Pinpoint para criar e gerenciar campanhas de notificação por push que atinjam segmentos específicos.

Por exemplo, sua campanha pode estimular o retorno dos usuários que não o usam há muito tempo ou oferecer promoções especiais àqueles que não fizeram compras recentemente.

A campanha envia uma mensagem personalizada para um segmento de usuários específico. A campanha poderá enviar a mensagem a todos os usuários do segmento, ou você pode alocar um holdout, que é uma porcentagem de usuários que não recebem mensagens.

Você pode programar a campanha para enviar a mensagem uma vez só ou de forma recorrente, como uma vez por semana. Para evitar que os usuários recebam-na em momentos inconvenientes, você pode programar um período de silêncio, em que a campanha não envie nada.

Para experimentar estratégias alternativas, configure sua campanha como um teste A/B. Esse teste inclui dois ou mais tratamento da mensagem ou programação. Esses tratamentos são variações do conteúdo enviado. À medida que seus usuários reagem à campanha, você poderá visualizar e fazer análises para comparar a eficácia de cada tratamento.

Para mais informações, acesse Campanhas.

Criar campanhas padrão

Campanhas padrão enviam notificações por push personalizadas a segmentos especificados, de acordo com a programação definida.

Criação de campanhas com o AWS SDK for Java

O exemplo a seguir demonstra como criar uma campanha com o AWS SDK for Java.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
```

```
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>

                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
```

```
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId, String
segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
    }

    public static CampaignResponse createCampaign(PinpointClient client, String appID,
String segmentID) {

        try {
            Schedule schedule = Schedule.builder()
                    .startTime("IMMEDIATE")
                    .build();

            Message defaultMessage = Message.builder()
                    .action(Action.OPEN_APP)
                    .body("My message body.")
                    .title("My message title.")
                    .build();

            MessageConfiguration messageConfiguration = MessageConfiguration.builder()
                    .defaultMessage(defaultMessage)
                    .build();

            WriteCampaignRequest request = WriteCampaignRequest.builder()
                    .description("My description")
                    .schedule(schedule)
                    .name("MyCampaign")
                    .segmentId(segmentID)
                    .messageConfiguration(messageConfiguration)
                    .build();

            CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
```

```
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());

            System.out.println("Campaign ID: " + result.campaignResponse().id());
            return result.campaignResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }
}
```

Quando você executa este exemplo, o conteúdo abaixo é impresso na janela do console do seu IDE:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Para ver o exemplo completo do SDK, consulte CreateCampaign.java on. GitHub

Criar campanhas de teste A/B

Testes A/B agem como campanhas padrão, mas permitem que você defina tratamentos diferentes para a mensagem ou programação.

Criação de campanhas de teste A/B com o AWS SDK for Java

O exemplo a seguir demonstra como criar uma campanha de teste A/B com o AWS SDK for Java.

```
import com.amazonaws.services.pinpoint.AmazonPinpointClient;
import com.amazonaws.services.pinpoint.model.Action;
import com.amazonaws.services.pinpoint.model.CampaignResponse;
import com.amazonaws.services.pinpoint.model.CreateCampaignRequest;
import com.amazonaws.services.pinpoint.model.CreateCampaignResult;
import com.amazonaws.services.pinpoint.model.Message;
import com.amazonaws.services.pinpoint.model.MessageConfiguration;
import com.amazonaws.services.pinpoint.model.Schedule;
import com.amazonaws.services.pinpoint.model.WriteCampaignRequest;
import com.amazonaws.services.pinpoint.model.WriteTreatmentResource;

import java.util.ArrayList;
```

```
import java.util.List;

public class PinpointCampaignSample {

    public CampaignResponse createAbCampaign(AmazonPinpointClient client, String appId,
 String segmentId) {
        Schedule schedule = new Schedule()
                .withStartTime("IMMEDIATE");

        // Default treatment.
        Message defaultMessage = new Message()
                .withAction(Action.OPEN_APP)
                .withBody("My message body.")
                .withTitle("My message title.");

        MessageConfiguration messageConfiguration = new MessageConfiguration()
                .withDefaultMessage(defaultMessage);

        // Additional treatments
        WriteTreatmentResource treatmentResource = new WriteTreatmentResource()
                .withMessageConfiguration(messageConfiguration)
                .withSchedule(schedule)
                .withSizePercent(40)
                .withTreatmentDescription("My treatment description.")
                .withTreatmentName("MyTreatment");

        List<WriteTreatmentResource> additionalTreatments = new
 ArrayList<WriteTreatmentResource>();
        additionalTreatments.add(treatmentResource);

        WriteCampaignRequest request = new WriteCampaignRequest()
                .withDescription("My description.")
                .withSchedule(schedule)
                .withSegmentId(segmentId)
                .withName("MyCampaign")
                .withMessageConfiguration(messageConfiguration)
                .withAdditionalTreatments(additionalTreatments)
                .withHoldoutPercent(10); // Hold out of A/B test

        CreateCampaignRequest createCampaignRequest = new CreateCampaignRequest()
                .withApplicationId(appId).withWriteCampaignRequest(request);

        CreateCampaignResult result = client.createCampaign(createCampaignRequest);
```

```
        System.out.println("Campaign ID: " + result.getCampaignResponse().getId());

        return result.getCampaignResponse();
    }

}
```

Quando você executa este exemplo, o conteúdo abaixo é impresso na janela do console do seu IDE:

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

Usar a API de SMS e voz, versão 2, do Amazon Pinpoint

> **Note**
>
> O Guia do usuário do Amazon Pinpoint foi atualizado. Para obter as informações mais recentes sobre como criar, configurar e gerenciar os recursos de voz e SMS do Amazon Pinpoint, consulte o novo Guia do usuário de SMS do Amazon Pinpoint.
> O tópico a seguir foi movido para o novo Guia do usuário do Amazon Pinpoint SMS.
>
> - Gerenciando números de telefone
> - Gerenciando IDs de remetente
> - Gerenciando pools
> - Gerenciando listas de exclusão
> - Gerenciando conjuntos de configurações
> - Gerenciando palavras-chave
> - Gerenciando destinos de eventos
> - Enviar mensagens

O Amazon Pinpoint inclui uma API (chamada de API de SMS e voz, versão 2) que foi projetada para enviar SMS e mensagens de voz. Enquanto a API do Amazon Pinpoint se concentra no envio de mensagens por meio de campanhas e jornadas programadas e orientadas por eventos, a API de SMS e voz fornece novos recursos e capacidades para enviar SMS e mensagens de voz diretamente para destinatários individuais. Você pode usar a API de SMS e voz independentemente dos recursos de campanha e jornada do Amazon Pinpoint, ou usar os dois ao mesmo tempo para acomodar diferentes casos de uso. Se você já usa o Amazon Pinpoint para enviar SMS ou mensagens de voz, sua conta já está configurada para usar essa API.

Essa API é uma boa solução para usuários que têm uma arquitetura multilocatária, como provedores de software independentes (ISVs). Essa API facilita a garantia de que os dados do evento, os números de telefone de origem e as listas de cancelamento sejam separadas para diferentes locatários.

Ao usar a API de SMS e voz, recomendamos que você configure conjuntos de configurações e destinos de eventos. A API de SMS e voz não emite automaticamente dados de eventos para as

mensagens que você envia. A configuração de destinos de eventos garante que você capture dados importantes do evento, como entrega de mensagens e eventos de falha.

A versão 2 dessa API foi precedida pela versão 1. Se você usa atualmente a versão 1 dessa API, ela continuará disponível e você poderá continuar a usá-la. No entanto, se você migrar para a versão 2, obterá recursos adicionais, como a capacidade de criar grupos de números de telefone, solicitar novos números de telefone de forma programática e ativar ou desativar determinados recursos dos números de telefone.

> **Note**
>
> Atualmente, existem algumas tarefas que só podem ser concluídas usando o console do Amazon Pinpoint. Por exemplo, se você quiser verificar um número de telefone para usar enquanto sua conta estiver na sandbox de SMS, ou se quiser se registrar para usar o 10DLC, você deve usar o console do Amazon Pinpoint.

Esta seção fornece informações sobre essa API e inclui exemplos de como usá-la. Você também pode encontrar a documentação de referência na Referência da API de SMS e voz, versão 2.

Envio e validação de senhas de uso único (OTPs) com o Amazon Pinpoint

O Amazon Pinpoint inclui um atributo de gerenciamento de senha de uso único (OTP). Você pode usar esse atributo para gerar novas senhas de uso único e enviá-las aos destinatários como mensagens SMS. Seus aplicativos podem então chamar a API do Amazon Pinpoint para verificar essas senhas.

> **Important**
>
> Para usar esse recurso, sua conta deve ter acesso à produção e uma identidade de originação ativa. Para obter mais informações, consulte Sobre o sandbox do Amazon Pinpoint SMS e Solicitar um número de telefone no Guia do usuário do Amazon Pinpoint SMS.
>
> Em alguns países e regiões, você deve obter um número de telefone dedicado ou ID de origem antes de poder enviar mensagens SMS. Por exemplo, ao enviar mensagens para os destinatários nos Estados Unidos, você deve ter um número de ligação gratuita dedicado, um número 10DLC ou um código curto. Ao enviar mensagens para destinatários na Índia, você deve ter um ID de remetente registrado, que inclui um ID de entidade principal (PEID) e um ID de modelo. Esses requisitos ainda se aplicam ao usar o atributo OTP.
>
> Para usar esse atributo, você precisa de permissões para enviar e verificar mensagens OTP. Consulte Senhas de uso único. Se precisar de ajuda para determinar as permissões, consulte Solução de problemas de gerenciamento de identidade e acesso do Amazon Pinpoint.

Envio de uma mensagem OTP

Você pode usar a operação `SendOtpMessages` na API do Amazon Pinpoint para enviar um código OTP a um usuário do aplicativo. Quando você usa essa API, o Amazon Pinpoint gera um código aleatório e o envia ao usuário como uma mensagem SMS. Sua solicitação pode incluir os seguintes parâmetros:

- `Channel`: o canal de comunicação pelo qual o código OTP é enviado. Atualmente, somente mensagens SMS são compatíveis, então o único valor aceitável é SMS.

- `BrandName`: o nome da marca, empresa ou produto associado ao código OTP. O nome pode conter até 20 caracteres.

> **Note**
>
> Quando o Amazon Pinpoint envia a mensagem OTP, o nome da marca é inserido automaticamente no seguinte modelo de mensagem:
>
> ```
> This is your One Time Password: {{otp}} from {{brand}}
> ```
>
> Portanto, se você especificar ExampleCorp o nome da sua marca e o Amazon Pinpoint gerar uma senha de uso único de 123456, ele enviará a seguinte mensagem ao seu usuário:
>
> ```
> This is your One Time Password: 123456 from ExampleCorp
> ```

- `CodeLength`: o número de dígitos que estarão no código OTP enviado ao destinatário. Os códigos OTP podem conter entre 5 e 8 dígitos, inclusive.
- `ValidityPeriod`: a quantidade de tempo, em minutos, em que o código OTP será válido. O período de validade pode ser entre 5 e 60 minutos.
- `AllowedAttempts`: o número de vezes que o destinatário pode tentar, sem sucesso, verificar a OTP. Se o número de tentativas exceder esse valor, a OTP se tornará automaticamente inválida. A quantidade máxima de tentativas é 5.
- `Language`: o idioma, no formato IETF BCP-47, a ser usado ao enviar a mensagem. Os valores aceitáveis são:
  - `de-DE`: Alemão
  - `en-GB`: Inglês (Reino Unido)
  - `en-US`: Inglês (EUA)
  - `es-419`: Espanhol (América Latina)
  - `es-ES`: Espanhol
  - `fr-CA`: Francês (Canadá)
  - `fr-FR`: Francês
  - `it-IT`: Italiano
  - `ja-JP`: Japonês

  - `ko-KR`: Coreano
  - `pt-BR`: Português (Brasil)
  - `zh-CN`: Chinês (simplificado)
  - `zh-TW`: Chinês (tradicional)
- `OriginationIdentity`: a identidade de origem (como código longo, código curto ou ID do remetente) usada para enviar o código OTP. Se você usar um código longo ou um número gratuito para enviar a OTP, o número de telefone deverá estar no formato E.164.
- `DestinationIdentity`: o número de telefone, no formato E.164, para o qual o código OTP foi enviado.
- `ReferenceId`: um ID exclusivo de referência para a solicitação. O ID de referência corresponde exatamente ao ID de referência que você fornece ao verificar a OTP. O ID de referência pode conter entre 1 e 48 caracteres.
- `EntityId`: um ID de entidade registrado em uma agência reguladora. Atualmente, esse parâmetro é usado somente ao enviar mensagens a destinatários na Índia. Se você não estiver enviando para destinatários na Índia, poderá omitir esse parâmetro.
- `TemplateId`: um ID de modelo registrado em uma agência reguladora. Atualmente, esse parâmetro é usado somente ao enviar mensagens a destinatários na Índia. Se você não estiver enviando para destinatários na Índia, poderá omitir esse parâmetro.

> **Note**
>
> Para obter mais informações sobre os requisitos de envio de mensagens para destinatários na Índia, consulte Requisitos especiais para envio de mensagens SMS para destinatários na Índia no Guia do usuário do Amazon Pinpoint.

Para garantir que sua conta do Amazon Pinpoint esteja configurada corretamente para enviar mensagens OTP, você pode usar o AWS CLI para enviar uma mensagem de teste. Para obter mais informações sobre o AWS CLI, consulte o Guia AWS Command Line Interface do usuário.

Para enviar uma mensagem OTP de teste usando o AWS CLI, execute o send-otp-messagecomando no terminal:

```
aws pinpoint send-otp-message --application-id 7353f53e6885409fa32d07cedexample --send-
otp-message-request-parameters
 Channel=SMS,BrandName=ExampleCorp,CodeLength=5,ValidityPeriod=20,AllowedAttempts=5,Origination
```

No comando anterior, faça o seguinte:

- Substitua *7353f53e6885409fa32d07cedexample* pelo ID do seu aplicativo.
- *ExampleCorp*Substitua pelo nome da sua empresa.
- `CodeLegth`Substitua *5* in pelo número de dígitos que estarão no código OTP enviado ao destinatário.
- Substitua *20* minutos pela quantidade de tempo, em minutos, em que o código OTP será válido. `ValidityPeriod`
- Substitua *5* em `AllowedAttempts` pelo número de vezes que o destinatário pode tentar verificar a OTP sem sucesso.
- Substitua *+18555550142* pela identidade de `OriginationIdentity` origem usada para enviar o código OTP.
- Substitua *+12065550007* pelo número de telefone para o `DestinationIdentity` qual enviar o código OTP.
- *SampleReferenceId*`ReferenceId`Substitua por um ID de referência exclusivo para a solicitação.

Resposta do **`SendOtpMessage`**

Ao enviar uma mensagem OTP com êxito, você recebe uma resposta semelhante ao seguinte exemplo:

```
{
    "MessageResponse": {
        "ApplicationId": "7353f53e6885409fa32d07cedexample",
        "RequestId": "255d15ea-75fe-4040-b919-096f2example",
        "Result": {
            "+12065550007": {
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "nvrmgq9kq4en96qgp0tlqli3og1at6aexample",
                "StatusCode": 200,
                "StatusMessage": "MessageId: nvrmgq9kq4en96qgp0tlqli3og1at6aexample"
            }
        }
    }
}
```

Validar uma mensagem OTP

Para verificar um código OTP, chame a API `VerifyOtpMessages`. Essa solicitação precisam incluir os parâmetros a seguir:

- `DestinationIdentity`: o número de telefone, no formato E.164, para o qual o código OTP foi enviado.
- `ReferenceId`: o ID de referência que você usou ao enviar o código OTP ao destinatário. O ID de referência deve corresponder exatamente.
- `Otp`: o código OTP que você está validando.

Você pode usar o AWS CLI para testar o processo de validação. Para obter mais informações sobre como instalar e configurar o AWS CLI, consulte o Guia do AWS Command Line Interface usuário.

Para verificar uma OTP usando o AWS CLI, execute o verify-otp-messagecomando no terminal:

```
aws pinpoint verify-otp-message --application-id 7353f53e6885409fa32d07cedexample --
verify-otp-message-request-parameters
 DestinationIdentity=+12065550007,ReferenceId=SampleReferenceId,Otp=01234
```

No comando anterior, faça o seguinte:

- Substitua *7353f53e6885409fa32d07cedexample* pelo ID do seu aplicativo.
- Substitua *+12065550007* pelo número de telefone para o `DestinationIdentity` qual o código OTP foi enviado.
- *SampleReferenceId*`ReferenceId`Substitua por um ID de referência exclusivo para a solicitação. Esse valor deve corresponder ao `ReferenceID` que foi usado para enviar a solicitação.
- Substitua *01234* in `Otp` por um Otp que foi enviado para o. `DestinationIdentity`

Resposta do **`VerifyOtpMessage`**

Quando você envia uma solicitação para a API `VerifyOTPMessage`, ela retorna um objeto `VerificationResponse`, que contém uma única propriedade, `Valid`. Se o ID de referência, o número de telefone e a OTP corresponderem aos valores esperados pelo Amazon Pinpoint e se a OTP não tiver expirado, o valor de `Valid` será `true`; caso contrário, será `false`. Veja a seguir um exemplo de uma verificação de OTP bem sucedida:

```
{
    "VerificationResponse": {
        "Valid": true
    }
}
```

Exemplos de código

Esta seção contém exemplos de código que mostram como usar o SDK para Python (Boto3) para enviar e verificar códigos OTP.

Gerar um ID de referência

A função a seguir gera um ID de referência exclusivo para cada destinatário, com base no número de telefone do destinatário, no produto ou marca para o qual o destinatário está recebendo uma OTP e na origem da solicitação (que pode ser o nome de uma página em um site ou aplicativo, por exemplo). Ao verificar o código OTP, você deve passar um ID de referência idêntico para que a validação seja bem-sucedida. Os exemplos de código de envio e validação usam essa função utilitária.

Essa função não é obrigatória, mas é uma forma útil de definir o escopo do processo de envio e verificação de OTP para uma transação específica, de forma que possa ser facilmente reenviada durante a etapa de verificação. Você pode usar qualquer ID de referência que quiser. Esse é apenas um exemplo básico. No entanto, os outros exemplos de código nesta seção dependem dessa função.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import hashlib

def generate_ref_id(destinationNumber,brandName,source):
    refId = brandName + source + destinationNumber
    return hashlib.md5(refId.encode()).hexdigest()
```

Envio de códigos OTP

O exemplo de código a seguir mostra como usar o SDK para Python (Boto3) para enviar um código OTP.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

### Some variables that are unlikely to change from request to request. ###

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The phone number or short code to send the message from.
originationNumber = "+18555550142"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# The number of times the user can unsuccessfully enter the OTP code before it becomes
 invalid.
allowedAttempts = 3

# Function that sends the OTP as an SMS message.
def send_otp(destinationNumber,codeLength,validityPeriod,brandName,source,language):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.send_otp_message(
            ApplicationId=appId,
            SendOTPMessageRequestParameters={
                'Channel': 'SMS',
                'BrandName': brandName,
                'CodeLength': codeLength,
                'ValidityPeriod': validityPeriod,
                'AllowedAttempts': allowedAttempts,
                'Language': language,
                'OriginationIdentity': originationNumber,
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source)
            }
        )

    except ClientError as e:
        print(e.response)
```

```
    else:
        print(response)

# Send a message to +14255550142 that contains a 6-digit OTP that is valid for 15
 minutes. The
# message will include the brand name "ExampleCorp", and the request originated from a
 part of your
# site or application called "CreateAccount". The US English message template should be
 used to
# send the message.
send_otp("+14255550142",6,15,"ExampleCorp","CreateAccount","en-US")
```

Validar códigos OTP

O exemplo de código a seguir mostra como usar o SDK para Python (Boto3) para verificar um código OTP já enviado. Para que a etapa de validação seja bem-sucedida, sua solicitação deve incluir um ID de referência que corresponda exatamente ao ID de referência usado para enviar a mensagem.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# Function that verifies the OTP code.
def verify_otp(destinationNumber,otp,brandName,source):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.verify_otp_message(
            ApplicationId=appId,
            VerifyOTPMessageRequestParameters={
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source),
                'Otp': otp
            }
        )
```

```
    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Verify the OTP 012345, which was sent to +14255550142. The brand name ("ExampleCorp")
 and the
# source name ("CreateAccount") are used to generate the correct reference ID.
verify_otp("+14255550142","012345","ExampleCorp","CreateAccount")
```

Enviar e recuperar mensagens no aplicativo no Amazon Pinpoint

Você pode usar mensagens no aplicativo para enviar mensagens direcionadas aos usuários de seus aplicativos. As mensagens no aplicativo são altamente personalizáveis. Elas podem incluir botões que abrem sites ou levam os usuários a partes específicas do seu aplicativo. Você pode configurar as cores de fundo e do texto, posicionar o texto e adicionar botões e imagens à notificação. Você pode enviar uma única mensagem ou criar um carrossel que contenha até cinco mensagens exclusivas. Para uma visão geral das mensagens no aplicativo, incluindo instruções para criar modelos de mensagens no aplicativo, consulte Criação de modelos no aplicativo no Guia do usuário do Amazon Pinpoint.

Você pode usar o AWS Amplify para integrar perfeitamente os recursos de mensagens no aplicativo do Amazon Pinpoint ao seu aplicativo. O Amplify pode gerenciar automaticamente os processos de busca de mensagens, renderização de mensagens e envio de dados analíticos para o Amazon Pinpoint. Atualmente, essa integração é compatível com aplicativos React Native. Para obter mais informações, consulte Mensagens no aplicativo na documentação do Amplify Framework.

Esta seção fornece informações sobre como solicitar mensagens no aplicativo para um endpoint em seu aplicativo e como interpretar o resultado dessa solicitação.

Recuperação de mensagens no aplicativo para um endpoint

Seus aplicativos podem chamar a API GetInAppMessages para recuperar todas as mensagens no aplicativo às quais um determinado endpoint tem direito. Quando você chama a API `GetInAppMessages`, fornece os seguintes parâmetros:

- `ApplicationId`: o ID exclusivo do projeto do Amazon Pinpoint ao qual a campanha de mensagem no aplicativo está associada.
- `EndpointId`: o ID exclusivo do endpoint para o qual você está recuperando mensagens.

Quando você chama a API com esses valores, ela retorna uma lista de mensagens. Para obter mais informações sobre a resposta produzida por essa operação, consulte Entendimento das respostas da API `GetInAppMessages`

Você também pode usar AWS SDKs para chamar a operação `GetInAppMessages`. Os exemplos de código a seguir incluem funções que recuperam mensagens no aplicativo.

JavaScript

Crie o cliente em um módulo separado e exporte-o:

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
const REGION = "us-east-1";
const pinClient = new PinpointClient({ region: REGION });
export { pinClient };
```

Recupere mensagens no aplicativo para um endpoint:

```

// Import required AWS SDK clients and commands for Node.js
import { PinpointClient, GetInAppMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./lib/pinClient.js";

("use strict");

//The Amazon Pinpoint application ID.
const projectId = "4c545b28d21a490cb51b0b364example";

//The ID of the endpoint to retrieve messages for.
const endpointId = "c5ac671ef67ee3ad164cf7706example";

const params = {
  ApplicationId: projectId,
  EndpointId: endpointId
};

const run = async () => {
  try {
    const data = await pinClient.send(new GetInAppMessagesCommand(params));
    console.log(JSON.stringify(data, null, 4));
    return data;
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Python

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

logger = logging.getLogger(__name__)

def retrieve_inapp_messages(
            pinpoint_client, project_id, endpoint_id):
    """
    Retrieves the in-app messages that a given endpoint is entitled to.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project ID.
    :param endpoint_id: The ID of the endpoint to retrieve messages for.
    :return: A JSON object that contains information about the in-app message.
    """

    try:
        response = pinpoint_client.get_in_app_messages(
            ApplicationId=project_id,
            EndpointId=endpoint_id)
    except ClientError:
        logger.exception("Couldn't retrieve messages.")
        raise
    else:
        return response

def main():
    project_id = "4c545b28d21a490cb51b0b364example"
    endpoint_id = "c5ac671ef67ee3ad164cf7706example"
    inapp_response = retrieve_inapp_messages(
        boto3.client('pinpoint'), project_id, endpoint_id)
    print(inapp_response)

if __name__ == '__main__':
    main()
```

Entendimento das respostas da API **`GetInAppMessages`**

Quando você chama a operação da API GetInAppMessages, ela retorna uma lista de mensagens às quais o endpoint especificado tem direito. Seu aplicativo pode então renderizar a mensagem com base nos valores da resposta.

A lista a seguir é um exemplo do objeto JSON retornado quando você chama a API `GetInAppMessages`:


```
{
  "InAppMessagesResponse":{
    "InAppMessageCampaigns":[
      {
        "CampaignId":"inAppTestCampaign-4c545b28d21a490cb51b0b364example",
        "DailyCap":0,
        "InAppMessage":{
          "Content":[
            {
              "BackgroundColor":"#f8e71c",
              "BodyConfig":{
                "Alignment":"CENTER",
                "Body":"This is a sample in-app message sent using Amazon Pinpoint.",
                "TextColor":"#d0021b"
              },
              "HeaderConfig":{
                "Alignment":"CENTER",
                "Header":"Sample In-App Message",
                "TextColor":"#d0021b"
              },
              "ImageUrl":"https://example.com/images/thumbnail.png",
              "PrimaryBtn":{
                "DefaultConfig":{
                  "BackgroundColor":"#d0021b",
                  "BorderRadius":50,
                  "ButtonAction":"CLOSE",
                  "Text":"Dismiss",
                  "TextColor":"#f8e71c"
                }
              }
            }
          ],
          "Layout":"MIDDLE_BANNER"
        },
```

```
        "Priority":3,
        "Schedule":{
          "EndDate":"2021-11-06T00:08:05Z",
          "EventFilter":{
            "Dimensions":{
              "Attributes":{

              },
              "EventType":{
                "DimensionType":"INCLUSIVE",
                "Values":[
                  "_session.start"
                ]
              },
              "Metrics":{

              }
            }
          }
        },
        "SessionCap":0,
        "TotalCap":0,
        "TreatmentId":"0"
      }
    ]
  }
}
```

As seções a seguir fornecem mais informações sobre os componentes dessa resposta.

Objeto **InAppMessageCampaigns**

O objeto `InAppMessageCampaigns` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`CampaignId`	Uma cadeia de caracteres que contém o nome e o ID exclusivo da campanha do Amazon Pinpoint da qual a mensagem foi enviada. O nome precede o ID da	Criado automaticamente pelo Amazon Pinpoint quando você cria a campanha.

Atributo	Descrição	Onde é definido
	campanha. Os dois valores são separados por um hífen (-).	
`TreatmentId`	Um número inteiro que representa o ID do tratament o da campanha para essa mensagem. Se a campanha tiver apenas um tratamento, o valor será `0`.	
`Priority`	A prioridade da mensagem no aplicativo, expressa como um número inteiro entre 1 e 5, em que 1 indica a prioridade mais alta e 5 indica a prioridad e mais baixa.	Etapa 1 do processo de criação da campanha.
`InAppMessage`	Um Objeto `InAppMessage` que contém informações sobre como a mensagem é renderizada.	Com base no conteúdo do modelo de mensagem no aplicativo que foi especificado para a campanha.
`Schedule`	Um objeto de programaç ão que contém informações sobre quando a mensagem foi enviada.	Etapa 4 do processo de criação da campanha (se a campanha foi criada no console) ou do objeto `Schedule` (se a campanha foi criada usando a API ou um SDK).

Atributo	Descrição	Onde é definido
`DailyCap`	O número de vezes, mostrado como um número inteiro, em que uma mensagem no aplicativo pode ser exibida ao usuário durante um período de 24 horas.	Herdado das configurações em nível de projeto. Se a campanha incluir configurações que substituam as configurações do projeto, as primeiras serão utilizadas.
`SessionCap`	O número de vezes, expresso como um número inteiro, em que uma mensagem no aplicativo pode ser exibida ao usuário durante uma sessão do aplicativo.	
`TotalCap`	O número total de vezes, expresso como um número inteiro, em que qualquer mensagem no aplicativo pode ser exibida para um endpoint por campanha.	

Objeto **`InAppMessage`**

O objeto `InAppMessage` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`Content`	Uma matriz contendo um objeto InAppMessageContent, que descreve o conteúdo da mensagem.	Com base no conteúdo do modelo de mensagem no aplicativo que foi especificado para a campanha.
`Layout`	Uma cadeia de caracteres que descreve como a mensagem no aplicativo será exibida no	

Atributo	Descrição	Onde é definido
	dispositivo do destinatário. Os valores possíveis são: • `BOTTOM_BANNER` : a mensagem exibida como um banner na parte inferior da página. • `TOP_BANNER` : a mensagem exibida como um banner na parte superior da página. • `OVERLAYS`: a mensagem cobre toda a tela. • `MOBILE_FEED` : a mensagem é mostrada em uma janela na frente da página. • `MIDDLE_BANNER` : a mensagem exibida como um banner no meio da página. • `CAROUSEL`: um layout rolável de até cinco mensagens exclusivas.	

Objeto **`HeaderConfig`**

O objeto `HeaderConfig` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`Alignment`	Uma cadeia de caracteres que especifica o alinhamento do texto do cabeçalho. Os	Com base no conteúdo do modelo de mensagem no

Atributo	Descrição	Onde é definido
	valores possíveis são `LEFT`, `CENTER` e `RIGHT`.	aplicativo que foi especificado para a campanha.
`Header`	O texto do cabeçalho da mensagem.	
`TextColor`	A cor do texto do cabeçalho , expressa como um código de cor hexadecimal (como “#000000” para preto).	

Objeto **`BodyConfig`**

O objeto `BodyConfig` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`Alignment`	Uma cadeia de caracteres que especifica o alinhamento do texto do corpo da mensagem. Os valores possíveis são `LEFT`, `CENTER` e `RIGHT`.	Com base no conteúdo do modelo de mensagem no aplicativo que foi especificado para a campanha.
`Body`	O corpo do texto principal da mensagem.	
`TextColor`	A cor do corpo do texto, expressa como uma cadeia de caracteres que consiste em um código de cor hexadecimal (como “#000000” para preto).	

Objeto **`InAppMessageContent`**

O objeto `InAppMessageContent` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`BackgroundColor`	A cor de fundo da mensagem no aplicativo, expressa como uma cadeia de caracteres que contém um código de cor hexadecimal (como “#000000” para preto).	Com base no conteúdo do modelo de mensagem no aplicativo que foi especificado para a campanha.
`BodyConfig`	Um objeto BodyConfig, que contém informações relacionadas ao conteúdo principal do corpo da mensagem.	
`HeaderConfig`	Um objeto HeaderConfig, que contém informações relacionadas ao cabeçalho ou título da mensagem.	
`ImageUrl`	O URL da imagem que aparece na mensagem.	
`PrimaryBtn`	Um objeto InAppMessageButton que contém informações sobre o botão principal na mensagem.	
`SecondaryBtn`	Um objeto InAppMessageButton que contém informações sobre o botão secundário na mensagem. Não está presente se o modelo de mensagem no aplicativo não especificar um botão secundário.	

Objeto `Schedule`

O objeto `Schedule` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`EndDate`	O horário programado, no formato ISO 8601, quando a campanha terminou ou terminará.	Etapa 4 do processo de criação da campanha (se a campanha foi criada no console) ou do objeto `Schedule` (se a campanha foi criada usando a API ou um SDK).
`EventFilter`	Informações sobre o evento que faz com que a mensagem no aplicativo seja exibida. Quando você gera um evento que corresponde a uma campanha no aplicativo do Amazon Pinpoint, a mensagem é exibida.	

Objeto `InAppMessageButton`

Um objeto `InAppMessageButton` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`DefaultConfig`	Um objeto DefaultButtonConfig contém informações sobre as configurações padrão de um botão em uma mensagem no aplicativo.	Com base no conteúdo do modelo de mensagem no aplicativo que foi especificado para a campanha.
`Android`	Um objeto OverrideButtonConfig que especifica a forma como o botão se comporta em dispositivos Android. Isso substitui a configuração de	

Atributo	Descrição	Onde é definido
	botão padrão detalhada no objeto `DefaultConfig` .	
`IOS`	Um objeto OverrideButtonConfig que especifica a forma como o botão se comporta em dispositivos iOS. Isso substitui a configuração de botão padrão detalhada no objeto `DefaultConfig` .	
`Web`	Um objeto OverrideButtonConfig que especifica a forma como o botão se comporta em aplicativos web. Isso substitui a configuração de botão padrão detalhada no objeto `DefaultConfig` .	

Objeto **`DefaultButtonConfig`**

Um objeto `DefaultButtonConfig` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`BackgroundColor`	A cor de fundo do botão, expressa como uma cadeia de caracteres que contém o código de cor hexadecimal (como “#000000” para preto).	Com base no conteúdo do modelo de mensagem no aplicativo que foi especificado para a campanha.
`BorderRadius`	O raio da borda do botão em pixels, expresso como um número inteiro. Um número	

Atributo	Descrição	Onde é definido
	maior resulta em cantos mais arredondados.	
`ButtonAction`	Uma cadeia de caractere s que descreve a ação que ocorre quando um destinatá rio escolhe um botão em uma mensagem no aplicativo. Os valores possíveis são: • `LINK`: um link para um destino na Web. • `DEEP_LINK` : um link para uma página específica em uma aplicação. • `CLOSE`: ignora a mensagem.	
`Link`	O URL de destino de um botão. Não está presente nos botões em que o ButtonAction está `CLOSE`.	
`Text`	O texto que aparece no botão.	
`TextColor`	A cor do texto no botão, expressa como uma cadeia de caracteres que contém um código de cor hexadecimal (como “#000000” para preto).	

Objeto **`OverrideButtonConfig`**

O objeto `OverrideButtonConfig` só estará presente se o modelo de mensagem no aplicativo usar botões de substituição. Um botão de substituição é um botão que tem uma configuração específica

para um determinado tipo de dispositivo, como um dispositivo iOS, um dispositivo Android ou um navegador da web.

Um objeto `OverrideButtonConfig` contém os seguintes atributos:

Atributo	Descrição	Onde é definido
`ButtonAction`	A ação que ocorre quando um destinatário escolhe um botão em uma mensagem no aplicativo. Os valores possíveis são: • `LINK`: um link para um destino na Web. • `DEEP_LINK` : um link para uma página específica em uma aplicação. • `CLOSE`: ignora a mensagem.	Com base no conteúdo do modelo de mensagem no aplicativo que foi especificado para a campanha.
`Link`	O URL de destino de um botão. Não está presente nos botões em que `ButtonAction` está `CLOSE`.	
`Text`	O texto que aparece no botão.	
`TextColor`	A cor do texto no botão, expressa como uma cadeia de caracteres que contém um código de cor hexadecimal (como “#000000” para preto).	

Validar números de telefone no Amazon Pinpoint

O Amazon Pinpoint inclui um serviço de validação de números de telefone que você pode usar para determinar se um número de telefone é válido, além de obter informações adicionais sobre o próprio número de telefone. Por exemplo, quando você usa o serviço de validação de números de telefone, ele retorna as seguintes informações:

- O número de telefone em formato E.164.
- O tipo de número de telefone (como celular, fixo ou VoIP).
- A cidade e o país onde o número telefone está localizado.
- A operadora associada ao número de telefone.

Há um custo adicional pelo uso do serviço de validação de números de telefone. Para mais informações, consulte Definição de preços do Amazon Pinpoint.

> **Important**
>
> Para números de telefone originados nos Estados Unidos e Canadá, a API de validação de número de telefone não retornará mais dados de `City`, `County`, `Timezone` e `ZipCode`.

Casos de uso da validação de números de telefone

Você pode aplicar o serviço de validação de números de telefone a vários casos de uso, incluindo os seguintes:

- Verificação de números de telefone fornecidos em um formulário da Web: se você usa formulários na web para coletar informações de contato de seus clientes, é possível validar os números de telefone fornecidos pelos clientes antes do formulário ser enviado. Use o back-end do site para validar o número usando a API do Amazon Pinpoint. A resposta da API indica se o número é inválido, por exemplo, se o número de telefone estiver formatado incorretamente. Se você determinar que o número de telefone fornecido pelo cliente é inválido, seu formulário da Web poderá notificar o cliente para que ele forneça outro número.
- Limpeza de seu banco de dados de contatos: se você tem um banco de dados de números de telefone de clientes, é possível validar cada número de telefone e atualizar o banco de dados de

acordo com as descobertas. Por exemplo, caso encontre endpoints com números de telefone que não podem receber mensagens SMS, você poderá alterar a propriedade `ChannelType` do endpoint de `SMS` para `VOICE`. É possível validar o número de telefone primeiro e depois a propriedade `ChannelType` para endpoints novos ou existentes seguindo as instruções em Adicionar endpoints ao Amazon Pinpoint para um único endpoint ou Adição de um lote de endpoints ao Amazon Pinpoint para vários endpoints.

- Escolha do canal adequado antes de enviar a mensagem: se você pretende enviar uma mensagem SMS, mas constata que o número de destino é inválido, você pode enviar a mensagem ao destinatário por um canal diferente. Por exemplo, se o endpoint não puder receber mensagens SMS, você pode enviar uma mensagem de voz em vez disso.

Usar o serviço de validação de números de telefone

O exemplo a seguir mostra como validar um número de telefone usando o. AWS CLI Para obter mais informações, consulte phone-number-validatena Referência de AWS CLI Comandos. Por exemplo, respostas de validação, consulteRespostas da validação de números de telefone. Para obter mais informações sobre como configurar o AWS CLI, consulte Configurar o AWS CLI no Guia do AWS Command Line Interface Usuário.

Para usar o serviço de validação de número de telefone usando o AWS CLI

- Na linha de comando, insira o seguinte comando:

```
aws pinpoint phone-number-validate --number-validate-request
 PhoneNumber=+442079460881,IsoCountryCode=GB
```

No comando anterior, substitua *+442079460881* pelo número de telefone que você deseja validar e *GB* pelo código ISO de país ou região de dois dígitos.

> **Note**
>
> Você sempre deve incluir o código do país ao fornecer o número de telefone para o serviço de validação de números de telefone. Se você não incluir o código do país, o serviço poderá retornar informações de um número de telefone em outro país. Você pode ter traços no número de telefone, por exemplo, *+44-207-946-0881*.

Respostas da validação de números de telefone

As informações fornecidas pelo serviço de validação de números de telefone podem variar um pouco de acordo com os dados disponíveis para o número de telefone. Esta seção contém exemplos de respostas do serviço de validação de números de telefone.

> **Note**
>
> Os dados fornecidos pelo serviço de validação de números de telefone são baseados nas informações fornecidas por operadoras de telefonia e outras entidades em todo o mundo. As operadoras em alguns países podem atualizar essas informações com menos frequência do que em outros países. Por exemplo, se você emitir uma solicitação para validar um número de celular, e o número que você forneceu migrou de uma operadora para outra, a resposta do serviço de validação de números de telefone pode incluir o nome da operadora original, em vez de o atual.

Números de telefone celular válidos

Quando você envia uma solicitação para o serviço de validação de números de telefone, e o número de telefone é um celular válido, ele retorna informações semelhantes às do seguinte exemplo:

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp Mobile",
        "City": "Seattle",
        "CleansedPhoneNumberE164": "+12065550142",
        "CleansedPhoneNumberNational": "2065550142",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+12065550142",
        "PhoneType": "MOBILE",
        "PhoneTypeCode": 0,
        "Timezone": "America/Los_Angeles",
        "ZipCode": "98101"
    }
}
```

Números de telefone fixo válidos

Se a solicitação contiver um número de telefone fixo válido, o serviço de validação de números de telefone retornará informações semelhantes às do seguinte exemplo:

```
{
    "CountryCodeIso2": "US",
    "CountryCodeNumeric": "1",
    "Country": "United States",
    "City": "Santa Clara",
    "ZipCode": "95037",
    "Timezone": "America/Los_Angeles",
    "CleansedPhoneNumberNational": "4085550101",
    "CleansedPhoneNumberE164": "14085550101",
    "Carrier": "AnyCompany",
    "PhoneTypeCode": 1,
    "PhoneType": "LANDLINE",
    "OriginalPhoneNumber": "+14085550101"
}
```

Números de telefone VoIP válidos

Se a solicitação contiver um número de telefone VoIP (Voz sobre IP) válido, o serviço de validação de números de telefone retornará informações semelhantes às do seguinte exemplo:

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+441514960001",
        "CleansedPhoneNumberNational": "1514960001",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+441514960001",
        "PhoneType": "VOIP",
        "PhoneTypeCode": 2
    }
}
```

Números de telefone inválidos

Se a solicitação contiver um número de telefone inválido, o serviço de validação de números de telefone retornará informações semelhantes às do seguinte exemplo:

```
{
    "NumberValidateResponse": {
        "CleansedPhoneNumberE164": "+44163296076",
        "CleansedPhoneNumberNational": "163296076",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+440163296076",
        "PhoneType": "INVALID",
        "PhoneTypeCode": 3
    }
}
```

Observe que a propriedade `PhoneType` nessa resposta indica que esse número de telefone é `INVALID` e não inclui informações sobre a operadora ou a localização associadas ao número de telefone. Evite enviar mensagens SMS ou por voz para números de telefone cuja propriedade `PhoneType` é `INVALID`, pois é improvável que esses números pertençam a destinatários reais.

Outros números de telefone

Ocasionalmente, a resposta do serviço de validação de números de telefone inclui um valor `PhoneType` de `OTHER`. O serviço pode retornar esse tipo de resposta nas seguintes situações:

- O número de telefone é gratuito (público).
- O número de telefone está reservado para uso em programas de TV e filmes, como os números de telefone da América do Norte que começam com 555.
- O número de telefone inclui um código de área que não está em uso no momento, como o código de área 999 na América do Norte.
- O número de telefone está reservado para qualquer outra finalidade.

O exemplo a seguir mostra a resposta que o serviço de validação de números de telefone fornece quando a solicitação inclui um número de telefone norte-americano fictício:

```
{
    "NumberValidateResponse": {
        "Carrier": "Multiple OCN Listing",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
```

```
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "OTHER",
        "PhoneTypeCode": 4,
        "Timezone": "America/Los_Angeles"
    }
}
```

Números de telefone pré-pagos

Se a solicitação contiver um número de telefone pré-pago válido, o serviço de validação de números de telefone retornará informações semelhantes às do seguinte exemplo:

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "PREPAID",
        "PhoneTypeCode": 5
    }
}
```

Para obter mais informações sobre as informações contidas nessas respostas, consulte Validação de número de telefone na Referência de API do Amazon Pinpoint.

Enviar mensagens transacionais dos seus aplicativos

Você pode usar a API do Amazon Pinpoint e os AWS SDKs para enviar mensagens transacionais diretamente de seus aplicativos. Mensagens transacionais são mensagens que você envia para destinatários específicos, ao contrário das mensagens enviadas aos segmentos. Há várias razões pelas quais você pode querer enviar mensagens transacionais em vez de mensagens baseadas em campanhas. Por exemplo, você pode enviar uma confirmação de pedido por e-mail quando um cliente fizer um pedido. Você também pode enviar uma senha única por SMS ou voz que um cliente pode usar para concluir o processo de criação de uma conta para seu serviço.

Esta seção inclui exemplos de código em várias linguagens de programação que você pode usar para iniciar o envio de e-mails transacionais, mensagens SMS e mensagens de voz.

Tópicos nesta seção:

- Enviar mensagens de e-mail transacionais
- Enviar mensagens SMS
- Enviar mensagens de voz
- Enviar notificações por push

Enviar mensagens de e-mail transacionais

Esta seção fornece exemplos de código completos que você pode usar para enviar mensagens de e-mail transacionais por meio do Amazon Pinpoint:

- Usando a SendMessages operação na API do Amazon Pinpoint: você pode usar a `SendMessages` operação na API do Amazon Pinpoint para enviar mensagens em todos os canais compatíveis com o Amazon Pinpoint, incluindo os canais de notificação push, SMS, voz e e-mail.

  A vantagem de usar essa operação é que a sintaxe de solicitação para o envio de mensagens é muito semelhante em todos os canais. Isso facilita a redefinição de seu código existente. A operação `SendMessages` também permite substituir conteúdo em suas mensagens de e-mail e permite enviar e-mails para IDs de endpoint do Amazon Pinpoint em vez de endereços de e-mail específicos.

Esta seção inclui um código de exemplo em várias linguagens de programação que você pode usar para iniciar o envio de e-mails transacionais.

Tópicos nesta seção:

- Escolher um método de envio de e-mail
- Escolher entre o Amazon Pinpoint e o Amazon Simple Email Service (SES)
- Envie e-mails usando a API do Amazon Pinpoint
- Enviando um e-mail com cabeçalhos de cancelamento de inscrição

Escolher um método de envio de e-mail

O melhor método a ser usado para enviar e-mails transacionais depende do seu caso de uso. Por exemplo, se você precisar enviar e-mails usando um aplicativo de terceiros ou se não houver um AWS SDK disponível para sua linguagem de programação, talvez seja necessário usar a interface SMTP. Caso você queira enviar mensagens em outros canais compatíveis com o Amazon Pinpoint e queira usar um código consistente para fazer essas solicitações, use a operação `SendMessages` na API do Amazon Pinpoint.

Escolher entre o Amazon Pinpoint e o Amazon Simple Email Service (SES)

Se você enviar um grande número de e-mails transacionais, como confirmações de compra ou mensagens de redefinição de senha, considere usar o Amazon SES. O Amazon SES tem uma API e uma interface SMTP, ambas adequadas para enviar e-mails de seus aplicativos ou serviços. Ele também oferece recursos adicionais de e-mail, incluindo recursos de recebimento de e-mail, conjuntos de configuração e recursos de autorização de envio.

O Amazon SES também inclui uma interface SMTP que você pode integrar aos seus aplicativos de terceiros existentes, incluindo serviços de gerenciamento de relacionamento com o cliente (CRM), como o Salesforce. Para obter mais informações sobre como enviar e-mail usando o Amazon SES, consulte o Guia do desenvolvedor do Amazon Simple Email Service.

Envie e-mails usando a API do Amazon Pinpoint

Esta seção contém exemplos de código completos que você pode usar para enviar e-mails por meio da API do Amazon Pinpoint usando um AWS SDK.

C#

Use esse exemplo para enviar e-mail usando o AWS SDK for .NET. Esse exemplo pressupõe que você já tenha instalado e configurado o AWS SDK for .NET. Para obter mais informações,

consulte Conceitos básicos do AWS SDK for .NET no Guia do desenvolvedor do AWS SDK for .NET .

Esse exemplo 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. Para obter mais informações, consulte Configurar credenciais da AWS no Guia do desenvolvedor do AWS SDK for .NET .

Esse exemplo de código foi testado usando a AWS SDK for .NET versão 3.3.29.13 e a versão 2.1.2 do runtime do.NET Core.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;
```

```
        // The Amazon Pinpoint project/application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<MessageResponse> SendEmailMessage(
        string region, string appId, string toAddress, string senderAddress)
    {
        var client = new
 AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        // The subject line of the email.
        string subject = "Amazon Pinpoint Email test";

        // The body of the email for recipients whose email clients don't
        // support HTML content.
        string textBody = @"Amazon Pinpoint Email Test (.NET)"
                          + "\n---------------------------------"
                          + "\nThis email was sent using the Amazon Pinpoint API
 using the AWS SDK for .NET.";

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK for .NET)</
h1>"
                          + "\n  <p>This email was sent using the "
                          + "\n    <a href='https://aws.amazon.com/pinpoint/'>Amazon
 Pinpoint</a> API "
```

```
                            + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                            + "\n  </p>"
                            + "\n</body>"
                            + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
                },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
                            },
                            Subject = new SimpleEmailPart
                            {
```

```
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
        return response.MessageResponse;
    }
}
```

Java

Use esse exemplo para enviar e-mail usando o AWS SDK for Java. Esse exemplo pressupõe que você já tenha instalado e configurado o AWS SDK for Java 2.x. Para obter mais informações, consulte Conceitos básicos no Guia do desenvolvedor do AWS SDK for Java 2.x .

Esse exemplo 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. Para mais informações, consulte Configurar as credenciais padrão e a Região no Guia do desenvolvedor do AWS SDK for Java .

Esse exemplo de código foi testado usando a AWS SDK for Java versão 2.3.1 e a versão 11.0.1 do OpenJDK.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
```

```
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendEmailMessage {

        // The character encoding the you want to use for the subject line and
        // message body of the email.
```

```
        public static String charset = "UTF-8";

   // The body of the email for recipients whose email clients support HTML
content.
   static final String body = """
       Amazon Pinpoint test (AWS SDK for Java 2.x)

       This email was sent through the Amazon Pinpoint Email API using the AWS SDK
for Java 2.x

       """;

       public static void main(String[] args) {
              final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
<toAddress>

           Where:
              subject - The email subject to use.
              senderAddress - The from address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
              toAddress - The to address. This address has to be verified in Amazon
Pinpoint in the region you're using to send email\s
           """;

       if (args.length != 3) {
           System.out.println(usage);
           System.exit(1);
       }

       String subject = args[0];
       String senderAddress = args[1];
       String toAddress = args[2];
       System.out.println("Sending a message");
       PinpointEmailClient pinpoint = PinpointEmailClient.builder()
           .region(Region.US_EAST_1)
           .build();

       sendEmail(pinpoint, subject, senderAddress, toAddress);
       System.out.println("Email was sent");
       pinpoint.close();
   }
```

```
    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
 subject, String senderAddress, String toAddress) {
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Para ver o exemplo completo do SDK, consulte SendEmailMessage.java on. GitHub

JavaScript (Node.js)

Use este exemplo para enviar e-mails usando o AWS SDK para JavaScript em Node.js. Este exemplo pressupõe que você já tenha instalado e configurado o SDK para JavaScript o Node.js. Para obter mais informações, consulte Introdução ao AWS SDK ou JavaScript no Guia do desenvolvedor do Node.js.

Esse exemplo 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. Para obter mais informações, consulte Configuração de credenciais no AWS SDK para JavaScript o Guia do Desenvolvedor do Node.js.

Esse exemplo de código foi testado usando o SDK para JavaScript Node.js versão 2.388.0 e Node.js versão 11.7.0.

```
"use strict";

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

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
```

```
---------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
```

```
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
  }
});
```

Python

Use esse exemplo para enviar e-mail usando o AWS SDK for Python (Boto3). Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para Python (Boto3). Para obter mais informações, consulte Início rápido na Referência da API do AWS SDK para Python (Boto3).

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

```
logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
 account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and message
                     body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
 can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
```

```
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data": html_message},
                            "TextPart": {"Charset": char_set, "Data": text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -------------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
```

```
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Você também pode usar modelos de mensagem para enviar mensagens de e-mail, conforme mostrado no exemplo a seguir:

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

logger = logging.getLogger(__name__)


def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
                         account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.
```

```
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress": sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
```

```
            template_version,
        )
        print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Esses exemplos pressupõem 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. Para obter mais informações, consulte Credenciais na Referência da API do AWS SDK para Python (Boto3).

Enviando um e-mail com cabeçalhos de cancelamento de inscrição

> **Note**
>
> Antes de usar cabeçalhos de e-mail, você deve configurar uma função de envio de orquestração de e-mail se estiver enviando e-mails de uma campanha ou de uma jornada. Para envio direto de e-mail, você deve ter permissões para `ses:SendEmail` `ses:SendRawEmail` e. Para obter mais informações, consulte Criação de uma função de envio de orquestração de e-mail no Guia do usuário do Amazon Pinpoint.

Incluir um link de cancelamento de inscrição no e-mail é uma prática recomendada e, em alguns países, é uma exigência legal. Para adicionar um link de cancelamento de inscrição com um clique, adicione os seguintes cabeçalhos:

1. Defina o nome do cabeçalho `List-Unsubscribe` e defina o valor no link de cancelamento da assinatura. O link deve oferecer suporte a solicitações HTTP POST para processar a solicitação de cancelamento de assinatura do destinatário.
2. Defina o nome do cabeçalho como `List-Unsubscribe-Post` e defina o valor como`List-Unsubscribe=One-Click`.

Você pode adicionar até 15 cabeçalhos a uma mensagem de e-mail. Para obter uma lista de cabeçalhos compatíveis, consulte os campos de cabeçalho do Amazon SES no Amazon Simple Email Service Developer Guide.

O exemplo a seguir mostra como enviar uma mensagem de e-mail com cabeçalhos de cancelamento de assinatura usando o. AWS Command Line Interface Para obter mais informações sobre como configurar o AWS CLI, consulte Configurar o AWS CLI no Guia do AWS Command Line Interface Usuário.

No comando a seguir, faça o seguinte:

- *AppId*Substitua pelo ID do seu aplicativo.
- Substitua *richard_roe@example.com* pelo endereço de e-mail do destinatário.
- Substitua *https://example.com/unsub* pelo seu link de cancelamento de inscrição.
- Substitua *example123456* por um identificador exclusivo para o destinatário.

```
aws pinpoint send-messages --application-id AppId --message-request '{
 "Addresses": {
   "richard_roe@example.com": {
     "ChannelType": "EMAIL"
   }
 },
 "MessageConfiguration": {
   "EmailMessage": {
     "Substitutions": {
       "url": [
         "https://example.com/unsub"
       ],
        "id1": [
          "/example123456"
        ]
     },
     "SimpleEmail": {
       "TextPart": {
         "Data": "Sample email message with an subscribe header",
         "Charset": "UTF-8"
       },
       "Subject": {
         "Data": "Hello",
         "Charset": "UTF-8"
       },
       "Headers": [
         {
           "Name": "List-Unsubscribe",
           "Value": "{{url}}{{id1}}"
```

```
          },
          {
            "Name": "List-Unsubscribe-Post",
            "Value": "List-Unsubscribe=One-Click"
          }
        ]
      }
    }
  }
}'
```

Enviar mensagens SMS

Você pode usar a API do Amazon Pinpoint para enviar mensagens SMS (mensagens de texto) para números de telefone específicos ou IDs de endpoint. Esta seção contém exemplos de código completos que você pode usar para enviar mensagens SMS por meio da API do Amazon Pinpoint usando um AWS SDK.

C#

Use esse exemplo para enviar uma mensagem SMS usando o AWS SDK for .NET. Esse exemplo pressupõe que você já tenha instalado e configurado o AWS SDK for .NET. Para obter mais informações, consulte Conceitos básicos no Guia do desenvolvedor do AWS SDK for .NET .

Esse exemplo 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 IAM. Para obter mais informações, consulte Configurar credenciais da AWS no Guia do desenvolvedor do AWS SDK for .NET .

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
```

```
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;

        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
```

```
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
{response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
ChannelType.SMS }
                        }
```

```
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

Java

Use esse exemplo para enviar uma mensagem SMS usando o AWS SDK for Java. Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para Java. Para obter mais informações, consulte Conceitos básicos no Guia do desenvolvedor do AWS SDK for Java .

Esse exemplo 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 IAM. Para mais informações, consulte Configurar as credenciais padrão e a Região no Guia do desenvolvedor do AWS SDK for Java .

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
```

```
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """
```

```
                                Usage:   <message> <appId> <originationNumber>
<destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application ID
to use when you send this message.
                                  originationNumber - The phone number or short code
that you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                                  destinationNumber - The recipient's phone number.
For best results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
                                  """;

              if (args.length != 4) {
                      System.out.println(usage);
                      System.exit(1);
              }

              String message = args[0];
              String appId = args[1];
              String originationNumber = args[2];
              String destinationNumber = args[3];
              System.out.println("Sending a message");
              PinpointClient pinpoint = PinpointClient.builder()
                              .region(Region.US_EAST_1)
                              .build();

              sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
              pinpoint.close();
       }

       public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                      String originationNumber,
                      String destinationNumber) {
              try {
                      Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                      AddressConfiguration addConfig =
AddressConfiguration.builder()
                                      .channelType(ChannelType.SMS)
```

```
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
 DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request = SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" + v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Para ver o exemplo completo do SDK, consulte SendMessage.java on. GitHub

JavaScript (Node.js)

Use este exemplo para enviar uma mensagem SMS usando o AWS SDK para JavaScript em Node.js. Este exemplo pressupõe que você já tenha instalado e configurado o SDK JavaScript em Node.js. Para obter mais informações, consulte Introdução ao AWS SDK ou JavaScript no Guia do desenvolvedor do Node.js.

Esse exemplo 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 IAM. Para obter mais informações, consulte Configuração de credenciais no AWS SDK para JavaScript o Guia do Desenvolvedor do Node.js.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";
```

```
// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
var senderId = "MySenderID";

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

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
```

```
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

Python

Use esse exemplo para enviar uma mensagem SMS usando o AWS SDK for Python (Boto3). Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para Python. Para obter mais informações, consulte Início rápido no AWS SDK for Python (Boto3) Getting Started.

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

logger = logging.getLogger(__name__)


def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
```

```
    :param app_id: The Amazon Pinpoint project/application ID to use when you send
                   this message. The SMS channel must be enabled for the project or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This phone
                               number must be associated with your Amazon Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you send
                         time-sensitive content, specify TRANSACTIONAL. If you send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK for
 "
        "Python (Boto 3)."
    )
```

```
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

Você também pode usar modelos de mensagem para enviar mensagens SMS, conforme mostrado no exemplo a seguir:

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

logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
```

```
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )

    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
```

```
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

Esses exemplos pressupõem que você esteja usando um arquivo de credenciais compartilhadas para especificar a chave de acesso e a chave de acesso secreta de um usuário do IAM existente. Para obter mais informações, consulte Credenciais na Referência da API do AWS SDK para Python (Boto3).

Enviar mensagens de voz

Você pode usar a API do Amazon Pinpoint para enviar mensagens de voz para números de telefone específicos. Esta seção contém exemplos de código completos que você pode usar para enviar mensagens de voz por meio da API de SMS e voz do Amazon Pinpoint usando um AWS SDK.

Java

Use esse exemplo para enviar uma mensagem de voz usando o AWS SDK for Java. Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para Java. Para obter mais informações, consulte Conceitos básicos no Guia do desenvolvedor do AWS SDK for Java.

Esse exemplo 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. Para mais informações, consulte Configurar as credenciais e a Região da AWS para desenvolvimento no Guia do desenvolvedor do AWS SDK for Java.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
```

```
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendVoiceMessage {

        // The Amazon Polly voice that you want to use to send the message. For a
 list
        // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
        static final String voiceName = "Matthew";

        // The language to use when sending the message. For a list of supported
        // languages, see
        // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
        static final String languageCode = "en-US";
```

```
        // The content of the message. This example uses SSML to customize and
control
        // certain aspects of the message, such as by adding pauses and changing
        // phonation. The message can't contain any line breaks.
        static final String ssmlMessage = "<speak>This is a test message sent from "
                        + "<emphasis>Amazon Pinpoint</emphasis> "
                        + "using the <break strength='weak'/>AWS "
                        + "SDK for Java. "
                        + "<amazon:effect phonation='soft'>Thank "
                        + "you for listening.</amazon:effect></speak>";

        public static void main(String[] args) {

                final String usage = """

                                Usage:   <originationNumber> <destinationNumber>\s

                                Where:
                                  originationNumber - The phone number or short code
that you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                                  destinationNumber - The recipient's phone number.
For best results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
                                  """;

                if (args.length != 2) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String originationNumber = args[0];
                String destinationNumber = args[1];
                System.out.println("Sending a voice message");

                // Set the content type to application/json.
                List<String> listVal = new ArrayList<>();
                listVal.add("application/json");
                Map<String, List<String>> values = new HashMap<>();
                values.put("Content-Type", listVal);

                ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()
                                .headers(values)
```

```
                                .build();

                PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
                                .overrideConfiguration(config2)
                                .region(Region.US_EAST_1)
                                .build();

                sendVoiceMsg(client, originationNumber, destinationNumber);
                client.close();
        }

        public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
 originationNumber,
                        String destinationNumber) {
                try {
                        SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                                        .languageCode(languageCode)
                                        .text(ssmlMessage)
                                        .voiceId(voiceName)
                                        .build();

                        VoiceMessageContent content = VoiceMessageContent.builder()
                                        .ssmlMessage(ssmlMessageType)
                                        .build();

                        SendVoiceMessageRequest voiceMessageRequest =
 SendVoiceMessageRequest.builder()
                                        .destinationPhoneNumber(destinationNumber)
                                        .originationPhoneNumber(originationNumber)
                                        .content(content)
                                        .build();

                        client.sendVoiceMessage(voiceMessageRequest);
                        System.out.println("The message was sent successfully.");

                } catch (PinpointSmsVoiceException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Para ver o exemplo completo do SDK, consulte SendVoiceMessage.java no GitHub.

JavaScript (Node.js)

Use este exemplo para enviar uma mensagem de voz usando o AWS SDK para JavaScript no Node.js. Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para JavaScript no Node.js.

Esse exemplo 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. Para obter mais informações, consulte Definir credenciais no Node.js no Guia do desenvolvedor do AWS SDK para JavaScript no Node.js.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best results,
 you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
```

```
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

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

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
```

```
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

Python

Use esse exemplo para enviar uma mensagem de voz usando o AWS SDK for Python (Boto3). Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para Python (Boto3).

Esse exemplo 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. Para obter mais informações, consulte Credenciais na Referência da API do AWS SDK para Python (Boto3).

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

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your Amazon
                      Pinpoint account and be in E.164 format.
```

```
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
 message.
    :param ssml_message: The content of the message. This example uses SSML to
 control
                         certain aspects of the message, such as the volume and the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
```

```
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

Enviar notificações por push

A API do Amazon Pinpoint pode enviar notificações por push transacionais para identificadores de dispositivo específicos. Esta seção contém exemplos de código completos que podem ser usados para enviar notificações por push por meio da API do Amazon Pinpoint usando um AWS SDK.

É possível usar esses exemplos para enviar notificações por push por meio de qualquer serviço de notificações por push compatível com o Amazon Pinpoint. No momento, o Amazon Pinpoint é compatível com os seguintes canais: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push e Amazon Device Messaging (ADM).

> **Note**
>
> Ao enviar notificações por push por meio do serviço Firebase Cloud Messaging (FCM), use o nome do serviço `GCM` em sua chamada para a API do Amazon Pinpoint. O serviço Google Cloud Messaging (GCM) foi descontinuado pelo Google em 10 de abril de 2018. No entanto, a API do Amazon Pinpoint usa o nome do serviço `GCM` para mensagens que envia por meio

do serviço FCM para manter a compatibilidade com o código da API que foi escrito antes da descontinuação do serviço GCM.

JavaScript (Node.js)

Use este exemplo para enviar notificações por push usando o AWS SDK para JavaScript no Node.js. Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para 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 Definir credenciais no Node.js no Guia do desenvolvedor do AWS SDK para JavaScript no Node.js.

```
'use strict';

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

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const region = 'us-east-1';

// The title that appears at the top of the push notification.
var title = 'Test message sent from Amazon Pinpoint.';

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

// The Amazon Pinpoint project 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). Esse exemplo pressupõe que você já tenha instalado e configurado o SDK para Python (Boto3).

Esse exemplo também pressupõe que você esteja usando um arquivo de credenciais compartilhadas para especificar a chave de acesso e a chave de acesso secreta de um usuário existente do . Para obter mais informações, consulte Credenciais na Referência da API do AWS SDK para Python (Boto3).

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

# The AWS Region that you want to use to send the message. For a list of
# AWS Regions where the Amazon Pinpoint API is available, see
# https://docs.aws.amazon.com/pinpoint/latest/apireference/
region = "us-east-1"

# The title that appears at the top of the push notification.
title = "Test message sent from Amazon Pinpoint."

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

# The Amazon Pinpoint project/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()
```

Criação de canais personalizados no Amazon Pinpoint

O Amazon Pinpoint inclui suporte integrado para o envio de mensagens por meio de canais de notificação por push, e-mail, SMS e voz. Também é possível configurar o Amazon Pinpoint para enviar mensagens por meio de outros canais criando canais personalizados. Os canais personalizados no Amazon Pinpoint permitem que você envie mensagens por meio de qualquer serviço que tenha uma API, incluindo serviços de terceiros. É possível interagir com as APIs usando um webhook ou chamando uma função do AWS Lambda.

Os segmentos para os quais você enviar campanhas de canal personalizado poderão conter endpoints de todos os tipos (ou seja, endpoints em que o valor do atributo `ChannelType` seja EMAIL (E-mail), VOICE (Voz), SMS, CUSTOM (Personalizado) ou um dos vários tipos de endpoint de notificação por push).

Criar uma campanha que envia mensagens por meio de um canal personalizado

Para atribuir uma função do Lambda ou webhook a uma campanha específica, use a API do Amazon Pinpoint para criar ou atualizar um objeto Campanha.

O objeto `MessageConfiguration` na campanha também deve conter um objeto `CustomMessage`. Este objeto tem um membro, `Data`. O valor de `Data` é uma string JSON que contém a carga da mensagem que você deseja enviar para o canal personalizado.

A campanha deve conter um objeto `CustomDeliveryConfiguration`. No objeto `CustomDeliveryConfiguration`, especifique o seguinte:

- `EndpointTypes`: uma matriz que contém todos os tipos de endpoint para os quais a campanha de canal personalizado deve ser enviada. Ela pode conter qualquer um ou todos os seguintes tipos de canal:
  - `ADM`
  - `APNS`
  - `APNS_SANDBOX`
  - `APNS_VOIP`
  - `APNS_VOIP_SANDBOX`
  - `BAIDU`

- `CUSTOM`
- `EMAIL`
- `GCM`
- `SMS`
- `VOICE`

- `DeliveryUri`: o destino para o qual os endpoints são enviados. É possível especificar um dos seguintes:
  - O nome do recurso da Amazon (ARN) de uma função do Lambda que você deseja realizar quando a campanha for executada.
  - O URL do webhook para o qual você deseja enviar dados de endpoint quando a campanha for executada.

> **Note**
>
> O objeto `Campaign` também pode conter um objeto `Hook`. Esse objeto é usado somente para criar segmentos personalizados por uma função do Lambda quando uma campanha for executada. Para obter mais informações, consulte Personalizar segmentos com o AWS Lambda.

Noções básicas sobre os dados de eventos que o Amazon Pinpoint envia para os canais personalizados

Antes de criar uma função do Lambda que envia mensagens por um canal personalizado, familiarize-se com os dados que o Amazon Pinpoint emite. Quando uma campanha do Amazon Pinpoint envia mensagens por um canal personalizado, ela envia uma carga para a função do Lambda de destino semelhante ao seguinte exemplo:

```
{
  "Message":{},
  "Data":"The payload that's provided in the CustomMessage object in
 MessageConfiguration",
  "ApplicationId":"3a9b1f4e6c764ba7b031e7183example",
  "CampaignId":"13978104ce5d6017c72552257example",
  "TreatmentId":"0",
  "ActivityId":"575cb1929d5ba43e87e2478eeexample",
```

```
  "ScheduledTime":"2020-04-08T19:00:16.843Z",
  "Endpoints":{
    "1dbcd396df28ac6cf8c1c2b7fexample":{
      "ChannelType":"EMAIL",
      "Address":"mary.major@example.com",
      "EndpointStatus":"ACTIVE",
      "OptOut":"NONE",
      "Location":{
        "City":"Seattle",
        "Country":"USA"
      },
      "Demographic":{
        "Make":"OnePlus",
        "Platform":"android"
      },
      "EffectiveDate":"2020-04-01T01:05:17.267Z",
      "Attributes":{
        "CohortId":[
          "42"
        ]
      },
      "CreationDate":"2020-04-01T01:05:17.267Z"
    }
  }
}
```

Os dados de eventos fornecem os seguintes atributos:

- `ApplicationId`: o ID do projeto do Amazon Pinpoint ao qual a campanha pertence.
- `CampaignId`: o ID da campanha do Amazon Pinpoint que invocou a função do Lambda.
- `TreatmentId`: o ID da variante da campanha. Se você criou uma campanha padrão, esse valor será sempre 0. Se você criou uma campanha de teste A/B, esse valor será um número inteiro entre 0 e 4.
- `ActivityId`: o ID da atividade sendo executada pela campanha.
- `ScheduledTime`: a hora em que a campanha do Amazon Pinpoint foi executada, mostrada no formato ISO 8601.
- `Endpoints`: uma lista dos endpoints que foram alvo da campanha. Cada carga pode conter até 50 endpoints. Se o segmento para o qual a campanha foi enviada contiver mais de 50 endpoints, o Amazon Pinpoint invocará a função repetidamente, com até 50 endpoints por vez, até que todos os endpoints sejam processados.

É possível usar esses dados de exemplo ao criar e testar sua função do Lambda do canal personalizado.

Configurar webhooks

Se você usar um webhook para enviar mensagens de canal personalizado, o URL do webhook deve começar com "https://". O URL do webhook pode conter somente caracteres alfanuméricos, além dos seguintes símbolos: hífen (-), ponto (.), sublinhado (_), til (~), ponto de interrogação (?), barra (/), cerquilha ou hashtag (#) e ponto e vírgula (;). O URL deve estar em conformidade com RFC3986.

Quando você cria uma campanha que especifica o URL de um webhook, o Amazon Pinpoint emite um HTTP `HEAD` para esse URL. A resposta à solicitação `HEAD` deve conter um cabeçalho chamado `X-Amz-Pinpoint-AccountId`. O valor desse cabeçalho deve ser igual ao ID da sua conta da AWS.

Configurar as funções do Lambda

Esta seção fornece uma visão geral das etapas que devem ser seguidas ao criar uma função do Lambda que envia mensagens por um canal personalizado. Primeiro, crie a função. Depois disso, adicione uma política de execução à função. Essa política permite que o Amazon Pinpoint execute a política quando uma campanha for executada.

Para uma introdução à criação de funções do Lambda, consulte Como criar funções do Lambda no Guia do desenvolvedor do AWS Lambda.

Exemplo de função do Lambda

O exemplo de código a seguir processa a carga e registra o número de endpoints de cada tipo de endpoint no CloudWatch.

```
import boto3
import random
import pprint
import json
import time

cloudwatch = boto3.client('cloudwatch')

def lambda_handler(event, context):
    customEndpoints = 0
```

```
    smsEndpoints = 0
    pushEndpoints = 0
    emailEndpoints = 0
    voiceEndpoints = 0
    numEndpoints = len(event['Endpoints'])

    print("Payload:\n", event)
    print("Endpoints in payload: " + str(numEndpoints))

    for key in event['Endpoints'].keys():
        if event['Endpoints'][key]['ChannelType'] == "CUSTOM":
            customEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "SMS":
            smsEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "EMAIL":
            emailEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "VOICE":
            voiceEndpoints += 1
        else:
            pushEndpoints += 1

    response = cloudwatch.put_metric_data(
        MetricData = [
            {
                'MetricName': 'EndpointCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': len(event['Endpoints'])
            },
            {
                'MetricName': 'CustomCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
```

```
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': customEndpoints
        },
        {
            'MetricName': 'SMSCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': smsEndpoints
        },
        {
            'MetricName': 'EmailCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': emailEndpoints
        },
        {
            'MetricName': 'VoiceCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
```

```
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': voiceEndpoints
        },
        {
            'MetricName': 'PushCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': pushEndpoints
        },
        {
            'MetricName': 'EndpointCount',
            'Dimensions': [
            ],
            'Unit': 'None',
            'Value': len(event['Endpoints'])
        },
        {
            'MetricName': 'CustomCount',
            'Dimensions': [
            ],
            'Unit': 'None',
            'Value': customEndpoints
        },
        {
            'MetricName': 'SMSCount',
            'Dimensions': [
            ],
            'Unit': 'None',
```

```
                'Value': smsEndpoints
            },
            {
                'MetricName': 'EmailCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': emailEndpoints
            },
            {
                'MetricName': 'VoiceCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': voiceEndpoints
            },
            {
                'MetricName': 'PushCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': pushEndpoints
            }
        ],
        Namespace = 'PinpointCustomChannelExecution'
    )
    print("cloudwatchResponse:\n",response)
```

Quando uma campanha do Amazon Pinpoint executa essa função do Lambda, o Amazon Pinpoint envia à função uma lista de membros do segmento. A função conta o número de endpoints de cada `ChannelType`. Em seguida, envia esses dados para o Amazon CloudWatch. É possível visualizar essas métricas na seção Métricas do console do CloudWatch. As métricas estão disponíveis no namespace PinpointCustomChannelExecution.

É possível modificar esse exemplo de código para que ele também se conecte à API de um serviço externo para enviar mensagens por meio desse serviço.

Formato de resposta da função do Lambda para o Amazon Pinpoint

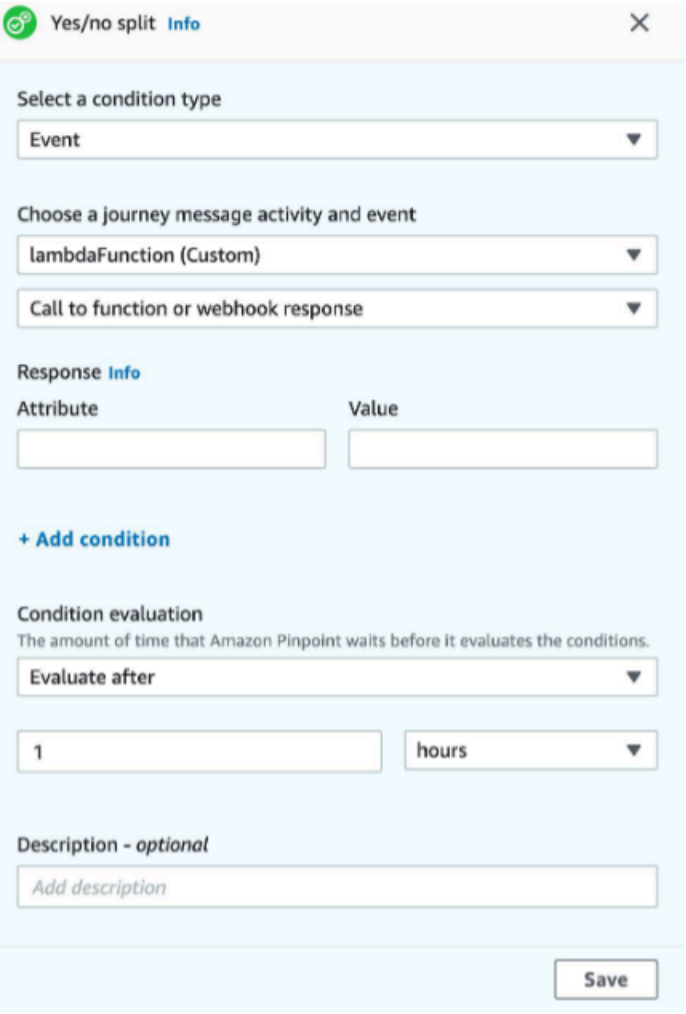
Se você quiser usar a jornada multivariada ou a divisão sim/não para determinar o caminho do endpoint após uma atividade de canal personalizada, você deve estruturar sua resposta da função

do Lambda em um formato que o Amazon Pinpoint possa entender e, depois, enviar os endpoints pelo caminho correto.

A estrutura da resposta deve estar no seguinte formato:

```
{
    <Endpoint ID 1>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
    <Endpoint ID 2>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
...
}
```

Isso permitirá que você selecione uma chave e um valor para determinar o caminho dos endpoints.



Conceder permissão ao Amazon Pinpoint para invocar a função do Lambda

Você pode usar a AWS Command Line Interface (AWS CLI) para adicionar permissões à política de função do Lambda atribuída à função do Lambda. Para permitir que o Amazon Pinpoint invoque uma função, use o comando do Lambda add-permission, conforme mostrado no exemplo a seguir:

```
aws lambda add-permission \
--function-name myFunction \
--statement-id sid0 \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:111122223333:apps/* 
--source-account 111122223333
```

No comando anterior, faça o seguinte:

- Substitua *myFunction* pelo nome da função do Lambda.
- Substitua *us-east-1* pela Região da AWS em que você usa o Amazon Pinpoint.
- Substitua *111122223333* pelo ID de sua conta da AWS.

Ao executar o comando `add-permission`, o Lambda retornará o seguinte resultado:

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:myFunction\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/*\"}},
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
}
```

O valor `Statement` é uma versão em cadeia de caracteres JSON de uma instrução adicionada à política de função do Lambda.

Restringir ainda mais a política de execução

É possível modificar a política de execução restringindo-a a um projeto do Amazon Pinpoint específico. Para fazer isso, substitua o * no exemplo anterior pelo ID exclusivo do projeto. É possível restringir ainda mais a política limitando-a a uma campanha específica. Por exemplo, para restringir a política para permitir somente uma campanha com o ID de campanha `95fee4cd1d7f5cd67987c1436example` em um projeto com o ID de projeto `dbaf6ec2226f0a9a8615e3ea5example`, use o seguinte valor para o atributo `source-arn`:

```
arn:aws:mobiletargeting:us-east-1:111122223333:apps/dbaf6ec2226f0a9a8615e3ea5example/
campaigns/95fee4cd1d7f5cd67987c1436example
```

> **Note**
>
> Se restringir a execução da função do Lambda a uma campanha específica, primeiramente será necessário criar a função com uma política menos restritiva. Depois, você terá de criar a campanha no Amazon Pinpoint e escolher a função. Por fim, atualize a política de execução para fazer referência à campanha especificada.

Transmissão de eventos do Amazon Pinpoint para o Kinesis

No Amazon Pinpoint, um evento é uma ação que ocorre quando: um usuário interage com um de seus aplicativos; quando você envia uma mensagem de uma campanha ou jornada; ou quando você envia uma mensagem transacional SMS ou de e-mail. Por exemplo, se enviar uma mensagem de e-mail, ocorrem vários eventos:

- Quando envia a mensagem, ocorre um evento de envio .
- Quando a mensagem chega à caixa de entrada do destinatário, ocorre um evento de entrega .
- Quando o destinatário abre a mensagem, ocorre um evento abertura .

Você pode configurar o Amazon Pinpoint para enviar informações sobre eventos para o Amazon Kinesis. A plataforma Kinesis oferece serviços que você pode usar para coletar, processar e analisar dados de AWS serviços em tempo real. O Amazon Pinpoint pode enviar dados de eventos para o Firehose, que transmite esses AWS dados para datastores como Amazon S3 ou Amazon Redshift. O Amazon Pinpoint também pode transmitir dados para o Kinesis Data Streams, que ingere e armazena vários fluxos de dados para processamento por aplicativos de análise.

A transmissão de evento do Amazon Pinpoint inclui informações sobre interações dos usuários com seus aplicativos (apps) conectados ao Amazon Pinpoint. Ele também inclui informações sobre todas as mensagens de campanhas que você envia, seja através de um canal e de jornadas. Isso também pode incluir qualquer evento personalizado que você tenha definido. Por fim, inclui informações sobre todas as mensagens de SMS e e-mail transacionais enviadas por você.

> **Note**
>
> O Amazon Pinpoint não transmite informações sobre mensagens de voz ou notificações por push transacionais.

Este capítulo fornece informações sobre como configurar o Amazon Pinpoint para transmitir dados de eventos para o Kinesis. Ela também contém exemplos dos dados de eventos que o Amazon Pinpoint transmite.

Tópicos

- Configurar streaming de eventos

- Eventos de aplicativo
- Eventos de campanha
- Eventos de jornada
- Eventos de e-mail
- Eventos de SMS

Configurar streaming de eventos

Você pode configurar o Amazon Pinpoint para enviar dados de eventos para um stream do Amazon Kinesis ou para um stream de entrega do Amazon Data Firehose. O Amazon Pinpoint pode enviar dados de eventos para campanhas, viagens e mensagens de e-mail e SMS transacionais.

Esta seção inclui informações sobre como configurar os eventos de streaming de forma programática. Também é possível usar o console do Amazon Pinpoint para configurar a transmissão de eventos. Para obter informações sobre como configurar a transmissão de eventos usando o console do Amazon Pinpoint, consulte Configurações de transmissão de eventos no Guia do usuário do Amazon Pinpoint.

Pré-requisitos

Os exemplos nesta seção exigem a seguinte entrada:

- O ID de um aplicativo que está integrado com o Amazon Pinpoint e eventos de relatório. Para obter informações sobre como integrar, leia Integrar o Amazon Pinpoint ao aplicativo.
- O Amazon Resource Name (ARN) de um stream do Kinesis ou stream de entrega do Firehose em sua conta. AWS Para obter informações sobre a criação desses recursos, consulte Criação e gerenciamento de streams no Guia do desenvolvedor do Amazon Kinesis Data Streams ou Criação de um stream de entrega do Amazon Data Firehose no Guia do desenvolvedor do Amazon Data Firehose.
- O ARN de uma função AWS Identity and Access Management (IAM) que autoriza o Amazon Pinpoint a enviar dados para o stream. Para obter informações sobre como criar uma função, leia Perfil do IAM para transmissão de eventos para o Kinesis.

AWS CLI

O AWS CLI exemplo a seguir usa o put-event-streamcomando. Este comando configura o Amazon Pinpoint para enviar eventos para um fluxo do Kinesis:

```
aws pinpoint put-event-stream \
--application-id projectId \
--write-event-stream DestinationStreamArn=streamArn,RoleArn=roleArn
```

AWS SDK for Java

O exemplo de Java a seguir configura o Amazon Pinpoint para enviar eventos para um fluxo do Kinesis:

```
public PutEventStreamResult createEventStream(AmazonPinpoint pinClient,
        String appId, String streamArn, String roleArn) {

    WriteEventStream stream = new WriteEventStream()
            .withDestinationStreamArn(streamArn)
            .withRoleArn(roleArn);

    PutEventStreamRequest request = new PutEventStreamRequest()
            .withApplicationId(appId)
            .withWriteEventStream(stream);

    return pinClient.putEventStream(request);
}
```

Este exemplo estrutura um objeto `WriteEventStream` que armazena os ARNs do fluxo do Kinesis e do perfil do IAM. O objeto `WriteEventStream` é passado para um objeto `PutEventStreamRequest` para configurar o Amazon Pinpoint para enviar fluxos de eventos a um aplicativo específico. O objeto `PutEventStreamRequest` é passado para o método `putEventStream` do cliente Amazon Pinpoint.

Um fluxo do Kinesis pode ser atribuído a vários aplicativos. Se você fizer isso, o Amazon Pinpoint enviará dados de eventos codificados em base64 de cada aplicativo para o fluxo, o que permite analisar os dados como uma coleção. O método do exemplo a seguir aceita uma lista de IDs de aplicativo (app) e usa o método do exemplo anterior, `createEventStream`, para atribuir um stream para cada aplicativo:

```
public List<PutEventStreamResult> createEventStreamFromAppList(
        AmazonPinpoint pinClient, List<String> appIDs,
        String streamArn, String roleArn) {

    return appIDs.stream()
            .map(appId -> createEventStream(pinClient, appId, streamArn,
                    roleArn))
            .collect(Collectors.toList());
}
```

Apesar ser possível atribuir um fluxo a vários aplicativos, não é possível atribuir vários fluxos a um mesmo aplicativo.

Desabilitar o streaming de eventos

Se você atribuiu um fluxo do Kinesis a um aplicativo, pode desabilitar a transmissão de eventos para esse aplicativo. O Amazon Pinpoint interrompe a transmissão dos eventos para o Kinesis, mas você pode visualizar a análise de eventos usando o console do Amazon Pinpoint.

AWS CLI

Use o comando delete-event-stream:

```
aws pinpoint delete-event-stream --application-id application-id
```

AWS SDK for Java

Use o deleteEventStreammétodo do cliente Amazon Pinpoint:

```
pinClient.deleteEventStream(new DeleteEventStreamRequest().withApplicationId(appId));
```

Eventos de aplicativo

Depois de integrar seu aplicativo (app) com o Amazon Pinpoint, o Amazon Pinpoint pode transmitir dados de eventos sobre a atividade do usuário, eventos personalizados e entregas de mensagens para o app.

Exemplo

O objeto JSON de um evento de aplicativo contém os dados mostrados no exemplo a seguir.

```
{
  "event_type": "_session.stop",
  "event_timestamp": 1487973802507,
  "arrival_timestamp": 1487973803515,
  "event_version": "3.0",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "cognito_identity_pool_id": "us-east-1:a1b2c3d4-e5f6-g7h8-i9j0-k1l2m3n4o5p6",
    "package_name": "main.page",
    "sdk": {
      "name": "aws-sdk-mobile-analytics-js",
      "version": "0.9.1:2.4.8"
    },
    "title": "title",
    "version_name": "1.0",
    "version_code": "1"
  },
  "client": {
    "client_id": "m3n4o5p6-a1b2-c3d4-e5f6-g7h8i9j0k1l2",
    "cognito_id": "us-east-1:i9j0k1l2-m3n4-o5p6-a1b2-c3d4e5f6g7h8"
  },
  "device": {
    "locale": {
      "code": "en_US",
      "country": "US",
      "language": "en"
    },
    "make": "generic web browser",
    "model": "Unknown",
    "platform": {
      "name": "android",
      "version": "10.10"
    }
  },
  "session": {
    "session_id": "f549dea9-1090-945d-c3d1-e4967example",
    "start_timestamp": 1487973202531,
    "stop_timestamp": 1487973802507
  },
  "attributes": {},
  "metrics": {}
}
```

Atributos de eventos do aplicativo

Esta seção define os atributos incluídos nos eventos de streaming do aplicativo.

Atributo	Descrição
`event_type`	O tipo de evento. Os valores possíveis são: • _session.start: o endpoint iniciou uma nova sessão. • _session.stop: o endpoint encerrou uma sessão. • _userauth.sign_in: o endpoint fez login no aplicativo. • _userauth.sign_up: um novo endpoint concluiu o processo de registro no aplicativo. • _userauth.auth_fail: o endpoint tentou fazer login em seu aplicativo, mas não conseguiu concluir o processo. • _monetization.purchase: o endpoint fez uma compra em seu aplicativo. • _session.pause: o endpoint pausou uma sessão. As sessões pausadas podem ser retomadas para que seja possível continuar a coletar métricas sem iniciar uma sessão totalmente nova. • _session.resume: o endpoint retomou uma sessão.
`event_timestamp`	A hora em que o evento foi relatado, mostrada como horário do Unix em milissegundos.
`arrival_timestamp`	A hora em que o evento foi recebido pelo Amazon Pinpoint, mostrada como horário do Unix em milissegundos.
`event_version`	A versão do esquema do evento JSON.

Atributo	Descrição
	Tip Marque esta versão no seu aplicativo de processamento de eventos para que você saiba quando atualizar o aplicativo em resposta a uma atualização de esquema.
`application`	Informações sobre o projeto do Amazon Pinpoint associado ao evento. Para obter mais informações, consulte a tabela Aplicativo.
`client`	Informações sobre o endpoint que relatou o evento. Para obter mais informações, consulte a tabela Cliente.
`device`	Informações sobre o dispositivo que relatou o evento. Para obter mais informações, consulte a tabela Dispositivo.
`session`	Informações sobre a sessão que gerou o evento. Consulte a tabela Sessão para obter mais informações.
`attributes`	Atributos associados ao evento. Para eventos relatados por seus aplicativos, esse objeto inclui atributos personalizados definidos por você.
`metrics`	Métricas relacionadas ao evento. Também é possível configurar seus aplicativos para enviar métricas personalizadas ao Amazon Pinpoint.

Aplicação

Inclui informações sobre o projeto do Amazon Pinpoint ao qual o evento está associado.

Atributo	Descrição
`app_id`	O ID exclusivo do projeto do Amazon Pinpoint que relatou o evento.
`cognito_identity_pool_id`	O ID do banco de identidades do Amazon Cognito ao qual o endpoint está associado.
`package_name`	O nome do pacote do aplicativo, como, por exemplo, `com.example.my_app` .
`sdk`	Informações sobre o SDK usado para relatar o evento. Para obter mais informações, consulte a tabela SDK.
`title`	O nome do app.
`version_name`	O nome da versão do aplicativo, como `V2.5`.
`version_code`	O número da versão do aplicativo, como `3`.

SDK

Inclui informações sobre o SDK usado para relatar o evento.

Atributo	Descrição
`name`	O nome do SDK usado para relatar o evento.
`version`	A versão do SDK.

Cliente

Inclui informações sobre o endpoint que gerou o evento.

Atributo	Descrição
`client_id`	O ID do endpoint.

Atributo	Descrição
`cognito_id`	O token do ID do Amazon Cognito associado ao endpoint.

Dispositivo

Inclui informações sobre o dispositivo do endpoint que gerou o evento.

Atributo	Descrição
`locale`	Informações sobre configurações de região e idioma do dispositivo do endpoint. Consulte a tabela Local para obter mais informações.
`make`	O fabricante do dispositivo do endpoint.
`model`	O identificador de modelo do dispositivo do endpoint.
`platform`	Informações sobre o sistema operacional no dispositivo do endpoint. Consulte a tabela Plataforma para obter mais informações.

Locale

Inclui informações sobre as configurações de região e idioma do dispositivo do endpoint.

Atributo	Descrição
`code`	O identificador do local associado ao dispositivo.
`country`	O país ou a região associada ao local do dispositivo.
`language`	O idioma associado ao local do dispositivo.

Plataforma

Inclui informações sobre o sistema operacional no dispositivo do endpoint.

Atributo	Descrição
`name`	O nome do sistema operacional no dispositivo.
`version`	A versão do sistema operacional no dispositivo.

Sessão

Inclui informações sobre a sessão que gerou o evento.

Atributo	Descrição
`session_id`	Um ID exclusivo que identifica a sessão.
`start_timestamp`	A data e a hora em que a sessão começou, mostradas como horário do Unix em milissegundos.
`stop_timestamp`	A data e a hora em que a sessão terminou, mostradas como horário do Unix em milissegundos.

Eventos de campanha

Se o Amazon Pinpoint for usado para enviar campanhas através de qualquer canal, o Amazon Pinpoint pode transmitir dados de eventos sobre essas campanhas. Isso inclui dados de eventos para quaisquer mensagens de email ou SMS enviadas de uma campanha. Para obter informações detalhadas sobre os dados transmitidos pelo Amazon Pinpoint para esses tipos de mensagens, consulte the section called “Eventos de e-mail” e the section called “Eventos de SMS”.

Evento de exemplo

O objeto JSON para uma campanha contém os dados mostrados no exemplo a seguir.

```
{
  "event_type": "_campaign.send",
  "event_timestamp": 1562109497426,
  "arrival_timestamp": 1562109497494,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "d8dcf7c5-e81a-48ae-8313-f540cexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "treatment_id": "0",
    "campaign_activity_id": "5473285727f04865bc673e527example",
    "delivery_type": "GCM",
    "campaign_id": "4f8d6097c2e8400fa3081d875example",
    "campaign_send_status": "SUCCESS"
  },
  "client_context": {
    "custom": {
      "endpoint": "{\"ChannelType\":\"GCM\",\"EndpointStatus\":\"ACTIVE\",
          #\"OptOut\":\"NONE\",\"RequestId\":\"ec229696-9d1e-11e9-8bf1-85d0aexample\",
          #\"EffectiveDate\":\"2019-07-02T23:12:54.836Z\",\"User\":{}}"
    }
  },
  "awsAccountId": "123456789012"
}
```

Atributos de eventos de campanha

Esta seção define os atributos incluídos nos eventos de streaming da campanha.

Atributo	Descrição
`event_type`	O tipo de evento. Os valores possíveis são:

Atributo	Descrição
	• _campaign.send: o Amazon Pinpoint executou a campanha. • _campaign.opened_notification: para campanhas de notificação por push, esse tipo de evento indica que o destinatário tocou na notificação para abri-la. • _campaign.received_foreground: para campanhas de notificação por push, esse tipo de evento indica que o destinatário recebeu a mensagem como uma notificação em primeiro plano. • _campaign.received_background: para campanhas de notificação por push, esse tipo de evento indica que o destinatário recebeu a mensagem como uma notificação em segundo plano. **Note** _campaign.opened_notification, _campaign.received_foreground e _campaign.received_background são retornados somente se você usar o AWS Amplify. Para obter mais informações sobre como integrar seu aplicativo com o AWS Amplify. Consulte Conectar o aplicativo de front-end ao Amazon Pinpoint com o AWS Amplify.
`event_timestamp`	A hora em que o evento foi relatado, mostrada como horário do Unix em milissegundos.

Atributo	Descrição
`arrival_timestamp`	A hora em que o evento foi recebido pelo Amazon Pinpoint, mostrada como horário do Unix em milissegundos.
`event_version`	A versão do esquema do evento JSON. **Tip** Marque esta versão no seu aplicativo de processamento de eventos para que você saiba quando atualizar o aplicativo em resposta a uma atualização de esquema.
`application`	Informações sobre o projeto do Amazon Pinpoint associado ao evento. Para obter mais informações, consulte a tabela Aplicativo.
`client`	Informações sobre o endpoint ao qual o evento está associado. Para obter mais informações, consulte a tabela Cliente.
`device`	Informações sobre o dispositivo que relatou o evento. Para campanhas e mensagens transacionais, esse objeto está vazio.
`session`	Informações sobre a sessão que gerou o evento. Para campanhas, esse objeto está vazio.

Atributo	Descrição
`attributes`	Atributos associados ao evento. Para eventos relatados por um de seus aplicativos, esse objeto pode incluir atributos personalizados que são definidos pelo aplicativo. Para eventos criados ao enviar uma campanha, esse objeto contém atributos associados à campanha. Para eventos gerados ao enviar mensagens transacionais, esse objeto contém informações relacionadas à mensagem em si. Para obter mais informações, consulte a tabela Atributos.
`client_context`	Contém um objeto `custom`, que contém uma propriedade `endpoint`. A proprieda de `endpoint` tem o conteúdo do registro de endpoint para o endpoint ao qual a campanha foi enviada.
`awsAccountId`	O ID da conta da AWS usada para enviar a mensagem.

Aplicação

Inclui informações sobre o projeto do Amazon Pinpoint ao qual o evento está associado.

Atributo	Descrição
`app_id`	O ID exclusivo do projeto do Amazon Pinpoint que relatou o evento.
`sdk`	O SDK usado para relatar o evento.

Atributos

Inclui informações sobre a campanha que produziu o evento.

Atributo	Descrição
`treatment_id`	Se a mensagem foi enviada usando uma campanha de teste A/B, esse valor represent a o número de tratamento da mensagem. Para campanhas padrão, esse valor é `0`.
`campaign_activity_id`	O ID exclusivo que o Amazon Pinpoint gera quando o evento ocorre.
`delivery_type`	O método de entrega da campanha. Não confunda esse atributo com o campo `ChannelType` especificado na propriedade `endpoint` do `client_context` . O campo `ChannelType` geralmente se baseia no endpoint para o qual a mensagem está sendo enviada. Para canais que oferecem suporte a apenas um tipo de endpoint, os campos `delivery_type` e `ChannelType` têm o mesmo valor. Por exemplo, para o canal de e-mail, os campos `delivery_type` e `ChannelType` têm o mesmo valor de EMAIL. No entanto, essa condição nem sempre é verdadeira para canais que oferecem suporte a diferentes tipos de endpoints, como canais personalizados. Você pode usar um canal personalizado para diferentes endpoints, como EMAIL, SMS, CUSTOM e assim por diante. Nesse caso, `delivery_type` identifica um evento de entrega personalizado, CUSTOM, e `ChannelType` especifica o tipo de endpoint

Atributo	Descrição
	para o qual a campanha foi enviada, como EMAIL, SMS, CUSTOM e assim por diante. Para obter mais informações sobre a criação de canais personalizados, consulte Criar canais personalizados. Os valores possíveis são: • EMAIL • SMS • ADM • APNS • APNS_SANDBOX • APNS_VOIP • APNS_VOIP_SANDBOX • VOICE • GCM • BAIDU • PUSH • CUSTOM
`campaign_id`	O ID exclusivo da campanha da qual a mensagem foi enviada.

Atributo	Descrição
`campaign_send_status`	Indica o status da campanha para o endpoint de destino. Os possíveis valores incluem: • SUCCESS: a campanha foi enviada com êxito para o endpoint. • FAILURE: a campanha não foi enviada para o endpoint. • DAILY_CAP: a campanha não foi enviada ao endpoint porque o endpoint já recebeu o número máximo de mensagens diárias. • EXPIRED: a campanha não foi enviada ao endpoint porque esse envio excederia as configurações da taxa de envio ou da duração máxima da campanha. • QUIET_TIME: a campanha não foi enviada ao endpoint devido às restrições de período de silêncio. • HOLDOUT: a campanha não foi enviada ao endpoint porque o endpoint era membro do grupo de holdout. • DUPLICATE_ADDRESS: há endereços de endpoint duplicados no segmento. A campanha foi enviada uma vez para o endereço do endpoint. • QUIET_TIME: a campanha não foi enviada ao endpoint devido às restrições de período de silêncio. • CAMPAIGN_CAP: a campanha não foi enviada ao endpoint porque o endpoint já recebeu o número máximo de mensagens desta campanha. • FAILURE_PERMANENT: ocorreu uma falha permanente ao enviar para o endpoint.

Atributo	Descrição
	• TRANSIENT_FAILURE: ocorreu uma falha transitória ao enviar para o endpoint. • THROTTLED: o envio foi limitado. • UNKNOWN: falha desconhecida. • HOOK_FAILURE: falha no hook da campanha. • CUSTOM_DELIVERY_FAILURE: falha na entrega personalizada. • RECOMMENDATION_FAILURE: falha na recomendação. • UNSUPPORTED_CHANNEL: o canal não é suportado.

Cliente

Inclui informações sobre o endpoint que foi direcionado pela campanha.

Atributo	Descrição
`client_id`	O ID do endpoint para o qual a campanha foi enviada.

Eventos de jornada

Se você publicar uma jornada, o Amazon Pinpoint pode transmitir dados de eventos sobre a jornada. Isso inclui dados de eventos para qualquer mensagem de e-mail, por push ou personalizada que você enviar da jornada.

Consulte o seguinte para obter informações sobre os dados que o Amazon Pinpoint transmite:

- Para mensagens de e-mail, consulte the section called “Eventos de e-mail”.
- Para mensagens SMS, consulte Eventos SMS.

Evento de exemplo

O objeto JSON para um evento de jornada contém os dados mostrados no exemplo a seguir.

```
{
   "event_type":"_journey.send",
   "event_timestamp":1572989078843,
   "arrival_timestamp":1572989078843,
   "event_version":"3.1",
   "application":{
      "app_id":"a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
      "sdk":{

      }
   },
   "client":{
      "client_id":"d8dcf7c5-e81a-48ae-8313-f540cexample"
   },
   "device":{
      "platform":{

      }
   },
   "session":{

   },
   "attributes":{
      "journey_run_id":"edc9a0b577164d1daf72ebd15example",
      "journey_send_status":"SUCCESS",
      "journey_id":"546401670c5547b08811ac6a9example",
      "journey_activity_id":"0yKexample",
      "journey_activity_type": "EMAIL",
      "journey_send_status_message": "200",
      "journey_send_status_code": "200"
   },
   "client_context":{
      "custom":{
         "endpoint":"{\"ChannelType\":\"EMAIL\",\"EndpointStatus\":\"ACTIVE\",\"OptOut
\":\"NONE\",\"Demographic\":{\"Timezone\":\"America/Los_Angeles\"}}"
      }
   },
   "awsAccountId":"123456789012"
}
```

Atributos de eventos de jornadas

Esta seção define os atributos incluídos nos dados de fluxo de eventos que o Amazon Pinpoint gera para uma jornada.

Atributo	Descrição
`event_type`	O tipo de evento. Para eventos de jornada, o valor para esse atributo é sempre `_journey.send` , o que indica que o Amazon Pinpoint executou a jornada.
`event_timestamp`	A hora em que o evento foi relatado, mostrada como horário do Unix em milissegundos.
`arrival_timestamp`	A hora em que o evento foi recebido pelo Amazon Pinpoint, mostrada como horário do Unix em milissegundos.
`event_version`	A versão do esquema do evento JSON. **Tip** Marque esta versão no seu aplicativo de processamento de eventos para que você saiba quando atualizar o aplicativo em resposta a uma atualização de esquema.
`application`	Informações sobre o projeto do Amazon Pinpoint associado ao evento. Para obter mais informações, consulte a tabela Aplicativo.
`client`	Informações sobre o endpoint ao qual o evento está associado. Para obter mais informações, consulte a tabela Cliente.

Atributo	Descrição
`device`	Informações sobre o dispositivo que relatou o evento. Para jornadas, este objeto está vazio.
`session`	Informações sobre a sessão que gerou o evento. Para jornadas, este objeto está vazio.
`attributes`	Atributos associados à jornada e à atividade de jornada que gerou o evento. Para obter mais informações, consulte a tabela Atributos.
`client_context`	Contém um objeto `custom`, que contém uma propriedade `endpoint`. A propriedade `endpoint` contém o conteúdo do registro de endpoint para o endpoint que está associado ao evento.
`awsAccountId`	O ID da AWS conta que foi usada para executar a viagem.

Aplicativo

Inclui informações sobre o projeto do Amazon Pinpoint ao qual o evento está associado.

Atributo	Descrição
`app_id`	O ID exclusivo do projeto do Amazon Pinpoint que relatou o evento.
`sdk`	O SDK usado para relatar o evento.

Cliente

Inclui informações sobre o endpoint que está associado ao evento.

Atributo	Descrição
`client_id`	O ID do endpoint.

Atributos

Inclui informações sobre a jornada sessão que gerou o evento.

Atributo	Descrição
`journey_run_id`	O ID exclusivo da execução da jornada que gerou o evento. O Amazon Pinpoint gera e atribui esse ID automaticamente a cada nova execução de uma jornada.
`journey_send_status`	Indica o status de entrega da mensagem que está associada ao evento. Os possíveis valores incluem: • SUCCESS: a mensagem foi enviada com êxito para o endpoint. • FAILURE: a mensagem não foi enviada para o endpoint porque ocorreu um erro. • CUSTOM_DELIVERY_FAILURE: falha na entrega personalizada. • FAILURE_PERMANENT: ocorreu uma falha permanente ao enviar para o endpoint. **Tip** Você pode filtrar eventos com o status FAILURE_PERMANENT e `journey_send_status_code` definir como 403 para determinar se há uma violação de política de acesso e função. Para campanhas

Atributo	Descrição
	externas com voz, essas exceções são típicas dos casos em que a função de execução da campanha Connect que vincula as viagens do Amazon Pinpoint às campanhas do Amazon Connect é excluída inadvertidamente para execuções de viagens em voo. • THROTTLED: o envio foi limitado. • UNSUPPORTED_CHANNEL: o canal não é compatível. • DAILY_CAP: a mensagem não foi enviada para o endpoint porque o envio da mensagem excederia o número máximo de mensagens que a jornada ou o projeto pode enviar para um único endpoint durante um período de 24 horas. • QUIET_TIME: a mensagem não foi enviada devido às restrições de tempo de silêncio para a jornada ou projeto. • QUIET_TIME_MISSING_TIMEZONE: a mensagem não foi enviada porque não foi possível estimar um fuso horário para o endpoint e o horário de silêncio está ativado.
`journey_id`	O ID exclusivo da jornada que gerou o evento.
`journey_activity_id`	O ID exclusivo da atividade de jornada que gerou o evento.

Atributo	Descrição
`journey_activity_type`	O tipo de atividade de jornada do evento. Isso pode ser EMAIL, SMS, PUSH, CONTACT_C ENTER ou CUSTOM. **Note** VOICE não é um tipo de atividade de jornada compatível.
`journey_send_status_message`	A descrição do status do evento de envio.
`journey_send_status_code`	O código de status HTTP da solicitação.

Eventos de e-mail

Ao enviar mensagens de e-mail, o Amazon Pinpoint pode transmitir dados que fornecem informações adicionais sobre os seguintes tipos de eventos para essas mensagens:

- Envios
- Entregas
- Devoluções
- Reclamações
- Aberturas
- Cliques
- Rejeições
- Cancelamento de assinaturas
- Falhas de processamento

Os tipos de eventos na lista anterior são explicados em detalhes em Atributos de eventos de e-mail.

Dependendo da API e das configurações usadas para enviar mensagens por e-mail, será possível ver tipos de eventos adicionais ou dados diferentes. Por exemplo, se você enviar mensagens usando conjuntos de configurações que publicam dados de eventos no Amazon Kinesis, como os fornecidos

pelo Amazon Simple Email Service (Amazon SES), os dados também podem incluir eventos para falhas de renderização de modelo. Para obter informações sobre esses dados, consulte Monitorar usando a publicação de eventos do Amazon SES no Guia do desenvolvedor do Amazon Simple Email.

Eventos de exemplo

Envio de e-mail

O objeto JSON de um evento de envio de e-mail contém os dados mostrados no exemplo a seguir.


```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
```

```
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "send": {}
      }
    }
  }
}
```

E-mail entregue

O objeto JSON de um evento de e-mail entregue contém os dados mostrados no exemplo a seguir.

```
{
  "event_type": "_email.delivered",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622690,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "e9a3000d-daa2-40dc-ac47-1cd34example"
  },
  "device": {
    "platform": {}
  },
```

```
  "session": {},
  "attributes": {
    "feedback": "delivered"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "delivery": {
          "smtp_response": "250 ok:  Message 82080542 accepted",
          "reporting_mta": "a8-53.smtp-out.amazonses.com",
          "recipients": ["recipient@example.com"],
          "processing_time_millis": 1310
        }
      }
    }
```

```
  }
}
```

Clique em e-mail

O objeto JSON de um evento de clique em e-mail contém os dados mostrados no exemplo a seguir.

```
{
  "event_type": "_email.click",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618713751,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "49c1413e-a69c-46dc-b1c4-6470eexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "https://aws.amazon.com/pinpoint/"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
```

```
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "click": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15",
          "link": "https://aws.amazon.com/pinpoint/"
        }
      }
    }
  }
}
```

E-mail aberto

O objeto JSON de um evento de e-mail aberto contém os dados mostrados no exemplo a seguir.

```
{
  "event_type": "_email.open",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618712316,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "8dc1f651-b3ec-46fc-9b67-2a050example"
  },
```

```
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "opened"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "open": {
          "ip_address": "72.21.198.67",
```

```
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko)"
        }
      }
    }
  }
}
```

Atributos de eventos de e-mail

Esta seção define os atributos incluídos nos dados de fluxo do evento que o Amazon Pinpoint gera ao enviar mensagens de e-mails.

Atributo	Descrição
`event_type`	O tipo de evento. Os valores possíveis são: • _email.send: o Amazon Pinpoint aceitou a mensagem e tentou entregá-la ao destinatário. • _email.delivered: a mensagem foi entregue ao destinatário. • _email.rejected: o Amazon Pinpoint determinou que a mensagem continha malware e não tentou enviá-la. • _email.hardbounce: um problema permanente impediu que o Amazon Pinpoint entregasse a mensagem. O Amazon Pinpoint não tentará entregar a mensagem novamente. • _email.softbounce: um problema temporário impediu que o Amazon Pinpoint entregasse a mensagem. O Amazon Pinpoint tentará entregar a mensagem novamente por determinado período. Se a mensagem ainda não puder ser entregue, não haverá mais tentativas. O estado final do e-mail será então SOFTBOUNCE.

Atributo	Descrição
	• _email.complaint: o destinatário recebeu a mensagem e reportou a mensagem para o provedor de e-mail como spam (por exemplo, usando o atributo “Denunciar spam” do cliente de e-mail). • _email.open: o destinatário recebeu a mensagem e a abriu. • _email.click: o destinatário recebeu a mensagem e clicou em um link nela. • _email.unsubscribe: o destinatário recebeu a mensagem e clicou em um link de cancelamento de inscrição contido nela. • _email.rendering_failure: o e-mail não foi enviado devido a uma falha na renderiza ção. Isso pode ocorrer quando estão faltando dados no modelo ou quando há uma incompatibilidade entre os parâmetros e os dados do modelo.
`event_timestamp`	A hora em que a mensagem foi enviada, mostrada como tempo Unix em milissegundos. Esse valor geralmente é o mesmo para todos os eventos gerados para uma mensagem.
`arrival_timestamp`	A hora em que o evento foi recebido pelo Amazon Pinpoint, mostrada como horário do Unix em milissegundos.

Atributo	Descrição
`event_version`	A versão do esquema do evento JSON. **Tip** Marque esta versão no seu aplicativo de processamento de eventos para que você saiba quando atualizar o aplicativo em resposta a uma atualização de esquema.
`application`	Informações sobre o projeto do Amazon Pinpoint associado ao evento. Consulte a tabela Aplicativo para obter mais informações.
`client`	Informações sobre o cliente de aplicativo instalado no dispositivo que relatou o evento. Para obter mais informações, consulte a tabela Cliente.
`device`	Informações sobre o dispositivo que relatou o evento. Para obter mais informações, consulte a tabela Dispositivo. Para eventos de e-mail, esse objeto está vazio.
`session`	Para eventos de e-mail, esse objeto está vazio.

Atributo	Descrição
`attributes`	Atributos associados ao evento. Para obter mais informações, consulte a tabela Atributos. Para eventos relatados por um de seus aplicativos, esse objeto pode incluir atributos personalizados que são definidos pelo aplicativo. Para eventos criados quando você envia uma mensagem de uma campanha ou jornada, esse objeto contém atributos associados à campanha ou jornada. Para eventos gerados ao enviar mensagens transacionais, esse objeto contém informações relacionadas à mensagem em si.
`client_context`	Para eventos de email, esse objeto contém um objeto de `custom` que contém um atributo `legacy_identifier` . O valor do atributo `legacy_identifier` é o ID do projeto do qual a mensagem foi enviada.
`facets`	Informações adicionais sobre a mensagem, como os cabeçalhos de e-mail. Consulte a tabela Facetas para obter mais informações.
`awsAccountId`	O ID da conta da AWS usada para enviar a mensagem.

Aplicação

Inclui informações sobre o projeto do Amazon Pinpoint ao qual o evento está associado.

Atributo	Descrição
`app_id`	O ID exclusivo do projeto do Amazon Pinpoint que relatou o evento.

Atributo	Descrição
`sdk`	O SDK usado para relatar o evento. Se você enviar uma mensagem de e-mail transacional chamando a API do Amazon Pinpoint diretamente ou usando o console do Amazon Pinpoint, esse objeto estará vazio.

Atributos

Inclui informações sobre a campanha ou jornada que produziu o evento.

Campanha

Inclui informações sobre a campanha que produziu o evento.

Atributo	Descrição
`feedback`	Para eventos `_email.click` , o valor desse atributo é o URL do link em que o destinatário clicou para gerar o evento. Para outros eventos, o valor representa o tipo de evento (como, por exemplo, `received`, `opened` ou `clicked`).
`treatment_id`	Se a mensagem foi enviada usando uma campanha de teste A/B, esse valor representa o número de tratamento da mensagem. Para campanhas padrão e mensagens de e-mail transacionais, esse valor é `0`.
`campaign_activity_id`	O ID exclusivo que o Amazon Pinpoint gera quando o evento ocorre.
`campaign_id`	O ID exclusivo da campanha do que enviou a mensagem.

Jornada

Inclui informações sobre a jornada que produziu o evento.

Atributo	Descrição
`journey_run_id`	O ID exclusivo da execução da jornada que enviou a mensagem. O Amazon Pinpoint gera e atribui esse ID automaticamente a cada nova execução de uma jornada.
`feedback`	Para eventos `_email.click` , o valor desse atributo é o URL do link em que o destinatá rio clicou para gerar o evento. Para outros eventos, o valor representa o tipo de evento (como, por exemplo, `received`, `delivered` ou `opened`).
`journey_id`	O ID exclusivo da jornada que enviou a mensagem.
`journey_activity_id`	O ID exclusivo da atividade de jornada que enviou a mensagem.

Cliente

O identificador exclusivo do cliente que foi alvo da campanha ou da jornada.

Atributo	Descrição
`client_id`	O ID do cliente. O valor é o ID do endpoint para campanhas e jornadas e para envio transacio nal é um UUID.

Facetas

Inclui informações sobre a mensagem e o tipo de evento.

Atributo	Descrição
`email_channel`	Contém um objeto `mail_event` , que contém dois objetos: `mail` e um objeto que correspon de ao tipo de evento.

Mail

Inclui informações sobre o conteúdo da mensagem de e-mail, bem como metadados sobre a mensagem.

Atributo	Descrição
`message_id`	O ID exclusivo da mensagem. O Amazon Pinpoint gera automaticamente esse ID quando aceita a mensagem.
`message_send_timestamp`	A data e a hora em que a mensagem foi enviada, no formato especificado em RFC 822.
`from_address`	O endereço de e-mail do qual a mensagem foi enviada.
`destination`	Uma matriz que contém os endereços de e-mail para os quais a mensagem foi enviada.
`headers_truncated`	Um valor booliano que indica se os cabeçalhos de e-mail foram truncados.
`headers`	Um objeto que contém vários pares de nome-valor que correspondem aos cabeçalhos na mensagem. Esse objeto normalmente contém informações sobre os seguintes cabeçalhos: • `From`: o endereço de e-mail do remetente. • `To`: o endereço de e-mail do destinatário. • `Subject`: a linha de assunto do e-mail.

Atributo	Descrição
	Tip O cabeçalho do assunto não está incluído nos eventos de campanha _email.send. • `MIME-Version` : indica que a mensagem está formato MIME. Se esse cabeçalho estiver presente, o valor será sempre `1.0`. • `Content-Type` : o tipo de mídia MIME do conteúdo da mensagem.
`common_headers`	Contém informações sobre vários cabeçalhos comuns de mensagens de e-mail. As informaçõ es podem incluir a data em que a mensagem foi enviada e as linhas de para, de e assunto da mensagem.

Eventos de SMS

Se o canal SMS estiver habilitado para um projeto, o Amazon Pinpoint pode transmitir dados de evento sobre entregas de mensagens SMS para o projeto. Os eventos de SMS gerados pelas operadoras podem levar até 72 horas para serem recebidos e não devem ser usados para determinar se há um atraso na entrega das mensagens enviadas. Depois de 72 horas, se o Amazon Pinpoint não tiver recebido um evento final de uma operadora, o serviço retornará automaticamente um record_status UNKNOWN, pois não sabemos o que aconteceu com essa mensagem.

Exemplo

O objeto JSON de um evento de SMS contém os dados mostrados no exemplo a seguir.

```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
```

```
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "123456789012"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
    "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
}
```

Atributos de eventos de SMS

Esta seção define os atributos incluídos nos dados de fluxo do evento que o Amazon Pinpoint gera ao enviar mensagens SMS.

Evento

Atributo	Descrição
`event_type`	O tipo de evento. Os valores possíveis são:

Atributo	Descrição
	• _SMS.BUFFERED: a mensagem ainda está em processo de entrega ao destinatário. • _SMS.SUCCESS: a mensagem foi aceita pela operadora e entregue ao destinatário. • _SMS.FAILURE: o Amazon Pinpoint não conseguiu entregar a mensagem ao destinatário. Para saber mais sobre o erro que impediu a entrega da mensagem, consulte `attributes.record_status` . • _SMS.OPTOUT: o cliente recebeu a mensagem e respondeu enviando a palavra-chave de cancelamento (geralmente "STOP").
`event_timestamp`	A hora em que o evento foi relatado, mostrada como horário do Unix em milissegundos.
`arrival_timestamp`	A hora em que o evento foi recebido pelo Amazon Pinpoint, mostrada como horário do Unix em milissegundos.
`event_version`	A versão do esquema do evento JSON. **Tip** Marque esta versão no seu aplicativo de processamento de eventos para que você saiba quando atualizar o aplicativo em resposta a uma atualização de esquema.
`application`	Informações sobre o projeto do Amazon Pinpoint associado ao evento. Para obter mais informações, consulte a tabela Aplicativo.

Atributo	Descrição
`client`	Informações sobre o cliente de aplicativo instalado no dispositivo que relatou o evento. Para obter mais informações, consulte a tabela Cliente.
`device`	Informações sobre o dispositivo que relatou o evento. Para obter mais informações, consulte a tabela Dispositivo. Para eventos de SMS, esse objeto está vazio.
`session`	Para eventos de SMS, esse objeto está vazio.
`attributes`	Atributos associados ao evento. Para eventos relatados por um de seus aplicativos, esse objeto pode incluir atributos personalizados que são definidos pelo aplicativo. Para eventos criados ao enviar uma campanha, esse objeto contém atributos associados à campanha. Para eventos gerados ao enviar mensagens transacionais, esse objeto contém informações relacionadas à mensagem em si. Para obter mais informações, consulte a tabela Atributos.
`metrics`	Métricas adicionais associadas ao evento. Consulte a tabela Métricas para obter mais informações.
`awsAccountId`	O ID da conta da AWS usada para enviar a mensagem.

Aplicação

Inclui informações sobre o projeto do Amazon Pinpoint ao qual o evento está associado e, se aplicável, o SDK usado para relatar o evento.

Atributo	Descrição
`app_id`	O ID exclusivo do projeto do Amazon Pinpoint que relatou o evento.
`sdk`	O SDK usado para relatar o evento. Se você enviar uma mensagem SMS transacional chamando a API do Amazon Pinpoint diretamente ou usando o console do Amazon Pinpoint, esse objeto estará vazio.

Atributos

Inclui informações sobre os atributos associados ao evento.

Atributo	Descrição
`sender_request_id`	Um ID exclusivo associado à solicitação para enviar a mensagem SMS.
`campaign_activity_id`	O ID exclusivo da atividade na campanha.
`origination_phone_number`	O número de telefone do qual a mensagem foi enviada.
`destination_phone_number`	O número de telefone para o qual você tentou enviar a mensagem.
`record_status`	Informações adicionais sobre o status da mensagem. Os possíveis valores incluem: • SUCCESSFUL/DELIVERED: a mensagem foi entregue.

Atributo	Descrição
	• PENDING: a mensagem ainda não foi entregue ao dispositivo do destinatário. • INVALID: o número de telefone de destino é inválido. • UNREACHABLE: o dispositivo do destinatário está inacessível ou indisponível no momento. Por exemplo, o dispositivo pode estar desligado ou desconectado da rede. É possível tentar enviar a mensagem novamente mais tarde. • UNKNOWN: ocorreu um erro que impediu a entrega da mensagem. Esse erro geralmente é temporário, e é possível tentar enviar a mensagem novamente mais tarde. • BLOCKED: o dispositivo do destinatário está bloqueando mensagens SMS do número de origem. • CARRIER_UNREACHABLE: um problema com a rede móvel do destinatário impediu que a mensagem fosse entregue. Esse erro geralmente é temporário, e é possível tentar enviar a mensagem novamente mais tarde. • SPAM: a operadora móvel do destinatário identificou o conteúdo da mensagem como spam e bloqueou a entrega da mensagem. • INVALID_MESSAGE: o corpo da mensagem SMS é inválido e não pode ser entregue. • CARRIER_BLOCKED: a operadora do destinatário bloqueou a entrega dessa mensagem. Isso geralmente ocorre quando a operadora identifica o conteúdo da mensagem como não solicitado ou mal-intencionado.

Atributo	Descrição
	• TTL_EXPIRED: não foi possível entregar a mensagem SMS dentro de um determina do prazo. Esse erro geralmente é temporári o, e é possível tentar enviar a mensagem novamente mais tarde. • MAX_PRICE_EXCEEDED: o envio da mensagem resultaria em uma cobrança que excederia a cota mensal de gastos de SMS de sua conta. É possível solicitar um aumento dessa cota concluindo o procedime nto em Solicitar aumentos de sua cota de gastos mensais de SMS no Guia do usuário do Amazon Pinpoint. • OPTED_OUT: a mensagem SMS não foi enviada porque o destinatário optou por não receber suas mensagens. • NO_QUOTA_LEFT_ON_ACCOUNT: não há cota de gastos suficiente em sua conta para enviar a mensagem. É possível solicitar um aumento dessa cota concluindo o procedime nto em Solicitar aumentos de sua cota de gastos mensais de SMS no Guia do usuário do Amazon Pinpoint. • NO_ORIGINATION_IDENTITY_AVA ILABLE_TO_SEND: sua conta não contém um número de telefone que possa ser usado para enviar a mensagem ao destino. • DESTINATION_COUNTRY_NOT_SUP PORTED: o país de destino está bloqueado . Para saber todos os países que oferecem suporte, consulte Países e regiões que oferecem suporte (canal SMS) • ACCOUNT_IN_SANDBOX: sua conta está na sandbox e só pode ser enviada para

Atributo	Descrição
	números de destino verificados. Você pode verificar o número de destino no console do Amazon Pinpoint ou iniciar o processo para tirar a conta da sandbox, consulte Migração da sandbox de SMS do Amazon Pinpoint para a produção. • RATE_EXCEEDED: você tentou enviar a mensagem muito rápido e foi bloqueado. Você precisa diminuir sua taxa de chamadas. Para obter detalhes sobre nossos limites, consulte Limites de partes de mensagem por segundo (MPS). • INVALID_ORIGINATION_IDENTITY: a identidade de origem fornecida é inválida. • ORIGINATION_IDENTITY_DOES_NOT_EXIST: a identidade de origem fornecida não existe. • INVALID_DLT_PARAMETERS: foram fornecidos parâmetros DLT inválidos (necessários para destinos na Índia). • INVALID_PARAMETERS: foram fornecidos parâmetros inválidos. • ACCESS_DENIED: sua conta está bloqueada para enviar mensagens. Entre em contato com o suporte ao cliente para descobrir a causa e resolver o problema. • INVALID_KEYWORD: a palavra-chave fornecida é inválida. A palavra-chave pode estar no formato incorreto ou não estar definida na sua conta. • INVALID_SENDER_ID: O ID de remetente fornecido é inválido. O ID de remetente pode estar em formato ou tamanho incorretos.

Atributo	Descrição
	• INVALID_POOL_ID: O ID de grupo fornecido é inválido. O ID de grupo pode estar no formato incorreto ou não pertencer à sua conta. • SENDER_ID_NOT_SUPPORTED_FOR_DESTINATION: o país de destino não oferece suporte ao ID do remetente. Você precisa usar um número de telefone ou outra identidade de origem para enviar. • INVALID_PHONE_NUMBER: o número de telefone de origem fornecido é inválido. O número de telefone pode estar em formato ou tamanho incorretos.
`iso_country_code`	O país associada ao número de telefone do destinatário, mostrado no formato ISO 3166-1 alpha-2.
`treatment_id`	O ID do tratamento da mensagem, se a mensagem tiver sido enviada em uma campanha A/B.
`treatment_id`	Se a mensagem foi enviada usando uma campanha de teste A/B, esse valor represent a o número de tratamento da mensagem. Para mensagens SMS transacionais, esse valor é 0.

Atributo	Descrição
`number_of_message_parts`	O número de partes da mensagem que o Amazon Pinpoint criou para enviar a mensagem. Geralmente, as mensagens SMS podem conter apenas 160 caracteres GSM-7 ou 67 caracteres não GSM, embora esses limites possam variar de acordo com o país . Se você enviar uma mensagem que exceda esses limites, o Amazon Pinpoint dividirá a mensagem automaticamente em partes menores. Você será cobrado com base no número de partes de mensagens enviadas.
`message_id`	O ID exclusivo que o Amazon Pinpoint gera quando aceita a mensagem.
`message_type`	O tipo de mensagem. Os valores possíveis são Promocional e Transacional. Especifique esse valor ao criar uma campanha ou ao enviar mensagens transacionais usando a operação SendMessages na API do Amazon Pinpoint.
`campaign_id`	O ID exclusivo da campanha do Amazon Pinpoint que enviou a mensagem.

Cliente

Inclui informações sobre o cliente do aplicativo instalado no dispositivo que relatou o evento.

Atributo	Descrição
`client_id`	Para eventos gerados por aplicativos, esse valor é o ID exclusivo do cliente do aplicativo instalado no dispositivo. Esse ID é gerado

Atributo	Descrição
	automaticamente pelo AWS Mobile SDK for iOS e AWS Mobile SDK for Android. Para eventos gerados ao enviar campanhas e mensagens transacionais, esse valor é igual ao ID do endpoint para o qual você enviou a mensagem.
`cognito_id`	O ID exclusivo atribuído para o cliente do aplicativo no banco de identidades do Amazon Cognito usado pelo aplicativo.

Dispositivo

Inclui informações sobre o dispositivo que relatou o evento.

Atributo	Descrição
`locale`	A localidade do dispositivo.
`make`	O make do dispositivo, como `Apple` ou `Samsung`.
`model`	O modelo do dispositivo, como `iPhone`.
`platform`	A plataforma do dispositivo, como `ios` ou `android`.

Métricas

Inclui informações sobre as métricas associadas ao evento.

Atributo	Descrição
`price_in_millicents_usd`	O valor cobrado pelo envio da mensagem. Esse preço é mostrado em milésimos de um

Atributo	Descrição
	centavo dos Estados Unidos. Por exemplo, se o valor desse atributo for 645, cobraremos USD 0,645 para enviar a mensagem (645/1000 = 0,645 = 0,00645). **Note** Esta propriedade não aparece para mensagens com um `event_type` de _SMS.BUFFERED.

Consultar dados de análise do Amazon Pinpoint

Além de usar as páginas de análises no console do Amazon Pinpoint, é possível usar as APIs do Amazon Pinpoint Analytics para consultar dados de análise para um subconjunto de métricas padrão que fornecem informações sobre tendências relacionadas ao envolvimento de usuários, ao alcance da campanha e muito mais. Essas métricas, também conhecidas como os principais indicadores de desempenho (KPIs), são valores mensuráveis que podem ajudar você a monitorar e avaliar o desempenho de seus projetos, campanhas e jornadas.

Se você usar as APIs para consultar dados analíticos, poderá analisar os dados usando a ferramenta de relatório de sua escolha, sem precisar entrar no console do Amazon Pinpoint ou analisar dados brutos de eventos de fontes como fluxos do Amazon Kinesis. Por exemplo, você pode criar um painel personalizado que exiba resultados semanais da campanha ou forneça análises detalhadas sobre as taxas de entrega para suas campanhas.

Você pode consultar os dados usando a API REST do Amazon Pinpoint, o AWS Command Line Interface (AWS CLI) ou um AWS SDK. Para consultar os dados, envie uma solicitação para a API do Amazon Pinpoint e use parâmetros compatíveis para especificar os dados que deseja e quaisquer filtros que deseja aplicar. Depois de enviar sua consulta, o Amazon Pinpoint retorna os resultados dela em uma resposta JSON. Em seguida, você pode passar os resultados para outro serviço ou aplicativo para uma análise mais aprofundada, armazenamento ou relatórios.

Métricas compatíveis

O Amazon Pinpoint fornece acesso programático aos dados analíticos para diversos tipos de métricas padrão:

- Métricas de aplicativo: essas métricas fornecem informações sobre tendências para todas as campanhas e mensagens transacionais associadas a um projeto. Por exemplo, você pode usar uma métrica de aplicativo para ver um detalhamento do número de mensagens que foram abertas pelos destinatários para cada campanha associada a um projeto. Para acessar os dados de uma métrica de aplicativo, use o recurso Métricas de aplicativo da API do Amazon Pinpoint.
- Métricas de campanha: essas métricas fornecem informações sobre o desempenho de campanhas individuais. Por exemplo, é possível usar uma métrica de campanha para determinar para quantos endpoints uma mensagem de campanha foi enviada. Para acessar os dados de uma métrica de campanha, use o recurso Métricas de campanha da API do Amazon Pinpoint.

- Métricas de engajamento de jornada: essas métricas fornecem informações sobre o desempenho de jornadas individuais. Por exemplo, você pode usar uma métrica de engajamento de jornada para obter um detalhamento do número de mensagens que foram abertas pelos participantes em cada atividade de uma jornada. Para acessar os dados de uma métrica de engajamento de jornada, use o recurso Métricas de engajamento de jornada da API do Amazon Pinpoint.
- Métricas de execução de jornada: essas métricas fornecem informações sobre as tendências de participação para jornadas individuais. Por exemplo, você pode usar uma métrica de execução de jornada para determinar quantos participantes estão realizando as atividades na jornada. Para acessar os dados de uma métrica de execução de jornada, use o recurso Métricas de execução de jornada da API do Amazon Pinpoint.
- Métricas de execução de atividades da jornada: essas métricas fornecem informações sobre as tendências de participação para atividades individuais de uma jornada. Por exemplo, você pode usar uma métrica de execução de atividades da jornada para determinar quantos participantes concluíram uma atividade. Para acessar os dados de uma métrica de execução de atividades da jornada, use o recurso Métricas de execução de atividades da jornada da API do Amazon Pinpoint.

Para obter uma lista completa de métricas padrão que você pode consultar programaticamente, consulte Métricas padrão.

O Amazon Pinpoint coleta e agrega dados automaticamente para todas as métricas compatíveis e para todos os seus projetos, campanhas e jornadas. Além disso, os dados são atualizados continuamente, o que resulta em um período de latência de dados que é limitado a cerca de duas horas. Observe, no entanto, que pode haver latência de dados adicional para determinadas métricas. Isso ocorre porque os dados de algumas métricas são baseados em informações que recebemos dos provedores de e-mail dos destinatários. Alguns provedores nos enviam essas informações imediatamente, enquanto outros enviam com menos frequência.

O Amazon Pinpoint armazena os dados por 90 dias. Para armazenar os dados por mais de 90 dias ou acessar dados analíticos brutos em tempo real, você pode configurar um projeto Amazon Pinpoint para transmitir dados de eventos para o Amazon Kinesis Data Streams ou o Amazon Data Firehose. Para obter informações sobre como configurar fluxos de eventos, consulte Transmissão de eventos do Amazon Pinpoint para o Kinesis.

Noções básicas de consulta

Para consultar os dados de uma métrica, você envia uma solicitação `get` para o recurso de métricas apropriado da API do Amazon Pinpoint. Na solicitação, você define sua consulta usando parâmetros compatíveis para os seguintes componentes de consulta:

- Projeto: especifique um projeto fornecendo o ID do projeto como o valor do parâmetro `application-id`. Esse parâmetro é necessário para todas as métricas.
- Campanha: especifique uma campanha fornecendo o ID da campanha como o valor do parâmetro `campaign-id`. Esse parâmetro é obrigatório somente para métricas de campanha.
- Jornada: especifique uma jornada fornecendo o ID da jornada como o valor do parâmetro `journey-id`. Esse parâmetro é necessário somente para métricas de engajamento e execução da jornada e métricas de execução de atividades da jornada.
- Atividade da jornada: especifique uma atividade da jornada fornecendo o ID da atividade da jornada como o valor do parâmetro `journey-activity-id`. Esse parâmetro é obrigatório somente para métricas de execução de atividades da jornada.
- Intervalo de datas: para filtrar opcionalmente os dados por intervalo de datas, forneça a primeira e última data e hora do intervalo de datas usando os parâmetros de hora inicial e final compatíveis. Os valores devem estar no formato ISO 8601 estendido e usar o Tempo Universal Coordenado (UTC), por exemplo, `2019-07-19T20:00:00Z` para 20h UTC 19 de julho de 2019.

  Os intervalos de datas são inclusivos e devem estar limitados a 31 dias ou menos. Adicionalmente, a primeira data e hora devem ser inferiores a 90 dias a partir do dia atual. Se você não especificar um intervalo de datas, o Amazon Pinpoint retornará os dados para os 31 dias corridos anteriores. Os parâmetros de intervalo de datas são suportados por todas as métricas, exceto métricas de execução de jornada e métricas de execução de atividades da jornada.
- Métrica: especifique a métrica fornecendo o nome da métrica como o valor do parâmetro `kpi-name`. Esse valor descreve a métrica associada e consiste em dois ou mais termos, que são compostos por caracteres alfanuméricos minúsculos, separados por um hífen. Os exemplos são `email-open-rate` e `successful-delivery-rate`. Esse parâmetro é necessário para todas as métricas, exceto métricas de execução de jornada e métricas de execução de atividades da jornada. Para obter uma lista completa das métricas compatíveis e o valor de `kpi-name` a ser usado para cada uma, consulte Métricas padrão.

Depois de enviar sua consulta, o Amazon Pinpoint retorna os resultados dela em uma resposta JSON. Na resposta, a estrutura dos resultados varia dependendo da métrica consultada.

Algumas métricas fornecem apenas um valor, por exemplo, o número de mensagens que foram entregues por uma campanha. Outras métricas fornecem vários valores e normalmente agrupam esses valores por um campo relevante, por exemplo, o número de mensagens que foram entregues em cada execução de uma campanha, agrupado por execução da campanha. Se uma métrica fornecer e agrupar vários valores, a resposta JSON incluirá um campo que indica qual campo foi usado para agrupar os dados. Para saber mais sobre a estrutura dos resultados da consulta, consulte Usar resultados da consulta.

Políticas do IAM para consulta de dados de análise do Amazon Pinpoint

Usando a API do Amazon Pinpoint, você pode consultar dados analíticos para um subconjunto de métricas padrão, também conhecidos como indicadores-chave de desempenho (KPIs) que se aplicam a projetos, campanhas e jornadas do Amazon Pinpoint. Essas métricas podem ajudá-lo a monitorar e avaliar o desempenho de projetos, campanhas e jornadas.

Para gerenciar o acesso a esses dados, você pode criar políticas (IAM) do AWS Identity and Access Management que definem permissões para perfis do IAM ou usuários autorizados a acessar os dados. Para oferecer suporte ao controle granular de acesso a esses dados, o Amazon Pinpoint fornece várias ações distintas que você pode especificar em políticas do IAM. Há uma ação distinta para visualizar dados analíticos no console do Amazon Pinpoint (`mobiletargeting:GetReports`) e há outras ações para acessar dados analíticos programaticamente usando a API do Amazon Pinpoint.

Para criar políticas do IAM que gerenciem o acesso aos dados analíticos, você pode usar o AWS Management Console, a AWS CLI ou a API do IAM. Observe que a guia Editor visual no AWS Management Console atualmente não inclui ações para visualização ou consulta de dados analíticos do Amazon Pinpoint. No entanto, você pode adicionar as ações necessárias às políticas do IAM manualmente usando a guia JSON no console.

Por exemplo, a política a seguir permite o acesso programático a todos os dados analíticos de todos os seus projetos, campanhas e jornadas em todas as regiões da AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "QueryAllAnalytics",
            "Effect": "Allow",
```

```
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:*:accountId:apps/*/kpis/*",
                "arn:aws:mobiletargeting:*:accountId:apps/*/campaigns/*/kpis/*",
                "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/kpis/*",
                "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/execution-
metrics",
                "arn:aws:mobiletargeting:*:accountId:apps/*/journeys/*/activities/*/
execution-metrics"
            ]
        }
    ]
}
```

Onde *accountId* é o ID da sua conta da AWS.

No entanto, como prática recomendada, você deve criar políticas que sigam o princípio de privilégio mínimo. Em outras palavras, você deve criar políticas que incluem somente as permissões necessárias para executar uma tarefa específica. Para apoiar esta prática e implementar um controle mais granular, você pode restringir o acesso programático aos dados analíticos para apenas um projeto específico em uma região específica da AWS, por exemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "QueryProjectAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/*",
```

```
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/*/
kpis/*",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/
kpis/*",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/
execution-metrics",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

Onde:

- *região* é o nome da região da AWS que hospeda o projeto.
- *accountId* é o ID da conta da AWS.
- *projectId* é o identificador do projeto ao qual você deseja fornecer acesso.

Da mesma forma, a política de exemplo a seguir permite acesso programático aos dados analíticos apenas para uma campanha específica:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "QueryCampaignAnalytics",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetCampaignDateRangeKpi",
            "Resource": "arn:aws:mobiletargeting:region:accountId:apps/projectId/
campaigns/campaignId/kpis/*"
        }
    ]
}
```

Onde:

- *região* é o nome da região da AWS que hospeda o projeto.
- *accountId* é seu ID Conta da AWS.
- *projectId* é o identificador do projeto associado à campanha.

- *campaignId* é o identificador da campanha à qual você deseja fornecer acesso.

E a política de exemplo a seguir permite o acesso programático a todos os dados analíticos, tanto de engajamento quanto de execução, para uma determinada jornada e as atividades que compõem essa jornada:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "QueryJourneyAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/
journeys/journeyId/kpis/*",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/
journeys/journeyId/execution-metrics",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/
journeys/journeyId/activities/*/execution-metrics"
            ]
        }
    ]
}
```

Onde:

- *região* é o nome da região da AWS que hospeda o projeto.
- *accountId* é o ID da conta da AWS.
- *projectId* é o identificador do projeto associado à jornada.
- *journeyId* é o identificador da jornada à qual você deseja fornecer acesso.

Para obter uma lista completa de ações da API do Amazon Pinpoint que você pode usar em políticas do IAM, consulte Ações do Amazon Pinpoint para políticas do IAM. Para obter informações detalhadas sobre como criar e gerenciar políticas do IAM, consulte o Guia do usuário do IAM.

Métricas de análise do Amazon Pinpoint padrão

Você pode usar as APIs do Amazon Pinpoint Analytics para consultar dados analíticos de um subconjunto de métricas padrão que se aplicam a projetos, campanhas e jornadas do Amazon Pinpoint. Essas métricas, também conhecidas como os principais indicadores de desempenho (KPIs), são valores mensuráveis que podem ajudar você a monitorar e avaliar o desempenho de projetos, campanhas e jornadas.

O Amazon Pinpoint fornece acesso programático aos dados analíticos para diversos tipos de métricas padrão:

- Métricas de aplicativo: essas métricas fornecem informações sobre tendências para todas as campanhas e mensagens transacionais associadas a um projeto, também conhecido como um aplicativo. Por exemplo, você pode usar uma métrica de aplicativo para ver um detalhamento do número de mensagens que foram abertas pelos destinatários para cada campanha associada a um projeto.
- Métricas de campanha: essas métricas fornecem informações sobre o desempenho de campanhas individuais. Por exemplo, você pode usar uma métrica de campanha para determinar para quantos endpoints uma mensagem de campanha foi enviada ou quantas dessas mensagens foram entregues nos endpoints desejados.
- Métricas de engajamento de jornada: essas métricas fornecem informações sobre o desempenho de jornadas individuais. Por exemplo, você pode usar uma métrica de engajamento de jornada para obter um detalhamento do número de mensagens que foram abertas pelos participantes em cada atividade de uma jornada.
- Métricas de execução de jornada: essas métricas fornecem informações sobre as tendências de participação para jornadas individuais. Por exemplo, você pode usar uma métrica de execução de jornada para determinar quantos participantes iniciaram uma jornada.
- Métricas de execução de atividades da jornada: essas métricas fornecem informações sobre as tendências de participação para atividades individuais de uma jornada. Por exemplo, você pode usar uma métrica de execução de atividades da jornada para determinar quantos participantes iniciaram uma atividade e quantos participantes concluíram cada caminho em uma atividade.

Os tópicos desta seção listam e descrevem as métricas individuais que você pode consultar para cada tipo de métrica.

Tópicos

- Métricas de aplicativo para campanhas
- Métricas de aplicativo para mensagens de e-mail transacionais
- Métricas de aplicativo para mensagens SMS transacionais
- Métricas de campanha
- Métricas de engajamento de jornada
- Métricas de execução de jornada
- Métricas de execução de atividades da jornada
- Métricas de jornada e execução de campanhas

Métricas de aplicativo para campanhas

A tabela a seguir lista e descreve as métricas de aplicativo padrão que você pode consultar para avaliar o desempenho de todas as campanhas associadas a um projeto do Amazon Pinpoint. Para consultar dados dessas métricas, use o recurso Métricas de aplicativo da API do Amazon Pinpoint. A coluna kpi-name da tabela indica o valor a ser usado para o parâmetro `kpi-name` na consulta.

Métrica	Kpi-name	Descrição
Taxa de entrega	`successful-delivery-rate`	Para todas as campanhas que estão associadas a um projeto, a porcentagem de mensagens que foram entregues aos destinatários. Essa métrica é calculada como o número de mensagens que foram enviadas por todas as campanhas para um projeto e entregues aos destinatá rios, dividido pelo número de mensagens que foram enviadas por todas essas campanhas.

Métrica	Kpi-name	Descrição
Taxa de entrega, agrupada por data	`successful-delivery-rate-grouped-by-date`	Para todas as campanhas que estão associadas a um projeto, a porcentagem de mensagens que foram entregues aos destinatários para cada dia no intervalo de datas especificado. Essa métrica é calculada como o número de mensagens que foram enviadas por todas as campanhas para um projeto e entregue aos destinatá rios, dividido pelo número de mensagens que foram enviadas por todas essas campanhas, para cada dia no intervalo de datas especific ado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Taxa de e-mails abertos	`email-open-rate`	Para todas as campanhas que estão associadas a um projeto, a porcentagem de mensagens de e-mail que foram abertas pelos destinatá rios. Essa métrica é calculada como o número de mensagens de e-mail que foram enviadas por todas as campanhas para um projeto e abertas pelos destinatá rios, dividido pelo número de mensagens de e-mail que foram enviadas por todas essas campanhas e entregues aos destinatários.

Métrica	Kpi-name	Descrição
Taxa de abertura de e-mail, agrupados por campanha	`email-open-rate-grouped-by-campaign`	Para cada campanha que está associada a um projeto, a porcentagem de mensagens de e-mail que foram abertas pelos destinatários. Essa métrica é calculada como o número de mensagens de e-mail que foram enviadas por uma campanha e aberta pelos destinatários, dividido pelo número de mensagens de e-mail que foram enviadas pela campanha e entregues aos destinatários. Os resultados da consulta para essa métrica são agrupados por ID de campanha (`CampaignId`), que é uma string que identifica exclusivamente uma campanha.
Entregas de endpoint	`unique-deliveries`	Para todas as campanhas associadas a um projeto, o número de endpoints exclusivos para os quais as mensagens foram enviadas.

Métrica	Kpi-name	Descrição
Entregas de endpoint, agrupadas por campanha	`unique-deliveries-grouped-by-campaign`	Para cada campanha associada a um projeto, o número de endpoints exclusivos para os quais as mensagens foram enviadas. Os resultados da consulta para essa métrica são agrupados por ID de campanha (`CampaignId`), que é uma string que identifica exclusivamente uma campanha.
Entregas de endpoint, agrupadas por data	`unique-deliveries-grouped-by-date`	Para todas as campanhas associadas a um projeto, o número de endpoints exclusivos para os quais as mensagens foram enviadas, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Mensagens entregues, agrupadas por campanha	`successful-deliveries-grouped-by-campaign`	Para cada campanha que está associada a um projeto, o número de mensagens que foram entregues aos destinatários. Essa métrica é calculada como o número de mensagens que foram enviadas por uma campanha, menos o número de mensagens que foram enviadas pela campanha e não puderam ser entregues aos destinatários devido a uma devolução definitiva. Os resultados da consulta para essa métrica são agrupados por ID de campanha (`CampaignId`), que é uma string que identifica exclusivamente uma campanha.

Métrica	Kpi-name	Descrição
Taxa de abertura de push	`push-open-rate`	Para todas as campanhas que estão associadas a um projeto, a porcentagem de notificações por push que foram abertas pelos destinatá rios. Essa métrica é calculada como o número de notificaç ões por push que foram enviadas por todas as campanhas para um projeto e abertas pelos destinatá rios, dividido pelo número de notificações por push que foram enviadas por todas essas campanhas e entregues aos destinatários.

Métrica	Kpi-name	Descrição
Taxa de abertura de notificaç ões por push, agrupada por campanha	`push-open-rate-gro uped-by-campaign`	Para cada campanha que está associada a um projeto, a porcentagem de notificações por push que foram abertas pelos destinatários. Essa métrica é calculada como o número de notificaç ões por push que foram enviadas por uma campanha e abertas pelos destinatá rios, dividido pelo número de notificações por push que foram enviadas pela campanha e entregues aos destinatários. Os resultados da consulta para essa métrica são agrupados por ID de campanha (`CampaignI d`), que é uma string que identifica exclusivamente uma campanha.

Métricas de aplicativo para mensagens de e-mail transacionais

A tabela a seguir lista e descreve métricas de aplicativo padrão que você pode consultar para monitorar as tendências de todas as mensagens de e-mail transacionais associadas a um projeto do Amazon Pinpoint. Para consultar dados dessas métricas, use o recurso Métricas de aplicativo da API do Amazon Pinpoint. A coluna kpi-name da tabela indica o valor a ser usado para o parâmetro `kpi-name` na consulta.

Observe que essas métricas não fornecem dados sobre mensagens de e-mail que foram enviadas por campanhas. Elas fornecem dados somente sobre mensagens de e-mail transacionais. Para

consultar dados de mensagens que foram enviadas por uma ou mais campanhas, use uma métrica de campanha ou uma métrica de aplicativo para campanhas.

Métrica	Kpi-name	Descrição
Cliques	`txn-emails-clicked`	O número de vezes que os destinatários clicaram em links nas mensagens: Se um único destinatário clicou em vários links em uma mensagem, ou clicou no mesmo link mais de uma vez, cada clique será incluído na contagem.
Cliques, agrupados por data	`txn-emails-clicked-grouped-by-date`	O número de vezes que os destinatários clicaram em links nas mensagens, para cada dia no intervalo de datas especificado. Se um único destinatário clicou em vários links em uma mensagem, ou clicou no mesmo link mais de uma vez, cada clique será incluído na contagem. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Taxa de reclamações	`txn-emails-complaint-rate`	A porcentagem de mensagens que foram relatadas pelos destinatários como e-mails não solicitados ou indesejados. Essa métrica é calculada como o número de

Métrica	Kpi-name	Descrição
		mensagens que foram relatadas pelos destinatários como e-mails não solicitados ou indesejados, dividido pelo número de mensagens que foram enviadas.
Taxa de reclamações, agrupada por data	`txn-emails-complaint-rate-grouped-by-date`	A porcentagem de mensagens que foram relatadas pelos destinatários como e-mails não solicitados ou indesejados, para cada dia no intervalo de datas especificado. Essa métrica é calculada como o número de mensagens que foram relatadas pelos destinatários como e-mails não solicitados ou indesejados, dividido pelo número de mensagens que foram enviadas, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Reclamações	`txn-emails-with-complaints`	O número de mensagens que foram relatadas pelos destinatários como e-mail não solicitado ou indesejado.

Métrica	Kpi-name	Descrição
Reclamações, agrupadas por data	`txn-emails-with-complaints-grouped-by-date`	O número de mensagens que foram relatadas pelos destinatários como e-mail não solicitado ou indesejado, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Entregas	`txn-emails-delivered`	O número de mensagens entregues aos destinatários: Essa métrica é calculada como o número de mensagens que foram enviadas, menos o número de mensagens que não puderam ser entregues devido a uma devolução flexível ou definitiva ou porque foram rejeitadas. Uma mensagem será rejeitada se o Amazon Pinpoint determinar que a mensagem contém malware. O Amazon Pinpoint não tenta enviar mensagens rejeitadas.

Métrica	Kpi-name	Descrição
Entregas, agrupadas por data	`txn-emails-delivered-grouped-by-date`	O número de mensagens que foram entregues aos destinatários, para cada dia no intervalo de datas especificado. Essa métrica é calculada como o número de mensagens que foram enviadas, menos o número de mensagens que não puderam ser entregues devido a uma devolução flexível ou definitiva ou porque foram rejeitadas, para cada dia no intervalo de datas especificado. Uma mensagem será rejeitada se o Amazon Pinpoint determinar que a mensagem contém malware. O Amazon Pinpoint não tenta enviar mensagens rejeitadas. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Taxa de entrega	`txn-emails-delivery-rate`	A porcentagem de mensagens que foram entregues aos destinatários. Essa métrica é calculada como o número de mensagens que foram enviadas e entregues aos destinatários, dividido pelo número de mensagens que foram enviadas.
Taxa de entrega, agrupada por data	`txn-emails-delivery-rate-grouped-by-date`	A porcentagem de mensagens que foram entregues aos destinatários, para cada dia no intervalo de datas especific ado. Essa métrica é calculada como o número de mensagens que foram enviadas e entregues aos destinatários, dividido pelo número de mensagens que foram enviadas, para cada dia no intervalo de datas especific ado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Devoluções definitivas	`txn-emails-hard-bounced`	O número de mensagens que não puderam ser entregues aos destinatários devido a uma devolução definitiva. Uma devolução definitiva ocorre quando um problema persistente impede que uma mensagem seja entregue, por exemplo, se o endereço de e-mail do destinatário não existir.
Devoluções definitivas, agrupadas por data	`txn-emails-hard-bounced-grouped-by-date`	O número de mensagens que não puderam ser entregues aos destinatários devido a uma devolução definitiva, para cada dia no intervalo de datas especificado. Uma devolução definitiva ocorre quando um problema persistente impede que uma mensagem seja entregue, por exemplo, se o endereço de e-mail do destinatário não existir. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Aberturas	`txn-emails-opened`	O número de mensagens que foram abertas pelos destinatários.

Métrica	Kpi-name	Descrição
Aberturas, agrupadas por data	`txn-emails-opened-grouped-by-date`	O número de mensagens que foram abertas pelos destinatários, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Envios	`txn-emails-sent`	O número de mensagens enviadas.
Envios, agrupados por data	`txn-emails-sent-grouped-by-date`	O número de mensagens que foram enviadas, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Devoluções flexíveis	`txn-emails-soft-bounced`	O número de mensagens que não puderam ser entregues aos destinatários devido a uma devolução flexível. Uma devolução flexível ocorre se um problema temporário impede que uma mensagem seja entregue, por exemplo, se a caixa de entrada do destinatário está cheia ou o servidor receptor está temporariamente indisponível.

Métrica	Kpi-name	Descrição
Devoluções flexíveis, agrupadas por data	`txn-emails-soft-bounced-grouped-by-date`	O número de mensagens que não puderam ser entregues aos destinatários devido a uma devolução flexível, para cada dia no intervalo de datas especificado. Uma devolução flexível ocorre se um problema temporário impede que uma mensagem seja entregue, por exemplo, se a caixa de entrada do destinatário está cheia ou o servidor receptor está temporariamente indisponível. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Eventos exclusivos de clique de usuário	`txn-emails-unique-clicks`	O número de destinatá rios exclusivos (endpoints) que clicaram em links nas mensagens. Ao contrário da métrica Clicks (Cliques), essa métrica informa o número de destinatá rios exclusivos que clicaram em links, e não o número de eventos de clique que ocorreram. Por exemplo, se um único destinatário clicou em vários links na mesma mensagem ou clicou no mesmo link mais de uma vez, essa métrica relatará apenas um evento de clique para esse destinatário.

Métrica	Kpi-name	Descrição
Eventos exclusivos de clique de usuário, agrupados por data	`txn-emails-unique-clicks-grouped-by-date`	O número de destinatá rios exclusivos (endpoints) que clicaram em links em mensagens, para cada dia no intervalo de datas especific ado. Ao contrário da métrica Clicks, grouped by date (Cliques, agrupados por data) essa métrica informa o número de destinatários exclusivo s que clicaram em links, e não o número de eventos de clique que ocorreram. Por exemplo, se um único destinatário clicou em vários links na mesma mensagem ou clicou no mesmo link mais de uma vez, essa métrica relatará apenas um evento de clique para esse destinatário. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Eventos de abertura de usuário exclusivos	`txn-emails-unique-opens`	O número de destinatários exclusivos (endpoints) que abriram mensagens. Ao contrário da métrica Opens (Aberturas), essa métrica informa o número de destinatários exclusivos que abriram mensagens, não o número de eventos de abertura que ocorreram. Por exemplo, se um único destinatário abrir a mesma mensagem várias vezes, essa métrica relatará apenas um evento de abertura para esse destinatário.

Métrica	Kpi-name	Descrição
Eventos de abertura de usuário exclusivos, agrupados por data	`txn-emails-unique-opens-grouped-by-date`	O número de destinatários exclusivos (endpoints) que abriram mensagens, para cada dia no intervalo de datas especificado. Ao contrário da métrica Opens, grouped by date (Aberturas, agrupadas por data), essa métrica informa o número de destinatários exclusivos que abriram mensagens, não o número de eventos de abertura que ocorreram. Por exemplo, se um único destinatário abrir a mesma mensagem várias vezes, essa métrica relatará apenas um evento de abertura para esse destinatário. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métricas de aplicativo para mensagens SMS transacionais

A tabela a seguir lista e descreve métricas de aplicativo padrão que você pode consultar para monitorar as tendências de todas as mensagens SMS transacionais associadas a um projeto do Amazon Pinpoint. Para consultar dados dessas métricas, use o recurso Métricas de aplicativo da API do Amazon Pinpoint. A coluna kpi-name da tabela indica o valor a ser usado para o parâmetro `kpi-name` na consulta.

Observe que essas métricas não fornecem dados sobre mensagens SMS que foram enviadas por campanhas. Elas fornecem dados somente sobre mensagens SMS transacionais. Para consultar dados de mensagens que foram enviadas por uma ou mais campanhas, use uma métrica de campanha ou uma métrica de aplicativo para campanhas.

Métrica	Kpi-name	Descrição
Preço médio por mensagem, agrupado por país	`txn-sms-average-price-grouped-by-country`	O custo médio para o envio de cada mensagem, para cada país ou região de destino das mensagens. Esse preço é mostrado em milésimos de um centavo dos Estados Unidos. Por exemplo, se o valor desse atributo for 645, cobraremos 0,645 para enviar a mensagem (645/1000 = 0,645 = 0,00645). Essa métrica é calculada como o custo total de todas as mensagens enviadas aos destinatários em cada país ou região, dividido pelo número de mensagens enviadas aos destinatários em cada um desses países e regiões. Os resultados da consulta dessa métrica são agrupados por país ou região, no formato ISO 3166-1 alfa-2.
Preço médio por parte da mensagem, agrupado por país	`txn-sms-average-price-by-parts-grouped-by-country`	O custo médio para o envio de cada parte da mensagem, para cada país ou região de destino das mensagens. Uma parte da

Métrica	Kpi-name	Descrição
		mensagem é uma parte de uma mensagem SMS. Esse preço é mostrado em milésimos de um centavo dos Estados Unidos. Por exemplo, se o valor desse atributo for 645, cobraremos 0,645 para enviar a mensagem (645/1000 = 0,645 = 0,00645). Essa métrica é calculada como o custo total de todas as partes das mensagens enviadas aos destinatários em cada país ou região, dividido pelo número de partes das mensagens enviadas aos destinatários em cada um desses países e regiões. Os resultados da consulta dessa métrica são agrupados por país ou região, no formato ISO 3166-1 alfa-2.
Entregas	`txn-sms-delivered`	O número de mensagens entregues aos destinatários:

Métrica	Kpi-name	Descrição
Entregas, agrupadas por país	`txn-sms-delivered-grouped-by-country`	O número de mensagens que foram entregues aos destinatários, para cada país ou região de destino das mensagens. Os resultados da consulta dessa métrica são agrupados por país ou região, no formato ISO 3166-1 alfa-2.
Entregas, agrupadas por data	`txn-sms-delivered-grouped-by-date`	O número de mensagens que foram entregues aos destinatários, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Erros de entrega	`txn-sms-error-distribution`	O número de vezes que um erro ocorreu durante a tentativa de entrega das mensagens, para cada tipo de erro ocorrido. Os resultados da consulta dessa métrica são agrupados por código de erro, para cada tipo de erro ocorrido.

Métrica	Kpi-name	Descrição
Taxa de entrega	`txn-sms-delivery-rate`	A porcentagem de mensagens que foram entregues aos destinatários. Essa métrica é calculada como o número de mensagens que foram enviadas e entregues aos destinatários, dividido pelo número de mensagens que foram enviadas.
Taxa de entrega, agrupada por data	`txn-sms-delivery-rate-grouped-by-date`	A porcentagem de mensagens que foram entregues aos destinatários, para cada dia no intervalo de datas especificado. Essa métrica é calculada como o número de mensagens que foram enviadas e entregues aos destinatários, dividido pelo número de mensagens que foram enviadas, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Partes da mensagem entregues	`txn-sms-delivered-by-parts`	O número de partes das mensagens entregues. Uma parte da mensagem é uma parte de uma mensagem SMS. Se uma mensagem SMS contiver mais caractere s do que o permitido pelo protocolo SMS, o Amazon Pinpoint dividirá a mensagem em quantas partes forem necessárias para enviá-la ao destinatário.
Partes da mensagem entregues, agrupadas por país	`txn-sms-delivered-by-parts-grouped-by-country`	O número de mensagens que foram entregues aos destinatá rios, para cada país ou região de destino das mensagens. Uma parte da mensagem é uma parte de uma mensagem SMS. Os resultados da consulta dessa métrica são agrupados por país ou região, no formato ISO 3166-1 alfa-2.

Métrica	Kpi-name	Descrição
Partes da mensagem enviadas	`txn-sms-sent-by-parts`	O número de partes das mensagens enviadas. Uma parte da mensagem é uma parte de uma mensagem SMS. Se uma mensagem SMS contiver mais caracteres do que o permitido pelo protocolo SMS, o Amazon Pinpoint dividirá a mensagem em quantas partes forem necessárias para enviá-la ao destinatário.
Partes da mensagem enviadas, agrupadas por país	`txn-sms-sent-by-parts-grouped-by-country`	O número de partes das mensagens que foram enviadas, para cada país ou região de destino das mensagens. Uma parte da mensagem é uma parte de uma mensagem SMS. Os resultados da consulta dessa métrica são agrupados por país ou região, no formato ISO 3166-1 alfa-2.
Mensagens enviadas	`txn-sms-sent`	O número de mensagens enviadas.

Métrica	Kpi-name	Descrição
Mensagens entregues, agrupadas por país	`txn-sms-sent-grouped-by-country`	O número de mensagens que foram enviadas, para cada país ou região de destino das mensagens. Os resultados da consulta dessa métrica são agrupados por país ou região, no formato ISO 3166-1 alfa-2.
Mensagens enviadas, agrupadas por país	`txn-sms-sent-grouped-by-date`	O número de mensagens que foram enviadas, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Preço total, agrupado por país	`txn-sms-total-price-grouped-by-country`	O custo total do envio das mensagens, para cada país ou região de destino das mensagens. Esse preço é mostrado em milésimos de um centavo dos Estados Unidos. Por exemplo, se o valor desse atributo for 645, cobraremos 0,645 para enviar a mensagem (645/1000 = 0,645 = 0,00645). Os resultados da consulta dessa métrica são agrupados por país ou região, no formato ISO 3166-1 alfa-2.

Métricas de campanha

A tabela a seguir lista e descreve métricas de campanha padrão que você pode consultar para avaliar o desempenho de uma campanha individual. Para consultar dados para essas métricas, use o recurso Métricas de campanha da API do Amazon Pinpoint. A coluna kpi-name da tabela indica o valor a ser usado para o parâmetro `kpi-name` na consulta.

Métrica	Kpi-name	Descrição
Taxa de devolução	`hard-bounce-rate`	Para todas as execuções de campanha, a porcentagem de mensagens de e-mail que não puderam ser entregues aos destinatários. Essa métrica calcula apenas devoluções definitivas, ou seja, o endereço de e-mail do destinatário tinha um problema permanente que impediu a entrega da mensagem. Essa métrica é calculada como o número de mensagens de e-mail devolvidas que foram enviadas por todas as execuções de campanha, dividido pelo número de mensagens de e-mail que foram enviadas por todas essas execuções de campanha.
Taxa de devolução, agrupada por execução de campanha	`hard-bounce-rate-grouped-by-campaign-activity`	Para cada execução de campanha, a porcentagem de mensagens de e-

Métrica	Kpi-name	Descrição
		mail que não puderam ser entregues ao destinatá rios. Essa métrica calcula apenas devoluções definitiv as, ou seja, o endereço de e-mail do destinatário tinha um problema permanente que impediu a entrega da mensagem. Essa métrica é calculada como o número de mensagens de e-mail devolvidas que foram enviadas por uma execução de campanha, dividido pelo número de mensagens de e-mail que foram enviadas pela execução de campanha. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignA ctivityId`), que é uma string que identifica exclusiva mente uma execução de campanha.

Métrica	Kpi-name	Descrição
Taxa de entrega	`successful-delivery-rate`	Para todas as execuções de campanha, a porcentagem de mensagens que foram entregues aos destinatários. Essa métrica é calculado como o número de mensagens que foram enviadas por todas as execuções de campanha e entregues aos destinatá rios, dividido pelo número de mensagens que foram enviados por todas essas execuções de campanha.

Métrica	Kpi-name	Descrição
Taxa de entrega, agrupada por execução de campanha	`successful-delivery-rate-grouped-by-campaign-activity`	Para cada execução de campanha, a porcentagem de mensagens que foram entregues aos destinatários. Essa métrica é calculado como o número de mensagens que foram enviadas por uma execução de campanha e entregues aos destinatários, dividido pelo número de mensagens que foram enviadas pela execução de campanha. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignActivityId`), que é uma string que identifica exclusivamente uma execução de campanha.

Métrica	Kpi-name	Descrição
Taxa de entrega, agrupada por data	`successful-delivery-rate-grouped-by-date`	Para todas as execuções de campanha, a porcentagem de mensagens que foram entregues aos destinatários para cada dia no intervalo de datas especificado. Essa métrica é calculada como o número de mensagens que foram enviadas por todas as execuções de campanha e entregues aos destinatá rios, dividido pelo número de mensagens que foram enviadas por todas essas execuções de campanha, para cada dia no intervalo de datas especificado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.

Métrica	Kpi-name	Descrição
Taxa de e-mails abertos	`email-open-rate`	Para todas as execuções de campanha, a porcentagem de mensagens de e-mail que foram abertas pelos destinatá rios. Essa métrica é calculada como o número de mensagens de e-mail que foram enviadas por todas as execuções de campanha e abertas pelos destinatá rios, dividido pelo número de mensagens de e-mail que foram enviadas por todas essas execuções de campanha e entregues aos destinatários.

Métrica	Kpi-name	Descrição
Taxa de abertura de e-mail, agrupada por execução de campanha	`email-open-rate-grouped-by-campaign-activity`	Para cada execução de campanha, a porcentagem de mensagens de e-mail que foram abertas pelos destinatários. Essa métrica é calculado como o número de mensagens de e-mail que foram enviadas por uma execução de campanha e abertas pelos destinatários, dividido pelo número de mensagens de e-mail que foram enviadas pela execução de campanha e entregues aos destinatários. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignActivityId`), que é uma string que identifica exclusivamente uma execução de campanha.

Métrica	Kpi-name	Descrição
E-mails abertos, agrupados por execução de campanha	`direct-email-opens-grouped-by-campaign-activity`	Para cada execução de campanha, o número de mensagens de e-mail que foram abertas pelos destinatá rios. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignA ctivityId`), que é uma string que identifica exclusiva mente uma execução de campanha.
Entregas de endpoint	`unique-deliveries`	Para todas as execuções de campanha, o número de endpoints exclusivos aos quais as mensagens foram entregues.
Entregas de endpoint, agrupadas por execução de campanha	`unique-deliveries-grouped-by-campaig n-activity`	Para cada execução de campanha, o número de endpoints exclusivos aos quais as mensagens foram entregues. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignA ctivityId`), que é uma string que identifica exclusiva mente uma execução de campanha.

Métrica	Kpi-name	Descrição
Entregas de endpoint, agrupadas por data	`unique-deliveries-grouped-by-date`	Para todas as execuções de campanha, o número de endpoints exclusivos aos quais as mensagens foram entregues, para cada dia no intervalo de datas especific ado. Os resultados da consulta para essa métrica são agrupados por dia corrido, no formato estendido ISO 8601.
Links clicados, agrupados por execução de campanha	`clicks-grouped-by-campaign-activity`	Para cada execução de campanha, o número de vezes que os destinatá rios clicaram nos links na mensagem de e-mail. Se um único destinatário clicou em vários links na mensagem, ou clicou no mesmo link mais de uma vez, cada clique será incluído na contagem. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignA ctivityId`), que é uma string que identifica exclusiva mente uma execução de campanha.

Métrica	Kpi-name	Descrição
Mensagens entregues, agrupadas por execução de campanha	`successful-deliveries-grouped-by-campaign-activity`	Para cada execução de campanha, o número de mensagens que foram entregues aos destinatários. Essa métrica é calculada como o número de mensagens que foram enviadas por uma execução de campanha, menos o número de mensagens que não puderam ser entregues aos destinatários da execução devido a uma devolução definitiva. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignActivityId`), que é uma string que identifica exclusivamente uma execução de campanha.

Métrica	Kpi-name	Descrição
Mensagens enviadas, agrupadas por execução de campanha	`attempted-deliveries-grouped-by-campaign-activity`	Para cada execução de campanha, o número de mensagens que foram enviadas. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignActivityId`), que é uma string que identifica exclusivamente uma execução de campanha.
Taxa de abertura de push	`push-open-rate`	Para todas as execuções de campanha, a porcentagem de notificações por push que foram abertas pelos destinatários. Essa métrica é calculada como o número de notificações por push que foram enviadas por todas as execuções de campanha e abertas pelos destinatários, dividido pelo número de notificações por push que foram enviadas por todas essas execuções de campanha e entregues aos destinatários.

Métrica	Kpi-name	Descrição
Taxa de abertura de notificaç ões por push, agrupada por execução de campanha	`push-open-rate-gro uped-by-campaign-a ctivity`	Para cada execução de campanha, a porcentagem de notificações por push que foram abertas pelos destinatá rios. Essa métrica é calculada como o número de notificaç ões por push que foram enviadas por uma execução de campanha e abertas pelos destinatários, dividido pelo número de notificações por push que foram enviadas pela execução de campanha e entregues aos destinatários. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignA ctivityId`), que é uma string que identifica exclusiva mente uma execução de campanha.

Métrica	Kpi-name	Descrição
Total de notificações por push abertas, agrupadas por execução de campanha	`direct-push-opens-grouped-by-campaign-activity`	Para cada execução de campanha, o número de notificações por push que foram abertas pelos destinatários. Os resultados da consulta para essa métrica são agrupados por ID de atividade de campanha (`CampaignActivityId`), que é uma string que identifica exclusivamente uma execução de campanha.
Total SMS spend (Total de SMS gastas)	`sms-spend`	Para todas as campanhas, a quantidade total de dinheiro, em milésimos de centavos, gasto no envio de SMS.

Métricas de engajamento de jornada

A tabela a seguir lista e descreve métricas de engajamento de jornada padrão que você pode consultar para monitorar tendências de todas as mensagens de e-mail enviadas por uma jornada do Amazon Pinpoint. Para consultar dados dessas métricas, use o recurso Métricas de engajamento de jornada da API do Amazon Pinpoint. A coluna kpi-name da tabela indica o valor a ser usado para o parâmetro `kpi-name` na consulta.

Métrica	Kpi-name	Descrição
Cliques	`journey-emails-clicked`	O número de vezes que os participantes clicaram em links nas mensagens. Se um único participante clicou em vários links em uma mensagem, ou

Métrica	Kpi-name	Descrição
		clicou no mesmo link mais de uma vez, cada clique será incluído na contagem.
Cliques, agrupados por atividade	`emails-clicked-grouped-by-journey-activity`	Para cada atividade na jornada, o número de vezes que os participantes clicaram nos links nas mensagens . Se um único participante clicou em vários links em uma mensagem, ou clicou no mesmo link mais de uma vez, cada clique será incluído na contagem. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.
Reclamações	`journey-emails-complained`	O número de mensagens que foram relatadas pelos participantes como e-mail não solicitado ou indesejado.

Métrica	Kpi-name	Descrição
Reclamações, agrupadas por atividade	`emails-complained-grouped-by-journey-activity`	Para cada atividade na jornada, o número de mensagens que foram relatadas pelos participantes como e-mail não solicitado ou indesejado. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.
Entregas	`journey-emails-delivered`	O número de mensagens entregues aos participantes. Essa métrica é calculada como o número de mensagens que foram enviadas, menos o número de mensagens que não puderam ser entregues devido a uma devolução flexível ou definitiva ou porque foram rejeitadas.

Métrica	Kpi-name	Descrição
Entregas, agrupadas por atividade	`emails-delivered-grouped-by-journey-activity`	Para cada atividade na jornada, o número de mensagens que foram entregues aos participantes. Essa métrica é calculada como o número de mensagens que foram enviadas, menos o número de mensagens que não puderam ser entregues devido a uma devolução flexível ou definitiva ou porque foram rejeitadas, para cada atividade na jornada. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.
Devoluções definitivas	`journey-emails-hardbounced`	O número de mensagens que não puderam ser entregues aos participantes devido a uma devolução definitiva. Uma devolução definitiva ocorre quando um problema persistente impede que uma mensagem seja entregue, por exemplo, se o endereço de e-mail do participante não existir.

Métrica	Kpi-name	Descrição
Devoluções definitivas, agrupadas por atividade	`emails-hardbounced-grouped-by-journey-activity`	Para cada atividade na jornada, o número de mensagens que não puderam ser entregues aos participantes devido a uma devolução definitiva. Uma devolução definitiva ocorre quando um problema persistente impede que uma mensagem seja entregue, por exemplo, se o endereço de e-mail do participante não existir. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.
Aberturas	`journey-emails-opened`	O número de mensagens que foram abertas pelos participantes.
Aberturas, agrupadas por atividade	`emails-opened-grouped-by-journey-activity`	Para cada atividade na jornada, o número de mensagens que foram abertas pelos participantes. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.

Métrica	Kpi-name	Descrição
Rejeições	`journey-emails-rejected`	O número de mensagens que não foram enviadas aos participantes porque foram rejeitadas. Uma mensagem será rejeitada se o Amazon Pinpoint determinar que a mensagem contém malware. O Amazon Pinpoint não tenta enviar mensagens rejeitadas.
Rejeições, agrupadas por atividade	`emails-rejected-grouped-by-journey-activity`	Para cada atividade na jornada, o número de mensagens que não foram enviadas aos participantes porque foram rejeitadas. Uma mensagem será rejeitada se o Amazon Pinpoint determinar que a mensagem contém malware. O Amazon Pinpoint não tenta enviar mensagens rejeitadas. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.
Envios	`journey-emails-sent`	O número de mensagens enviadas.

Métrica	Kpi-name	Descrição
Envios, agrupados por atividade	`emails-sent-grouped-by-journey-activity`	Para cada atividade na jornada, o número de mensagens enviadas. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.
Devoluções flexíveis	`journey-emails-softbounced`	O número de mensagens que não puderam ser entregues aos participantes devido a uma devolução flexível. Uma devolução flexível ocorre se um problema temporário impede que uma mensagem seja entregue, por exemplo, se a caixa de entrada do participante está cheia ou o servidor receptor está temporariamente indisponível.

Métrica	Kpi-name	Descrição
Devoluções flexíveis, agrupadas por atividade	`emails-softbounced-grouped-by-journey-activity`	Para cada atividade na jornada, o número de mensagens que não puderam ser entregues aos participantes devido a uma devolução flexível. Uma devolução flexível ocorre se um problema temporário impede que uma mensagem seja entregue, por exemplo, se a caixa de entrada do participante está cheia ou o servidor receptor está temporariamente indisponível. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.
Cancelamento de assinaturas	`journey-emails-unsubscribed`	O número de vezes que os participantes clicaram em links de cancelamento de assinatura nas mensagens. Se um único participante clicou no mesmo link de cancelamento de assinatura mais de uma vez, cada clique será incluído na contagem.

Métrica	Kpi-name	Descrição
Cancelamentos de assinatura, agrupados por atividade	`emails-unsubscribed-grouped-by-journey-activity`	Para cada atividade na jornada, o número de vezes que os participantes clicaram nos links de cancelamento de assinatura nas mensagens . Se um único participante clicou no mesmo link de cancelamento de assinatura mais de uma vez, cada clique será incluído na contagem. Os resultados da consulta para essa métrica são agrupados por ID de atividade (`JourneyActivityId`), que é uma string que identifica exclusivamente uma atividade.

Métricas de execução de jornada

A tabela a seguir lista e descreve métricas de execução padrão que você pode consultar para avaliar o status dos participantes em uma jornada do Amazon Pinpoint. Para consultar dados dessas métricas, use o recurso Métricas de execução de jornada da API do Amazon Pinpoint. A coluna Campo na tabela identifica o nome do campo que aparece nos resultados da consulta para cada métrica.

Métrica	Campo	Descrição
Participantes ativos	`ENDPOINT_ACTIVE`	O número de participa ntes que estão ativamente realizando as atividades na jornada. Esta métrica é calculada como o número de participantes que

Métrica	Campo	Descrição
		iniciaram a jornada, menos o número de participantes que saíram da jornada e o número de participantes que foram removidos da jornada.
Cancelamentos de participantes	`CANCELLED`	O número de participantes que não concluíram a jornada porque a jornada foi cancelada.
Partidas de participantes	`ENDPOINT_LEFT`	O número de participantes que saíram da jornada.
Entradas de participantes	`ENDPOINT_ENTERED`	O número de participantes que iniciaram a jornada.
Exceções dos participantes, limites de reentrada	`REENTRY_CAP_EXCEEDED`	O número de participantes que não concluíram a jornada porque teriam excedido o número máximo de vezes que um único participante pode entrar novamente na jornada.

Métrica	Campo	Descrição
Exceções de participantes, rejeições	`ACTIVE_ENDPOINT_REJECTED`	O número de participantes que não podem iniciar a jornada porque já são participantes ativos da jornada. Um participante será rejeitado se iniciar uma jornada e, posteriormente, você atualizar a definição do endpoint do participante de tal maneira que afete a inclusão de participantes em um segmento (com base em critérios de segmento) ou na jornada (com base nas condições da atividade).

Métricas de execução de atividades da jornada

A tabela a seguir lista e descreve métricas de execução padrão que você pode consultar para avaliar o status dos participantes em cada tipo de atividade individual para uma jornada do Amazon Pinpoint. Para consultar dados dessas métricas, use o recurso Métricas de execução de jornada da API do Amazon Pinpoint. A coluna Métricas na tabela lista os campos que aparecem nos resultados da consulta para cada tipo de atividade. Ela também fornece uma breve descrição de cada campo.

Tipo de atividade	Indicadores
Divisão Sim/Não (`CONDITIONAL_SPLIT `)	As métricas são: • `Branch_FALSE` : o número de participantes que não atenderam às condições da atividade e prosseguiram para a atividade no caminho “Não”.

Tipo de atividade	Indicadores
	• `Branch_TRUE` : o número de participantes que atenderam às condições da atividade e prosseguiram para a atividade no caminho “Sim”. Para cada caminho na atividade, há métricas adicionais disponíveis. Para obter informações sobre essas métricas, consulte a linha desta tabela para esse tipo de atividade.
Retenção (`HOLDOUT`)	As métricas são: • `HOLDOUT`: o número de participantes que foram removidos da jornada como parte da porcentagem de retenção da atividade. • `PASSED`: o número de participantes que continuaram para a próxima atividade na jornada.

Tipo de atividade	Indicadores
E-mail (`MESSAGE`)	As métricas são: • `DAILY_CAP_EXCEEDED` : o número de mensagens que não foram enviadas porque teriam excedido o número máximo de mensagens que um único participante pode receber durante um período de 24 horas. • `FAILURE_PERMANENT` : o número de mensagens que não foram enviadas devido a um problema permanente. • `QUIET_TIME` : o número de mensagens que não foram enviadas porque elas teriam sido entregues durante o período de silêncio para o fuso horário do participante. • `SERVICE_FAILURE` : o número de mensagens que não foram enviadas devido a um problema com o Amazon Pinpoint. • `SUCCESS`: o número de mensagens que foram entregues com êxito aos participantes. • `THROTTLED` : o número de mensagens que não foram enviadas porque, se fossem, excederia as cotas de envio de sua conta do Amazon Pinpoint. • `TRANSIENT_FAILURE` : o número de mensagens que não foram enviadas devido a um problema temporário. • `UNKNOWN`: o número de mensagens que não foram enviadas devido a um problema desconhecido.

Tipo de atividade	Indicadores
Divisão multivariada (`MULTI_CONDITIONAL_SPLIT`)	Para cada caminho da atividade, o número de participantes que foram para a atividade no caminho. Os resultados da consulta para essa métrica são agrupados por caminho, `Branch_#` em que `#` é o identificador numérico de um caminho, por exemplo, `Branch_1` para o primeiro caminho da atividade. Para cada caminho na atividade, há métricas adicionais disponíveis. Para obter informações sobre essas métricas, consulte a linha desta tabela para esse tipo de atividade.
Divisão aleatória (`RANDOM_SPLIT`)	Para cada caminho da atividade, o número de participantes que foram para a atividade no caminho. Os resultados da consulta para essa métrica são agrupados por caminho, `Branch_#` em que `#` é o identificador numérico de um caminho, por exemplo, `Branch_1` para o primeiro caminho da atividade. Para cada caminho na atividade, há métricas adicionais disponíveis. Para obter informações sobre essas métricas, consulte a linha desta tabela para esse tipo de atividade.

Tipo de atividade	Indicadores
Aguardar (`WAIT`)	As métricas são: • `WAIT_FINISHED` : o número de participantes que terminaram de aguardar o tempo especificado. • `WAIT_SKIPPED` : o número de participantes que não esperaram pelo tempo especificado, normalmente porque iniciaram a atividade ou a jornada após a hora de término programada para a atividade. • `WAIT_STARTED` : o número de participantes que começaram a esperar e não pularam ou terminaram de aguardar o tempo especificado.

Tipo de atividade	Indicadores
Central de atendimento (`CONTACT_CENTER`)	As métricas são: • `CALL_QUEUED` : o número de participantes que foram contatados e colocados na fila do Amazon Connect. Inclui novas tentativas de contato. • `CONTINUE_WAITING` : o número de participantes que continuam aguardando tentativas de contato. • `DIAL_FAILURE` : o número de participantes com tentativas diárias fracassadas. • `DROPPED`: o número de participantes que não atendem mais às condições definidas nas atividades de jornada anteriores no horário de envio. • `TIMEOUT`: o número de participantes que não receberam nenhum código de disposição do Amazon Connect após várias tentativas de discagem. • `WAIT_FINISHED` : o número de participa ntes que terminaram de aguardar o tempo especificado. • `WAIT_FOR_QUIET_HOURS` : o número de participantes que aguardam o término do período de silêncio para fazer a entrega no canal. • `WAIT_STARTED` : o número de participantes que começaram a esperar e não pularam ou terminaram de aguardar o tempo especific ado.

Métricas de jornada e execução de campanhas

A tabela a seguir lista e descreve métricas de execução padrão para avaliar o status dos participantes em cada tipo de atividade individual para uma jornada ou campanha do Amazon Pinpoint. Para consultar dados dessas métricas, use o recurso Métricas de execução de jornada ou Métricas da campanha da API do Amazon Pinpoint. A tabela a seguir lista os campos que aparecem nos resultados da consulta para cada tipo de atividade.

Nome da métrica	Aplica-se a jornadas, campanhas ou ambas	Descrição
ENDPOINT_PRODUCED	Ambos	O número de endpoints produzidos inicialmente a partir do segmento ou evento antes de qualquer filtragem.
ENDPOINTS_FROM_USER	Ambos	Se o cliente tiver um segmento somente com ID de usuário, todos os endpoints desses usuários serão adicionados. Essa métrica mede o número de endpoints adicionados dessa forma.
ENDPOINT_OPT_OUT	Ambos	O endpoint foi excluído e não entrou na campanha ou na jornada.
ENDPOINT_INACTIVE	Ambos	O endpoint estava inativo e não entrou na campanha ou na jornada.
FILTERED_OUT_BY_SEGMENT	Ambos	O endpoint não correspondeu aos filtros do segmento e não entrou na campanha ou na jornada.

Nome da métrica	Aplica-se a jornadas, campanhas ou ambas	Descrição
ENDPOINT_MISSING_ADDRESS	Ambos	O endpoint não tinha um endereço e não entrou na campanha ou na jornada.
ENDPOINT_MISSING_CHANNEL	Ambos	O endpoint não tinha um canal e não entrou na campanha ou na jornada.
ENDPOINT_MISSING_TIMEZONE	Ambos	O endpoint não tinha um valor para o fuso horário e foi filtrado. Isso só acontece quando um valor de fuso horário é necessário.
ENDPOINT_TIMEZONE_MISMATCH	Ambos	O endpoint estava em um fuso horário que não estava incluído na execução naquele momento.
ENDPOINT_CHANNEL_MISMATCH	Campanhas	A campanha não tem uma mensagem configurada para o tipo de canal desse endpoint.
DUPLICATE_ENDPOINT	Ambos	Endpoints duplicados foram encontrados e as cópias eliminadas.
DUPLICATE_USER	Ambos	Usuários duplicados foram encontrados e as cópias foram eliminadas de um segmento somente com ID de usuário. Se eles tiverem o mesmo ID de usuário, uma métrica de 1 será emitida.

Nome da métrica	Aplica-se a jornadas, campanhas ou ambas	Descrição
PAUSED	Jornadas	Removido da execução porque a jornada foi interrompida.
ENDED	Jornadas	Removido da execução porque a jornada foi encerrada.
TREATMENT_HOLDOUT	Campanhas	Isso é emitido em campanhas A/B, para endpoints cujas coortes não correspondem ao tratamento atual. Por exemplo, em uma divisão A/B de 50/50, 50% dos endpoints emitirão essa métrica para cada tratamento
ENDPOINT_ESTIMATED_TIMEZONE	Jornadas	A estimativa do fuso horário foi capaz de estimar um fuso horário para o endpoint.

Consultar dados de análise do Amazon Pinpoint para campanhas

Além de usar as páginas de análise no console do Amazon Pinpoint, é possível usar as APIs do Amazon Pinpoint Analytics para consultar dados de análise de um subconjunto de métricas padrão que fornecem informações sobre as tendências de entrega e engajamento para campanhas.

Cada métrica é um valor mensurável, também chamado de indicador-chave de desempenho (KPI), que pode ajudá-lo a monitorar e avaliar o desempenho de uma ou mais campanhas. Por exemplo, você pode usar uma métrica a fim de descobrir para quantos endpoints uma mensagem de campanha foi enviada ou quantas dessas mensagens foram entregues nos endpoints desejados.

O Amazon Pinpoint coleta e agrega automaticamente esses dados para todas as suas campanhas. Os dados são armazenados por 90 dias. Se você integrou um aplicativo móvel ao Amazon Pinpoint usando um AWS Mobile SDK, o Amazon Pinpoint estenderá esse suporte para incluir métricas

adicionais, como a porcentagem de notificações por push que foram abertas pelos destinatários. Para obter informações sobre como integrar um aplicativo móvel, consulte Integrar o Amazon Pinpoint ao aplicativo.

Se você usar as APIs do Amazon Pinpoint Analytics para consultar dados, poderá escolher várias opções que definem o escopo, os dados, o agrupamento e os filtros para a consulta. Faça isso usando parâmetros que especificam o projeto, a campanha e a métrica que você deseja consultar, além de qualquer filtro baseado em datas que você deseja aplicar.

Este tópico explica e fornece exemplos de como escolher essas opções e consultar os dados de uma ou mais campanhas.

Pré-requisitos

Antes de consultar dados de análise para uma ou mais campanhas, é útil reunir as seguintes informações, usadas para definir a consulta:

- ID do projeto: o identificador exclusivo do projeto associado à campanha ou às campanhas. Na API do Amazon Pinpoint, esse valor é armazenado na propriedade `application-id`. No console do Amazon Pinpoint, esse valor é exibido como o ID do projeto na página Todos os projetos.
- ID da campanha: o identificador exclusivo da campanha, se você quiser consultar os dados de apenas uma campanha. Na API do Amazon Pinpoint, esse valor é armazenado na propriedade `campaign-id`. Esse valor não é exibido no console.
- Intervalo de datas: opcionalmente, a primeira e a última data e hora do intervalo de datas para o qual consultar dados. Os intervalos de datas são inclusivos e devem estar limitados a 31 dias ou menos. Além disso, eles devem começar em menos de 90 dias a partir do dia atual. Se você não especificar um intervalo de datas, o Amazon Pinpoint consultará automaticamente os dados dos 31 dias corridos anteriores.
- Tipo de métrica: o tipo de métrica a consultar. Existem dois tipos: métricas de aplicativo e métricas de campanha. Uma métrica de aplicativo fornece dados para todas as campanhas associadas a um projeto, também chamadas de aplicativo. Uma métrica de campanha fornece dados para apenas uma campanha.
- Métrica: o nome da métrica a ser consultada, mais especificamente o valor `kpi-name` da métrica. Para obter uma lista completa das métricas compatíveis e o valor `kpi-name` de cada uma, consulte Métricas padrão.

Também ajuda determinar se você deseja agrupar os dados por um campo relevante. Se o fizer, poderá simplificar a análise e a emissão de relatórios escolhendo uma métrica projetada para agrupar dados automaticamente para você. Por exemplo, o Amazon Pinpoint fornece várias métricas padrão que relatam a porcentagem de mensagens que foram entregues aos destinatários de uma campanha. Uma dessas métricas agrupa automaticamente os dados por data (`successful-delivery-rate-grouped-by-date`). Outra métrica agrupa automaticamente os dados por execução de campanha (`successful-delivery-rate-grouped-by-campaign-activity`). Uma terceira métrica simplesmente retorna um único valor: a porcentagem de mensagens que foram entregues aos destinatários por todas as execuções de campanha (`successful-delivery-rate`).

Se você não conseguir encontrar uma métrica padrão que agrupe dados da maneira desejada, poderá desenvolver uma série de consultas que retornem os dados desejados. Depois, é possível dividir manualmente ou combinar os resultados da consulta em grupos personalizados criados.

Por fim, é importante verificar se você está autorizado a acessar os dados que deseja consultar. Para obter mais informações, consulte Políticas do IAM para consulta de dados de análise do Amazon Pinpoint.

Consultar dados para uma campanha

Para consultar os dados para uma campanha, use a API de Métricas de campanha e especifique valores para os seguintes parâmetros necessários:

- application-id: o ID do projeto, que é o identificador exclusivo do projeto associado à campanha. No Amazon Pinpoint, os termos projeto e aplicativo têm o mesmo significado.
- campaign-id: o identificador exclusivo da campanha.
- kpi-name: o nome da métrica a ser consultada. Esse valor descreve a métrica associada e consiste em dois ou mais termos, que são compostos por caracteres alfanuméricos minúsculos, separados por um hífen. Para obter uma lista completa das métricas compatíveis e o valor `kpi-name` de cada uma, consulte Métricas padrão.

Também é possível aplicar um filtro que consulta os dados para um intervalo de datas específico. Se você não especificar um intervalo de datas, o Amazon Pinpoint retornará os dados para os 31 dias corridos anteriores. Para filtrar os dados por datas diferentes, use os parâmetros de intervalo de datas compatíveis para especificar a primeira e a última data e hora do intervalo de datas. Os valores devem estar no formato ISO 8601 estendido e usar o Tempo Universal Coordenado (UTC),

por exemplo, `2019-07-19T20:00:00Z` para 20h UTC 19 de julho de 2019. Os intervalos de datas são inclusivos e devem estar limitados a 31 dias ou menos. Adicionalmente, a primeira data e hora devem ser inferiores a 90 dias a partir do dia atual.

Os exemplos a seguir mostram como consultar dados de análise para uma campanha usando a API REST, a AWS CLI e o AWS SDK for Java do Amazon Pinpoint. É possível usar qualquer AWS SDK compatível para consultar dados de análise para uma campanha. Os exemplos da AWS CLI são formatados para o Microsoft Windows. Para Unix, Linux e macOS, substitua o caractere de continuação de linha circunflexo (^) por uma barra invertida (\).

REST API

Para consultar dados de análise para uma campanha usando a API REST do Amazon Pinpoint, envie uma solicitação HTTP(S) GET para o URI de Métricas de campanha. No URI, especifique os valores apropriados para os parâmetros de caminho necessários:

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

Onde:

- *endpoint* é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto associado à campanha.
- *application-id* é o identificador exclusivo do projeto associado à campanha.
- *campaign-id* é o identificador exclusivo da campanha.
- *kpi-name* é o valor `kpi-name` da métrica a consultar.

Todos os parâmetros devem ser codificados por URL.

Para aplicar um filtro que consulta os dados de um intervalo de datas específico, acrescente os valores e parâmetros de consulta `start-time` e `end-time` ao URI. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Use um "e" comercial (&) para separar os parâmetros.

Por exemplo, a seguinte solicitação recupera o número de endpoints exclusivos aos quais as mensagens foram entregues, por todas as execuções de uma campanha, de 19 de julho de 2019 até 26 de julho de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Onde:

- `pinpoint.us-east-1.amazonaws.com` é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto.
- `12345678901234567890123456example` é o identificador exclusivo do projeto associado à campanha.
- `80b8efd84042ff8d9c96ce2f8example` é o identificador exclusivo da campanha.
- `unique-deliveries` é o valor `kpi-name` da métrica de entregas de endpoint da campanha, que é a métrica que relata o número de endpoints exclusivos aos quais as mensagens foram entregues, por todas as execuções de uma campanha.
- `2019-07-19T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-07-26T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

AWS CLI

Para consultar dados de análise para uma campanha usando a AWS CLI, use o comando get-campaign-date-range-kpi e especifique os valores apropriados para os parâmetros necessários:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id application-id ^
    --campaign-id campaign-id ^
    --kpi-name kpi-name
```

Onde:

- *`application-id`* é o identificador exclusivo do projeto associado à campanha.
- *`campaign-id`* é o identificador exclusivo da campanha.
- *`kpi-name`* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que consulta os dados de um intervalo de datas específico, adicione os valores e parâmetros `start-time` e `end-time` à consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a seguinte solicitação recupera o número de endpoints exclusivos aos quais as mensagens foram entregues, por todas as execuções de uma campanha, de 19 de julho de 2019 até 26 de julho de 2019:

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
    --kpi-name unique-deliveries ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Onde:

- `12345678901234567890123456example` é o identificador exclusivo do projeto associado à campanha.
- `80b8efd84042ff8d9c96ce2f8example` é o identificador exclusivo da campanha.
- `unique-deliveries` é o valor `kpi-name` da métrica de entregas de endpoint da campanha, que é a métrica que relata o número de endpoints exclusivos aos quais as mensagens foram entregues, por todas as execuções de uma campanha.
- `2019-07-19T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-07-26T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

SDK for Java

Para consultar dados analíticos de uma campanha usando o AWS SDK for Java, use o método GetCampaignDateRangeKpiRequest da API Métricas de campanha. Especifique os valores apropriados para os parâmetros necessários:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withCampaignId("campaignId")
        .withKpiName("kpiName")
```

Onde:

- *applicationId* é o identificador exclusivo do projeto associado à campanha.
- *campaignId* é o identificador exclusivo da campanha.
- *kpiName* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que consulta os dados de um intervalo de datas específico, inclua os valores e parâmetros `startTime` e `endTime` na consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a seguinte solicitação recupera o número de endpoints exclusivos aos quais as mensagens foram entregues, por todas as execuções de uma campanha, de 19 de julho de 2019 até 26 de julho de 2019:

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withCampaignId("80b8efd84042ff8d9c96ce2f8example")
        .withKpiName("unique-deliveries")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Onde:

- `12345678901234567890123456example` é o identificador exclusivo do projeto associado à campanha.
- `80b8efd84042ff8d9c96ce2f8example` é o identificador exclusivo da campanha.
- `unique-deliveries` é o valor `kpi-name` da métrica de entregas de endpoint da campanha, que é a métrica que relata o número de endpoints exclusivos aos quais as mensagens foram entregues, por todas as execuções de uma campanha.
- `2019-07-19T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-07-26T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

Depois de enviar sua consulta, o Amazon Pinpoint retorna os resultados dela em uma resposta JSON. A estrutura dos resultados varia dependendo da métrica consultada. Algumas métricas retornam apenas um valor. Por exemplo, a métrica de entregas de endpoint (`unique-deliveries`)

da campanha, usada nos exemplos anteriores, retorna um valor: o número de endpoints exclusivos aos quais as mensagens foram entregues, por todas as execuções de uma campanha. Nesse caso, a resposta JSON é a seguinte:

```
{
    "CampaignDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "CampaignId":"80b8efd84042ff8d9c96ce2f8example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

Outras métricas retornam vários valores e agrupam os valores por um campo relevante. Se uma métrica retornar vários valores, a resposta JSON incluirá um campo que indica qual campo foi usado para agrupar os dados.

Para saber mais sobre a estrutura dos resultados da consulta, consulte Usar resultados da consulta.

Consultar dados para várias campanhas

Há duas maneiras de consultar os dados para várias campanhas. A melhor maneira depende se você deseja consultar os dados para campanhas associadas ao mesmo projeto. Fazer isso também depende de se você deseja consultar os dados para todas ou somente ou subconjunto dessas campanhas.

Para consultar os dados de campanhas associadas a projetos diferentes ou apenas para um subconjunto das campanhas associadas ao mesmo projeto, a melhor abordagem é criar e executar

uma série de consultas individuais, uma para cada campanha para a qual você deseja consultar os dados. A seção anterior explica como consultar os dados de apenas uma campanha.

Para consultar os dados de todas as campanhas associadas ao mesmo projeto, é possível usar a API de Métricas de aplicativo. Especifique valores para os seguintes parâmetros necessários:

- application-id: o ID do projeto, que é o identificador exclusivo do projeto. No Amazon Pinpoint, os termos projeto e aplicativo têm o mesmo significado.
- kpi-name: o nome da métrica a ser consultada. Esse valor descreve a métrica associada e consiste em dois ou mais termos, que são compostos por caracteres alfanuméricos minúsculos, separados por um hífen. Para obter uma lista completa das métricas compatíveis e o valor `kpi-name` de cada uma, consulte Métricas padrão.

Também é possível filtrar os dados por intervalo de datas. Se você não especificar um intervalo de datas, o Amazon Pinpoint retornará os dados para os 31 dias corridos anteriores. Para filtrar os dados por datas diferentes, use os parâmetros de intervalo de datas compatíveis para especificar a primeira e a última data e hora do intervalo de datas. Os valores devem estar no formato ISO 8601 estendido e usar o Tempo Universal Coordenado (UTC), por exemplo, `2019-07-19T20:00:00Z` para 20h UTC 19 de julho de 2019. Os intervalos de datas são inclusivos e devem estar limitados a 31 dias ou menos. Adicionalmente, a primeira data e hora devem ser inferiores a 90 dias a partir do dia atual.

Os exemplos a seguir mostram como consultar dados de análise para uma campanha usando a API REST do Amazon Pinpoint, o AWS CLI e o AWS SDK for Java. É possível usar qualquer AWS SDK compatível para consultar dados de análise para uma campanha. Os exemplos da AWS CLI são formatados para o Microsoft Windows. Para Unix, Linux e macOS, substitua o caractere de continuação de linha circunflexo (^) por uma barra invertida (\).

REST API

Para consultar dados de análise de várias campanhas usando a API REST do Amazon Pinpoint, envie uma solicitação HTTP(S) GET para o URI de Métricas de aplicativo. No URI, especifique os valores apropriados para os parâmetros de caminho necessários:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Onde:

- *endpoint* é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto associado às campanhas.
- *application-id* é o identificador exclusivo do projeto associado às campanhas.
- *kpi-name* é o valor kpi-name da métrica a consultar.

Todos os parâmetros devem ser codificados por URL.

Para aplicar um filtro que recupera os dados de um intervalo de datas específico, acrescente os valores e parâmetros de consulta start-time e end-time ao URI. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Use um "e" comercial (&) para separar os parâmetros.

Por exemplo, a solicitação a seguir recupera o número de endpoints exclusivos aos quais as mensagens foram entregues, por cada uma das campanhas de um projeto, de 19 de julho de 2019 até 26 de julho de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/
kpis/daterange/unique-deliveries-grouped-by-campaign?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

Onde:

- pinpoint.us-east-1.amazonaws.com é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto.
- 12345678901234567890123456789012345example é o identificador exclusivo do projeto associado às campanhas.
- unique-deliveries-grouped-by-campaign é o valor kpi-name da métrica de aplicativo de entregas de endpoint, agrupadas por campanha, que é a métrica que retorna o número de endpoints exclusivos aos quais as mensagens foram entregues, por cada campanha.
- 2019-07-19T00:00:00Z é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- 2019-07-26T23:59:59Z é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

AWS CLI

Para consultar dados de análise de várias campanhas usando a AWS CLI, use o comando get-application-date-range-kpi e especifique os valores apropriados para os parâmetros necessários:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Onde:

- *application-id* é o identificador exclusivo do projeto associado às campanhas.
- *kpi-name* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que recupera os dados de um intervalo de datas específico, inclua os valores e parâmetros `start-time` e `end-time` na consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a solicitação a seguir recupera o número de endpoints exclusivos aos quais as mensagens foram entregues, por cada uma das campanhas de um projeto, de 19 de julho de 2019 até 26 de julho de 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name unique-deliveries-grouped-by-campaign ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

Onde:

- `12345678901234567890123456example` é o identificador exclusivo do projeto associado à campanha.
- `unique-deliveries-grouped-by-campaign` é o valor `kpi-name` da métrica de aplicativo de entregas de endpoint, agrupadas por campanha, que é a métrica que retorna o número de endpoints exclusivos aos quais as mensagens foram entregues, por cada campanha.
- `2019-07-19T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-07-26T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

SDK for Java

Para consultar dados analíticos de diversas campanhas usando o AWS SDK for Java, use o método GetApplicationDateRangeKpiRequest da API Métricas de aplicativo. Especifique os valores apropriados para os parâmetros necessários:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Onde:

- *applicationId* é o identificador exclusivo do projeto associado às campanhas.
- *kpiName* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que recupera os dados de um intervalo de datas específico, inclua os valores e parâmetros `startTime` e `endTime` na consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a solicitação a seguir recupera o número de endpoints exclusivos aos quais as mensagens foram entregues, por cada uma das campanhas de um projeto, de 19 de julho de 2019 até 26 de julho de 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("unique-deliveries-grouped-by-campaign")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

Onde:

- `12345678901234567890123456example` é o identificador exclusivo do projeto associado às campanhas.
- `unique-deliveries-grouped-by-campaign` é o valor `kpi-name` da métrica de aplicativo de entregas de endpoint, agrupadas por campanha, que é a métrica que retorna o número de endpoints exclusivos aos quais as mensagens foram entregues, por cada campanha.
- `2019-07-19T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

- `2019-07-26T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

Depois de enviar sua consulta, o Amazon Pinpoint retorna os resultados dela em uma resposta JSON. A estrutura dos resultados varia dependendo da métrica consultada. Algumas métricas retornam apenas um valor. Outras métricas retornam vários valores, e esses valores são agrupados por um campo relevante. Se uma métrica retornar vários valores, a resposta JSON incluirá um campo que indica qual campo foi usado para agrupar os dados.

Por exemplo, a métrica de aplicativo de entregas de endpoint, agrupadas por campanha (`unique-deliveries-grouped-by-campaign`), que é usada nos exemplos anteriores, retorna vários valores: o número de endpoints exclusivos aos quais as mensagens foram entregues, para cada campanha associada a um projeto. Nesse caso, a resposta JSON é a seguinte:


```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
```

```
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```


Nesse caso, o campo `GroupedBys` indica que os valores são agrupados por ID de campanha (`CampaignId`).

Para saber mais sobre a estrutura dos resultados da consulta, consulte Usar resultados da consulta.

Consultar dados de análise do Amazon Pinpoint para mensagens transacionais

Além de usar as páginas de análise no console do Amazon Pinpoint, você pode usar as APIs do Amazon Pinpoint Analytics para consultar dados analíticos de um subconjunto de métricas padrão que fornecem informações sobre as tendências de entrega e engajamento para as mensagens transacionais enviadas para um projeto.

Cada métrica é um valor mensurável, também chamado de indicador-chave de desempenho (KPI), que pode ajudá-lo a monitorar e avaliar o desempenho de mensagens transacionais. Por exemplo, você pode usar uma métrica para descobrir quantas mensagens transacionais de e-mail ou SMS você enviou ou quantas dessas mensagens foram entregues aos destinatários. O Amazon Pinpoint coleta e agrega automaticamente esses dados para todas as mensagens transacionais de e-mail e SMS que você envia para um projeto. Os dados são armazenados por 90 dias.

Se você usar as APIs do Amazon Pinpoint Analytics para consultar dados, poderá escolher várias opções que definem o escopo, os dados, o agrupamento e os filtros para a consulta. Faça isso usando parâmetros que especificam o projeto e a métrica que você deseja consultar, além de qualquer filtro baseado em datas que você deseja aplicar.

Este tópico explica e fornece exemplos de como escolher essas opções e consultar dados de mensagens transacionais para um projeto.

Pré-requisitos

Antes de consultar dados analíticos para mensagens transacionais, ele ajuda a coletar as seguintes informações, usadas para definir sua consulta:

- ID do projeto: o identificador exclusivo do projeto do qual as mensagens foram enviadas. Na API do Amazon Pinpoint, esse valor é armazenado na propriedade `application-id`. No console do Amazon Pinpoint, esse valor é exibido como o ID do projeto na página Todos os projetos.
- Intervalo de datas: opcionalmente, a primeira e a última data e hora do intervalo de datas para o qual consultar dados. Os intervalos de datas são inclusivos e devem estar limitados a 31 dias ou menos. Além disso, eles devem começar em menos de 90 dias a partir do dia atual. Se você não especificar um intervalo de datas, o Amazon Pinpoint consultará automaticamente os dados dos 31 dias corridos anteriores.

- Métrica: o nome da métrica a ser consultada, mais especificamente o valor `kpi-name` da métrica. Para obter uma lista completa das métricas compatíveis e o valor `kpi-name` de cada uma, consulte Métricas padrão.

Também ajuda determinar se você deseja agrupar os dados por um campo relevante. Se o fizer, poderá simplificar a análise e a emissão de relatórios escolhendo uma métrica projetada para agrupar dados automaticamente para você. Por exemplo, o Amazon Pinpoint fornece várias métricas padrão que relatam o número de mensagens SMS transacionais que foram entregues aos destinatários. Uma dessas métricas agrupa automaticamente os dados por data (`txn-sms-delivered-grouped-by-date`). Outra métrica agrupa automaticamente os dados por país ou região (`txn-sms-delivered-grouped-by-country`). Uma terceira métrica simplesmente retorna um único valor: a quantidade de mensagens que foram entregues aos destinatários (`txn-sms-delivered`). Se você não conseguir encontrar uma métrica padrão que agrupe dados da maneira desejada, poderá desenvolver uma série de consultas que retornem os dados desejados. Depois, é possível dividir manualmente ou combinar os resultados da consulta em grupos personalizados criados.

Por fim, é importante verificar se você está autorizado a acessar os dados que deseja consultar. Para obter mais informações, consulte Políticas do IAM para consulta de dados de análise do Amazon Pinpoint.

Consultar dados para mensagens de e-mail transacionais

Para consultar os dados de mensagens de e-mail transacionais que foram enviadas para um projeto, use a API de Métricas de aplicativo e especifique valores para os seguintes parâmetros necessários:

- application-id: o ID do projeto, que é o identificador exclusivo do projeto. No Amazon Pinpoint, os termos projeto e aplicativo têm o mesmo significado.
- kpi-name: o nome da métrica a ser consultada. Esse valor descreve a métrica associada e consiste em dois ou mais termos, que são compostos por caracteres alfanuméricos minúsculos, separados por um hífen. Para obter uma lista completa das métricas compatíveis e o valor `kpi-name` de cada uma, consulte Métricas padrão.

Também é possível aplicar um filtro que consulta os dados para um intervalo de datas específico. Se você não especificar um intervalo de datas, o Amazon Pinpoint retornará os dados para os 31 dias corridos anteriores. Para filtrar os dados por datas diferentes, use os parâmetros de intervalo de datas compatíveis para especificar a primeira e a última data e hora do intervalo de datas. Os

valores devem estar no formato ISO 8601 estendido e usar o Tempo Universal Coordenado (UTC), por exemplo, `2019-09-06T20:00:00Z` para 20h UTC 6 de setembro de 2019. Os intervalos de datas são inclusivos e devem estar limitados a 31 dias ou menos. Adicionalmente, a primeira data e hora devem ser inferiores a 90 dias a partir do dia atual.

Os exemplos a seguir mostram como consultar dados de análise para mensagens de e-mail transacionais usando a API REST do Amazon Pinpoint, o AWS CLI e o AWS SDK for Java. Você pode usar qualquer AWS SDK compatível para consultar dados analíticos para mensagens transacionais. Os exemplos da AWS CLI são formatados para o Microsoft Windows. Para Unix, Linux e macOS, substitua o caractere de continuação de linha circunflexo (^) por uma barra invertida (\).

REST API

Para consultar dados de análise de mensagens de e-mail transacionais usando a API REST do Amazon Pinpoint, envie uma solicitação HTTP(S) GET para o URI de Métricas de aplicativo. No URI, especifique os valores apropriados para os parâmetros de caminho necessários:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Onde:

- *`endpoint`* é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto.
- *`application-id`* é o identificador exclusivo do projeto.
- *`kpi-name`* é o valor `kpi-name` da métrica a consultar.

Todos os parâmetros devem ser codificados por URL.

Para aplicar um filtro que consulta os dados de um intervalo de datas específico, acrescente os valores e parâmetros de consulta `start-time` e `end-time` ao URI. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Use um "e" comercial (&) para separar os parâmetros.

Por exemplo, a seguinte solicitação recupera o número de mensagens de e-mail transacionais que foram enviadas para um projeto de 6 de setembro de 2019 a 13 de setembro de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-13T23:59:59Z
```

Onde:

- `pinpoint.us-east-1.amazonaws.com` é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto.
- `12345678901234567890123456example` é o identificador exclusivo do projeto.
- `txn-emails-sent` é o valor `kpi-name` da métrica do aplicativo envios, que é a métrica que relata o número de mensagens de e-mail transacionais enviadas para um projeto.
- `2019-09-06T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-09-13T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

AWS CLI

Para consultar dados de análise de mensagens de e-mail transacionais usando a AWS CLI, use o comando get-application-date-range-kpi e especifique os valores apropriados para os parâmetros necessários:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Onde:

- *`application-id`* é o identificador exclusivo do projeto.
- *`kpi-name`* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que consulta os dados de um intervalo de datas específico, adicione os valores e parâmetros `start-time` e `end-time` à consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a seguinte solicitação recupera o número de mensagens de e-mail transacionais que foram enviadas para um projeto de 6 de setembro de 2019 a 13 de setembro de 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-emails-sent ^
```

```
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-13T23:59:59Z
```

Onde:

- `12345678901234567890123example` é o identificador exclusivo do projeto.
- `txn-emails-sent` é o valor `kpi-name` da métrica do aplicativo envios, que é a métrica que relata o número de mensagens de e-mail transacionais enviadas para um projeto.
- `2019-09-06T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-09-13T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

SDK for Java

Para consultar dados analíticos para mensagens de e-mail transacionais usando o AWS SDK for Java, use o método GetApplicationDateRangeKpiRequest da API Métricas de aplicativo. Especifique os valores apropriados para os parâmetros necessários:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Onde:

- *`applicationId`* é o identificador exclusivo do projeto.
- *`kpiName`* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que consulta os dados de um intervalo de datas específico, inclua os valores e parâmetros `startTime` e `endTime` na consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a seguinte solicitação recupera o número de mensagens de e-mail transacionais que foram enviadas para um projeto de 6 de setembro de 2019 a 13 de setembro de 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123example")
```

```
        .withKpiName("txn-emails-sent")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

Onde:

- `12345678901234567890123456example` é o identificador exclusivo do projeto.
- `txn-emails-sent` é o valor `kpi-name` da métrica do aplicativo envios, que é a métrica que relata o número de mensagens de e-mail transacionais enviadas para um projeto.
- `2019-09-06T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-09-13T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

Depois de enviar sua consulta, o Amazon Pinpoint retorna os resultados dela em uma resposta JSON. A estrutura dos resultados varia dependendo da métrica consultada. Algumas métricas retornam apenas um valor. Por exemplo, a métrica do aplicativo sends (`txn-emails-sent`), usada nos exemplos anteriores, retorna um valor: o número de mensagens de e-mail transacionais enviadas de um projeto. Nesse caso, a resposta JSON é a seguinte:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-13T23:59:59Z",
        "KpiName":"txn-emails-sent",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"TxnEmailsSent",
                            "Type":"Double",
                            "Value":"62.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
```

```
}
```

Outras métricas retornam vários valores e agrupam os valores por um campo relevante. Se uma métrica retornar vários valores, a resposta JSON incluirá um campo que indica qual campo foi usado para agrupar os dados.

Para saber mais sobre a estrutura dos resultados da consulta, consulte Usar resultados da consulta.

Consultar dados para mensagens SMS transacionais

Para consultar os dados de mensagens SMS transacionais que foram enviadas para um projeto, use a API de Métricas de aplicativo e especifique valores para os seguintes parâmetros necessários:

- application-id: o ID do projeto, que é o identificador exclusivo do projeto. No Amazon Pinpoint, os termos projeto e aplicativo têm o mesmo significado.
- kpi-name: o nome da métrica a ser consultada. Esse valor descreve a métrica associada e consiste em dois ou mais termos, que são compostos por caracteres alfanuméricos minúsculos, separados por um hífen. Para obter uma lista completa das métricas compatíveis e o valor `kpi-name` de cada uma, consulte Métricas padrão.

Também é possível aplicar um filtro que consulta os dados para um intervalo de datas específico. Se você não especificar um intervalo de datas, o Amazon Pinpoint retornará os dados para os 31 dias corridos anteriores. Para filtrar os dados por datas diferentes, use os parâmetros de intervalo de datas compatíveis para especificar a primeira e a última data e hora do intervalo de datas. Os valores devem estar no formato ISO 8601 estendido e usar o Tempo Universal Coordenado (UTC), por exemplo, `2019-09-06T20:00:00Z` para 20h UTC 6 de setembro de 2019. Os intervalos de datas são inclusivos e devem estar limitados a 31 dias ou menos. Adicionalmente, a primeira data e hora devem ser inferiores a 90 dias a partir do dia atual.

Os exemplos a seguir mostram como consultar dados de análise para mensagens SMS transacionais usando a API REST do Amazon Pinpoint, o AWS CLI e o AWS SDK for Java. Você pode usar qualquer AWS SDK compatível para consultar dados analíticos para mensagens transacionais. Os exemplos da AWS CLI são formatados para o Microsoft Windows. Para Unix, Linux e macOS, substitua o caractere de continuação de linha circunflexo (^) por uma barra invertida (\).

REST API

Para consultar dados de análise de mensagens SMS transacionais usando a API REST do Amazon Pinpoint, envie uma solicitação HTTP(S) GET para o URI de Métricas de aplicativo. No URI, especifique os valores apropriados para os parâmetros de caminho necessários:

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

Onde:

- *endpoint* é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto.
- *application-id* é o identificador exclusivo do projeto.
- *kpi-name* é o valor `kpi-name` da métrica a consultar.

Todos os parâmetros devem ser codificados por URL.

Para aplicar um filtro que recupera os dados de um intervalo de datas específico, acrescente os valores e parâmetros de consulta `start-time` e `end-time` ao URI. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Use um "e" comercial (&) para separar os parâmetros.

Por exemplo, a seguinte solicitação recupera o número de mensagens SMS transacionais que foram enviadas todos os dias de 6 de setembro de 2019 a 8 de setembro de 2019:

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-08T23:59:59Z
```

Onde:

- `pinpoint.us-east-1.amazonaws.com` é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o projeto.
- `12345678901234567890123456example` é o identificador exclusivo do projeto.
- `txn-sms-sent-grouped-by-date` é o valor `kpi-name` da métrica do aplicativo envios, agrupados por data, que é a métrica que retorna o número de mensagens SMS transacionais que foram enviadas durante cada dia do intervalo de datas.

- `2019-09-06T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-09-08T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

AWS CLI

Para consultar dados de análise de mensagens SMS transacionais usando a AWS CLI, use o comando get-application-date-range-kpi e especifique os valores apropriados para os parâmetros necessários:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

Onde:

- *`application-id`* é o identificador exclusivo do projeto.
- *`kpi-name`* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que recupera os dados de um intervalo de datas específico, inclua os valores e parâmetros `start-time` e `end-time` na consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a seguinte solicitação recupera o número de mensagens SMS transacionais que foram enviadas todos os dias de 6 de setembro de 2019 a 8 de setembro de 2019:

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-sms-sent-grouped-by-date ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-08T23:59:59Z
```

Onde:

- `12345678901234567890123456example` é o identificador exclusivo do projeto.

- `txn-sms-sent-grouped-by-date` é o valor `kpi-name` da métrica do aplicativo envios, agrupados por data, que é a métrica que retorna o número de mensagens SMS transacionais que foram enviadas durante cada dia do intervalo de datas.
- `2019-09-06T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-09-08T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

SDK for Java

Para consultar dados analíticos para mensagens SMS transacionais usando o AWS SDK for Java, use o método GetApplicationDateRangeKpiRequest da API de Métricas de aplicativo e especifique os valores apropriados para os parâmetros necessários:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

Onde:

- *`applicationId`* é o identificador exclusivo do projeto.
- *`kpiName`* é o valor `kpi-name` da métrica a consultar.

Para aplicar um filtro que recupera os dados de um intervalo de datas específico, inclua os valores e parâmetros `startTime` e `endTime` na consulta. Ao usar esses parâmetros, é possível especificar a primeira e a última data e hora, no formato ISO 8601 estendido, de um intervalo de datas inclusivo para o qual recuperar os dados. Por exemplo, a seguinte solicitação recupera o número de mensagens SMS transacionais que foram enviadas todos os dias de 6 de setembro de 2019 a 8 de setembro de 2019:

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123example")
        .withKpiName("txn-sms-sent-grouped-by-date")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-08T23:59:59Z")));
```

Onde:

- `12345678901234567890123456example` é o identificador exclusivo do projeto.
- `txn-sms-sent-grouped-by-date` é o valor `kpi-name` da métrica do aplicativo envios, agrupados por data, que é a métrica que retorna o número de mensagens SMS transacionais que foram enviadas durante cada dia do intervalo de datas.
- `2019-09-06T00:00:00Z` é a primeira data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.
- `2019-09-08T23:59:59Z` é a última data e hora para as quais recuperar dados, como parte de um intervalo de datas inclusivo.

Depois de enviar sua consulta, o Amazon Pinpoint retorna os resultados dela em uma resposta JSON. A estrutura dos resultados varia dependendo da métrica consultada. Algumas métricas retornam apenas um valor. Outras métricas retornam vários valores e agrupam esses valores por um campo relevante. Se uma métrica retornar vários valores, a resposta JSON incluirá um campo que indica qual campo foi usado para agrupar os dados.

Por exemplo, a métrica do aplicativo sends, grouped by date (`txn-sms-sent-grouped-by-date`), usada nos exemplos anteriores, retorna vários valores: o número de mensagens SMS transacionais enviadas durante cada dia do intervalo de datas especificado. Nesse caso, a resposta JSON é a seguinte:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-08T23:59:59Z",
        "KpiName":"txn-sms-sent-grouped-by-date",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-06"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
                            "Type":"Double",
```

```
                        "Value":"29.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-07"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"35.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-08"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"10.0"
                    }
                ]
            }
        ]
    },
    "StartTime":"2019-09-06T00:00:00Z"
  }
}
```

Nesse caso, o campo `GroupedBys` indica que os valores são agrupados por dia do calendário (`Date`). Isto significa que:

- 29 mensagens foram enviadas em 6 de setembro de 2019.
- 35 mensagens foram enviadas em 7 de setembro de 2019.
- 10 mensagens foram enviadas em 8 de setembro de 2019.

Para saber mais sobre a estrutura dos resultados da consulta, consulte Usar resultados da consulta.

Usar os resultados da consulta de análises do Amazon Pinpoint

Quando você usa APIs do Amazon Pinpoint Analytics para consultar dados analíticos, o Amazon Pinpoint retorna os resultados em uma resposta JSON. Para métricas de aplicativos, métricas de campanha e métricas de engajamento de jornada, os dados na resposta aderem a um esquema JSON padrão para relatórios de dados analíticos do Amazon Pinpoint.

Isso significa que você pode usar a linguagem de programação ou a ferramenta de sua escolha para implementar uma solução personalizada que consulta os dados de uma ou mais métricas, captura os resultados de cada consulta e, depois, grava os resultados em uma tabela, um objeto ou outro local. Em seguida, você pode trabalhar com os resultados da consulta nesse local usando outro serviço ou aplicativo.

Por exemplo, é possível:

- Crie um painel personalizado que consulta regularmente um conjunto de métricas e exiba os resultados usando sua estrutura de visualização de dados preferida.
- Crie um relatório que rastreia as taxas de engajamento consultando as métricas apropriadas e exibindo os resultados em um gráfico ou outro tipo de relatório criado por você.
- Analise e grave dados analíticos em um formato de armazenamento específico e, em seguida, migre os resultados para uma solução de armazenamento de longo prazo.

Observe que as APIs do Amazon Pinpoint Analytics não foram projetadas para criar ou armazenar objetos persistentes que você pode ler ou usar posteriormente em um projeto do Amazon Pinpoint ou em sua conta do Amazon Pinpoint. Em vez disso, as APIs são projetadas para ajudá-lo a recuperar dados analíticos e transferir esses dados para outros serviços e aplicativos para análise, armazenamento ou emissão de relatórios adicionais. Elas fazem isso em parte usando a mesma estrutura de resposta JSON e o mesmo esquema para todos os dados analíticos que você pode

consultar programaticamente para métricas de aplicativos, métricas de campanha e métricas de engajamento de jornada.

Este tópico explica a estrutura, os objetos e os campos em uma resposta JSON a uma consulta para uma métrica de aplicativo, métrica de campanha ou métrica de engajamento de jornada. Para obter informações sobre os campos em uma resposta JSON a uma consulta para uma métrica de execução de jornada ou métrica de execução de atividade de jornada, consulte Métricas de análise do Amazon Pinpoint padrão.

Estrutura JSON

Para ajudá-lo a analisar e usar os resultados da consulta, as APIs do Amazon Pinpoint Analytics usam a mesma estrutura de resposta JSON para todos os dados analíticos do Amazon Pinpoint que você pode consultar programaticamente para métricas de aplicativos, métricas de campanha e métricas de engajamento de jornada. Cada resposta JSON especifica os valores que definiram a consulta, como o ID do projeto (`ApplicationId`). A resposta também inclui um (e apenas um) objeto `KpiResult`. O objeto `KpiResult` contém o conjunto geral de resultados para uma consulta.

Cada objeto `KpiResult` contém um objeto `Rows`. Esta é uma matriz de objetos que contêm resultados de consulta e metadados relevantes sobre os valores nesses resultados. A estrutura e o conteúdo de um objeto `Rows` têm as seguintes características gerais:

- Cada linha de resultados da consulta é um objeto JSON separado, chamado `Values`, no objeto `Rows`. Por exemplo, se uma consulta retornar três valores, o objeto `Rows` conterá três objetos `Values`. Cada objeto `Values` contém um resultado individual para a consulta.
- Cada coluna de resultados de consulta é uma propriedade do objeto `Values` ao qual ela se aplica. O nome da coluna é armazenado no campo `Key` do objeto `Values`.
- Para resultados de consulta agrupados, cada objeto `Values` tem um objeto `GroupedBys` associado. O objeto `GroupedBys` indica qual campo foi usado para agrupar os resultados. Ele também fornece o valor de agrupamento para o objeto `Values` associado.
- Se os resultados da consulta para uma métrica forem nulos, o objeto `Rows` estará vazio.

Além dessas características gerais, a estrutura e o conteúdo do objeto `Rows` variam dependendo da métrica. Isso ocorre porque o Amazon Pinpoint oferece suporte a dois tipos de métrica: métricas de valor único e métricas de vários valores.

Uma métrica de valor único fornece apenas um valor cumulativo. Um exemplo é a porcentagem de mensagens que foram entregues aos destinatários por todas as execuções de uma campanha. Uma

métrica de vários valores fornece mais de um valor e agrupa esses valores por um campo relevante. Um exemplo é a porcentagem de mensagens que foram entregues aos destinatários para cada execução de uma campanha, agrupadas por execução de campanha.

Você pode determinar rapidamente se uma métrica é uma métrica de valor único ou uma métrica de vários valores referindo-se ao nome da métrica. Se o nome não contiver `grouped-by`, será uma métrica de valor único. Se isso acontecer, é uma métrica de vários valores. Para obter uma lista completa das métricas que você pode consultar programaticamente, consulte Métricas de análise do Amazon Pinpoint padrão.

Métricas de valor único

Para uma métrica de valor único, o objeto `Rows` contém um objeto `Values` que:

- Especifica o nome amigável da métrica que foi consultada.
- Fornece o valor para a métrica que foi consultada.
- Identifica o tipo de dados do valor retornado.

Por exemplo, a seguinte resposta JSON contém os resultados da consulta para uma métrica de valor único. Essa métrica registra o número de endpoints exclusivos aos quais as mensagens foram entregues por todas as campanhas associadas a um projeto, de 1 de agosto de 2019 a 31 de agosto de 2019:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"1368.0"
                        }
                    ]
                }
            ]
```

```
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Neste exemplo, a resposta indica que todas as campanhas do projeto entregaram mensagens para 1.368 endpoints exclusivos de 1 de agosto de 2019 a 31 de agosto de 2019, onde:

- `Key` é o nome amigável da métrica cujo valor é especificado no campo `Value` (`UniqueDeliveries`).
- `Type` é o tipo de dados do valor especificado no campo `Value` (`Double`).
- `Value` é o valor real para a métrica que foi consultada, incluindo quaisquer filtros que foram aplicados (`1368.0`).

Se os resultados da consulta para uma métrica de valor único forem nulos (não maiores ou iguais a zero), o objeto `Rows` estará vazio. O Amazon Pinpoint retornará um valor nulo para uma métrica se não houver nenhum dado a ser retornado para a métrica. Por exemplo:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"234567890123456789012345example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[

            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Métricas de vários valores

A estrutura e o conteúdo do objeto `Rows` para uma métrica de vários valores são principalmente os mesmos que uma métrica de valor único. O objeto `Rows` para uma métrica de vários valores também contém um objeto `Values`. O objeto `Values` especifica o nome amigável da métrica que foi consultada, fornece o valor para essa métrica e identifica o tipo de dados desse valor.

No entanto, o objeto `Rows` para uma métrica de vários valores também contém um ou mais objetos `GroupedBy`. Há um objeto `GroupedBy` para cada objeto `Values` nos resultados da consulta. O objeto `GroupedBy` indica qual campo foi usado para agrupar os dados nos resultados e o tipo de dados desse campo. Ele também indica o valor de agrupamento para esse campo (para o objeto `Values` associado).

Por exemplo, a seguinte resposta JSON contém os resultados da consulta para uma métrica de vários valores que relata o número de endpoints exclusivos aos quais as mensagens foram entregues, para cada campanha associada a um projeto, de 1 de agosto de 2019 a 31 de agosto de 2019:

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
```

```
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

Neste exemplo, a resposta indica que três das campanhas do projeto entregaram mensagens para endpoints exclusivos de 1 de agosto de 2019 a 31 de agosto de 2019. Para cada uma dessas campanhas, a repartição das contagens de entregas é:

- Campanha `80b8efd84042ff8d9c96ce2f8example` entregou mensagens para 123 endpoints exclusivos.
- Campanha `810c7aab86d42fb2b56c8c966example` entregou mensagens para 456 endpoints exclusivos.
- Campanha `42d8c7eb0990a57ba1d5476a3example` entregou mensagens para 789 endpoints exclusivos.

Onde a estrutura geral dos objetos e campos é:

- `GroupedBys.Key`: o nome da propriedade ou campo que armazena o valor de agrupamento especificado no campo `GroupedBys.Value` (`CampaignId`).
- `GroupedBys.Type`: o tipo de dados do valor especificado no campo `GroupedBys.Value` (`String`).
- `GroupedBys.Value`: o valor real para o campo que foi usado para agrupar os dados, conforme especificado no campo `GroupedBys.Key` (ID da campanha).
- `Values.Key`: o nome amigável da métrica cujo valor é especificado no campo `Values.Value` (`UniqueDeliveries`).
- `Values.Type`: o tipo de dados do valor especificado no campo `Values.Value` (`Double`).
- `Values.Value`: o valor real para a métrica que foi consultada, incluindo quaisquer filtros que foram aplicados.

Se os resultados da consulta para uma métrica de vários valores forem nulos (não maiores ou iguais a zero) para um projeto específico, campanha ou outro recurso, o Amazon Pinpoint não retornará nenhum objeto ou campo para o recurso. Se os resultados da consulta para uma métrica de vários valores forem nulos para todos os recursos, o Amazon Pinpoint retornará um objeto `Rows` vazio.

Objetos e campos JSON

Além de especificar os valores que definiram uma consulta, como o ID do projeto (`ApplicationId`), cada resposta JSON a uma consulta para uma métrica de aplicativo, métrica de campanha ou métrica de engajamento de jornada inclui um objeto `KpiResult`. Este objeto contém o conjunto de resultados geral de uma consulta, que você pode analisar para enviar dados analíticos para outro serviço ou aplicativo. Cada objeto `KpiResult` contém alguns ou todos os seguintes objetos e campos padrão, dependendo da métrica.

Objeto ou campo	Descrição
`Rows`	Uma matriz de objetos que contém o conjunto de resultados para uma consulta.
`Rows.GroupedBys`	Para uma métrica de vários valores, uma matriz de campos que define o campo e os

Objeto ou campo	Descrição
	valores que foram usados para agrupar dados nos resultados da consulta.
`Rows.GroupedBys.Key`	Para uma métrica de vários valores, o nome da propriedade ou campo que armazena o valor especificado no campo `GroupedBys.Value` .
`Rows.GroupedBys.Type`	Para uma métrica de vários valores, o tipo de dados do valor especificado no campo `GroupedBys.Value` .
`Rows.GroupedBys.Value`	Para uma métrica de vários valores, o valor real para o campo que foi usado para agrupar dados nos resultados da consulta. Esse valor se correlaciona a um objeto `Values` associado .
`Rows.Values`	Uma matriz de campos que contém resultados de consulta.
`Rows.Values.Key`	O nome amigável da métrica que foi consultada. O valor da métrica é especificado no campo `Values.Value` .
`Rows.Values.Type`	O tipo de dados do valor especificado no campo `Values.Value` .
`Rows.Values.Value`	O valor real para a métrica que foi consultada, incluindo quaisquer filtros que foram aplicados.

Para obter informações sobre os campos em uma resposta JSON a uma consulta para uma métrica de execução de jornada ou métrica de execução de atividade de jornada, consulte Métricas de análise do Amazon Pinpoint padrão.

Registro de chamadas de API do Amazon Pinpoint com AWS CloudTrail

O Amazon Pinpoint é integrado com AWS CloudTrail, que é um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon Pinpoint. CloudTrail captura chamadas de API para o Amazon Pinpoint como eventos. As chamadas capturadas incluem as chamadas do console do Amazon Pinpoint e as chamadas de código para as operações da API do Amazon Pinpoint.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon Simple Storage Service (Amazon S3), incluindo eventos para o Amazon Pinpoint. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes usando o histórico de eventos no CloudTrail console. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Pinpoint, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o Guia AWS CloudTrail do usuário.

Informações do Amazon Pinpoint em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando uma atividade de evento suportada ocorre no Amazon Pinpoint, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos do Amazon Pinpoint, 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, a trilha se aplica a todas as AWS regiões. 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 mais informações, consulte:

- Visão geral da criação de uma trilha

- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e Recebendo arquivos de CloudTrail log de várias contas

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 raiz ou de AWS Identity and Access Management usuário.
- 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 Elemento userIdentity do CloudTrail .

Você poder criar uma trilha e armazenar os arquivos de log no bucket do Amazon S3 pelo tempo que quiser. Além disso, você pode definir as regras de ciclo de vida do Amazon S3 para arquivar ou excluir os arquivos de log automaticamente. Por padrão, os arquivos de log são criptografados com criptografia do lado do servidor (SSE) do Amazon S3.

Para ser notificado sobre a entrega do arquivo de log, configure CloudTrail para publicar notificações do Amazon SNS quando novos arquivos de log forem entregues. Para obter mais informações, consulte Configuração de notificações do Amazon SNS para. CloudTrail

Você também pode agregar arquivos de log do Amazon Pinpoint de AWS várias regiões e AWS várias contas em um único bucket do Amazon S3. Para obter mais informações, consulte Recebimento de arquivos de CloudTrail log de várias regiões e Recebimento de arquivos de CloudTrail log de várias contas.

Você pode usar CloudTrail para registrar ações para as seguintes APIs do Amazon Pinpoint:

- API do Amazon Pinpoint
- API de SMS e voz do Amazon Pinpoint

Ações da API Amazon Pinpoint que podem ser registradas por CloudTrail

A API do Amazon Pinpoint oferece suporte ao registro das seguintes ações como eventos em arquivos de CloudTrail log:

- CreateApp
- CreateCampaign
- CreateEmailTemplate
- CreateExportJob
- CreateImportJob
- CreateJourney
- CreatePushTemplate
- CreateRecommenderConfiguration
- CreateSegment
- CreateSmsTemplate
- CreateVoiceTemplate
- DeleteAdmChannel
- DeleteApnsChannel
- DeleteApnsSandboxChannel
- DeleteApnsVoipChannel
- DeleteApnsVoipSandboxChannel
- DeleteApp
- DeleteBaiduChannel
- DeleteCampaign
- DeleteEmailChannel
- DeleteEmailTemplate
- DeleteEndpoint
- DeleteEventStream
- DeleteGcmChannel

- DeleteJourney
- DeletePushTemplate
- DeleteRecommenderConfiguration
- DeleteSegment
- DeleteSmsChannel
- DeleteSmsTemplate
- DeleteUserEndpoints
- DeleteVoiceChannel
- DeleteVoiceTemplate
- GetAdmChannel
- GetApnsChannel
- GetApnsSandboxChannel
- GetApnsVoipChannel
- GetApnsVoipSandboxChannel
- GetApp
- GetApplicationDateRangeKpi
- GetApplicationSettings
- GetApps
- GetBaiduChannel
- GetCampaign
- GetCampaignActivities
- GetCampaignDateRangeKpi
- GetCampaignVersion
- GetCampaignVersions
- GetCampaigns
- GetChannels
- GetEmailChannel
- GetEmailTemplate
- GetEndpoint
- GetEventStream

- GetExportJob
- GetExportJobs
- GetGcmChannel
- GetImportJob
- GetImportJobs
- GetJourney
- GetJourneyDateRangeKpi
- GetJourneyExecutionActivityMetrics
- GetJourneyExecutionMetrics
- GetPushTemplate
- GetRecommenderConfiguration
- GetRecommenderConfigurations
- GetSegment
- GetSegmentExportJobs
- GetSegmentImportJobs
- GetSegmentVersion
- GetSegmentVersions
- GetSegments
- GetSmsChannel
- GetSmsTemplate
- GetUserEndpoints
- GetVoiceChannel
- GetVoiceTemplate
- ListJourneys
- ListTagsForResource
- ListTemplates
- ListTemplateVersions
- PhoneNumberValidate
- PutEventStream

- RemoveAttributes
- TagResource
- UntagResource
- UpdateAdmChannel
- UpdateApnsChannel
- UpdateApnsSandboxChannel
- UpdateApnsVoipChannel
- UpdateApnsVoipSandboxChannel
- UpdateApplicationSettings
- UpdateBaiduChannel
- UpdateCampaign
- UpdateEmailChannel
- UpdateEmailTemplate
- UpdateEndpoint
- UpdateEndpointsBatch
- UpdateGcmChannel
- UpdateJourney
- UpdateJourneyState
- UpdatePushTemplate
- UpdateRecommenderConfiguration
- UpdateSegment
- UpdateSmsChannel
- UpdateSmsTemplate
- UpdateTemplateActiveVersion
- UpdateVoiceChannel
- UpdateVoiceTemplate

As seguintes ações da API do Amazon Pinpoint não estão registradas: CloudTrail

- PutEvents

- SendMessages
- SendUsersMessages

Ações da API de e-mail do Amazon Pinpoint que podem ser registradas por CloudTrail

A API de e-mail do Amazon Pinpoint suporta o registro das seguintes ações como eventos em arquivos de CloudTrail log:

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- CreateDedicatedIpPool
- CreateEmailIdentity
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- DeleteDedicatedIpPool
- DeleteEmailIdentity
- GetAccount
- GetConfigurationSet
- GetConfigurationSetEventDestinations
- GetDedicatedIp
- GetDedicatedIps
- GetEmailIdentity
- ListConfigurationSets
- ListDedicatedIpPools
- ListEmailIdentities
- PutAccountDedicatedIpWarmupAttributes
- PutAccountSendingAttributes
- PutConfigurationSetDeliveryOptions
- PutConfigurationSetReputationOptions
- PutConfigurationSetSendingOptions

- PutConfigurationSetTrackingOptions
- PutDedicatedIpInPool
- PutDedicatedIpWarmupAttributes
- PutEmailIdentityDkimAttributes
- PutEmailIdentityFeedbackAttributes
- PutEmailIdentityMailFromAttributes
- UpdateConfigurationSetEventDestination

A seguinte ação da API de e-mail do Amazon Pinpoint não está conectada: CloudTrail

- SendEmail

Ações da API de voz e SMS do Amazon Pinpoint versão 1 que podem ser registradas por CloudTrail

A API Amazon Pinpoint SMS and Voice versão 1 suporta o registro das seguintes ações como eventos em arquivos de CloudTrail log:

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- GetConfigurationSetEventDestinations
- UpdateConfigurationSetEventDestination

A seguinte ação da API Amazon Pinpoint SMS and Voice versão 1 não está conectada: CloudTrail

- SendVoiceMessage

Exemplos: entradas de arquivo de log do Amazon Pinpoint

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 solicitação única de qualquer fonte. Ele inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra as `CreateCampaign` ações `GetCampaigns` e as ações da API do Amazon Pinpoint.


```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Jersey/${project.version} (HttpUrlConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        },
        "type": "Root"
      }
    },
    {
```

```
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "CreateCampaign",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T01:05:16Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": false,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "Description": "***",
        "HoldoutPercent": 0,
        "IsPaused": false,
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
        },
        "SegmentId": "exampleda204adf991a80281aa0e591",
        "SegmentVersion": 1,
        "application-id": "example71dfa4c1aab66332a5839798f"
      },
      "responseElements": {
        "ApplicationId": "example71dfa4c1aab66332a5839798f",
        "CreationDate": "2018-02-03T01:05:16.425Z",
        "Description": "***",
        "HoldoutPercent": 0,
        "Id": "example54a654f80948680cbba240ede",
        "IsPaused": false,
        "LastModifiedDate": "2018-02-03T01:05:16.425Z",
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
        },
        "SegmentId": "example4da204adf991a80281example",
        "SegmentVersion": 1,
```

```
        "State": {
          "CampaignStatus": "SCHEDULED"
        },
        "Version": 1
      },
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/userName",
        "principalId": "AIDAIHTHRCDA62EXAMPLE",
        "type": "IAMUser",
        "userName": "userName"
      }
    }
  ]
}
```

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra as `CreateConfigurationSetEventDestination` ações `CreateConfigurationSet` e na API de voz e SMS do Amazon Pinpoint.

```
{
  "Records": [
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:45:55Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSet",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "ConfigurationSetName":"MyConfigurationSet"
```

```
      },
      "responseElements":null,
      "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
      "eventID":"725843fc-8846-41f4-871a-7c52dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"123456789012"
    },
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:47:08Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSetEventDestination",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "EventDestinationName":"CloudWatchEventDestination",
        "ConfigurationSetName":"MyConfigurationSet",
        "EventDestination":{
          "Enabled":true,
          "MatchingEventTypes":[
            "INITIATED_CALL",
            "INITIATED_CALL"
          ],
          "CloudWatchLogsDestination":{
            "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
            "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
          }
        }
      },
      "responseElements":null,
      "requestID":"81de1e73-e20d-11e8-b158-d5536example",
      "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
      "readOnly":false,
```

```
      "eventType":"AwsApiCall",
      "recipientAccountId":"111122223333"
    }
  ]
}
```

Marcar recursos do Amazon Pinpoint

Uma tag é um rótulo que você pode definir e associar a recursos da AWS, incluindo certos tipos de recursos do Amazon Pinpoint. As tags podem ajudar a categorizar e gerenciar recursos de diferentes maneiras, como por finalidade, proprietário, ambiente ou outros critérios. Por exemplo, você pode usar tags para aplicar políticas ou automação ou para identificar recursos sujeitos a determinados requisitos de conformidade. Você pode adicionar tags aos seguintes tipos de recurso do Amazon Pinpoint:

- Campaigns (Campanhas)
- Modelos de mensagens
- Projetos (aplicativos)
- Segmentos

Um recurso pode ter até 50 tags.

Gerenciar tags

Cada tag consiste em uma chave de tag obrigatória e um valor de tag associado, ambos definidos por você. Uma chave de tag é um rótulo geral que atua como uma categoria para valores de tags mais específicos. Um valor de tag age como um descritor de uma chave de tag.

Uma chave de tag podem conter até 128 caracteres. Um valor de tag podem conter até 256 caracteres. Os caracteres podem ser letras Unicode, números, espaços em branco ou um dos seguintes símbolos: _ . : / = + -. As seguintes restrições adicionais se aplicam a tags:

- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.
- Para cada recurso associado, cada chave de tag deve ser exclusiva e pode ter apenas um valor.
- O prefixo `aws:` é reservado para uso pela AWS; você não pode usá-lo em nenhuma chave de tag ou valores definidos por você. Além disso, você não pode editar ou remover chaves de tag ou valores que usam esse prefixo. As tags que usam esse prefixo não contam com a cota de 50 tags por recurso.
- Você não pode atualizar ou excluir um recurso com base apenas em suas tags. Você também deve especificar o nome de recurso da Amazon (ARN) ou o ID do recurso, dependendo da operação usada.

- Você pode associar tags a recursos públicos ou compartilhados. No entanto, as tags estão disponíveis apenas para sua conta da AWS, e não para nenhuma outra conta que compartilhe o recurso. Além disso, as tags estão disponíveis apenas para recursos localizados na região da AWS especificada para sua conta da AWS.

Para adicionar, exibir, atualizar e remover chaves de tag e valores de recursos do Amazon Pinpoint, você pode usar a AWS Command Line Interface (AWS CLI), a API do Amazon Pinpoint, a API de marcação do AWS Resource Groups ou um AWS SDK. Para gerenciar chaves e valores de tags em todos os recursos da AWS localizados em uma Região específica da AWS para sua conta da AWS (incluindo os recursos do Amazon Pinpoint), use a API de marcação do AWS Resource Groups.

Usar tags nas políticas do IAM

Depois de começar a implementar tags, você pode aplicar permissões de nível de recurso baseadas em tags para políticas do AWS Identity and Access Management (IAM) e operações de API. Isto inclui operações que ão compatíveis com acrescentar tags a recursos quando estes são criados. Usando os tags dessa forma, é possível implementar um controle granular de quais grupos e usuários em sua conta da AWS têm permissão para criar e marcar recursos e quais grupos e usuários têm permissão para criar, atualizar e remover tags de maneira mais geral.

Por exemplo, é possível criar uma política que permita que um usuário tenha acesso completo a todos os recursos do Amazon Pinpoint, no qual o nome é um valor na tag `Owner` do recurso:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "ModifyResourceIfOwner",
         "Effect": "Allow",
         "Action": "mobiletargeting:*",
         "Resource": "*",
         "Condition": {
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

Se você definir permissões em nível de recurso e baseadas em tag, as permissões entrarão em vigor imediatamente. Isso significa que seus recursos ficam mais seguros assim que são criados, e você pode começar a aplicar rapidamente o uso de tags para novos recursos. Você também pode usar permissões em nível de recurso para controlar quais valores e chaves de tag podem ser associados a recursos novos e existentes. Para obter mais informações, consulte Controlar o acesso usando tags no Guia do usuário do IAM da AWS.

Adicionar tags a recursos

Os exemplos a seguir mostram como adicionar uma tag a um recurso do Amazon Pinpoint usando a AWS CLI e a API REST do Amazon Pinpoint. Você também pode usar qualquer SDK da AWS compatível para adicionar uma tag a um recurso.

Para adicionar uma tag a vários recursos do Amazon Pinpoint em uma única operação, use as operações de marcação dos grupos de recursos da AWS CLI ou da API de marcação do AWS Resource Groups.

Adicionar tags usando a API

Para criar um recurso e adicionar uma tag a ele usando a API REST do Amazon Pinpoint, envie uma solicitação POST ao URI do recurso apropriado. No corpo da solicitação, inclua o parâmetro `tags` e os valores. O exemplo a seguir mostra como especificar uma tag ao criar um novo projeto.

```
POST /v1/apps HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Cache-Control: no-cache

{
   "Name":"MyProject",
   "tags":{
      "key1":"value1"
   }
}
```

Para adicionar uma tag a um recurso existente, envie uma solicitação POST ao URI de Tags. Inclua o nome de recurso da Amazon (ARN) do recurso no URI. O ARN deve ser codificado por URL. No corpo da solicitação, inclua o parâmetro `tags` e os valores, conforme mostrado no exemplo a seguir.

```
POST /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache

{
   "tags":{
      "key1":"value1"
   }
}
```

Adição de tags usando a AWS CLI

Para criar um recurso e adicionar uma tag a ele usando a AWS CLI, use o comando `create` apropriado para o recurso. Inclua o parâmetro `tags` e os valores. O exemplo a seguir mostra como especificar tags ao criar um novo projeto.

Linux, macOS, or Unix

```
$ aws pinpoint create-app \
  --create-application-request '{
    "Name":"MyProject",
    "tags": {
      "key1":"value1",
      "key2":"value2"
    }
  }'
```

Windows Command prompt

```
C:\> aws pinpoint create-app ^
     --create-application-request Name=MyProject,tags={key1=value1,key2=value2}
```

No exemplo anterior, faça o seguinte:

- Substitua *MyProject* pelo nome que você deseja dar ao projeto.
- Substitua *key1 e key2* pelas chaves das tags que você deseja adicionar ao recurso.

- Substitua *value1* e *value2* pelos valores das tags que você deseja adicionar às respectivas chaves.

Para obter informações sobre os comandos que podem ser usados para criar um recurso do Amazon Pinpoint, consulte a Referência de comandos da AWS CLI.

Para adicionar uma tag a um recurso existente, empregue o comando `tag-resource` e especifique os valores apropriados para os parâmetros necessários:

Linux, macOS, or Unix

```
$ aws pinpoint tag-resource \
  --resource-arn resource-arn \
  --tags-model '{
    "tags": {
      "key1":"value1",
      "key2":"value2"
    }
  }'
```

Windows Command Prompt

```
C:\> aws pinpoint tag-resource ^
     --resource-arn resource-arn ^
     --tags-model tags={key1=value1,key2=value2}
```

No exemplo anterior, faça o seguinte:

- Substitua *resource-arn* pelo nome do recurso da Amazon (ARN) do recurso ao qual você deseja adicionar uma tag.
- Substitua *key1 e key2* pelas chaves das tags que você deseja adicionar ao recurso.
- Substitua *value1* e *value2* pelos valores das tags que você deseja adicionar às respectivas chaves.

Exibir tags para recursos

Os exemplos a seguir mostram como usar o AWS CLI e a API REST do Amazon Pinpoint para exibir uma lista de todas as tags (chaves e valores) que estão associadas a um recurso do Amazon

Pinpoint. Você também pode usar qualquer AWS SDK compatível para exibir as tags que estão associadas a um recurso.

Exibição de tags usando a API

Para usar a API REST do Amazon Pinpoint para exibir todas as tags que estão associadas a um recurso específico, envie uma solicitação GET ao URI de Tags, incluindo o nome do recurso da Amazon (ARN) do recurso no URI. O ARN deve ser codificado por URL. Por exemplo, a solicitação a seguir recupera todas as tags que estão associadas a uma campanha especificada (*resource-arn*):

```
GET /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

A resposta JSON à solicitação inclui um objeto `tags`. O objeto `tags` lista todas as chaves de tag e valores que estão associados à campanha.

Para exibir todas as tags que estão associadas a mais de um recurso do mesmo tipo, envie uma solicitação GET ao URI apropriado para aquele tipo de recurso. Por exemplo, a solicitação a seguir recupera informações sobre todas as campanhas no projeto especificado (*application-id*):

```
GET /v1/apps/application-id/campaigns HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

A resposta JSON à solicitação lista todas as campanhas no projeto. O objeto `tags` de cada campanha lista todas as chaves e valores de tag associados à campanha.

Exibição de tags usando a AWS CLI

Para usar a AWS CLI para exibir uma lista das tags que estão associadas a um recurso específico, execute o comando `list-tags-for-resource` e especifique o nome do recurso da Amazon (ARN) para o parâmetro `resource-arn`, conforme exibido no exemplo a seguir.

Linux, macOS, or Unix

```
$ aws pinpoint list-tags-for-resource \
  --resource-arn resource-arn
```

Windows Command Prompt

```
C:\> aws pinpoint list-tags-for-resource ^
     --resource-arn resource-arn
```

Para exibir uma lista de todos os recursos do Amazon Pinpoint que possuem tags, e todas as tags que estão associadas a cada um desses recursos, use o comando get-resources da API de marcação do AWS Resource Groups. Defina o parâmetro `resource-type-filters` como `mobiletargeting`, conforme mostrado no exemplo a seguir.

Linux, macOS, or Unix

```
$ aws resourcegroupstaggingapi get-resources \
     --resource-type-filters "mobiletargeting"
```

Windows Command Prompt

```
C:\> aws resourcegroupstaggingapi get-resources ^
     --resource-type-filters "mobiletargeting"
```

A saída do comando é uma lista de ARNs para todos os recursos do Amazon Pinpoint que possuem tags. A lista inclui todas as chaves de tag e valores associados a cada recurso.

Atualizar tags para recursos

Há várias formas de atualizar (substituir) uma tag para um recurso do Amazon Pinpoint. A melhor maneira de atualizar uma tag depende de:

- O tipo de recurso para o qual você deseja atualizar tags.
- Isto se aplica tanto se você quiser atualizar uma tag para um recurso quanto para vários recursos ao mesmo tempo.

- Se você quiser atualizar uma chave de tag e um valor de tag, ou ambos.

Para atualizar uma tag para um projeto do Amazon Pinpoint ou para vários recursos ao mesmo tempo, use as operações de marcação de grupos de recursos da AWS CLI ou a API de marcação do AWS Resource Groups. Atualmente, a API do Amazon Pinpoint não oferece suporte direto a nenhuma dessas tarefas.

Para atualizar uma chave de tag para um recurso, você pode remover a tag atual e adicionar uma nova tag usando a API do Amazon Pinpoint.

Remover tags de recursos

Os exemplos a seguir mostram como remover uma tag (a chave e o valor) de um recurso do Amazon Pinpoint usando a AWS CLI e a API REST do Amazon Pinpoint. Para remover uma tag de um recurso, também é possível qualquer SDK da AWS compatível.

Para remover uma tag de vários recursos do Amazon Pinpoint em uma única operação, use as operações de marcação dos grupos de recursos da AWS CLI ou da API de marcação do AWS Resource Groups. Para remover apenas um valor de tag específico, e não uma chave de tag, de um recurso, atualize a tag para o recurso.

Remover tags usando a API

Para remover uma tag de um recurso usando a API REST do Amazon Pinpoint, envie uma solicitação DELETE para o URI de Tags. No URI, inclua o nome de recurso da Amazon (ARN) do recurso do qual você deseja remover uma tag, seguido pelo parâmetro `tagKeys` e a tag a ser removida. Por exemplo:

```
https://endpoint/v1/tags/resource-arn?tagKeys=key
```

Onde:

- *`endpoint`* é o endpoint do Amazon Pinpoint da Região da AWS que hospeda o recurso.
- *`resource-arn`* é o ARN do recurso do qual você deseja remover uma tag.
- *`key`* é a tag que você deseja remover do recurso.

Todos os parâmetros devem ser codificados por URL.

Para remover várias chaves de tags e seus valores associados de um recurso, anexe o parâmetro e argumento de `tagKeys` para cada tag adicional a ser removida, separada por um E comercial (&). Por exemplo:

```
https://endpoint/v1/tags/resource-arn?tagKeys=key1&tagKeys=key2
```

Todos os parâmetros devem ser codificados por URL.

Remoção de tags usando a AWS CLI

Para remover uma tag de um recurso usando a AWS CLI, execute o comando `untag-resource`. Inclua o parâmetro e argumento `tag-keys`, conforme mostrado no exemplo a seguir.

Linux, macOS, or Unix

```
$ aws pinpoint untag-resource \
  --resource-arn resource-arn \
  --tag-keys key1 key2
```

Windows Command Prompt

```
C:\> aws pinpoint untag-resource ^
     --resource-arn resource-arn ^
     --tag-keys key1 key2
```

No exemplo anterior, faça as seguintes alterações:

- Substitua *`resource-arn`* pelo ARN do recurso do qual você deseja remover as tags.
- Substitua *`key1`* e *`key2`* pelas chaves das tags que você deseja remover do recurso.

Informações relacionadas

Para obter mais informações sobre os comandos da CLI que você pode usar para gerenciar os recursos do Amazon Pinpoint, consulte a seção do Amazon Pinpoint da Referência de comandos da AWS CLI.

Para obter mais informações sobre recursos na API do Amazon Pinpoint, incluindo suporte de métodos HTTP(S), parâmetros e esquemas, consulte a Referência da API do Amazon Pinpoint.

Personalizar recomendações com o AWS Lambda

No Amazon Pinpoint, você pode recuperar recomendações personalizadas de um modelo de recomendação e adicioná-las a mensagens enviadas de campanhas e jornadas. Um modelo de recomendação é um tipo de modelo de machine learning (ML) que encontra padrões em dados e gera previsões e recomendações com base nos padrões encontrados. Ele prevê o que um determinado usuário preferirá de um determinado conjunto de produtos ou itens, e fornece essas informações como um conjunto de recomendações para o usuário.

Usando modelos de recomendação com o Amazon Pinpoint, você pode enviar recomendações personalizadas para destinatários de mensagens com base nos atributos e no comportamento de cada destinatário. Com o AWS Lambda, você também pode personalizar e aprimorar essas recomendações. Por exemplo, você pode transformar dinamicamente uma recomendação de um único valor de texto (como um nome ou ID de produto) para conteúdo mais sofisticado (como nome de produto, descrição e imagem). E você pode fazer isso em tempo real, quando o Amazon Pinpoint envia a mensagem.

Esse atributo está disponível nas seguintes Regiões da AWS: Leste dos EUA (Norte da Virgínia); Oeste dos EUA (Oregon); Ásia-Pacífico (Mumbai); Ásia-Pacífico (Sydney) e Europa (Irlanda).

Tópicos

- Usar recomendações em mensagens
- Criar a função do Lambda
- Atribuir uma política de função do Lambda
- Autorizar o Amazon Pinpoint a invocar a função
- Configurar o modelo do recomendação

Usar recomendações em mensagens

Para usar um modelo de recomendação com o Amazon Pinpoint, comece criando uma solução do Amazon Personalize e implantando essa solução como uma campanha do Amazon Personalize. Depois disso, crie uma configuração para o modelo de recomendação no Amazon Pinpoint. Na configuração, você especifica configurações que determinam como recuperar e processar dados de recomendação da campanha do Amazon Personalize. Isso inclui se uma função do AWS Lambda deve ser invocada para executar processamento adicional dos dados recuperados.

O Amazon Personalize é um serviço da AWS projetado para ajudar você a criar modelos de ML que fornecem recomendações personalizadas em tempo real para clientes que usam os aplicativos criados por você. O Amazon Personalize fornece orientação no processo de criação e treinamento de um modelo de ML e, em seguida, na preparação e implantação do modelo como uma campanha do Amazon Personalize. Depois, você pode recuperar recomendações personalizadas em tempo real da campanha. Para saber mais sobre o Amazon Personalize, consulte o Guia do desenvolvedor do Amazon Personalize.

O AWS Lambda é um serviço de computação que pode ser usado para executar código sem provisionamento ou gerenciamento de servidores. Você empacota o código e faz upload dele no AWS Lambda como uma função do Lambda. O AWS Lambda então executa a função quando ela é invocada. Você pode invocar uma função manual e automaticamente em resposta a eventos ou em resposta a solicitações de aplicativos ou serviços, incluindo o Amazon Pinpoint. Para obter mais informações sobre criar e invocar funções do Lambda, consulte o Guia do desenvolvedor do AWS Lambda.

Depois de criar uma configuração do Amazon Pinpoint para um modelo de recomendação, você pode adicionar recomendações do modelo às mensagens enviadas de campanhas e jornadas. Faça isso usando modelos de mensagens que contêm variáveis de mensagem para os atributos recomendados. Um atributo recomendado é um endpoint dinâmico ou um atributo de usuário projetado para armazenar dados de recomendação. Você define esses atributos ao criar a configuração para um modelo de recomendação.

Você pode usar variáveis para atributos recomendados nos seguintes tipos de modelos de mensagens:

- Modelos de e-mail, para mensagens de e-mail que você envia de campanhas ou jornadas.
- Modelos de notificação por push, para as notificações por push enviadas de campanhas.
- Modelos de SMS, para mensagens de texto SMS enviadas de campanhas.

Para obter mais informações sobre como usar modelos de recomendação com o Amazon Pinpoint, consulte Modelos de machine learning no Guia do usuário do Amazon Pinpoint.

Se você configurar o Amazon Pinpoint para invocar uma função do Lambda que processa dados de recomendação, o Amazon Pinpoint executará as seguintes tarefas gerais sempre que enviar recomendações personalizadas em uma mensagem para uma campanha ou jornada:

1. Avalia e processa as definições de configuração e o conteúdo da mensagem e do modelo de mensagem.
2. Determina que o modelo de mensagem está conectado a um modelo de recomendação.
3. Avalia as definições da configuração de conexão e uso do modelo. Essas são definidas pelo recurso Modelo de recomendação do modelo.
4. Detecta uma ou mais variáveis de mensagem para atributos recomendados definidos pelas definições de configuração do modelo.
5. Recupera dados de recomendação da campanha do Amazon Personalize especificada nas configurações do modelo. Ele usa a operação GetRecommendations da API de runtime do Amazon Personalize para executar essa tarefa.
6. Adiciona os dados de recomendação apropriados a um atributo dinâmico recomendado (`RecommendationItems`) para cada destinatário da mensagem.
7. Invoca sua função do Lambda e envia os dados de recomendação para cada destinatário dessa função para processamento.

   Os dados são enviados como um objeto JSON que contém a definição do endpoint de cada destinatário. Cada definição de endpoint inclui um campo `RecommendationItems` que contém uma matriz ordenada de valores de 1 a 5. O número de valores na matriz depende das definições de configuração do modelo.
8. Aguarda que sua função do Lambda processe os dados e retorne os resultados.

   Os resultados são um objeto JSON que contém uma definição de endpoint atualizada para cada destinatário. Cada definição de endpoint atualizada contém um novo objeto `Recommendations` Esse objeto contém de 1 a 10 campos, um para cada atributo personalizado recomendado que você definiu nas configurações do modelo. Cada um desses campos armazena dados de recomendação aprimorados para o endpoint.
9. Usa a definição do endpoint atualizada para cada destinatário para substituir cada variável de mensagem pelo valor apropriado para esse destinatário.
10. Envia uma versão da mensagem que contém as recomendações personalizadas para cada destinatário da mensagem.

Para personalizar e aprimorar as recomendações dessa maneira, comece criando uma função do Lambda que processa as definições de endpoint enviadas pelo Amazon Pinpoint e retorna definições de endpoint atualizadas. Depois disso, atribua uma política de função do Lambda à função e autorize o Amazon Pinpoint a invocar a função. Depois, configure o modelo de recomendação no

Amazon Pinpoint. Ao configurar o modelo, especifique a função a ser invocada e defina os atributos recomendados a serem usados.

Criar a função do Lambda

Para saber como criar uma função do Lambda, consulte Conceitos básicos no Guia do desenvolvedor do AWS Lambda. Ao projetar e desenvolver sua função, lembre-se dos seguintes requisitos e diretrizes.

Dados de evento de entrada

Quando o Amazon Pinpoint invoca uma função do Lambda para um modelo de recomendação, ele envia uma carga que contém a configuração e outras definições da campanha ou jornada que está enviando a mensagem. A carga útil inclui um objeto `Endpoints`, que é um mapa que associa IDs de endpoint a definições de endpoint para os destinatários de mensagem.

As definições de endpoint usam a estrutura definida pelo recurso Endpoint da API do Amazon Pinpoint. No entanto, elas também incluem um campo para um atributo recomendado dinâmico chamado `RecommendationItems`. O campo `RecommendationItems` contém um ou mais itens recomendados para o endpoint, conforme retornado pela campanha do Amazon Personalize. O valor desse campo é uma matriz ordenada de 1 a 5 itens recomendados (como cadeias de caracteres). O número de itens na matriz depende do número de itens recomendados que você configurou o Amazon Pinpoint para recuperar para cada endpoint ou usuário.

Por exemplo:

```
"Endpoints": {
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
```

```
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z"
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
```

```
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z"
    }
}
```

No exemplo anterior, as configurações relevantes do Amazon Pinpoint são:

- O modelo de recomendação é configurado para recuperar três itens recomendados para cada endpoint ou usuário. (O valor da propriedade `RecommendationsPerMessage` é definido como 3.) Com essa configuração, o Amazon Pinpoint recupera e adiciona apenas o primeiro, o segundo e o terceiro item recomendado para cada endpoint ou usuário.
- O projeto está configurado para usar atributos de usuário personalizados que armazenam o nome, o sobrenome e a região em que vivem. (O objeto `UserAttributes` contém os valores desses atributos.)
- O projeto está configurado para usar um atributo de endpoint personalizado (`AddressType`) que indica se o endpoint é o endereço preferido do usuário (canal) para receber mensagens do projeto. (O objeto `Attributes` contém o valor desse atributo.)

Quando o Amazon Pinpoint invoca a função do Lambda e envia essa carga útil como os dados do evento, o AWS Lambda passa os dados para a função do Lambda para processamento.

Cada carga pode conter dados para até 50 endpoints. Se um segmento contiver mais de 50 endpoints, o Amazon Pinpoint invocará a função repetidamente, para até 50 endpoints de cada vez, até que a função processe todos os dados.

Dados e requisitos de resposta

Ao projetar e desenvolver sua função do Lambda, lembre-se das cotas de modelos de machine learning. Se a função não atender às condições definidas por essas cotas, o Amazon Pinpoint não poderá processar e enviar a mensagem.

Lembre-se também dos seguintes requisitos:

- A função deve retornar definições de endpoint atualizadas no mesmo formato fornecido pelos dados do evento de entrada.

- Cada definição de endpoint atualizada pode conter de 1 a 10 atributos personalizados recomendados para o endpoint ou usuário. Os nomes desses atributos devem corresponder aos nomes de atributos especificados ao configurar o modelo de recomendação no Amazon Pinpoint.
- Todos os atributos recomendados personalizados devem ser retornados em um único objeto `Recommendations` para cada endpoint ou usuário. Esse requisito ajuda a garantir que não ocorram conflitos de nomenclatura. Você pode adicionar o objeto `Recommendations` a qualquer local em uma definição de endpoint.
- O valor de cada atributo recomendado personalizado precisa ser uma string (valor único) ou uma matriz de strings (vários valores). Se o valor for uma matriz de cadeia de caracteres, recomendamos manter a ordem dos itens recomendados retornados pelo Amazon Personalize, conforme indicado no campo `RecommendationItems`. Caso contrário, seu conteúdo pode não refletir as previsões do modelo para um endpoint ou usuário.
- A função não deve modificar outros elementos nos dados do evento, incluindo outros valores de atributos para um endpoint ou usuário. Ele só deve adicionar e retornar valores para atributos personalizados recomendados. O Amazon Pinpoint não aceitará atualizações em nenhum outro valor na resposta da função.
- A função deve ser hospedada na mesma Região da AWS que o projeto do Amazon Pinpoint que está invocando a função. Se a função e o projeto não estiverem na mesma Região, o Amazon Pinpoint não poderá enviar dados de evento para a função.

Se qualquer um dos requisitos anteriores não for atendido, o Amazon Pinpoint não poderá processar e enviar a mensagem para um ou mais endpoints. Isso pode fazer com que uma atividade de campanha ou de jornada falhe.

Finalmente, recomendamos reservar 256 execuções simultâneas para a função.

No geral, a função do Lambda deve processar os dados de evento enviados pelo Amazon Pinpoint e retornar definições de endpoint modificadas. Ela pode fazer isso percorrendo cada endpoint no objeto `Endpoints` e, para cada endpoint, criando e definindo valores para os atributos recomendados personalizados que você deseja usar. O manipulador de exemplo a seguir, escrito em Python e continuando com o exemplo anterior de dados de evento de entrada, mostra o seguinte:

```
import json
import string

def lambda_handler(event, context):
    print("Received event: " + json.dumps(event))
```

```
    print("Received context: " +  str(context))
    segment_endpoints = event["Endpoints"]
    new_segment = dict()
    for endpoint_id in segment_endpoints.keys():
        endpoint = segment_endpoints[endpoint_id]
        if supported_endpoint(endpoint):
            new_segment[endpoint_id] = add_recommendation(endpoint)

    print("Returning endpoints: " + json.dumps(new_segment))
    return new_segment

def supported_endpoint(endpoint):
    return True

def add_recommendation(endpoint):
    endpoint["Recommendations"] = dict()

    customTitleList = list()
    customGenreList = list()
    for i,item in enumerate(endpoint["RecommendationItems"]):
        item = int(item)
        if item == 1210:
            customTitleList.insert(i, "Hanna")
            customGenreList.insert(i, "Action")
        elif item == 1527:
            customTitleList.insert(i, "Catastrophe")
            customGenreList.insert(i, "Comedy")
        elif item == 1815:
            customTitleList.insert(i, "Fleabag")
            customGenreList.insert(i, "Comedy")
        elif item == 2009:
            customTitleList.insert(i, "Late Night")
            customGenreList.insert(i, "Drama")
        elif item == 4582:
            customTitleList.insert(i, "Agatha Christie\'s The ABC Murders")
            customGenreList.insert(i, "Crime")
        elif item == 6542:
            customTitleList.insert(i, "Hunters")
            customGenreList.insert(i, "Drama")

    endpoint["Recommendations"]["Title"] = customTitleList
    endpoint["Recommendations"]["Genre"] = customGenreList
```

```
    return endpoint
```

No exemplo anterior, o AWS Lambda passa os dados do evento para o manipulador como o parâmetro `event`. O manipulador percorre cada endpoint no objeto `Endpoints` e define valores para os atributos recomendados personalizados nomeados `Recommendations.Title` e `Recommendations.Genre`. A instrução `return` retorna cada definição de endpoint atualizada para o Amazon Pinpoint.

Continuando com o exemplo anterior de dados de evento de entrada, as definições de endpoint atualizadas são:

```
"Endpoints":{
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z",
```

```
        "Recommendations":{
            "Title":[
                "Fleabag",
                "Late Night",
                "Catastrophe"
            ],
            "Genre":[
                "Comedy",
                "Comedy",
                "Comedy"
            ]
        }
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z",
```

```
        "Recommendations":{
            "Title":[
                "Hanna",
                "Hunters",
                "Agatha Christie\'s The ABC Murders"
            ],
            "Genre":[
                "Action",
                "Drama",
                "Crime"
            ]
        }
    }
}
```

No exemplo anterior, a função modificou o objeto `Endpoints` que recebeu e retornou os resultados. O objeto `Endpoint` de cada endpoint agora contém um novo objeto `Recommendations` que contém os campos `Title` e `Genre`. Cada um desses campos armazena uma matriz ordenada de três valores (como strings), onde cada valor fornece conteúdo aprimorado para um item recomendado correspondente no campo `RecommendationItems`.

Atribuir uma política de função do Lambda

Antes de usar a função do Lambda para processar dados de recomendação, você deve autorizar o Amazon Pinpoint a invocar a função. Para conceder permissão de invocação, atribua uma política de função do Lambda à função. A política de função do Lambda é uma política de permissões baseada em recursos que designa as entidades que podem usar uma função e quais ações essas entidades podem executar. Para mais informações, consulte Uso de políticas com base em recursos para o AWS Lambda no Guia do usuário do AWS Lambda.

A política de exemplo a seguir permite que a entidade principal do serviço do Amazon Pinpoint use a ação `lambda:InvokeFunction` para uma campanha específica do Amazon Pinpoint (*`campaignId`*) em um projeto específico do Amazon Pinpoint (*`projectId`*):

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
```

```
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

A política de função requer um bloco `Condition` que inclua uma chave `AWS:SourceArn`. Essa chave especifica qual recurso tem permissão para invocar a função. No exemplo anterior, a política permite que uma campanha específica chame a função.

Você também pode escrever uma política que permita que a entidade principal do serviço do Amazon Pinpoint use a ação `lambda:InvokeFunction` para todas as campanhas e jornadas em um projeto específico do Amazon Pinpoint (*`projectId`*). A política de exemplo a seguir mostra isso:

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

Ao contrário do primeiro exemplo, a chave `AWS:SourceArn` no bloco `Condition` deste exemplo permite que um projeto específico chame a função. Essa permissão aplica-se a todas as campanhas e jornadas do projeto.

Para escrever uma política mais genérica, use um caractere curinga (*) que corresponda a vários caracteres. Por exemplo, você pode usar o seguinte bloco `Condition` para permitir que qualquer projeto do Amazon Pinpoint invoque a função:

```
"Condition": {
```

```
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
```

Para usar a função do Lambda com todos os projetos da sua conta do Amazon Pinpoint, recomendamos configurar o bloco `Condition` da política da maneira anterior. No entanto, para usar melhores práticas, crie políticas que incluem apenas as permissões necessárias para executar uma ação específica em um recurso específico.

Autorizar o Amazon Pinpoint a invocar a função

Depois de atribuir uma política de função do Lambda à função, você pode adicionar permissões que permitem que o Amazon Pinpoint invoque a função para um projeto, campanha ou jornada específico. Você pode fazer isso usando o AWS Command Line Interface (AWS CLI) e o comando do Lambda `add-permission`. O exemplo a seguir mostra como fazer isso para um projeto específico (*`projectId`*):

```
$ aws lambda add-permission \
--function-name function-name \
--statement-id sid \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*
```

O exemplo anterior está formatado para Unix, Linux e macOS. Para o Microsoft Windows, substitua o caractere de continuação de linha de barra invertida (\) por um circunflexo (^).

Se o comando for executado com êxito, você verá um resultado semelhante a este:

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:recommenders/*\"}}}"
```

```
}
```

O valor de `Statement` é uma versão da cadeia de caracteres JSON da instrução adicionada à política da função do Lambda.

Configurar o modelo do recomendação

Para configurar o Amazon Pinpoint para invocar a função do Lambda de um modelo de recomendação, especifique as seguintes definições de configuração específicas ao Lambda para o modelo:

- `RecommendationTransformerUri`: esta propriedade especifica o nome ou o nome do recurso da Amazon (ARN) da função do Lambda.
- `Attributes`: este objeto é um mapa que define os atributos recomendados personalizados que a função adiciona a cada definição de endpoint. Cada um desses atributos pode ser usado como uma variável de mensagem em um modelo de mensagem.

É possível especificar essas configurações usando o recurso Modelos de recomendação da API do Amazon Pinpoint (ao criar a configuração de um modelo) ou o recurso Modelo de recomendação da API do Amazon Pinpoint (se você atualizar a configuração de um modelo). Também é possível definir essas configurações usando o console do Amazon Pinpoint.

Para obter mais informações sobre como usar modelos de recomendação com o Amazon Pinpoint, consulte Modelos de machine learning no Guia do usuário do Amazon Pinpoint.

Exclusão de dados do Amazon Pinpoint

Dependendo de como você o usa, o Amazon Pinpoint pode armazenar determinados dados que podem ser considerados pessoais. Por exemplo, um endpoint no Amazon Pinpoint contém informações de contato de um usuário final, como o endereço de e-mail ou número de telefone celular dessa pessoa.

Use o console ou a API do Amazon Pinpoint para excluir permanentemente esses dados pessoais. Este tópico inclui procedimentos para excluir vários tipos de dados que podem ser considerados pessoais.

Excluir endpoints

Um endpoint representa um único método de contato com um de seus clientes. Cada endpoint pode se referir ao endereço de e-mail de cliente, identificador de dispositivo móvel, número de telefone ou outro tipo de destino ao qual você pode enviar mensagens. Em várias jurisdições, esse tipo de informação pode ser considerada pessoal.

Para excluir todos os dados de um endpoint específico, você pode usar a API do Amazon Pinpoint para excluir o endpoint. O procedimento a seguir mostra como excluir um endpoint usando a AWS CLI para interagir com a API do Amazon Pinpoint. Esse procedimento pressupõe que você já tenha instalado e configurado a AWS CLI. Para obter mais informações, consulte Instalar a AWS CLI no Guia do usuário da AWS Command Line Interface .

Para excluir um endpoint usando o AWS CLI

- Na linha de comando, insira o seguinte comando:

  ```
  aws pinpoint delete-endpoint --application-id 810c7aab86d42fb2b56c8c966example --
  endpoint-id ad015a3bf4f1b2b0b82example
  ```

  No comando anterior, substitua *810c7aab86d42fb2b56c8c966example* pelo ID do projeto ao qual o endpoint está associado. Além disso, substitua *ad015a3bf4f1b2b0b82example* pelo ID exclusivo do próprio endpoint.

Para localizar o ID de um endpoint específico, determine a qual segmento o endpoint pertence e exporte o segmento do Amazon Pinpoint. Os dados exportados incluem o ID de cada endpoint. Você

pode exportar um segmento para um arquivo usando o console do Amazon Pinpoint. Para saber como, consulte Exportar segmentos no Guia do usuário do Amazon Pinpoint. Você também pode exportar um segmento a um bucket do Amazon Simple Storage Service (Amazon S3) usando a API do Amazon Pinpoint. Para saber como, consulte Exportação de endpoints neste guia.

Excluir dados de segmentos e endpoints armazenados no Amazon S3

É possível importar segmentos de um arquivo armazenado em um bucket do Amazon S3 usando o console ou a API do Amazon Pinpoint. Também é possível exportar dados do aplicativo, segmento ou endpoint do Amazon Pinpoint para um bucket do Amazon S3. Os arquivos importados e exportados podem conter dados pessoais, inclusive endereços de e-mail, números de telefone celular e informações sobre a localização física de um endpoint. Você pode excluir esses arquivos do Amazon S3.

O conteúdo entregue aos buckets do Amazon S3 pode conter conteúdo do cliente. Para obter mais informações sobre a remoção de dados confidenciais, consulte Como faço para esvaziar um bucket do S3? ou Como faço para excluir um bucket do S3?.

Excluir todos os dados do projeto

É possível excluir permanentemente todos os dados armazenados para um projeto do Amazon Pinpoint. Você pode fazer isso excluindo o projeto.

> **Warning**
>
> Se você excluir um projeto, o Amazon Pinpoint excluirá todas as configurações e dados específicos do projeto. As informações não podem ser recuperadas.

Quando você exclui um projeto, o Amazon Pinpoint exclui todas as configurações específicas do projeto para os canais de notificação por push e de sistema de mensagens SMS bidirecionais, além de todos os segmentos, campanhas, jornadas e análises específicas do projeto armazenados no Amazon Pinpoint, como a seguir:

- Segmentos: todas as configurações e dados do segmento. Para segmentos dinâmicos, isso inclui os grupos de segmentos e os filtros que você definiu. Para segmentos importados, isso inclui endpoints, IDs de usuários e outros dados importados e quaisquer filtros aplicados.

- Campanhas: todas as mensagens, tratamentos e variáveis de mensagens, dados de análise, programações, e outras configurações.
- Jornadas: todas as atividades, dados de análise, programações e outras configurações.
- Análise: dados de todas as métricas de engajamento, como o número de mensagens enviadas e entregues para campanhas e jornadas, e todas as métricas de execução de jornada. Para aplicativos móveis e web, todos os dados de eventos que não foram transmitidos para outro AWS serviço, como o Amazon Kinesis, todos os funis e dados de uso do aplicativo, receita e métricas demográficas. Antes de excluir um projeto, recomendamos que você exporte esses dados para outro local.

É possível excluir um projeto usando o console do Amazon Pinpoint. Para saber mais, consulte Excluir um projeto no Guia do usuário do Amazon Pinpoint. Você também pode excluir um projeto programaticamente usando o recurso App da API do Amazon Pinpoint.

Excluindo todos os AWS dados fechando sua conta AWS

Também é possível excluir todos os dados armazenados no Amazon Pinpoint fechando sua conta da AWS . No entanto, essa ação também exclui todos os outros dados, pessoais ou não pessoais, que você armazenou em todos os outros serviços. AWS Após o período pós-encerramento, fecha AWS permanentemente sua AWS conta e você não poderá mais reabri-la. Qualquer conteúdo que você não excluiu será excluído permanentemente, e todos AWS os serviços que você não interrompeu serão interrompidos. Para obter mais informações, consulte Fechar uma AWS conta no Guia AWS Account Management de referência.

> **Warning**
>
> O procedimento a seguir remove completamente todos os dados armazenados em sua AWS conta em todos os AWS serviços e AWS regiões.

Você pode fechar sua AWS conta usando AWS Management Console o.

Para fechar sua AWS conta

1. Abra o AWS Management Console em https://console.aws.amazon.com.
2. Acesse a página Configurações da conta em https://console.aws.amazon.com/billing/home?#/account.

> **Warning**
>
> As etapas a seguir excluem permanentemente todos os dados que você armazenou em todos os AWS serviços em todas as AWS regiões.

3. Em Fechar conta, leia o aviso legal que descreve as consequências do encerramento de sua AWS conta. Se concordar com os termos, marque a caixa de seleção e selecione Encerrar conta.
4. Na caixa de diálogo de confirmação, selecione Encerrar conta.

Exemplos de código para o Amazon Pinpoint usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Amazon Pinpoint com um kit de desenvolvimento de software (SDK) da AWS.

Para obter uma lista completa dos Guias do desenvolvedor do AWS SDK e exemplos de código, consulte Usando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Exemplos de código para o Amazon Pinpoint usando SDKs AWS
  - Ações para o Amazon Pinpoint usando SDKs AWS
    - Use CreateApp com um AWS SDK ou CLI
    - Use CreateCampaign com um AWS SDK ou CLI
    - Use CreateExportJob com um AWS SDK ou CLI
    - Use CreateImportJob com um AWS SDK ou CLI
    - Use CreateSegment com um AWS SDK ou CLI
    - Use DeleteApp com um AWS SDK ou CLI
    - Use DeleteEndpoint com um AWS SDK ou CLI
    - Use GetEndpoint com um AWS SDK ou CLI
    - Use GetSegments com um AWS SDK ou CLI
    - Use GetSmsChannel com um AWS SDK ou CLI
    - Use GetUserEndpoints com um AWS SDK ou CLI
    - Use SendMessages com um AWS SDK ou CLI
    - Use UpdateEndpoint com um AWS SDK ou CLI
- Exemplos de código para Amazon Pinpoint SMS e Voice API usando SDKs AWS
  - Ações para a API Amazon Pinpoint SMS e Voice usando SDKs AWS
    - Use SendVoiceMessage com um AWS SDK ou CLI

Exemplos de código para o Amazon Pinpoint usando SDKs AWS

Os exemplos de código a seguir mostram como usar o Amazon Pinpoint com um kit de desenvolvimento AWS de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço específicas, é possível ver as ações contextualizadas em seus devidos cenários e exemplos entre serviços.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Ações para o Amazon Pinpoint usando SDKs AWS
  - Use CreateApp com um AWS SDK ou CLI
  - Use CreateCampaign com um AWS SDK ou CLI
  - Use CreateExportJob com um AWS SDK ou CLI
  - Use CreateImportJob com um AWS SDK ou CLI
  - Use CreateSegment com um AWS SDK ou CLI
  - Use DeleteApp com um AWS SDK ou CLI
  - Use DeleteEndpoint com um AWS SDK ou CLI
  - Use GetEndpoint com um AWS SDK ou CLI
  - Use GetSegments com um AWS SDK ou CLI
  - Use GetSmsChannel com um AWS SDK ou CLI
  - Use GetUserEndpoints com um AWS SDK ou CLI
  - Use SendMessages com um AWS SDK ou CLI
  - Use UpdateEndpoint com um AWS SDK ou CLI

Ações para o Amazon Pinpoint usando SDKs AWS

Os exemplos de código a seguir demonstram como realizar ações individuais do Amazon Pinpoint com AWS SDKs. Esses trechos chamam a API do Amazon Pinpoint e são trechos de código de programas maiores que devem ser executados no contexto. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a Referência da API do Amazon Pinpoint.

Exemplos

- Use CreateApp com um AWS SDK ou CLI
- Use CreateCampaign com um AWS SDK ou CLI
- Use CreateExportJob com um AWS SDK ou CLI
- Use CreateImportJob com um AWS SDK ou CLI
- Use CreateSegment com um AWS SDK ou CLI
- Use DeleteApp com um AWS SDK ou CLI
- Use DeleteEndpoint com um AWS SDK ou CLI
- Use GetEndpoint com um AWS SDK ou CLI
- Use GetSegments com um AWS SDK ou CLI
- Use GetSmsChannel com um AWS SDK ou CLI
- Use GetUserEndpoints com um AWS SDK ou CLI
- Use SendMessages com um AWS SDK ou CLI
- Use UpdateEndpoint com um AWS SDK ou CLI

Use **`CreateApp`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `CreateApp`.

CLI

AWS CLI

Exemplo 1: Como criar uma aplicação

O exemplo de `create-app` a seguir cria uma aplicação (projeto).

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

Saída:

```
{
    "ApplicationResponse": {
```

```
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

Exemplo 2: como criar uma aplicação marcada

O exemplo de `create-app` a seguir cria uma aplicação (projeto) e associa uma tag (chave e valor) a ela.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

Saída:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- Para obter detalhes da API, consulte CreateAppna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateAppRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateAppResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateApplicationRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appName>

                Where:
                 appName - The name of the application to create.

               """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
        String appName = args[0];
        System.out.println("Creating an application with name: " + appName);

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String appID = createApplication(pinpoint, appName);
        System.out.println("App ID is: " + appID);
        pinpoint.close();
    }
```

```
    public static String createApplication(PinpointClient pinpoint, String
 appName) {
        try {
            CreateApplicationRequest appRequest =
 CreateApplicationRequest.builder()
                    .name(appName)
                    .build();

            CreateAppRequest request = CreateAppRequest.builder()
                    .createApplicationRequest(appRequest)
                    .build();

            CreateAppResponse result = pinpoint.createApp(request);
            return result.applicationResponse().id();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Para obter detalhes da API, consulte CreateAppa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun createApplication(applicationName: String?): String? {

    val createApplicationRequestOb = CreateApplicationRequest {
        name = applicationName
    }
```

```
    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.createApp(
            CreateAppRequest {
                createApplicationRequest = createApplicationRequestOb
            }
        )
        return result.applicationResponse?.id
    }
}
```

- Para obter detalhes da API, consulte a CreateAppreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CreateCampaign`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `CreateCampaign`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Crie uma campanha.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>

                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId,
 String segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
```

```
    }

    public static CampaignResponse createCampaign(PinpointClient client, String
appID, String segmentID) {

        try {
            Schedule schedule = Schedule.builder()
                    .startTime("IMMEDIATE")
                    .build();

            Message defaultMessage = Message.builder()
                    .action(Action.OPEN_APP)
                    .body("My message body.")
                    .title("My message title.")
                    .build();

            MessageConfiguration messageConfiguration =
MessageConfiguration.builder()
                    .defaultMessage(defaultMessage)
                    .build();

            WriteCampaignRequest request = WriteCampaignRequest.builder()
                    .description("My description")
                    .schedule(schedule)
                    .name("MyCampaign")
                    .segmentId(segmentID)
                    .messageConfiguration(messageConfiguration)
                    .build();

            CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());

            System.out.println("Campaign ID: " + result.campaignResponse().id());
            return result.campaignResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }
```

```
}
```

- Para obter detalhes da API, consulte CreateCampaigna Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun createPinCampaign(appId: String, segmentIdVal: String) {

    val scheduleOb = Schedule {
        startTime = "IMMEDIATE"
    }

    val defaultMessageOb = Message {
        action = Action.OpenApp
        body = "My message body"
        title = "My message title"
    }

    val messageConfigurationOb = MessageConfiguration {
        defaultMessage = defaultMessageOb
    }

    val writeCampaign = WriteCampaignRequest {
        description = "My description"
        schedule = scheduleOb
        name = "MyCampaign"
        segmentId = segmentIdVal
        messageConfiguration = messageConfigurationOb
    }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
```

```
        val result: CreateCampaignResponse = pinpoint.createCampaign(
            CreateCampaignRequest {
                applicationId = appId
                writeCampaignRequest = writeCampaign
            }
        )
        println("Campaign ID is ${result.campaignResponse?.id}")
    }
}
```

- Para obter detalhes da API, consulte a CreateCampaignreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CreateExportJob`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `CreateExportJob`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Exportar um endpoint.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
```

```
                3. Parses the endpoints files to obtain the endpoint IDs and
prints them.
                Usage: ExportEndpoints <applicationId> <s3BucketName>
<iamExportRoleArn> <path>

                Where:
                  applicationId - The ID of the Amazon Pinpoint application that
has the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the
JSON file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint write permissions to the S3 bucket.  path - The path where the files
downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
                """;

       if (args.length != 4) {
           System.out.println(usage);
           System.exit(1);
       }

       String applicationId = args[0];
       String s3BucketName = args[1];
       String iamExportRoleArn = args[2];
       String path = args[3];
       System.out.println("Deleting an application with ID: " + applicationId);

       Region region = Region.US_EAST_1;
       PinpointClient pinpoint = PinpointClient.builder()
               .region(region)
               .build();

       S3Client s3Client = S3Client.builder()
               .region(region)
               .build();

       exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
       pinpoint.close();
       s3Client.close();
   }

   public static void exportAllEndpoints(PinpointClient pinpoint,
           S3Client s3Client,
           String applicationId,
```

```
            String s3BucketName,
            String path,
            String iamExportRoleArn) {

        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
 s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
 o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint,
 S3Client s3Client, String s3BucketName,
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
                    .build();
```

```
            System.out.format("Exporting endpoints from Amazon Pinpoint
application %s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);

            CreateExportJobResponse exportResult =
pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                    .jobId(jobId)
```

```
                    .applicationId(applicationId)
                    .build();

            do {
                getExportJobResult =
pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path
location.
    public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();

                // Write the data to a local file.
```

```
                String fileSuffix = new
 SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte CreateExportJoba Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CreateImportJob`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `CreateImportJob`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Importar um segmento.

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the
 segment definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon
 Pinpoint to access S3. You need to set trust management for this
 to work. See https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];
```

```
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket,
key, roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for
import job status.");
        System.out.println("See application " + response.jobStatus() + " for
import job status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse =
client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
```

```
}
```

- Para obter detalhes da API, consulte CreateImportJoba Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use `CreateSegment` com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `CreateSegment`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
                final String usage = """

                                Usage:   <appId>

                                Where:
                                  appId - The application ID to create a segment
 for.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
 appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
 HashMap<>();
                        segmentAttributes.put("Team",
 AttributeDimension.builder()
```

```
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());

                        RecencyDimension recencyDimension =
RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors =
SegmentBehaviors.builder()
                                        .recency(recencyDimension)
                                        .build();

                        SegmentDemographics segmentDemographics =
SegmentDemographics
                                        .builder()
                                        .build();

                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();
```

```
                        CreateSegmentResponse createSegmentResult =
 client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
 createSegmentResult.segmentResponse().id());
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
        }
}
```

- Para obter detalhes da API, consulte CreateSegmenta Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun createPinpointSegment(applicationIdVal: String?): String? {

    val segmentAttributes = mutableMapOf<String, AttributeDimension>()
    val myList = mutableListOf<String>()
    myList.add("Lakers")

    val atts = AttributeDimension {
        attributeType = AttributeType.Inclusive
        values = myList
    }

    segmentAttributes["Team"] = atts
```

```
    val recencyDimension = RecencyDimension {
        duration = Duration.fromValue("DAY_30")
        recencyType = RecencyType.fromValue("ACTIVE")
    }

    val segmentBehaviors = SegmentBehaviors {
        recency = recencyDimension
    }

    val segmentLocation = SegmentLocation {}
    val dimensionsOb = SegmentDimensions {
        attributes = segmentAttributes
        behavior = segmentBehaviors
        demographic = SegmentDemographics {}
        location = segmentLocation
    }

    val writeSegmentRequestOb = WriteSegmentRequest {
        name = "MySegment101"
        dimensions = dimensionsOb
    }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val createSegmentResult: CreateSegmentResponse = pinpoint.createSegment(
            CreateSegmentRequest {
                applicationId = applicationIdVal
                writeSegmentRequest = writeSegmentRequestOb
            }
        )
        println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
        return createSegmentResult.segmentResponse?.id
    }
}
```

- Para obter detalhes da API, consulte a CreateSegmentreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteApp`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DeleteApp`.

CLI

AWS CLI

Como excluir uma aplicação

O exemplo de `delete-app` a seguir exclui uma aplicação (projeto).

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

Saída:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- Para obter detalhes da API, consulte DeleteAppna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Excluir um aplicativo.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appId>

                Where:
                 appId - The ID of the application to delete.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        System.out.println("Deleting an application with ID: " + appId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinApp(pinpoint, appId);
        System.out.println("Done");
        pinpoint.close();
    }

    public static void deletePinApp(PinpointClient pinpoint, String appId) {
```

```
        try {
            DeleteAppRequest appRequest = DeleteAppRequest.builder()
                    .applicationId(appId)
                    .build();

            DeleteAppResponse result = pinpoint.deleteApp(appRequest);
            String appName = result.applicationResponse().name();
            System.out.println("Application " + appName + " has been deleted.");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DeleteAppa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun deletePinApp(appId: String?) {

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteApp(
            DeleteAppRequest {
                applicationId = appId
            }
        )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
}
```

- Para obter detalhes da API, consulte a DeleteAppreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteEndpoint`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DeleteEndpoint`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Excluir um endpoint

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:   <appName> <endpointId >

                Where:
                  appId - The id of the application to delete.
                  endpointId - The id of the endpoint to delete.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
 String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- Para obter detalhes da API, consulte DeleteEndpointa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun deletePinEncpoint(appIdVal: String?, endpointIdVal: String?) {

    val deleteEndpointRequest = DeleteEndpointRequest {
        applicationId = appIdVal
        endpointId = endpointIdVal
    }

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteEndpoint(deleteEndpointRequest)
        val id = result.endpointResponse?.id
        println("The deleted endpoint is  $id")
    }
}
```

- Para obter detalhes da API, consulte a DeleteEndpointreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`GetEndpoint`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `GetEndpoint`.

CLI

AWS CLI

Como recuperar informações sobre as configurações e os atributos de um endpoint específico de uma aplicação

O exemplo de `get-endpoint` a seguir recupera informações sobre as configurações e os atributos de um endpoint específico de uma aplicação.

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

Saída:

```
{
    "EndpointResponse": {
        "Address": "+11234567890",
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "Attributes": {},
        "ChannelType": "SMS",
        "CohortId": "63",
        "CreationDate": "2019-01-28T23:55:11.534Z",
        "EffectiveDate": "2021-08-06T00:04:51.763Z",
        "EndpointStatus": "ACTIVE",
        "Id": "testendpoint",
        "Location": {
            "Country": "USA"
        },
        "Metrics": {
            "SmsDelivered": 1.0
        },
        "OptOut": "ALL",
        "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",
        "User": {
            "UserAttributes": {
                "Age": [
                    "24"
                ]
            },
        "UserId": "testuser"
```

```
        }
    }
}
```

- Para obter detalhes da API, consulte GetEndpointna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <endpoint>

                Where:
```

```
                appId - The ID of the application to delete.
                endpoint - The ID of the endpoint.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String
appId, String endpoint) {
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
            System.out.println(endpointJson);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
```

```
        System.out.println("Done");
    }
}
```

- Para obter detalhes da API, consulte GetEndpointa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun lookupPinpointEndpoint(appId: String?, endpoint: String?) {

    PinpointClient { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.getEndpoint(
            GetEndpointRequest {
                applicationId = appId
                endpointId = endpoint
            }
        )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson = GsonBuilder()
            .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
            .setPrettyPrinting()
            .create()

        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- Para obter detalhes da API, consulte a GetEndpointreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`GetSegments`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `GetSegments`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Listar segmentos.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListSegments {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:   <appId>

                Where:
                  appId - The ID of the application that contains a segment.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listSegs(pinpoint, appId);
        pinpoint.close();
    }

    public static void listSegs(PinpointClient pinpoint, String appId) {
        try {
            GetSegmentsRequest request = GetSegmentsRequest.builder()
                    .applicationId(appId)
                    .build();

            GetSegmentsResponse response = pinpoint.getSegments(request);
            List<SegmentResponse> segments = response.segmentsResponse().item();
            for (SegmentResponse segment : segments) {
                System.out
                        .println("Segement " + segment.id() + " " +
 segment.name() + " " + segment.lastModifiedDate());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte GetSegmentsa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun listSegs(appId: String?) {

    PinpointClient { region = "us-west-2" }.use { pinpoint ->

        val response = pinpoint.getSegments(
            GetSegmentsRequest {
                applicationId = appId
            }
        )
        response.segmentsResponse?.item?.forEach { segment ->
            println("Segement id is ${segment.id}")
        }
    }
}
```

- Para obter detalhes da API, consulte a GetSegmentsreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use `GetSmsChannel` com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `GetSmsChannel`.

CLI

AWS CLI

Como recuperar informações sobre o status e as configurações do canal de SMS de uma aplicação

O exemplo de `get-sms-channel` a seguir recupera o status e as configurações do canal de SMS de uma aplicação.

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

Saída:

```
{
    "SMSChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:39:18.511Z",
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:39:18.511Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 1
    }
}
```

- Para obter detalhes da API, consulte GetSmsChannelna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelResponse;
import software.amazon.awssdk.services.pinpoint.model.GetSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateChannel {
    public static void main(String[] args) {
        final String usage = """

                Usage: CreateChannel <appId>

                Where:
                  appId - The name of the application whose channel is updated.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
```

```
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        SMSChannelResponse getResponse = getSMSChannel(pinpoint, appId);
        toggleSmsChannel(pinpoint, appId, getResponse);
        pinpoint.close();
    }

    private static SMSChannelResponse getSMSChannel(PinpointClient client, String
appId) {
        try {
            GetSmsChannelRequest request = GetSmsChannelRequest.builder()
                    .applicationId(appId)
                    .build();

            SMSChannelResponse response =
client.getSmsChannel(request).smsChannelResponse();
            System.out.println("Channel state is " + response.enabled());
            return response;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void toggleSmsChannel(PinpointClient client, String appId,
SMSChannelResponse getResponse) {
        boolean enabled = !getResponse.enabled();
        try {
            SMSChannelRequest request = SMSChannelRequest.builder()
                    .enabled(enabled)
                    .build();

            UpdateSmsChannelRequest updateRequest =
UpdateSmsChannelRequest.builder()
                    .smsChannelRequest(request)
                    .applicationId(appId)
                    .build();
```

```
            UpdateSmsChannelResponse result =
 client.updateSmsChannel(updateRequest);
            System.out.println("Channel state: " +
 result.smsChannelResponse().enabled());

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte GetSmsChannela Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`GetUserEndpoints`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `GetUserEndpoints`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <applicationId> <userId>

                Where:
                   applicationId - The ID of the Amazon Pinpoint application that
 has the endpoint.
                   userId - The user id applicable to the endpoints""";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String userId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllEndpoints(pinpoint, applicationId, userId);
        pinpoint.close();
    }

    public static void listAllEndpoints(PinpointClient pinpoint,
            String applicationId,
            String userId) {

        try {
            GetUserEndpointsRequest endpointsRequest =
 GetUserEndpointsRequest.builder()
```

```
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

            GetUserEndpointsResponse response =
 pinpoint.getUserEndpoints(endpointsRequest);
            List<EndpointResponse> endpoints =
 response.endpointsResponse().item();

            // Display the results.
            for (EndpointResponse endpoint : endpoints) {
                System.out.println("The channel type is: " +
 endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte GetUserEndpointsa Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`SendMessages`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `SendMessages`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Envie uma mensagem de e-mail.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon
 Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
```

```
        string toAddress = configuration["ToAddress"]!;

        // The Amazon Pinpoint project/application ID to use when you send this
message.
        // Make sure that the SMS channel is enabled for the project or
application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<MessageResponse> SendEmailMessage(
        string region, string appId, string toAddress, string senderAddress)
    {
        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        // The subject line of the email.
        string subject = "Amazon Pinpoint Email test";

        // The body of the email for recipients whose email clients don't
        // support HTML content.
        string textBody = @"Amazon Pinpoint Email Test (.NET)"
                          + "\n---------------------------------"
                          + "\nThis email was sent using the Amazon Pinpoint API
using the AWS SDK for .NET.";

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK
for .NET)</h1>"
                          + "\n  <p>This email was sent using the "
```

```
                            + "\n    <a href='https://aws.amazon.com/
pinpoint/'>Amazon Pinpoint</a> API "
                            + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                            + "\n  </p>"
                            + "\n</body>"
                            + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
                },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
                            },
```

```
                            Subject = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await
 client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
        return response.MessageResponse;
    }
}
```

Envie uma mensagem SMS.

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
```

```
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone
 number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify
 the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or
 application
        // that you choose.
        string appId = configuration["AppId"]!;

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;

        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
```

```
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
```

```
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await
 client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

- Para obter detalhes da API, consulte SendMessagesa Referência AWS SDK for .NET da API.

CLI

AWS CLI

Como enviar uma mensagem SMS usando o endpoint de uma aplicação

O exemplo de `send-messages` a seguir envia uma mensagem direta a uma aplicação com um endpoint.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

Conteúdo de `myfile.json`:

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
```

```
        "testendpoint": {}
    }
}
```

Saída:

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

Para obter mais informações, consulte Canal de SMS do Amazon Pinpoint no Guia do usuário do Amazon Pinpoint.

- Para obter detalhes da API, consulte SendMessagesna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Envie uma mensagem de e-mail.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
```

```
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendEmailMessage {

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        public static String charset = "UTF-8";

    // The body of the email for recipients whose email clients support HTML
 content.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Java 2.x

        """;
```

```
       public static void main(String[] args) {
               final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
<toAddress>

           Where:
              subject - The email subject to use.
              senderAddress - The from address. This address has to be verified
in Amazon Pinpoint in the region you're using to send email\s
              toAddress - The to address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
           """;

       if (args.length != 3) {
           System.out.println(usage);
           System.exit(1);
       }

       String subject = args[0];
       String senderAddress = args[1];
       String toAddress = args[2];
       System.out.println("Sending a message");
       PinpointEmailClient pinpoint = PinpointEmailClient.builder()
           .region(Region.US_EAST_1)
           .build();

       sendEmail(pinpoint, subject, senderAddress, toAddress);
       System.out.println("Email was sent");
       pinpoint.close();
   }

   public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress) {
       try {
           Content content = Content.builder()
               .data(body)
               .build();

           Body messageBody = Body.builder()
               .text(content)
               .build();
```

```
            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Envie uma mensagem de e-mail com cópia para outros destinatários.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import  software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;
import java.util.ArrayList;
```

```
/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendEmailMessageCC {

    // The body of the email.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Java 2.x

        """;
    public static void main(String[] args) {
        final String usage = """

            Usage:    <subject> <senderAddress> <toAddress> <ccAddress>

            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified
 in Amazon Pinpoint in the region you're using to send email\s
               toAddress - The to address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email\s
               ccAddress - The CC address.
            """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        String ccAddress = args[3];

        System.out.println("Sending a message");
```

```
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        ArrayList<String> ccList = new ArrayList<>();
        ccList.add(ccAddress);
        sendEmail(pinpoint, subject, senderAddress, toAddress, ccList);
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress, ArrayList<String> ccAddresses)
{
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .ccAddresses(ccAddresses)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");
```

```
        } catch (PinpointException e) {
            // Handle exception
            e.printStackTrace();
        }
    }
}
```

Envie uma mensagem SMS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";
```

```
        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application
 ID to use when you send this message.
                                  originationNumber - The phone number or
 short code that you specify has to be associated with your Amazon Pinpoint
 account. For best results, specify long codes in E.164 format (for example,
 +1-555-555-5654).
                                  destinationNumber - The recipient's phone
 number.  For best results, you should specify the phone number in E.164 format
 (for example, +1-555-555-5654).\s
                                  """;

                if (args.length != 4) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
 destinationNumber);
                pinpoint.close();
        }
```

```
       public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request =
SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();
```

```
                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" +
 v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Envie mensagens SMS em lote.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessageBatch {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
```

```
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber> <destinationNumber1>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application
 ID to use when you send this message.
                                  originationNumber - The phone number or
 short code that you specify has to be associated with your Amazon Pinpoint
 account. For best results, specify long codes in E.164 format (for example,
 +1-555-555-5654).
                                  destinationNumber - The recipient's phone
 number.  For best results, you should specify the phone number in E.164 format
 (for example, +1-555-555-5654).
                                  destinationNumber1 - The second recipient's
 phone number.  For best results, you should specify the phone number in E.164
 format (for example, +1-555-555-5654).\s
                                  """;

                if (args.length != 5) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                String destinationNumber1 = args[4];
                System.out.println("Sending a message");
```

```
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber, destinationNumber1);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
                        String originationNumber,
                        String destinationNumber, String destinationNumber1) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        // Add an entry to the Map object for each number to whom
you want to send a
                        // message.
                        addressMap.put(destinationNumber, addConfig);
                        addressMap.put(destinationNumber1, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();
```

```
                        // Create a SendMessagesRequest object.
                        SendMessagesRequest request =
 SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" +
 v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

- Para obter detalhes da API, consulte SendMessagesa Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Crie o cliente em um módulo separado e exporte-o.

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
// Set the AWS Region.
```

```
const REGION = "us-east-1";
export const pinClient = new PinpointClient({ region: REGION });
```

Envie uma mensagem de e-mail.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
var subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https://aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint Email API</a>
 using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding for the subject line and message body of the email.
var charset = "UTF-8";

const params = {
  ApplicationId: projectId,
```

```
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

const run = async () => {
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }

    const recipientResult = MessageResponse.Result[toAddress];
```

```
    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
    } else {
      console.log(recipientResult.MessageId);
    }
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

Envie uma mensagem SMS.

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

/* The phone number or short code to send the message from. The phone number
 or short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number.  For best results, you should specify the phone
 number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX

// The content of the SMS message.
const message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

/*The Amazon Pinpoint project/application ID to use when you send this message.
Make sure that the SMS channel is enabled for the project or application
that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXX

/* The type of SMS message that you want to send. If you plan to send
time-sensitive content, specify TRANSACTIONAL. If you plan to send
marketing-related content, specify PROMOTIONAL.*/
```

```
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

/* The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html.*/

var senderId = "MySenderID";

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"],
    );
  } catch (err) {
    console.log(err);
  }
};
```

```
run();
```

- Para obter detalhes da API, consulte SendMessagesa Referência AWS SDK for JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Envie uma mensagem de e-mail.

```
"use strict";

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

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
```

```
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using
 the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
```

```
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
  }
});
```

Envie uma mensagem SMS.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
```

```
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
var senderId = "MySenderID";

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

// Specify the region.
```

```
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

- Para obter detalhes da API, consulte SendMessagesa Referência AWS SDK for JavaScript da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */

val body: String = """
        Amazon Pinpoint test (AWS SDK for Kotlin)

        This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Kotlin.

""".trimIndent()

suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <subject> <appId> <senderAddress> <toAddress>

    Where:
        subject - The email subject to use.
        senderAddress - The from address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email
        toAddress - The to address. This address has to be verified in Amazon
 Pinpoint in the region you're using to send email
    """
```

```
    if (args.size != 3) {
        println(usage)
        exitProcess(0)
    }

    val subject = args[0]
    val senderAddress = args[1]
    val toAddress = args[2]
    sendEmail(subject, senderAddress, toAddress)
}

suspend fun sendEmail(subjectVal: String?, senderAddress: String, toAddressVal:
 String) {
    var content = Content {
        data = body
    }

    val messageBody = Body {
        text = content
    }

    val subContent = Content {
        data = subjectVal
    }

    val message = Message {
        body = messageBody
        subject = subContent
    }

    val destinationOb = Destination {
        toAddresses = listOf(toAddressVal)
    }

    val emailContent = EmailContent {
        simple = message
    }

    val sendEmailRequest = SendEmailRequest {
        fromEmailAddress = senderAddress
        destination = destinationOb
        this.content = emailContent
    }
```

```
    PinpointEmailClient { region = "us-east-1" }.use { pinpointemail ->
        pinpointemail.sendEmail(sendEmailRequest)
        println("Message Sent")
    }
}
```

- Para obter detalhes da API, consulte a SendMessagesreferência da API AWS SDK for Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Envie uma mensagem de e-mail.

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

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
```

```
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this
message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and
message
                     body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data":
html_message},
                            "TextPart": {"Charset": char_set, "Data":
text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
```

```
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -------------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python
 (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
```

```
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Envie uma mensagem SMS.

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

logger = logging.getLogger(__name__)


def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you
 send
                   this message. The SMS channel must be enabled for the project
 or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This
 phone
                               number must be associated with your Amazon
 Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
```

```
    :param message_type: The type of SMS message that you want to send. If you
 send
                         time-sensitive content, specify TRANSACTIONAL. If you
 send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK
 for "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
```

```
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

Enviar uma mensagem com um modelo de e-mail existente.

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

logger = logging.getLogger(__name__)


def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
                         account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return: A dict of to_addresses and their message IDs.
    """
    try:
```

```
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress":
 sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
```

```
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

Envie uma mensagem de texto com um modelo de SMS existente.

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

logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
```

```
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )

    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

- Para obter detalhes da API, consulte a SendMessagesReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`UpdateEndpoint`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `UpdateEndpoint`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.EndpointDemographic;
import software.amazon.awssdk.services.pinpoint.model.EndpointLocation;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.UUID;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import java.util.Map;
import java.util.Date;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appId>

                Where:
                  appId - The ID of the application to create an endpoint for.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        EndpointResponse response = createEndpoint(pinpoint, appId);
        System.out.println("Got Endpoint: " + response.id());
        pinpoint.close();
    }

    public static EndpointResponse createEndpoint(PinpointClient client, String
 appId) {
        String endpointId = UUID.randomUUID().toString();
        System.out.println("Endpoint ID: " + endpointId);

        try {
```

```
            EndpointRequest endpointRequest = createEndpointRequestData();
            UpdateEndpointRequest updateEndpointRequest =
UpdateEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .endpointRequest(endpointRequest)
                    .build();

            UpdateEndpointResponse updateEndpointResponse =
client.updateEndpoint(updateEndpointRequest);
            System.out.println("Update Endpoint Response: " +
updateEndpointResponse.messageBody());

            GetEndpointRequest getEndpointRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            GetEndpointResponse getEndpointResponse =
client.getEndpoint(getEndpointRequest);
            System.out.println(getEndpointResponse.endpointResponse().address());

System.out.println(getEndpointResponse.endpointResponse().channelType());

System.out.println(getEndpointResponse.endpointResponse().applicationId());

System.out.println(getEndpointResponse.endpointResponse().endpointStatus());

System.out.println(getEndpointResponse.endpointResponse().requestId());
            System.out.println(getEndpointResponse.endpointResponse().user());

            return getEndpointResponse.endpointResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static EndpointRequest createEndpointRequestData() {
        try {
            List<String> favoriteTeams = new ArrayList<>();
            favoriteTeams.add("Lakers");
```

```
            favoriteTeams.add("Warriors");
            HashMap<String, List<String>> customAttributes = new HashMap<>();
            customAttributes.put("team", favoriteTeams);

            EndpointDemographic demographic = EndpointDemographic.builder()
                    .appVersion("1.0")
                    .make("apple")
                    .model("iPhone")
                    .modelVersion("7")
                    .platform("ios")
                    .platformVersion("10.1.1")
                    .timezone("America/Los_Angeles")
                    .build();

            EndpointLocation location = EndpointLocation.builder()
                    .city("Los Angeles")
                    .country("US")
                    .latitude(34.0)
                    .longitude(-118.2)
                    .postalCode("90068")
                    .region("CA")
                    .build();

            Map<String, Double> metrics = new HashMap<>();
            metrics.put("health", 100.00);
            metrics.put("luck", 75.00);

            EndpointUser user = EndpointUser.builder()
                    .userId(UUID.randomUUID().toString())
                    .build();

            DateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm'Z'"); //
Quoted "Z" to indicate UTC, no timezone
                                                                           //
offset
            String nowAsISO = df.format(new Date());

            return EndpointRequest.builder()
                    .address(UUID.randomUUID().toString())
                    .attributes(customAttributes)
                    .channelType("APNS")
                    .demographic(demographic)
                    .effectiveDate(nowAsISO)
                    .location(location)
```

```
                    .metrics(metrics)
                    .optOut("NONE")
                    .requestId(UUID.randomUUID().toString())
                    .user(user)
                    .build();

            } catch (PinpointException e) {
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            }
            return null;
        }
    }
```

- Para obter detalhes da API, consulte UpdateEndpointa Referência AWS SDK for Java 2.x da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código para Amazon Pinpoint SMS e Voice API usando SDKs AWS

Os exemplos de código a seguir mostram como usar a API de voz e SMS do Amazon Pinpoint com um kit de desenvolvimento de AWS software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço específicas, é possível ver as ações contextualizadas em seus devidos cenários e exemplos entre serviços.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Ações para a API Amazon Pinpoint SMS e Voice usando SDKs AWS
  - Use SendVoiceMessage com um AWS SDK ou CLI

Ações para a API Amazon Pinpoint SMS e Voice usando SDKs AWS

Os exemplos de código a seguir demonstram como realizar ações individuais da API de voz e SMS do Amazon Pinpoint com AWS SDKs. Esses trechos chamam a API de SMS e voz do Amazon Pinpoint e são trechos de código de programas maiores que devem ser executados no contexto. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para uma lista completa, consulte a Referência da API de SMS e voz do Amazon Pinpoint.

Exemplos

- Use SendVoiceMessage com um AWS SDK ou CLI

Use **SendVoiceMessage** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `SendVoiceMessage`.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendVoiceMessage {

        // The Amazon Polly voice that you want to use to send the message. For a
 list
        // of voices, see https://docs.aws.amazon.com/polly/latest/dg/
voicelist.html
        static final String voiceName = "Matthew";

        // The language to use when sending the message. For a list of supported
        // languages, see
        // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
        static final String languageCode = "en-US";

        // The content of the message. This example uses SSML to customize and
 control
        // certain aspects of the message, such as by adding pauses and changing
        // phonation. The message can't contain any line breaks.
        static final String ssmlMessage = "<speak>This is a test message sent
 from "
                        + "<emphasis>Amazon Pinpoint</emphasis> "
                        + "using the <break strength='weak'/>AWS "
                        + "SDK for Java. "
                        + "<amazon:effect phonation='soft'>Thank "
                        + "you for listening.</amazon:effect></speak>";

        public static void main(String[] args) {

                final String usage = """

                                Usage:   <originationNumber> <destinationNumber>
\s
```

```
                                Where:
                                  originationNumber - The phone number or
short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format (for example,
+1-555-555-5654).
                                  destinationNumber - The recipient's phone
number.  For best results, you should specify the phone number in E.164 format
(for example, +1-555-555-5654).\s
                                  """;

                if (args.length != 2) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String originationNumber = args[0];
                String destinationNumber = args[1];
                System.out.println("Sending a voice message");

                // Set the content type to application/json.
                List<String> listVal = new ArrayList<>();
                listVal.add("application/json");
                Map<String, List<String>> values = new HashMap<>();
                values.put("Content-Type", listVal);

                ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()
                                .headers(values)
                                .build();

                PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
                                .overrideConfiguration(config2)
                                .region(Region.US_EAST_1)
                                .build();

                sendVoiceMsg(client, originationNumber, destinationNumber);
                client.close();
        }

        public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                        String destinationNumber) {
                try {
```

```
                        SSMLMessageType ssmlMessageType =
 SSMLMessageType.builder()
                                        .languageCode(languageCode)
                                        .text(ssmlMessage)
                                        .voiceId(voiceName)
                                        .build();

                        VoiceMessageContent content =
 VoiceMessageContent.builder()
                                        .ssmlMessage(ssmlMessageType)
                                        .build();

                        SendVoiceMessageRequest voiceMessageRequest =
 SendVoiceMessageRequest.builder()

 .destinationPhoneNumber(destinationNumber)

 .originationPhoneNumber(originationNumber)
                                        .content(content)
                                        .build();

                        client.sendVoiceMessage(voiceMessageRequest);
                        System.out.println("The message was sent successfully.");

                } catch (PinpointSmsVoiceException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

- Para obter detalhes da API, consulte SendVoiceMessagea Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best
 results, you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/
SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
```

```
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint
 account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

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

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
```

```
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

- Para obter detalhes da API, consulte SendVoiceMessagea Referência AWS SDK for JavaScript da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

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

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
```

```
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your
Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your
Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
message.
    :param ssml_message: The content of the message. This example uses SSML to
control
                         certain aspects of the message, such as the volume and
the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
```

```
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

- Para obter detalhes da API, consulte a SendVoiceMessageReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Pinpoint com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Segurança no Amazon Pinpoint

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

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como a segurança da nuvem e segurança na nuvem:

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

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon Pinpoint. Os tópicos a seguir mostram como configurar o Amazon Pinpoint para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do Amazon Pinpoint.

Para obter mais informações, sobre arquiteturas de referência no Guia de arquitetura resiliente do Amazon Pinpoint.

Tópicos

- Proteção de dados no Amazon Pinpoint
- Gerenciamento de identidade e acesso para o Amazon Pinpoint
- Registrar em log e monitorar no Amazon Pinpoint
- Validação de conformidade do Amazon Pinpoint
- Resiliência no Amazon Pinpoint
- Segurança da infraestrutura no Amazon Pinpoint
- Análise de configuração e vulnerabilidade no Amazon Pinpoint

- Práticas recomendadas de segurança para o Amazon Pinpoint

Proteção de dados no Amazon Pinpoint

O modelo de responsabilidade AWS compartilhada de se aplica à proteção de dados no Amazon Pinpoint. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a AWS postagem do blog Shared Responsibility Model and GDPR no AWS Blog de segurança da.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as 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 uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalha com o Amazon Pinpoint ou outro Serviços da

AWS usando o console, a API ou AWS os AWS CLI 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, recomendemos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Dependendo de como você configura e usa o serviço, o Amazon Pinpoint pode armazenar os seguintes tipos de dados pessoais para você ou sobre seus clientes:

Dados de configuração

Isso inclui os dados de configuração do projeto, como credenciais e configurações que definem como e quando o Amazon Pinpoint envia mensagens por meio de canais compatíveis, e os segmentos do usuário para os quais envia mensagens. Para enviar mensagens, esses dados podem incluir endereços IP dedicados para mensagens de e-mail, códigos curtos e IDs de remetentes para mensagens de texto SMS, além de credenciais para a comunicação com serviços de notificação por push, como o Apple Push Notification Service (APNs) e o Firebase Cloud Messaging (FCM).

Dados do usuário e do endpoint

Isso inclui atributos padrão e personalizados que você usa para armazenar e gerenciar dados de usuários e endpoints de um projeto do Amazon Pinpoint. Um atributo pode armazenar informações de um usuário específico (como o nome de um usuário) ou de um endpoint de usuário específico (como o endereço de e-mail, o número do celular ou o token do dispositivo móvel do usuário). Esses dados também podem incluir IDs de usuários externos que correlacionam usuários de um projeto do Amazon Pinpoint com usuários em um sistema externo, como um sistema de gerenciamento de relacionamento com o cliente. Para obter mais informações sobre o que esses dados podem incluir, consulte os esquemas de Usuário e Endpoint na Referência de API do Amazon Pinpoint.

Dados de análise

Isso inclui dados de métricas, também chamados de indicadores-chave de desempenho (KPIs), que oferecem informações sobre o desempenho de um projeto do Amazon Pinpoint para áreas como envolvimento de usuários e atividades de compra. Isso também inclui dados de métricas que oferecem informações sobre dados demográficos do usuário para um projeto. Os dados podem derivar de atributos padrão ou personalizados para usuários e endpoints, como a cidade em que o usuário mora. Eles também podem derivar de eventos, como abrir e clicar, para mensagens de e-mail que você envia para um projeto.

Dados importados

Isso inclui todos os dados de usuário, segmentação e análise que você adiciona ou importa de fontes externas e usa no Amazon Pinpoint. Um exemplo é um arquivo JSON que você importa para o Amazon Pinpoint (diretamente do console ou de um bucket do Amazon S3) para criar um segmento estático. Outros exemplos são dados de endpoints que você adiciona programaticamente para criar um segmento dinâmico, endereços de endpoint para os quais você envia mensagens diretas e eventos que você configura um aplicativo para relatar para o Amazon Pinpoint.

Tópicos

- Criptografia de dados
- Privacidade do tráfego entre redes
- Criar um endpoint da VPC de interface para o Amazon Pinpoint

Criptografia de dados

Os dados do Amazon Pinpoint são criptografados em trânsito e em repouso. Quando você envia dados para o Amazon Pinpoint, ele criptografa os dados conforme os recebe e os armazena. Quando você recupera dados do Amazon Pinpoint, ele transmite os dados para você usando os protocolos de segurança atuais.

Criptografia inativa

O Amazon Pinpoint criptografa todos os dados e o armazena para você. Isso inclui os dados de configuração, os dados de usuário e do endpoint, dados de análise e todos os dados que você adicionar ou importar para o Amazon Pinpoint. Para criptografar seus dados, o Amazon Pinpoint usa chaves AWS Key Management Service internas AWS KMS() que o serviço possui e mantém em seu nome. Nós mudamos essas chaves regularmente. Para obter informações sobre AWS KMS, consulte o Guia do AWS Key Management Service desenvolvedor.

Criptografia em trânsito

O Amazon Pinpoint usa HTTPS e Transport Layer Security (TLS) 1.2 ou posterior para se comunicar com clientes e aplicativos. Para se comunicar com outros AWS serviços, o Amazon Pinpoint usa HTTPS e TLS 1.2. Além disso, quando você cria e gerencia recursos do Amazon Pinpoint usando o

console, um AWS SDK ou o. AWS Command Line Interface, todas as comunicações são protegidas usando HTTPS e TLS 1.2.

Gerenciamento de chaves

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

Privacidade do tráfego entre redes

A privacidade do tráfego entre redes se refere à proteção de conexões e tráfego entre o Amazon Pinpoint e seus clientes e aplicativos locais, e entre o Amazon Pinpoint e AWS outros recursos na mesma região. AWS Os seguintes atributos e práticas podem ajudar você a garantir a privacidade de tráfego entre redes para o Amazon Pinpoint.

Tráfego entre clientes do Amazon Pinpoint e clientes on-premises e aplicativos

Para estabelecer uma conexão privada entre o Amazon Pinpoint e clientes e aplicativos na rede on-premises, é possível usar o AWS Direct Connect. Isso permite vincular a rede a um local do AWS Direct Connect usando um cabo Ethernet de fibra ótica padrão. Uma extremidade do cabo é conectada ao roteador. A outra extremidade está conectada a um AWS Direct Connect roteador. Para obter mais informações, consulte O que é o AWS Direct Connect? no Guia do usuário do AWS Direct Connect .

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

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

Tráfego entre o Amazon Pinpoint e outros recursos AWS

Para proteger as comunicações entre o Amazon Pinpoint e outros AWS recursos na mesma AWS região, o Amazon Pinpoint usa HTTPS e TLS 1.2 por padrão.

Criar um endpoint da VPC de interface para o Amazon Pinpoint

Você pode estabelecer uma conexão privada entre a nuvem privada virtual (VPC) e um endpoint no Amazon Pinpoint criando um endpoint da VPC da interface.

Os endpoints de interface são alimentados por AWS PrivateLinkuma tecnologia que permite acessar de forma privada as APIs do Amazon Pinpoint sem um gateway de internet, dispositivo NAT, conexão VPN ou. AWS Direct Connect As instâncias na VPC não precisam de endereços IP públicos para a comunicação com APIs do Amazon Pinpoint que têm integração com AWS PrivateLink.

Para obter mais informações, consulte o Guia do AWS PrivateLink.

Criar endpoints da VPC de uma interface

Você pode criar um endpoint de interface usando o console Amazon VPC ou AWS Command Line Interface o AWS CLI(). Para obter mais informações, consulte Criar um endpoint de interface no AWS PrivateLink Guia.

O Amazon Pinpoint oferece suporte aos seguintes nomes de serviços:

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

Se você ativar o DNS privado para um endpoint de interface, poderá fazer solicitações de API para o Amazon Pinpoint usando o nome DNS padrão para, por Região da AWS exemplo,. `com.amazonaws.us-east-1.pinpoint` Para obter mais informações, consulte hostnames de DNS no Guia do usuário do AWS PrivateLink .

Para obter uma lista de todas as Regiões onde o Amazon Pinpoint está disponível no momento, consulte endpoints de serviço da AWS na Referência geral da Amazon Web Services.

Criar uma política de endpoint da VPC

É possível anexar uma política de endpoint ao endpoint da VPC que controla o acesso. Essa política especifica as seguintes informações:

- A entidade principal que pode executar ações.
- As ações que podem ser executadas.
- Os recursos sobre os quais as ações podem ser realizadas.

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

Exemplo: política de endpoint da VPC

A política de endpoint da VPC a seguir concede acesso às ações listadas do Amazon Pinpoint para todas as entidades principais em todos os recursos.

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

Gerenciamento de identidade e acesso para o Amazon Pinpoint

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

Tópicos

- Público
- Autenticando com identidades

- Gerenciando acesso usando políticas
- Como o Amazon Pinpoint funciona com o IAM
- Ações do Amazon Pinpoint para políticas do IAM
- Exemplos de políticas baseadas em identidade do Amazon Pinpoint
- Perfis do IAM para tarefas comuns do Amazon Pinpoint
- Solução de problemas de gerenciamento de identidade e acesso do Amazon Pinpoint

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no Amazon Pinpoint.

Usuário do serviço: se você usar o serviço do Amazon Pinpoint para fazer seu trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que mais atributos do Amazon Pinpoint forem usados para realizar o trabalho, talvez sejam necessárias 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 atributo no Amazon Pinpoint, consulte Solução de problemas de gerenciamento de identidade e acesso do Amazon Pinpoint.

Administrador do serviço: se você for a pessoa responsável pelos recursos do Amazon Pinpoint em sua empresa, provavelmente terá acesso total ao Amazon Pinpoint. Cabe a você determinar quais atributos e recursos do Amazon Pinpoint os usuários do seu serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como a empresa pode usar o IAM com o Amazon Pinpoint, consulte Como o Amazon Pinpoint funciona com o IAM.

Administrador do IAM: se você for um administrador do IAM, talvez deseje saber detalhes sobre como escrever políticas para gerenciar o acesso ao Amazon Pinpoint. Para visualizar exemplos de políticas baseadas em identidade do Amazon Pinpoint que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade do Amazon Pinpoint.

Autenticando com identidades

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

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (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 faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS 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 assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia do usuário do IAM.

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. Usar a autenticação multifator (MFA) na AWS no Guia do Usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada 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 pode executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do usuário do IAM.

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com 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 usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne Chaves de Acesso Regularmente para Casos de Uso que exijam Credenciais de Longo Prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. 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 nome de grupo IAMAdmins e atribuir a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a um aplicativo, mas uma função pode ser assumida por qualquer pessoa que precisar dela. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando Criar um Usuário do IAM (Ao Invés de uma Função) no Guia do Usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma função do IAM no AWS Management Console trocando de funções. Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte Usando Funções do IAM no Guia do Usuário do IAM.

Funções do IAM 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 mais informações sobre perfis para federação, consulte Criando um Perfil para um Provedor de Identidades Terceirizado no Guia do Usuário do IAM. Se você usa o IAM Identity Center, configure 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 um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no AWS IAM Identity Center Manual do Usuário.
- Permissões de usuários temporárias do IAM: um usuário ou perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.

- Acesso entre contas: você pode usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) acesse recursos na sua conta de uma conta diferente. As funções são a forma primária 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 aprender a diferença entre funções e políticas baseadas em recurso para acesso entre contas, consulte Como as Funções do IAM Diferem das Políticas Baseadas em Recurso no Guia do Usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões de chamada da entidade principal, uma função de serviço ou uma função vinculada ao serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.
  - Função de Serviço: uma função de serviço é uma função do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criando um Perfil para Delegar Permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
  - Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas a serviço.
- Aplicativos em execução no Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo AWS CLI solicitações de API. AWS É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de

instância contém a perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Usar uma função do IAM para conceder permissões a aplicativos em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para aprender se deseja usar perfis do IAM, consulte Quando Criar uma Função do IAM (em Vez de um Usuário) no Guia do Usuário do IAM.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define 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 AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão Geral das Políticas JSON no Guia do Usuário do IAM.

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

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

As políticas do IAM definem permissões para uma ação, independente do método usado para executar 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 da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em quais condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade também podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são incorporadas diretamente a um único usuário, grupo ou função. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como selecionar entre uma política gerenciada ou uma política em linha, consulte Selecionar entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

O Amazon Pinpoint oferece suporte ao uso de políticas baseadas em identidade para controlar o acesso aos recursos do Amazon Pinpoint.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de função do IAM e as políticas do 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 uma entidade principal especificada 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 atributos são políticas em linha que estão localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

O Amazon Pinpoint oferece suporte ao uso de políticas baseadas em recursos para controlar o acesso aos recursos do Amazon Pinpoint.

Listas de controle de acesso (ACLs)

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

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Saiba mais sobre ACLs em Configurações da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

O Amazon Pinpoint não oferece suporte ao uso de ACLs para controlar o acesso aos recursos do Amazon Pinpoint.

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 o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em atributo 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 Entidades do IAM no Guia do Usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON 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 atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte Como os SCPs Funcionam no AWS Organizations Manual do Usuário do.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para uma função 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 do usuário do IAM.

O Amazon Pinpoint oferece suporte ao uso desses tipos de políticas para controlar o acesso aos recursos do Amazon Pinpoint.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida

quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o Amazon Pinpoint funciona com o IAM

Para usar o Amazon Pinpoint, os usuários da sua AWS conta precisam de permissões que lhes permitam visualizar dados analíticos, criar projetos, definir segmentos de usuários, implantar campanhas e muito mais. Se você integrar um aplicativo móvel ou web ao Amazon Pinpoint, os usuários do aplicativo também precisarão acessar o Amazon Pinpoint. Esse acesso permite que o aplicativo registre endpoints e relate dados de uso ao Amazon Pinpoint. Para conceder acesso aos recursos do Amazon Pinpoint, crie políticas AWS Identity and Access Management (IAM) que permitam ações do Amazon Pinpoint para identidades do IAM ou recursos do Amazon Pinpoint.

O IAM é um serviço que ajuda os administradores a controlar com segurança o acesso aos recursos. AWS As políticas do IAM incluem instruções que permitem ou proíbem usuários de executar ações específicas em recursos específicos. O Amazon Pinpoint fornece um conjunto de ações que você pode usar em políticas do IAM para especificar permissões granulares a usuários e recursos do Amazon Pinpoint. Isso significa que você pode conceder o nível de acesso apropriado ao Amazon Pinpoint sem criar políticas excessivamente permissivas que possam expor dados importantes ou comprometer os recursos. Por exemplo, você pode conceder acesso irrestrito a um administrador do Amazon Pinpoint e conceder acesso somente leitura a indivíduos que precisam de acesso a apenas um projeto específico.

Antes de usar o IAM para gerenciar o acesso ao Amazon Pinpoint, você deve entender quais recursos do IAM estão disponíveis para uso com o Amazon Pinpoint. Para ter uma visão de alto nível de como o Amazon Pinpoint e AWS outros serviços funcionam com o IAM, AWS consulte os serviços que funcionam com o IAM no Guia do usuário do IAM.

Tópicos

- Políticas baseadas em identidade do Amazon Pinpoint
- Políticas de permissões baseadas em recursos do Amazon Pinpoint
- Autorização baseada em tags do Amazon Pinpoint
- Perfis do IAM do Amazon Pinpoint

Políticas baseadas em identidade do Amazon Pinpoint

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas.

O Amazon Pinpoint é compatível com ações, recursos e chaves de condição. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

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

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que 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.

Isso significa que as ações de políticas controlam o que os usuários podem fazer no console do Amazon Pinpoint. Eles também controlam o que os usuários podem fazer programaticamente usando diretamente os AWS SDKs, o AWS Command Line Interface (AWS CLI) ou as APIs do Amazon Pinpoint.

As ações de políticas no Amazon Pinpoint usam os seguintes prefixos:

- **`mobiletargeting`**: para ações que derivam da API do Amazon Pinpoint, que é a API principal do Amazon Pinpoint.
- **`sms-voice`**: para ações que derivam da API de voz e SMS do Amazon Pinpoint, que é uma API complementar que oferece opções avançadas para usar e gerenciar os canais de SMS e voz no Amazon Pinpoint.

Por exemplo, para conceder a alguém a permissão para visualizar informações sobre todos os segmentos de um projeto, que é uma ação que corresponde à operação `GetSegments` na API do Amazon Pinpoint, inclua a ação `mobiletargeting:GetSegments` na política. As instruções de política devem incluir um elemento `Action` ou `NotAction`. O Amazon Pinpoint define seu próprio conjunto de ações que descrevem as tarefas que os usuários podem executar com isso.

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

```
"Action": [
```

```
      "mobiletargeting:action1",
      "mobiletargeting:action2"
```

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Get`, inclua a seguinte ação:

```
"Action": "mobiletargeting:Get*"
```

No entanto, como prática recomendada, você deve criar políticas que sigam o princípio de privilégio mínimo. Em outras palavras, você deve criar políticas que incluem somente as permissões necessárias para executar uma ação específica.

Para obter uma lista de ações do Amazon Pinpoint que você pode usar em políticas do IAM, consulte Ações do Amazon Pinpoint para políticas do IAM.

Recursos

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

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

Para ações não compatíveis com permissões no nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Por exemplo, a ação `mobiletargeting:GetSegments` recupera informações sobre todos os segmentos que estão associados a um projeto do Amazon Pinpoint específico. Identifique um projeto com um ARN no seguinte formato:

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

Para obter mais informações sobre o formato de ARNs, consulte Nomes de recurso da Amazon (ARNs) na Referência geral da AWS.

Nas políticas do IAM, especifique os ARNs para os seguintes tipos de recursos do Amazon Pinpoint:

- Campanhas
- Jornadas
- Modelos de mensagens (referidos como modelos em alguns contextos)
- Projetos (referidos como apps ou aplicativos em alguns contextos)
- Modelos de recomendação (conhecidos como recomendações em alguns contextos)
- Segmentos

Por exemplo, para criar uma declaração de política para o projeto que tem o ID do projeto `810c7aab86d42fb2b56c8c966example`, use o seguinte ARN:

```
"Resource": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

Para especificar todos os projetos que pertencem a uma conta específica, use o caractere curinga (*):

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

Algumas ações do Amazon Pinpoint, como determinadas ações para criar recursos, não podem ser realizadas em um recurso específico. Nesses casos, você deve usar o caractere curinga (*):

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

Nas políticas do IAM, você também pode especificar ARNs para os seguintes tipos de recursos de SMS e voz do Amazon Pinpoint:

- Conjunto de configurações
- Lista de exclusão
- Número de telefone
- Grupo
- ID do remetente

Por exemplo, para criar uma declaração de política para um número de telefone que tenha o ID do número de telefone, `phone-12345678901234567890123456789012` use o seguinte ARN:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-12345678901234567890123456789012"
```

Para especificar todos os números de telefone que pertencem a uma conta específica, use um caractere curinga (*) no lugar do ID do número de telefone:

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

Algumas ações de SMS e voz do Amazon Pinpoint não são executadas em um recurso específico, como aquelas as que gerenciam configurações no nível da conta, por exemplo, limites de gastos. Nesses casos, você deve usar o caractere curinga (*):

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

Algumas ações da API do Amazon Pinpoint envolvem vários recursos. Por exemplo, a ação `TagResource` pode adicionar uma tag a vários projetos. Para especificar vários recursos em uma única instrução, separe os ARNs com vírgulas:

```
"Resource": [
      "resource1",
      "resource2"
```

Para ver uma lista dos tipos de recursos do Amazon Pinpoint e seus ARNs, consulte Recursos definidos pelo Amazon Pinpoint no Guia do usuário do IAM. Para saber as ações que podem ser especificadas com o ARN de cada tipo de recurso, consulte Ações definidas pelo Amazon Pinpoint no Guia de usuário do IAM.

Chaves de condição

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

O elemento `Condition` (ou bloco `Condition`) permite especificar condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. Você pode criar expressões

condicionais que usem operadores de condição, como “igual a” ou “menor que”, para corresponder a condição da política aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. 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, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de Política do IAM: Variáveis e Tags no Guia do Usuário do IAM.

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

O Amazon Pinpoint define o próprio conjunto de chaves de condição e também oferece suporte para algumas chaves de condição globais. Para ver uma lista de todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM. Para ver uma lista de chaves de condição do Amazon Pinpoint, consulte Chaves de condição para o Amazon Pinpoint no Guia do usuário do IAM. Para saber com quais ações e recursos é possível usar uma chave de condição, consulte Ações definidas pelo Amazon Pinpoint no Guia do usuário do IAM.

Exemplos

Para ver exemplos de políticas baseadas em identidade do Amazon Pinpoint, consulte Exemplos de políticas baseadas em identidade do Amazon Pinpoint.

Políticas de permissões baseadas em recursos do Amazon Pinpoint

As políticas baseadas em recursos são documentos de políticas JSON que especificam quais ações uma entidade principal pode executar em um recurso do Amazon Pinpoint e em quais condições. O Amazon Pinpoint oferece suporte a políticas de permissões baseadas em recursos para campanhas, jornadas, modelos de mensagens (modelos), modelos de recomendação (recomendações), projetos (aplicativos) e segmentos.

Exemplos

Para ver exemplos de políticas baseadas em recursos do Amazon Pinpoint, consulte the section called “Exemplos de políticas baseadas em identidade”.

Autorização baseada em tags do Amazon Pinpoint

É possível associar tags a determinados tipos de recursos do Amazon Pinpoint ou passar tags em uma solicitação para o Amazon Pinpoint. Para controlar o acesso baseado em tags, forneça informações sobre a tag no elemento de condição de uma política usando as chaves de condição `aws:ResourceTag/${TagKey}`, `aws:RequestTag/${TagKey}` ou `aws:TagKeys`.

Para obter informações sobre como marcar recursos do Amazon Pinpoint, incluindo uma política do IAM de exemplo, consulte Marcar recursos do Amazon Pinpoint.

Perfis do IAM do Amazon Pinpoint

Perfil do IAM é uma entidade dentro da sua conta da AWS que tem permissões específicas.

Usar credenciais temporárias com o Amazon Pinpoint

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir uma função entre contas. Você obtém credenciais de segurança temporárias chamando AWS Security Token Service (AWS STS) operações de API, como AssumeRoleou GetFederationToken.

O Amazon Pinpoint oferece suporte ao uso de credenciais temporárias.

Funções vinculadas a serviço

As funções vinculadas ao serviço permitem que AWS os serviços acessem recursos em outros serviços para concluir uma ação em seu nome. Os perfis vinculados a serviço aparecem na sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados a serviço.

O Amazon Pinpoint não usa funções vinculadas ao serviço.

Perfis de serviço

Esse atributo permite que um serviço assuma um perfil de serviço em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso indica que um administrador do IAM pode alterar as permissões para essa função. Porém, fazer isso pode alterar a funcionalidade do serviço.

O Amazon Pinpoint oferece suporte ao uso de funções de serviço.

Ações do Amazon Pinpoint para políticas do IAM

Para gerenciar o acesso aos recursos do Amazon Pinpoint em sua AWS conta, você pode adicionar ações do Amazon Pinpoint AWS Identity and Access Management às políticas (IAM). Ao usar ações em políticas, é possível controlar o que os usuários podem fazer no console do Amazon Pinpoint. Você também pode controlar o que os usuários podem fazer programaticamente usando diretamente os AWS SDKs, o AWS Command Line Interface (AWS CLI) ou as APIs do Amazon Pinpoint.

Em uma política, deve-se especificar cada ação com o namespace apropriado do Amazon Pinpoint, seguido de dois-pontos e do nome da ação, como `GetSegments`. A maioria dessas ações corresponde a uma solicitação à API do Amazon Pinpoint usando um método URI e HTTP específico. Por exemplo, se você permitir a ação `mobiletargeting:GetSegments` na política de um usuário, ele terá permissão para recuperar informações sobre todos os segmentos de um projeto ao enviar uma solicitação HTTP GET para o URI `/apps/projectId/segments`. Essa política também permite que o usuário visualize essas informações no console e recupere essas informações usando um AWS SDK ou o. AWS CLI

Cada ação é executada em um recurso do Amazon Pinpoint específico, que é identificado em uma declaração de política pelo nome do recurso da Amazon (ARN). Por exemplo, a ação `mobiletargeting:GetSegments` é executada em um projeto específico, identificado com o ARN, `arn:aws:mobiletargeting:region:accountId:apps/projectId`.

Este tópico identifica as ações do Amazon Pinpoint que você pode adicionar às políticas do IAM para a conta da AWS . Para ver exemplos que demonstram como você pode usar as ações nas políticas para gerenciar o acesso aos recursos do Amazon Pinpoint, consulte Exemplos de políticas baseadas em identidade do Amazon Pinpoint.

Tópicos

- Ações da API do Amazon Pinpoint
- Ações da API de SMS e voz, versão 1, do Amazon Pinpoint

Ações da API do Amazon Pinpoint

Esta seção identifica ações para atributos disponíveis na API do Amazon Pinpoint, que é a API principal do Amazon Pinpoint. Para saber mais sobre essa API, consulte Referência da API do Amazon Pinpoint.

Categorias:

- Análises e métricas
- Campanhas
- Canais
- Endpoints
- Streams de eventos
- Eventos
- Trabalhos de exportação
- Trabalhos de importação
- Jornadas
- Modelos de mensagens
- Mensagens
- Senhas de uso único
- Validação do número de telefone
- Projetos
- Modelos de recomendação
- Segmentos
- Tags
- Usuários

Análises e métricas

As permissões a seguir estão relacionadas à exibição de dados analíticos no console do Amazon Pinpoint. Elas também estão relacionadas à recuperação (consulta) de dados agregados para métricas padrão, também chamados de principais indicadores de desempenho (KPIs), que se aplicam a projetos, campanhas e jornadas.

`mobiletargeting:GetReports`

Visualizar dados de análise no console do Amazon Pinpoint. Essa permissão também é necessária para criar segmentos que contenham atributos personalizados usando o console do Amazon Pinpoint. Também é necessário obter uma estimativa do tamanho de um segmento no console do Amazon Pinpoint.

- URI – Não aplicável
- Método – Não aplicável
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:GetApplicationDateRangeKpi`

Recupere (consulte) dados agregados para uma métrica de aplicativo padrão. Esta é uma métrica que se aplica a todas as campanhas ou mensagens transacionais associadas a um projeto.

- URI – `/apps/projectId/kpis/daterange/kpi-name`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/kpis/daterange/kpi-name`

`mobiletargeting:GetCampaignDateRangeKpi`

Recupere (consulte) dados agregados para uma métrica de campanha padrão. Esta é uma métrica que se aplica a uma campanha individual.

- URI – `/apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/campaigns/campaignId/kpis/daterange/kpi-name`

`mobiletargeting:GetJourneyDateRangeKpi`

Recupere (consulte) dados agregados para uma métrica de engajamento de jornada padrão. Essa é uma métrica de engajamento que se aplica a uma jornada individual, por exemplo, o número de mensagens que foram abertas pelos participantes em todas as atividades de uma jornada.

- URI – `/apps/projectId/journeys/journeyId/kpis/daterange/kpi-name`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/kpis/daterange/kpi-name`

`mobiletargeting:GetJourneyExecutionMetrics`

Recupere (consulte) dados agregados para métricas de execução padrão que se aplicam a uma jornada individual, por exemplo, o número de participantes que estão realizando ativamente todas as atividades de uma jornada.

- URI – /apps/*projectId*/journeys/*journeyId*/execution-metrics
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/execution-metrics

mobiletargeting:GetJourneyExecutionActivityMetrics

Recupere (consulte) dados agregados para métricas de execução padrão que se aplicam a uma atividade individual em uma jornada, por exemplo, o número de participantes que começaram ou concluíram uma atividade.

- URI – /apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics

Campanhas

As permissões a seguir estão relacionadas ao gerenciamento de campanhas em sua conta do Amazon Pinpoint.

mobiletargeting:CreateCampaign

Crie uma campanha para um projeto.

- URI – /apps/*projectId*/campaigns
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns

mobiletargeting:DeleteCampaign

Excluir uma determinada campanha.

- URI – /apps/*projectId*/campaigns/*campaignId*
- Método – EXCLUIR
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaign

Recuperar informações sobre uma campanha específica.

- URI – /apps/*projectId*/campaigns/*campaignId*
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignActivities

Recuperar informações sobre as atividades executadas por uma campanha.

- URI – /apps/*projectId*/campaigns/*campaignId*/activities
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaigns

Recupere informações sobre todas as campanhas para um projeto.

- URI – /apps/*projectId*/campaigns
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetCampaignVersion

Recuperar informações sobre uma versão de campanha específica.

- URI – /apps/*projectId*/campaigns/*campaignId*/versions/*versionId*
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignVersions

Recuperar informações sobre as versões atual e anteriores de uma campanha.

- URI – /apps/*projectId*/campaigns/*campaignId*/versions
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:UpdateCampaign

Atualizar uma determinada campanha.

- URI – /apps/*projectId*/campaigns/*campaignId*
- Método – PUT
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

Canais

As permissões a seguir estão relacionadas ao gerenciamento de canais em sua conta do Amazon Pinpoint. No Amazon Pinpoint, os canais se referem aos métodos que você usa para entrar em contato com seus clientes, como enviando e-mail, mensagens SMS ou notificações por push.

mobiletargeting:DeleteAdmChannel

Desabilite o canal Amazon Device Messaging (ADM) de um projeto.

- URI – /apps/*projectId*/channels/adm
- Método – EXCLUIR
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:GetAdmChannel

Recupere informações sobre o canal ADM de um projeto.

- URI – /apps/*projectId*/channels/adm
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:UpdateAdmChannel

Ative ou atualize o canal ADM de um projeto.

- URI – /apps/*projectId*/channels/adm
- Método – PUT
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:DeleteApnsChannel

Desabilite o canal do serviço Apple Push Notification (APNs) de um projeto.

- URI – /apps/*projectId*/channels/apns
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns`

mobiletargeting:GetApnsChannel

Recupere informações sobre o canal APNs para um projeto.

- URI – /apps/*projectId*/channels/apns
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns`

mobiletargeting:UpdateApnsChannel

Ative ou atualize o canal de APNs de um projeto.

- URI – /apps/*projectId*/channels/apns
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns`

mobiletargeting:DeleteApnsSandboxChannel

Desabilite o canal sandbox dos APNs de um projeto.

- URI – /apps/*projectId*/channels/apns_sandbox
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_sandbox`

mobiletargeting:GetApnsSandboxChannel

Recupere informações sobre o canal sandbox dos APNs de um projeto.

- URI – /apps/*projectId*/channels/apns_sandbox
- Método – GET

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:UpdateApnsSandboxChannel`

Ative ou atualize o canal sandbox de APNs para um projeto.

- URI – `/apps/projectId/channels/apns_sandbox`
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_sandbox`

`mobiletargeting:DeleteApnsVoipChannel`

Desabilite o canal VoIP dos APNs de um projeto.

- URI – `/apps/projectId/channels/apns_voip`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:GetApnsVoipChannel`

Recupere informações sobre o canal VoIP dos APNs de um projeto.

- URI – `/apps/projectId/channels/apns_voip`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:UpdateApnsVoipChannel`

Ative ou atualize o canal VoIP de APNs de um projeto.

- URI – `/apps/projectId/channels/apns_voip`
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/apns_voip`

`mobiletargeting:DeleteApnsVoipSandboxChannel`

Desabilite o canal sandbox VoIP dos APNs de um projeto.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip_sandbox`

`mobiletargeting:GetApnsVoipSandboxChannel`

Recupere informações sobre o canal APNs VoIP Sandbox de um projeto.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip_sandbox`

`mobiletargeting:UpdateApnsVoipSandboxChannel`

Ative ou atualize o canal sandbox VoIP de APNs de um projeto.

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/apns_voip_sandbox`

`mobiletargeting:DeleteBaiduChannel`

Desabilite o canal Baidu Cloud Push de um projeto.

- URI – /apps/*projectId*/channels/baidu
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

`mobiletargeting:GetBaiduChannel`

Recupere informações sobre o canal Baidu Cloud Push de um projeto.

- URI – /apps/*projectId*/channels/baidu
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/channels/baidu`

mobiletargeting:UpdateBaiduChannel

Ative ou atualize o canal Baidu Cloud Push de um projeto.

- URI – /apps/*projectId*/channels/baidu
- Método – PUT
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:DeleteEmailChannel

Desabilite o canal de e-mail de um projeto.

- URI – /apps/*projectId*/channels/email
- Método – EXCLUIR
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:GetEmailChannel

Recupere informações sobre o canal de e-mail de um projeto.

- URI – /apps/*projectId*/channels/email
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:UpdateEmailChannel

Ative ou atualize o canal de e-mail para um projeto.

- URI – /apps/*projectId*/channels/email
- Método – PUT
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:DeleteGcmChannel

Desabilite o canal Firebase Cloud Messaging (FCM) de um projeto. Este canal permite ao Amazon Pinpoint enviar notificações por push para um aplicativo Android por meio do serviço FCM, que substitui o serviço Google Cloud Messaging (GCM).

- URI – /apps/*projectId*/channels/gcm
- Método – EXCLUIR

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

`mobiletargeting:GetGcmChannel`

Recupere informações sobre o canal FCM de um projeto. Este canal permite ao Amazon Pinpoint enviar notificações por push para um aplicativo Android por meio do serviço FCM, que substitui o serviço Google Cloud Messaging (GCM).

- URI – /apps/*projectId*/channels/gcm
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

`mobiletargeting:UpdateGcmChannel`

Ative ou atualize o canal FMC de um projeto. Este canal permite ao Amazon Pinpoint enviar notificações por push para um aplicativo Android por meio do serviço FCM, que substitui o serviço Google Cloud Messaging (GCM).

- URI – /apps/*projectId*/channels/gcm
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/gcm`

`mobiletargeting:DeleteSmsChannel`

Desabilite o canal de SMS de um projeto.

- URI – /apps/*projectId*/channels/sms
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

`mobiletargeting:GetSmsChannel`

Recupere informações sobre o canal de SMS para um projeto.

- URI – /apps/*projectId*/channels/sms
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/channels/sms`

mobiletargeting:UpdateSmsChannel

Ative ou atualize o canal de SMS para um projeto.

- URI – /apps/*projectId*/channels/sms
- Método – PUT
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetChannels

Obtém informações sobre o histórico e o status de cada canal de um aplicativo.

- URI – /apps/*application-id*/channels
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/channels

mobiletargeting:DeleteVoiceChannel

Desativa o canal de voz de um aplicativo e exclui todas as configurações existentes do canal.

- URI – /apps/*application-id*/channels/voice
- Método – EXCLUIR
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:GetVoiceChannel

Obtém informações sobre o status e as configurações do canal de voz de um aplicativo.

- URI – /apps/*application-id*/channels/voice
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:UpdateVoiceChannel

Habilita o canal de voz de um aplicativo ou atualiza o status e as configurações do canal de voz de um aplicativo.

- URI – /apps/*application-id*/channels/voice
- Método – PUT

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectid/channels/voice`

Endpoints

As permissões a seguir estão relacionadas ao gerenciamento de endpoints em sua conta do Amazon Pinpoint. No Amazon Pinpoint, um endpoint é um destino único para suas mensagens. Por exemplo, um endpoint pode ser um endereço de e-mail do cliente, número de telefone ou token de dispositivo móvel.

`mobiletargeting:DeleteEndpoint`

Excluir um endpoint

- URI – `/apps/projectId/endpoints/endpointId`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

`mobiletargeting:GetEndpoint`

Recuperar informações sobre um endpoint específico.

- URI – `/apps/projectId/endpoints/endpointId`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/endpoints/endpointId`

`mobiletargeting:RemoveAttributes`

Remove um ou mais atributos, do mesmo tipo de atributo, de todos os endpoints associados a um aplicativo.

- URI – `apps/application-id/attributes/attribute-type`
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/attributes/attribute-type`

`mobiletargeting:UpdateEndpoint`

Criar um endpoint ou atualizar as informações de um endpoint.

- URI – /apps/*projectId*/endpoints/*endpointId*
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/endpoints/`*endpointId*

`mobiletargeting:UpdateEndpointsBatch`

Criar ou atualizar endpoints como uma operação em lote.

- URI – /apps/*projectId*/endpoints
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*

Streams de eventos

As permissões a seguir estão relacionadas ao gerenciamento de fluxos de eventos para sua conta do Amazon Pinpoint.

`mobiletargeting:DeleteEventStream`

Exclua o fluxo de eventos de um projeto.

- URI – /apps/*projectId*/eventstream/
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/eventstream`

`mobiletargeting:GetEventStream`

Recupere informações sobre o fluxo de eventos de um projeto.

- URI – /apps/*projectId*/eventstream/
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:`*region*`:`*accountId*`:apps/`*projectId*`/eventstream`

`mobiletargeting:PutEventStream`

Crie ou atualize um fluxo de eventos de um projeto.

- URI – /apps/*projectId*/eventstream/

- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/eventstream`

Eventos

As permissões a seguir estão relacionadas ao gerenciamento de trabalhos de eventos em sua conta do Amazon Pinpoint. No Amazon Pinpoint, você cria trabalhos de importação para criar segmentos com base nas definições de endpoint armazenadas em um bucket do S3.

`mobiletargeting:PutEvents`

Cria um novo evento de registro para endpoints ou cria ou atualiza dados de endpoint aos quais os eventos existentes estão associados.

- URI – `/apps/application-id/events`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/events`

Trabalhos de exportação

As permissões a seguir estão relacionadas ao gerenciamento de trabalhos de exportação em sua conta do Amazon Pinpoint. No Amazon Pinpoint, você cria trabalhos de exportação para enviar informações sobre endpoints para um bucket do Amazon S3 para armazenamento ou análise.

`mobiletargeting:CreateExportJob`

Crie um trabalho de exportação para exportar definições de endpoint para o Amazon S3.

- URI – `/apps/projectId/jobs/export`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

`mobiletargeting:GetExportJob`

Recupere informações sobre um trabalho de exportação específico de um projeto.

- URI – `/apps/projectId/jobs/export/jobId`

- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export/jobId`

`mobiletargeting:GetExportJobs`

Recupere uma lista de todos os trabalhos de exportação de um projeto.

- URI – `/apps/projectId/jobs/export`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/export`

Trabalhos de importação

As permissões a seguir estão relacionadas ao gerenciamento de trabalhos de importação em sua conta do Amazon Pinpoint. No Amazon Pinpoint, você cria trabalhos de importação para criar segmentos com base nas definições de endpoint armazenadas em um bucket do S3.

`mobiletargeting:CreateImportJob`

Importar definições de endpoint do Amazon S3 para criar um segmento.

- URI – `/apps/projectId/jobs/import`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetImportJob`

Recupere informações sobre um trabalho de importação específico de um projeto.

- URI – `/apps/projectId/jobs/import/jobId`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/jobs/import/jobId`

`mobiletargeting:GetImportJobs`

Recupere informações sobre todas as tarefas de importação de um projeto.

- URI – `/apps/projectId/jobs/import`

- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Jornadas

As permissões a seguir estão relacionadas ao gerenciamento de jornadas em sua conta do Amazon Pinpoint.

`mobiletargeting:CreateJourney`

Crie uma jornada para um projeto.

- URI – `/apps/projectId/journeys`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

`mobiletargeting:GetJourney`

Recuperar informações sobre uma jornada específica.

- URI – `/apps/projectId/journeys/journeyId`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

`mobiletargeting:ListJourneys`

Recupere informações sobre todas as jornadas de um projeto.

- URI – `/apps/projectId/journeys`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys`

`mobiletargeting:UpdateJourney`

Atualize a configuração e outras configurações para uma jornada específica.

- URI – `/apps/projectId/journeys/journeyId`
- Método – PUT

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

`mobiletargeting:UpdateJourneyState`

Cancele uma jornada ativa.

- URI – `/apps/projectId/journeys/journeyId/state`
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId/state`

`mobiletargeting:DeleteJourney`

Excluir uma jornada específica.

- URI – `/apps/projectId/journeys/journeyId`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/journeys/journeyId`

Modelos de mensagens

As permissões a seguir estão relacionadas à criação e ao gerenciamento de modelos de mensagem para sua conta do Amazon Pinpoint. Um modelo de mensagem é um conjunto de conteúdo e configurações que você pode definir, salvar e reutilizar em mensagens enviadas para qualquer um dos seus projetos do Amazon Pinpoint.

`mobiletargeting:ListTemplates`

Recupere informações sobre todos os modelos de mensagem associados à sua conta do Amazon Pinpoint.

- URI – `/templates`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates`

`mobiletargeting:ListTemplateVersions`

Recuperar informações sobre todas as versões de um modelo de mensagem específico.

- URI – /templates/*template-name*/*template-type*/versions
- Método – GET
- ARN do recurso: não aplicável

mobiletargeting:UpdateTemplateActiveVersion

Designe uma versão específica de um modelo de mensagem como a versão ativa do modelo.

- URI – /templates/*template-name*/*template-type*/active-version
- Método – GET
- ARN do recurso: não aplicável

mobiletargeting:GetEmailTemplate

Recupere informações sobre um modelo de mensagem para mensagens enviadas por meio do canal de e-mail.

- URI – /templates/*template-name*/email
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:CreateEmailTemplate

Crie um modelo de mensagem para mensagens enviadas por meio do canal de e-mail.

- URI – /templates/*template-name*/email
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:UpdateEmailTemplate

Atualize um modelo de mensagem existente para mensagens enviadas pelo canal de e-mail.

- URI – /templates/*template-name*/email
- Método – PUT

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:DeleteEmailTemplate`

Exclua um modelo de mensagem para mensagens enviadas pelo canal de e-mail.

- URI – `/templates/template-name/email`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/EMAIL`

`mobiletargeting:GetPushTemplate`

Recupere informações sobre um modelo de mensagem para mensagens enviadas por meio de um canal de notificação por push.

- URI – `/templates/template-name/push`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:CreatePushTemplate`

Crie um modelo de mensagem para mensagens enviadas por meio de um canal de notificação por push.

- URI – `/templates/template-name/push`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:UpdatePushTemplate`

Atualize um modelo de mensagem existente para mensagens enviadas por meio de um canal de notificação por push.

- URI – `/templates/template-name/push`
- Método – PUT

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:DeletePushTemplate`

Exclua um modelo de mensagem para mensagens enviadas por meio de um canal de notificação por push.

- URI – `/templates/template-name/push`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/PUSH`

`mobiletargeting:GetSmsTemplate`

Recupere informações sobre um modelo de mensagem para mensagens enviadas por meio do canal SMS.

- URI – `/templates/template-name/sms`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:CreateSmsTemplate`

Crie um modelo de mensagem para mensagens enviadas por meio do canal SMS.

- URI – `/templates/template-name/sms`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:UpdateSmsTemplate`

Atualize um modelo de mensagem existente para mensagens enviadas por meio do canal SMS.

- URI – `/templates/template-name/sms`
- Método – PUT

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:DeleteSmsTemplate`

Exclua um modelo de mensagem para mensagens enviadas por meio do canal SMS.

- URI – `/templates/template-name/sms`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/SMS`

`mobiletargeting:GetVoiceTemplate`

Recupere informações sobre um modelo de mensagem para mensagens enviadas por meio do canal de voz.

- URI – `/templates/template-name/voice`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:CreateVoiceTemplate`

Crie um modelo de mensagem para mensagens enviadas por meio do canal de voz.

- URI – `/templates/template-name/voice`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:UpdateVoiceTemplate`

Atualize um modelo de mensagem existente para mensagens enviadas por meio do canal de voz.

- URI – `/templates/template-name/voice`
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:templates/template-name/VOICE`

`mobiletargeting:DeleteVoiceTemplate`

Exclua um modelo de mensagem para mensagens enviadas por meio do canal de voz.

- URI – /templates/*template-name*/voice
- Método – EXCLUIR
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

Mensagens

As permissões a seguir são relacionadas ao envio de mensagens e notificações por push de sua conta do Amazon Pinpoint. Você pode usar as operações SendUsersMessages e SendMessages para enviar mensagens para endpoints específicos sem criar segmentos e campanhas primeiro.

mobiletargeting:SendMessages

Envie uma mensagem ou notificação por push para endpoints específicos.

- URI – /apps/*projectId*/messages
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

mobiletargeting:SendUsersMessages

Envie uma mensagem ou notificação por push para todos os endpoints associados a um ID de usuário específico.

- URI – /apps/*projectId*/users-messages
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

Senhas de uso único

As permissões a seguir estão relacionadas ao envio e à validação de senhas de uso único (OTPs) no Amazon Pinpoint.

mobiletargeting:SendOTPMessage

Envie uma mensagem de texto que contenha uma senha de uso único.

- URI – /apps/*projectId*/otp
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/otp

mobiletargeting:VerifyOTPMessage

Verifique a validade de uma senha de uso único (OTP) que foi gerada usando a operação SendOTPMessage.

- URI – /apps/*projectId*/verify-otp
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/verify-otp

Validação do número de telefone

As permissões a seguir são relacionadas ao uso do serviço de validação do número de telefone no Amazon Pinpoint.

mobiletargeting:PhoneNumberValidate

Recupere informações sobre um número de telefone.

- URI – /phone/number/validate
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:phone/number/validate

Projetos

As permissões a seguir estão relacionadas ao gerenciamento de projetos em sua conta do Amazon Pinpoint. Inicialmente, os projetos eram chamados de aplicações. Para essas operações, um aplicativo do Amazon Pinpoint é o mesmo que um projeto do Amazon Pinpoint.

mobiletargeting:CreateApp

Crie um projeto do Amazon Pinpoint.

- URI – /apps

- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:DeleteApp`

Excluir um projeto do Amazon Pinpoint.

- URI – `/apps/projectId`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApp`

Recupere informações sobre um projeto do Amazon Pinpoint.

- URI – `/apps/projectId`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:GetApps`

Recupere informações sobre todos os projetos associados à sua conta do Amazon Pinpoint.

- URI – `/apps`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps`

`mobiletargeting:GetApplicationSettings`

Recupere as configurações padrão de um projeto do Amazon Pinpoint.

- URI – `/apps/projectId/settings`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

`mobiletargeting:UpdateApplicationSettings`

Atualize as configurações padrão de um projeto do Amazon Pinpoint.

- URI – `/apps/projectId/settings`
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId`

Modelos de recomendação

As permissões a seguir estão relacionadas ao gerenciamento das configurações do Amazon Pinpoint para recuperação e processamento de dados de recomendação de modelos de recomendação. Um modelo de recomendação é um tipo de modelo de machine learning que prevê e gera recomendações personalizadas encontrando padrões em dados.

`mobiletargeting:CreateRecommenderConfiguration`

Crie uma configuração do Amazon Pinpoint para um modelo de recomendação.

- URI – `/recommenders`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:recommenders`

`mobiletargeting:GetRecommenderConfigurations`

Recupere informações sobre todas as configurações de modelo de recomendação associadas à sua conta do Amazon Pinpoint.

- URI – `/recommenders`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:recommenders`

`mobiletargeting:GetRecommenderConfiguration`

Recupere informações sobre uma configuração individual do Amazon Pinpoint para um modelo de recomendação.

- URI – `/recommenders/`*`recommenderId`*
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:recommenders/`*`recommenderId`*

`mobiletargeting:UpdateRecommenderConfiguration`

Atualize uma configuração do Amazon Pinpoint para um modelo de recomendação.

- URI – `/recommenders/`*`recommenderId`*
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:recommenders/`*`recommenderId`*

mobiletargeting:DeleteRecommenderConfiguration

Excluir uma configuração do Amazon Pinpoint para um modelo de recomendação.

- URI – /recommenders/*recommenderId*
- Método – EXCLUIR
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

Segmentos

As permissões a seguir estão relacionadas ao gerenciamento de segmentos em sua conta do Amazon Pinpoint. No Amazon Pinpoint, os segmentos são grupos de destinatários para suas campanhas que compartilham determinados atributos que você define.

mobiletargeting:CreateSegment

Crie um segmento. Para permitir que um usuário crie um segmento importando dados de endpoint fora do Amazon Pinpoint, permita a ação mobiletargeting:CreateImportJob.

- URI – /apps/*projectId*/segments
- Método – POST
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:DeleteSegment

Exclua um segmento.

- URI – /apps/*projectId*/segments/*segmentId*
- Método – EXCLUIR
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegment

Recuperar informações sobre um segmento específico.

- URI – /apps/*projectId*/segments/*segmentId*
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentExportJobs

Recupere informações sobre tarefas que exportam definições de endpoint de um segmento.

- URI – /apps/*projectId*/segments/*segmentId*/jobs/export
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*/jobs/export

mobiletargeting:GetSegments

Recupere informações sobre todos os segmentos de um projeto.

- URI – /apps/*projectId*/segments
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetSegmentImportJobs

Recupere informações sobre trabalhos que criam segmentos importando definições de endpoint do Amazon S3.

- URI – /apps/*projectId*/segments/*segmentId*/jobs/import
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersion

Recuperar informações sobre uma versão de segmento específica.

- URI – /apps/*projectId*/segments/*segmentId*/versions/*versionId*
- Método – GET
- Recurso ARN: arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersions

Recuperar informações sobre as versões atual e anteriores de um segmento.

- URI – /apps/*projectId*/segments/*segmentId*/versions
- Método – GET

- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

`mobiletargeting:UpdateSegment`

Atualizar um segmento específico.

- URI – `/apps/projectId/segments/segmentId`
- Método – PUT
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/segments/segmentId`

Tags

As permissões a seguir estão relacionadas à visualização e ao gerenciamento de tags de recursos do Amazon Pinpoint.

`mobiletargeting:ListTagsForResource`

Recupere informações sobre as tags associadas a um projeto, campanha, modelo de mensagem ou segmento.

- URI – `/tags/resource-arn`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:TagResource`

Adicione uma ou mais tags a um projeto, campanha, modelo de mensagem ou segmento.

- URI – `/tags/resource-arn`
- Método – POST
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:*`

`mobiletargeting:UntagResource`

Remova uma ou mais tags de um projeto, campanha, modelo de mensagem ou segmento.

- URI – `/tags/resource-arn`

- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:*`

Usuários

As permissões a seguir estão relacionadas ao gerenciamento de usuários. No Amazon Pinpoint, os usuários correspondem às pessoas que recebem mensagens que você envia. Um único usuário pode estar associado a mais de um endpoint.

`mobiletargeting:DeleteUserEndpoints`

Exclua todos os endpoints associados a um ID de usuário.

- URI – `/apps/projectId/users/userId`
- Método – EXCLUIR
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

`mobiletargeting:GetUserEndpoints`

Recupere informações sobre todos os endpoints associados a um ID de usuário.

- URI – `/apps/projectId/users/userId`
- Método – GET
- Recurso ARN: `arn:aws:mobiletargeting:region:accountId:apps/projectId/users/userId`

Ações da API de SMS e voz, versão 1, do Amazon Pinpoint

Esta seção identifica ações de atributos disponíveis na API de voz e SMS do Amazon Pinpoint. Esta é uma API complementar que oferece opções avançadas para uso e gerenciamento de canais de SMS e voz no Amazon Pinpoint. Para saber mais sobre essa API, consulte Referência da API de SMS e voz do Amazon Pinpoint.

`sms-voice:CreateConfigurationSet`

Crie um conjunto de configurações para envio de mensagens de voz.

- URI – `/sms-voice/configuration-sets`

- Método – POST
- ARN do recurso: não disponível. Usar *.

sms-voice:DeleteConfigurationSet

Exclua um conjunto de configurações para envio de mensagens de voz.

- URI — /sms-voice/configuration-sets/ *ConfigurationSetName*
- Método – EXCLUIR
- ARN do recurso: não disponível. Usar *.

sms-voice:GetConfigurationSetEventDestinations

Recupere informações sobre um conjunto de configuração e os destinos de eventos que ele contém.

- URI — /sms-voice/configuration-sets/ /event-destinations *ConfigurationSetName*
- Método – GET
- ARN do recurso: não disponível. Usar *.

sms-voice:CreateConfigurationSetEventDestination

Crie um destino de evento para eventos de voz.

- URI — /sms-voice/configuration-sets/ /event-destinations *ConfigurationSetName*
- Método – POST
- ARN do recurso: não disponível. Usar *.

sms-voice:UpdateConfigurationSetEventDestination

Atualize um destino de evento para eventos de voz.

- URI — /sms-voice/configuration-sets/ /event-destinations/ *ConfigurationSetNameEventDestinationName*
- Método – PUT
- ARN do recurso: não disponível. Usar *.

`sms-voice:DeleteConfigurationSetEventDestination`

Exclua um destino de evento para eventos de voz.

- URI — /sms-voice/configuration-sets/ /event-destinations/ *`ConfigurationSetNameEventDestinationName`*
- Método – EXCLUIR
- ARN do recurso: não disponível. Usar *.

`sms-voice:SendVoiceMessage`

Crie e envie mensagens de voz.

- URI – /sms-voice/voice/message
- Método – POST
- ARN do recurso: não disponível. Usar *.

Exemplos de políticas baseadas em identidade do Amazon Pinpoint

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos do Amazon Pinpoint. Eles também não podem realizar tarefas usando o AWS Management Console AWS CLI, ou uma AWS API. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e funções permissão para executar operações de API específicas nos recursos de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM utilizando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

- Melhores práticas de política
- Usar o console do Amazon Pinpoint
- Exemplo: acessar um único projeto do Amazon Pinpoint
- Exemplo: visualizar recursos do Amazon Pinpoint com base em tags
- Exemplo: permitir que os usuários visualizem suas próprias permissões
- Exemplos: fornecer acesso às ações da API do Amazon Pinpoint
- Exemplos: fornecer acesso às ações da API de SMS e voz no Amazon Pinpoint
- Exemplo: restringir o acesso ao projeto do Amazon Pinpoint a endereços IP específicos

- Exemplo: restringir o acesso ao Amazon Pinpoint com base em tags
- Exemplo: permitir que o Amazon Pinpoint envie e-mails usando identidades que foram verificadas no Amazon SES

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon Pinpoint em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas Gerenciadas pela AWS ou AWS Políticas Gerenciadas para Funções de Trabalho no Guia do Usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e Permissões no IAM no Guia do Usuário do IAM.
- Utilize condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode gravar uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Condição de Elementos de Política JSON do IAM no Guia do Usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM para garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam o idioma de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de 100 verificações de política e ações recomendadas para ajudar você a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de Política do IAM Access Analyzer no Guia do Usuário do IAM.

- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configurando Acesso à API Protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas Recomendadas de Segurança no IAM no Guia do Usuário do IAM.

Usar o console do Amazon Pinpoint

Para acessar o console do Amazon Pinpoint, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amazon Pinpoint em sua AWS conta. Se você criar uma política baseada em identidade que aplique permissões mais restritivas do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou funções) com essa política. Para garantir que essas entidades possam usar o console do Amazon Pinpoint, anexe a política a elas. Para obter mais informações, consulte Adicionando Permissões a um Usuário no Guia do Usuário do IAM.

O exemplo de política a seguir fornece acesso somente de leitura ao console do Amazon Pinpoint em uma região específica. AWS Isso inclui acesso somente de leitura a outros serviços dos quais o console do Amazon Pinpoint depende, como Amazon Simple Email Service (Amazon SES), IAM e Amazon Kinesis.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "UseConsole",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
             ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
```

```
                "iam:ListRoles",
                "kinesis:ListStreams",
                "s3:List*",
                "ses:Describe*",
                "ses:Get*",
                "ses:List*",
                "sns:ListTopics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

No exemplo de política anterior, substitua *region* pelo nome de uma AWS região e substitua *accountId pelo ID* da sua conta. AWS

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

Exemplo: acessar um único projeto do Amazon Pinpoint

Você também pode criar políticas somente leitura que fornecem acesso somente a projetos específicos. O exemplo a seguir permite que os usuários entrem no console e visualizem uma lista de projetos. Ele também permite visualizar informações sobre recursos relacionados para outros serviços da AWS dos quais o console do Amazon Pinpoint depende, como Amazon SES, IAM e Amazon Kinesis. Contudo, a política apenas permite que os usuários visualizem informações adicionais sobre o projeto especificado nela. Você pode modificar essa política para permitir o acesso a projetos ou AWS regiões adicionais.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
```

```
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:region:accountId:apps/projectId",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/*",
                "arn:aws:mobiletargeting:region:accountId:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

No exemplo anterior, substitua *region* pelo nome de uma AWS região, substitua *accountId* pelo ID da AWS sua conta e substitua *projectID* pelo ID do projeto Amazon Pinpoint ao qual você deseja fornecer acesso.

Da mesma forma, você pode criar políticas que concedam a um usuário em sua AWS conta acesso de gravação limitado a um dos seus projetos do Amazon Pinpoint, por exemplo, o projeto que tem o ID do `810c7aab86d42fb2b56c8c966example` projeto. Nesse caso, permita que o usuário

visualize, adicione e atualize os componentes do projeto, como segmentos e campanhas, mas não exclua os componentes.

Além de conceder permissões para as ações `mobiletargeting:Get` e `mobiletargeting:List`, crie uma política que conceda permissões às seguintes ações: `mobiletargeting:Create`, `mobiletargeting:Update` e `mobiletargeting:Put`. Estas são as permissões adicionais necessárias para criar e gerenciar a maioria dos componentes do projeto. Por exemplo: .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "LimitedWriteProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*",
                "mobiletargeting:Create*",
                "mobiletargeting:Update*",
                "mobiletargeting:Put*"
            ],
            "Resource": [

 "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example",

 "arn:aws:mobiletargeting:region:accountId:apps/810c7aab86d42fb2b56c8c966example/*",
                "arn:aws:mobiletargeting:region:accountId:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
```

```
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

Exemplo: visualizar recursos do Amazon Pinpoint com base em tags

Você pode usar condições em uma política baseada em identidade para controlar o acesso aos recursos do Amazon Pinpoint com base em tags. Esta política de exemplo mostra como você pode criar esse tipo de política para permitir a visualização dos recursos do Amazon Pinpoint. No entanto, a permissão será concedida somente se a tag `Owner` tiver o valor do nome desse usuário. Essa política também concede as permissões necessárias concluir essa ação no console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewResourceIfOwner",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:*:*:*",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                },
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mobiletargeting:region:accountId:*"
                }
            }
        }
    ]
}
```

Você pode anexar esse tipo de política aos usuários do na sua conta. Se um usuário chamado `richard-roe` tentar visualizar um recurso do Amazon Pinpoint, o recurso deve ser marcado como `0wner=richard-roe` ou `owner=richard-roe`. Caso contrário, ele terá o acesso negado. A chave da tag de condição `0wner` corresponde a `0wner` e a `owner` porque os nomes das chaves de condição não fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulte Elementos de política JSON do IAM: condição no Guia do usuário do IAM.

Exemplo: permitir que os usuários visualizem suas próprias permissões

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

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

Exemplos: fornecer acesso às ações da API do Amazon Pinpoint

Esta seção fornece políticas de exemplo que permitem o acesso a recursos que estão disponíveis na API do Amazon Pinpoint, que é a API principal do Amazon Pinpoint. Para saber mais sobre essa API, consulte Referência da API do Amazon Pinpoint.

Acesso somente leitura

O exemplo de política a seguir permite acesso somente de leitura a todos os recursos em sua conta do Amazon Pinpoint em uma região específica. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewAllResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        }
    ]
```

```
}
```

No exemplo anterior, substitua *region* pelo nome de uma AWS região e substitua *accountId pelo ID da* sua conta. AWS

Acesso de administrador

A política de exemplo a seguir permite acesso completo a todas as ações e recursos do Amazon Pinpoint na conta do Amazon Pinpoint:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:*"
            ],
            "Resource": "arn:aws:mobiletargeting:region:accountId:*"
        }
    ]
}
```

No exemplo anterior, substitua *accountId* pelo ID da conta da AWS .

Exemplos: fornecer acesso às ações da API de SMS e voz no Amazon Pinpoint

Esta seção fornece políticas de exemplo que permitem acesso aos atributos que estão disponíveis na API de SMS e voz do Amazon Pinpoint. Esta é uma API complementar que oferece opções avançadas para uso e gerenciamento de canais de SMS e voz no Amazon Pinpoint. Para saber mais sobre essa API, consulte Referência da API de SMS e voz do Amazon Pinpoint.

Acesso somente leitura

O exemplo de política a seguir permite acesso somente de leitura a todas as ações e recursos da API Amazon Pinpoint SMS e Voice em sua conta: AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "SMSVoiceReadOnly",
            "Effect": "Allow",
            "Action": [
                "sms-voice:Get*",
                "sms-voice:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:region:accountId:*"
                }
            }
        }
    ]
}
```

Acesso de administrador

O exemplo de política a seguir permite acesso total a todas as ações e recursos da API de voz e SMS do Amazon Pinpoint em sua AWS conta:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SMSVoiceFullAccess",
            "Effect": "Allow",
            "Action": [
                "sms-voice:*",
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:region:accountId:*"
                }
            }
```

```
        }
    ]
}
```

Exemplo: restringir o acesso ao projeto do Amazon Pinpoint a endereços IP específicos

A política de exemplo a seguir concede permissões a todos os usuários para realizar qualquer ação do Amazon Pinpoint em um projeto especificado (*projectId*). No entanto, a solicitação deve se originar no intervalo de endereços IP que são especificados na condição.

A condição nesta instrução identifica o intervalo `54.240.143.*` de endereços do Internet Protocol versão 4 (IPv4), com uma exceção: `54.240.143.188`. O `Condition` bloco usa as `NotIpAddress` condições `IpAddress` e e a chave de `aws:SourceIp` condição, que é uma chave AWS de condição ampla. Para obter mais informações sobre essas chaves de condições, consulte Especificar condições em uma política no Guia do usuário do IAM. Os valores IPv4 `aws:SourceIp` usam a notação CIDR padrão. Para obter mais informações, consulte Operadores de condição de endereço IP no Guia do usuário do IAM.

```
{
    "Version":"2012-10-17",
    "Id":"AMZPinpointPolicyId1",
    "Statement":[
        {
            "Sid":"IPAllow",
            "Effect":"Allow",
            "Principal":"*",
            "Action":"mobiletargeting:*",
            "Resource":[
                "arn:aws:mobiletargeting:region:accountId:apps/projectId",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
            ],
            "Condition":{
                "IpAddress":{
                    "aws:SourceIp":"54.240.143.0/24"
                },
                "NotIpAddress":{
                    "aws:SourceIp":"54.240.143.188/32"
                }
            }
        }
```

```
    ]
}
```

Exemplo: restringir o acesso ao Amazon Pinpoint com base em tags

A política de exemplo a seguir concede permissões para realizar qualquer ação do Amazon Pinpoint em um projeto especificado (*projectId*). No entanto, as permissões são concedidas somente se a solicitação derivar de um usuário cujo nome seja um valor na tag de recurso `Owner` do projeto, conforme especificado na condição.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ModifyResourceIfOwner",
            "Effect": "Allow",
            "Action": "mobiletargeting:*",
            "Resource": [
                "arn:aws:mobiletargeting:region:accountId:apps/projectId",
                "arn:aws:mobiletargeting:region:accountId:apps/projectId/*"
                ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                }
            }
        }
    ]
}
```

Exemplo: permitir que o Amazon Pinpoint envie e-mails usando identidades que foram verificadas no Amazon SES

Quando você verifica uma identidade de e-mail (como um endereço de e-mail ou domínio) por meio do console do Amazon Pinpoint, essa identidade é configurada automaticamente para que possa ser usada tanto pelo Amazon Pinpoint quanto pelo Amazon SES. No entanto, se você verificar uma identidade de e-mail por meio do Amazon SES e quiser usar essa identidade com o Amazon Pinpoint, deverá aplicar uma política a essa identidade.

O exemplo de política a seguir concede ao Amazon Pinpoint permissão para enviar e-mails usando uma identidade de e-mail que foi verificada pelo Amazon SES.

```
{
    "Version":"2008-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws:ses:region:accountId:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"accountId"
                },
                "StringLike":{
                    "aws:SourceArn":"arn:aws:mobiletargeting:region:accountId:apps/*"
                }
            }
        }
    ]
}
```

Se você usa o Amazon Pinpoint na região AWS GovCloud (Oeste dos EUA), use o seguinte exemplo de política em vez disso:

```
{
    "Version":"2008-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws-us-gov:ses:us-gov-west-1:accountId:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"accountId"
                },
                "StringLike":{
```

```
                    "aws:SourceArn":"arn:aws-us-gov:mobiletargeting:us-gov-
west-1:accountId:apps/*"
                }
            }
        }
    ]
}
```

Perfis do IAM para tarefas comuns do Amazon Pinpoint

Uma função do IAM é uma identidade AWS Identity and Access Management (IAM) que você pode criar em sua AWS conta e conceder permissões específicas. Uma função do IAM é uma AWS identidade com políticas de permissão que determinam o que a identidade pode ou não fazer AWS. No entanto, em vez de ser exclusivamente associado a uma pessoa, o perfil pode ser assumido por qualquer pessoa que precisar dele.

Além disso, um perfil não tem credenciais de longo prazo padrão associadas a ele. Em vez disso, ela fornece credenciais de segurança temporárias para uma sessão. Você pode usar as funções do IAM para delegar acesso a usuários, aplicativos, aplicativos ou serviços que normalmente não têm acesso aos seus AWS recursos.

Por esses motivos, é possível usar perfis do IAM para integrar o Amazon Pinpoint a determinados serviços e recursos da AWS para sua conta. Por exemplo, pode ser conveniente permitir que o Amazon Pinpoint tenha acesso às definições de endpoint que você armazena em um bucket do Amazon Simple Storage Service (Amazon S3) e deseja usar em segmentos. Ou você pode permitir que o Amazon Pinpoint transmita dados de eventos a um fluxo do Amazon Kinesis para a sua conta. Da mesma forma, talvez você queira usar as funções do IAM para permitir que aplicativos web ou móveis registrem endpoints ou relatem dados de uso para projetos do Amazon Pinpoint, sem AWS incorporar chaves nos aplicativos (onde pode ser difícil alterná-las e os usuários podem potencialmente extraí-las).

Para esses cenários, é possível delegar acesso ao Amazon Pinpoint usando perfis do IAM. Esta seção explica e fornece exemplos de tarefas comuns do Amazon Pinpoint que usam perfis do IAM para operar com outros serviços da AWS . Para obter informações sobre como usar perfis do IAM com aplicativos web ou móveis mais especificamente, consulte Permitir acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.

Tópicos

- Perfil do IAM para importação de endpoints ou segmentos

- Perfil do IAM para exportação de endpoints ou de segmentos
- Perfil do IAM para recuperar recomendações do Amazon Personalize
- Perfil do IAM para transmissão de eventos para o Kinesis
- Função do IAM para enviar e-mails com o Amazon SES

Perfil do IAM para importação de endpoints ou segmentos

Com o Amazon Pinpoint, você pode definir um segmento de usuário importando definições de endpoint de um bucket do Amazon Simple Storage Service (Amazon S3) em sua conta. AWS Antes de importar, você deve delegar as permissões necessárias para o Amazon Pinpoint. Para fazer isso, você cria uma função AWS Identity and Access Management (IAM) e anexa as seguintes políticas à função:

- A política gerenciada `AmazonS3ReadOnlyAccess` AWS . Essa política é criada e gerenciada por AWS, e concede acesso somente de leitura ao seu bucket do Amazon S3.
- Uma política de confiança que permita que Amazon Pinpoint assuma o perfil.

Depois de criar o perfil, você pode usar o Amazon Pinpoint para importar segmentos de um bucket do Amazon S3. Para obter informações sobre como criar o bucket, criar arquivos de endpoint e importar um segmento usando o console, consulte Importação de segmentos no Guia do usuário do Amazon Pinpoint. Para obter um exemplo de como importar um segmento programaticamente usando o AWS SDK for Java, consulte este Importar segmentos guia.

Criação do perfil do IAM (AWS CLI)

Conclua as etapas a seguir para criar a função do IAM usando o AWS Command Line Interface (AWS CLI). Se você não instalou o AWS CLI, consulte Instalando o AWS CLI no Guia AWS Command Line Interface do Usuário.

Para criar a função do IAM usando o AWS CLI

1. Crie um arquivo JSON que contenha a política de confiança para a sua função, e salve-o localmente. Você pode usar a seguinte política de confiança.

```
{
     "Version": "2012-10-17",
     "Statement": [
```

```
          {
               "Action": "sts:AssumeRole",
               "Effect": "Allow",
               "Principal": {
                    "Service": "pinpoint.amazonaws.com"
               },
               "Condition": {
                    "StringEquals": {
                         "aws:SourceAccount": "accountId"
                    },
               "ArnLike": {
                    "arn:aws:mobiletargeting:region:accountId:apps/application-id"
                    }
               }
          }
     ]
}
```

No exemplo anterior, faça o seguinte:

- Substitua a *AWS região* pela região na qual você usa o Amazon Pinpoint.
- Substitua *accountId* pelo ID exclusivo da sua conta. AWS
- Substitua *application-id* pelo ID exclusivo do projeto.

2. Na linha de comando, use o comando `create-role` para criar a função e anexar a política de confiança:

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
 file://PinpointImportTrustPolicy.json
```

Seguindo o prefixo `file://`, especifique o caminho de acesso ao arquivo JSON que contém a política de confiança.

Depois de executar este comando, você verá uma saída semelhante à seguinte no terminal:

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
```

```
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "pinpoint.amazonaws.com"
                    },
                    "Condition": {
                        "StringEquals": {
                            "aws:SourceAccount": "accountId"
                         },
                         "ArnLike": {
                            "aws:SourceArn":
 "arn:aws:mobiletargeting:region:accountId:apps/application-id"
                         }
                    }
                }
            ]
        },
        "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
        "CreateDate": "2016-12-20T00:44:37.406Z",
        "RoleName": "PinpointSegmentImport",
        "Path": "/",
        "Arn": "arn:aws:iam::accountId:role/PinpointSegmentImport"
    }
}
```

3. Use o attach-role-policycomando para anexar a política `AmazonS3ReadOnlyAccess` AWS gerenciada à função:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

Perfil do IAM para exportação de endpoints ou de segmentos

Você pode obter uma lista de endpoints criando um trabalho de exportação. Ao criar um trabalho de exportação, você precisa especificar um ID de projeto e, opcionalmente, especificar um ID de segmento. Em seguida, o Amazon Pinpoint exporta uma lista dos endpoints associados ao projeto ou segmento para um bucket do Amazon Simple Storage Service (Amazon S3). O arquivo resultante contém uma lista de endpoints formatados por JSON e seus atributos, como canal, endereço, status de inclusão/exclusão, data de criação e ID de endpoint.

Para criar um trabalho de exportação, você precisa configurar um perfil do IAM que permita que o Amazon Pinpoint grave em um bucket do Amazon S3. O processo de configuração da função consiste em duas etapas:

1. Crie uma política do IAM que permita que uma entidade (neste caso, o Amazon Pinpoint) grave em um bucket específico do Amazon S3.
2. Crie um perfil do IAM e associe a política a ela.

Este tópico contém procedimentos para concluir as duas etapas. Esses procedimentos pressupõem que você já criou um bucket do Amazon S3 e uma pasta nesse bucket para armazenar segmentos exportados. Para obter mais informações sobre como criar buckets, consulte Criar um bucket, no Guia do usuário do Amazon Simple Storage Service.

Esses procedimentos também pressupõem que você já instalou e configurou a AWS Command Line Interface (AWS CLI). Para obter informações sobre como configurar o AWS CLI, consulte Instalando o AWS CLI no Guia AWS Command Line Interface do Usuário.

Etapa 1: criar a política do IAM

Uma política do IAM define as permissões de uma entidade, como uma identidade ou recurso. Para criar um perfil para exportar os endpoints do Amazon Pinpoint, você precisa criar uma política que conceda permissão para gravar em uma pasta específica em um bucket específico do Amazon S3. O exemplo de política a seguir segue a prática de segurança de concessão de privilégio mínimo, ou seja, apenas as permissões necessárias para a execução de uma única tarefa são concedidas.

Para criar a política do IAM

1. Em um editor de texto, crie um novo arquivo. Cole o seguinte código no arquivo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::*" ]
```

```
        },
        {
            "Sid": "AllowRootAndHomeListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::example-bucket" ],
            "Condition": {
                "StringEquals": {
                    "s3:delimiter": [ "/" ],
                    "s3:prefix": [
                        "",
                        "Exports/"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::example-bucket" ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "Exports/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Action": [ "s3:*" ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::example-bucket/Exports/*" ]
        }
    ]
}
```

No código anterior, substitua todas as instâncias de *example-bucket* pelo nome do bucket do Amazon S3 que contém a pasta para onde você deseja exportar as informações do segmento. Além disso, substitua todas as ocorrências de *Exports (Exportações)* pelo nome da pasta em si.

Ao concluir, salve o arquivo como `s3policy.json`.

2. Usando o AWS CLI, navegue até o diretório em que o `s3policy.json` arquivo está localizado. Insira o seguinte comando para criar a política:

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
 file://s3policy.json
```

Se a criação da política for bem-sucedida, você verá um resultado semelhante a este:

```
{
    "Policy": {
        "CreateDate": "2018-04-11T18:44:34.805Z",
        "IsAttachable": true,
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PolicyId": "ANPAJ2YJQRJCG3EXAMPLE",
        "UpdateDate": "2018-04-11T18:44:34.805Z",
        "Arn": "arn:aws:iam::123456789012:policy/s3ExportPolicy",
        "PolicyName": "s3ExportPolicy",
        "Path": "/"
    }
}
```

Copie o nome do recurso da Amazon (ARN) da política (`arn:aws:iam::123456789012:policy/s3ExportPolicy` no exemplo anterior). Na próxima seção, você deverá fornecer esse ARN quando criar a função.

> **Note**
>
> Se você receber uma mensagem informando que sua conta não está autorizada a executar a operação `CreatePolicy`, precisará anexar uma política à sua conta de usuário que permita a criação de novas políticas e perfis do IAM. Para obter mais

informações, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

Etapa 2: criar o perfil do IAM

Agora que você criou uma política do IAM, pode criar um perfil e anexar a política a ele. Cada perfil do IAM contém uma política de confiança, um conjunto de regras que especifica quais entidades têm permissão para assumir o perfil. Nesta seção, você cria uma política de confiança que permite ao Amazon Pinpoint assumir o perfil. Em seguida, você cria a função propriamente dita e depois anexa a ela a política que você criou na seção anterior.

Como criar o perfil do IAM

1. Em um editor de texto, crie um novo arquivo. Cole o seguinte código no arquivo:

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                    "aws:SourceArn":
 "arn:aws:mobiletargeting:region:accountId:apps/applicationId"
                }
            }
        }
    ]
}
```

   Salve o arquivo como `trustpolicy.json`.

2. Usando o AWS CLI, navegue até o diretório em que o `trustpolicy.json` arquivo está localizado. Insira o seguinte comando para criar uma nova função:

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
 file://trustpolicy.json
```

3. Na linha de comando, insira o seguinte comando para anexar a política que você criou na seção anterior à função que você acabou de criar:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

No comando anterior, substitua *arn:aws:iam: :123456789012:policy/s3 pelo ARN da política* que você criou na seção anterior. ExportPolicy

Perfil do IAM para recuperar recomendações do Amazon Personalize

Você pode configurar o Amazon Pinpoint para recuperar dados de recomendação de uma solução do Amazon Personalize que foi implantada como uma campanha do Amazon Personalize. Você pode usar esses dados para enviar recomendações personalizadas aos destinatários da mensagem com base nos atributos e no comportamento de cada destinatário. Para saber mais, consulte Modelos de machine learning no Guia do usuário do Amazon Pinpoint.

Para poder recuperar dados de recomendação de uma campanha do Amazon Personalize, é necessário criar um perfil do AWS Identity and Access Management (IAM) que permita que o Amazon Pinpoint recupere os dados da campanha. O Amazon Pinpoint pode criar esse perfil para você automaticamente quando você usa o console para configurar um modelo de recomendação no Amazon Pinpoint. Ou, é possível criar essa função manualmente.

Para criar o perfil manualmente, use a API do IAM para concluir as seguintes etapas:

1. Crie uma política do IAM que permita que uma entidade (neste caso, o Amazon Pinpoint) recupere dados de recomendação de uma campanha do Amazon Personalize.
2. Crie um perfil do IAM e anexe a política a ela.

Este tópico explica como concluir essas etapas usando o AWS Command Line Interface (AWS CLI). Ele pressupõe que você já tenha criado a solução do Amazon Personalize e tenha a implantado como uma campanha do Amazon Personalize. Para obter informações sobre como criar e implantar uma campanha, consulte Criação de uma campanha no Guia do desenvolvedor do Amazon Personalize.

Esse tópico também pressupõe que você já instalou e configurou a AWS CLI. Para obter informações sobre como configurar o AWS CLI, consulte Instalando o AWS CLI no Guia AWS Command Line Interface do Usuário.

Etapa 1: criar a política do IAM

Uma política do IAM define as permissões de uma entidade, como uma identidade ou recurso. Para criar um perfil que permita que o Amazon Pinpoint recupere dados de recomendação de uma campanha do Amazon Personalize, primeiro crie uma política do IAM para o perfil. Essa política precisa permitir que o Amazon Pinpoint:

- Recupere informações de configuração para a solução que é implantada pela campanha (`DescribeSolution`).
- Verifique o status da campanha (`DescribeCampaign`).
- Recupere dados de recomendação da campanha (`GetRecommendations`).

No procedimento a seguir, a política de exemplo permite esse acesso para uma solução específica do Amazon Personalize que foi implantada por uma campanha específica do Amazon Personalize.

Para criar a política do IAM

1. Em um editor de texto, crie um novo arquivo. Cole o seguinte código no arquivo:

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"RetrieveRecommendationsOneCampaign",
            "Effect":"Allow",
            "Action":[
                "personalize:DescribeSolution",
                "personalize:DescribeCampaign",
                "personalize:GetRecommendations"
            ],
            "Resource":[
                "arn:aws:personalize:region:accountId:solution/solutionId",
                "arn:aws:personalize:region:accountId:campaign/campaignId"
            ]
        }
    ]
```

```
}
```

No exemplo anterior, substitua o texto em *itálico* por suas informações:

- *região*: o nome da Região da AWS que hospeda a solução e a campanha do Amazon Personalize.
- *accountId*: seu ID da Conta da AWS .
- *solutionId*: o ID de recurso exclusivo da solução do Amazon Personalize implantada pela campanha.
- *campaignId*: o ID de recurso exclusivo da campanha do Amazon Personalize de onde recuperar dados de recomendação.

2. Ao concluir, salve o arquivo como `RetrieveRecommendationsPolicy.json`.
3. Usando a interface de linha de comando, navegue até o diretório onde salvou o arquivo `RetrieveRecommendationsPolicy.json`.
4. Digite o comando a seguir para criar uma política e nomeie-a como `RetrieveRecommendationsPolicy`. Para usar um nome diferente, altere *RetrieveRecommendationsPolicy*para o nome desejado.

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
 file://RetrieveRecommendationsPolicy.json
```

> **Note**
>
> Se você receber uma mensagem informando que sua conta não está autorizada a executar a operação `CreatePolicy`, será necessário anexar uma política ao usuário que permita a criação de novas políticas e perfis do IAM. Para obter mais informações, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

5. Copie o nome do recurso da Amazon (ARN) da política (`arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy` no exemplo anterior). Você precisará desse ARN para criar o perfil do IAM na próxima seção.

Etapa 2: criar o perfil do IAM

Depois de criar a política do IAM, você pode criar um perfil do IAM e anexar a política a ele.

Cada perfil do IAM contém uma política de confiança, que é um conjunto de regras que especifica quais entidades têm permissão para assumir o perfil. Nesta seção, você cria uma política de confiança que permite ao Amazon Pinpoint assumir o perfil. E também cria a própria função. Depois, você anexa a política à função.

Como criar o perfil do IAM

1. Em um editor de texto, crie um novo arquivo. Cole o seguinte código no arquivo:

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "accountId"
                },
                "ArnLike": {
                    "AWS:SourceArn":
 "arn:aws:mobiletargeting:region:accountId:apps/*"
                }
            }
        }
    ]
}
```

2. Salve o arquivo como `RecommendationsTrustPolicy.json`.
3. Usando a interface de linha de comando, navegue até o diretório onde salvou o arquivo `RecommendationsTrustPolicy.json`.
4. Insira o seguinte comando para criar uma função e nomeie-a como `PinpointRoleforPersonalize`: Para usar um nome diferente, altere *`PinpointRoleforPersonalize`*para o nome desejado.

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-
document file://RecommendationsTrustPolicy.json
```

5. Na linha de comando, insira o seguinte comando para anexar a política criada na seção anterior à função que você acabou de criar:

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

No comando anterior, substitua *arn:aws:iam: :123456789012:policy/ pelo ARN da política* que você criou na seção anterior. RetrieveRecommendationsPolicy Além disso, *PinpointRoleforPersonalize*substitua pelo nome da função que você especificou na etapa 4, se tiver especificado um nome diferente para a função.

Perfil do IAM para transmissão de eventos para o Kinesis

O Amazon Pinpoint pode enviar automaticamente dados de uso do aplicativo, ou dados de eventos, do seu aplicativo para um stream de dados do Amazon Kinesis ou stream de entrega do Amazon Data Firehose em sua conta. AWS Antes que o Amazon Pinpoint possa começar a transmitir os dados de eventos, você terá de delegar as permissões necessárias ao Amazon Pinpoint.

Se você usar o console para configurar a transmissão de eventos, o Amazon Pinpoint criará um perfil do AWS Identity and Access Management (IAM) automaticamente com as permissões necessárias. Para mais informações, consulte Transmissão de eventos do Amazon Pinpoint para o Amazon Kinesis no Guia do usuário do Amazon Pinpoint.

Se quiser criar a função manualmente, anexe as seguintes políticas a ela:

- Uma política de permissões que possibilita que o Amazon Pinpoint envie dados de eventos para o seu fluxo.
- Uma política de confiança que permita que Amazon Pinpoint assuma o perfil.

Após criar o perfil, você poderá configurar o Amazon Pinpoint para enviar eventos ao fluxo automaticamente. Para obter mais informações, consulte Transmissão de eventos do Amazon Pinpoint para o Kinesis neste guia.

Criação do perfil do IAM (AWS CLI)

Complete as seguintes etapas para criar manualmente um perfil do IAM usando a AWS Command Line Interface (AWS CLI). Para saber como criar o perfil usando o console do Amazon Pinpoint,

consulte Transmissão de eventos do Amazon Pinpoint para o Kinesis no Guia do usuário do Amazon Pinpoint.

Se você não instalou o AWS CLI, consulte Instalando o AWS CLI no Guia AWS Command Line Interface do Usuário. Você também precisa ter criado um stream do Kinesis ou do Firehose. Para obter informações sobre a criação desses recursos, consulte Criação e gerenciamento de streams no Guia do desenvolvedor do Amazon Kinesis Data Streams ou Criação de um stream de entrega do Amazon Data Firehose no Guia do desenvolvedor do Amazon Data Firehose.

Para criar a função do IAM usando o AWS CLI

1. Criar um novo arquivo. Cole a política a seguir no documento e faça as seguintes alterações:

   - Substitua a *AWS região* pela região na qual você usa o Amazon Pinpoint.
   - Substitua *accountId* pelo ID exclusivo da sua conta. AWS
   - *Substitua ApplicationID pelo ID exclusivo do projeto.*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                    "aws:SourceArn":
 "arn:aws:mobiletargeting:region:accountId:apps/applicationId"
                }
            }
        }
    ]
}
```

   Ao concluir, salve o arquivo como `PinpointEventStreamTrustPolicy.json`.

2. Use o comando `create-role` para criar a função e anexar a política de confiança:

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

3. Crie um arquivo que contenha a política de permissões para o perfil.

   Se você estiver configurando o Amazon Pinpoint para enviar dados para um stream do Kinesis, cole a seguinte política no arquivo e substitua a seguinte:

   - Substitua a *AWS região* pela região na qual você usa o Amazon Pinpoint.
   - Substitua *accountId* pelo ID exclusivo da sua conta. AWS
   - Substitua *streamName* pelo nome do seu stream do Kinesis.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Action": [
            "kinesis:PutRecords",
            "kinesis:DescribeStream"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:kinesis:region:accountId:stream/streamName"
        ]
    }
}
```

   Como alternativa, se você estiver configurando o Amazon Pinpoint para enviar dados para um stream do Firehose, cole a seguinte política no arquivo e substitua a seguinte:

   - Substitua a *AWS região* pela região na qual você usa o Amazon Pinpoint.
   - Substitua *accountId* pelo ID exclusivo da sua conta. AWS
   - *delivery-stream-name*Substitua pelo nome do seu stream do Firehose.

```
{
    "Version": "2012-10-17",
    "Statement": {
```

```
        "Effect": "Allow",
        "Action": [
         "firehose:PutRecordBatch",
         "firehose:DescribeDeliveryStream"
        ],
        "Resource": [
         "arn:aws:firehose:region:accountId:deliverystream/delivery-stream-name"
     ]
    }
}
```

Ao concluir, salve o arquivo como `PinpointEventStreamPermissionsPolicy.json`.

4. Use o comando `put-role-policy` para anexar a política de permissões à função:

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json
```

Função do IAM para enviar e-mails com o Amazon SES

O Amazon Pinpoint usa seus recursos do Amazon SES para enviar e-mails para sua campanha ou jornada. Antes que o Amazon Pinpoint possa usar seus recursos do Amazon SES para enviar e-mails, você deve conceder as permissões necessárias ao Amazon Pinpoint. Sua conta deve ter as `iam:UpdateAssumeRolePolicy` permissões `iam:PutRolePolicy` e as permissões para atualizar ou criar funções do IAM.

O console do Amazon Pinpoint pode criar automaticamente uma função AWS Identity and Access Management (IAM) com as permissões necessárias. Para obter mais informações, consulte Criação de uma função de envio de orquestração de e-mail no Guia do usuário do Amazon Pinpoint.

Se quiser criar a função manualmente, anexe as seguintes políticas a ela:

- Uma política de permissões que concede ao Amazon Pinpoint acesso aos seus recursos do Amazon SES.
- Uma política de confiança que permita que Amazon Pinpoint assuma o perfil.

Depois de criar a função, você pode configurar o Amazon Pinpoint para usar seus recursos do Amazon SES.

Você pode testar as políticas do IAM com o simulador de políticas do IAM Para obter mais informações, consulte Teste de políticas do IAM com o simulador de políticas do IAM no Guia do usuário do IAM.

Criação do perfil do IAM (AWS Management Console)

Conclua as etapas a seguir para criar manualmente uma função do IAM para sua campanha ou jornada para enviar e-mails.

1. Crie uma nova política de permissão seguindo as instruções em Como criar políticas usando o editor JSON no Guia do usuário do IAM.

   - Na etapa 5, use a seguinte política de permissão para a função do IAM.

     - Substitua a *partição* pela partição em que o recurso está. Por padrão Regiões da AWS, a partição é `aws`. Se você tem recursos em outras partições, a partição é `aws-partitionname`. Por exemplo, a partição para recursos no AWS GovCloud (Oeste dos EUA) é `aws-us-gov`.
     - Substitua a *região* pelo nome da Região da AWS que hospeda o projeto Amazon Pinpoint.
     - Substitua *accountId* pelo ID exclusivo do seu. Conta da AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PinpointUsesSESForEmailSends",
            "Effect": "Allow",
            "Action": [
                "ses:SendEmail",
                "ses:SendRawEmail"
            ],
            "Resource": [
                "arn:partition:ses:region:accountId:identity/*",
                "arn:partition:ses:region:accountId:configuration-set/*"
            ]
        }
    ]
}
```

2. Crie uma nova política de confiança seguindo as instruções em Como criar uma função usando políticas de confiança personalizadas no Guia do usuário do IAM.

   a. Na etapa 4, use a seguinte política de confiança.

      - Substitua *accountId* pelo ID exclusivo do seu. Conta da AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPinpoint",
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

   b. Na etapa 11, adicione a política de permissão que você criou na etapa anterior.

Solução de problemas de gerenciamento de identidade e acesso do Amazon Pinpoint

Use as informações a seguir para diagnosticar e corrigir problemas comuns que você possa encontrar ao trabalhar com o Amazon Pinpoint e o IAM.

Tópicos

- Não estou autorizado a executar uma ação no Amazon Pinpoint
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha AWS conta acessem meus recursos do Amazon Pinpoint

Não estou autorizado a executar uma ação no Amazon Pinpoint

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. Seu administrador é a pessoa que forneceu a você suas credenciais de login.

O erro de exemplo a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um projeto, mas não tem permissões `mobiletargeting:GetApp`.

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

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso `my-example-project` usando a ação `mobiletargeting:GetApp`.

Não estou autorizado a realizar iam: PassRole

Caso receba uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, as políticas deverão ser atualizadas para permitir a transmissão de um perfil ao Amazon Pinpoint.

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 erro exemplificado a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Amazon Pinpoint. 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 AWS conta acessem meus recursos do Amazon Pinpoint

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização possam 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 políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon Pinpoint é compatível com esses recursos, consulte Como o Amazon Pinpoint funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Saiba como conceder acesso por meio da federação de identidades consultando Concedendo Acesso a Usuários Autenticados Externamente (Federação de Identidades) no Guia do Usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Registrar em log e monitorar no Amazon Pinpoint

O registro e o monitoramento são importantes para manter a confiabilidade, a disponibilidade e o desempenho dos projetos do Amazon Pinpoint e outros tipos de recursos do Amazon Pinpoint. Você deve registrar e coletar dados de monitoramento de todas as partes dos seus projetos e recursos do Amazon Pinpoint para depurar com mais facilidade uma falha multiponto, caso ocorra. AWS fornece várias ferramentas que podem ajudá-lo a registrar e coletar esses dados e responder a possíveis incidentes:

AWS CloudTrail

O Amazon Pinpoint se integra com AWS CloudTrail, que é um serviço que fornece um registro das ações que foram realizadas no Amazon Pinpoint por um usuário, uma função ou outro

serviço. AWS Isso inclui ações do console do Amazon Pinpoint e chamadas programáticas às operações de API do Amazon Pinpoint. Usando as informações coletadas pelo CloudTrail, você pode determinar quais solicitações foram feitas ao Amazon Pinpoint. Para cada solicitação é possível identificar quando ela foi realizada, o endereço IP do qual foi feita, quem fez a solicitação e detalhes adicionais. Para obter mais informações, consulte Registro de chamadas de API do Amazon Pinpoint com AWS CloudTrail neste guia.

Amazon CloudWatch

Você pode usar CloudWatch a Amazon para coletar, visualizar e analisar várias métricas importantes relacionadas à sua conta e projetos do Amazon Pinpoint. Você também pode usar CloudWatch para criar alarmes que o notificam se o valor de uma métrica atende a determinadas condições e está dentro ou excede um limite definido por você. Se você criar um alarme, CloudWatch envia uma notificação para um tópico do Amazon Simple Notification Service (Amazon SNS) especificado por você. Para obter mais informações, consulte Monitoramento do Amazon Pinpoint com a Amazon CloudWatch no Guia do usuário do Amazon Pinpoint.

AWS Health Painéis

Ao usar AWS Health painéis, você pode verificar e monitorar o status do seu ambiente Amazon Pinpoint. Para verificar o status geral do serviço Amazon Pinpoint, use o AWS Service Health Dashboard. Para verificar, monitorar e visualizar dados históricos sobre quaisquer eventos ou problemas que possam afetar seu AWS ambiente mais especificamente, use o AWS Personal Health Dashboard. Para saber mais sobre esses painéis, consulte o Guia do usuário do AWS Health.

AWS Trusted Advisor

AWS Trusted Advisor inspeciona seu AWS ambiente e fornece recomendações de oportunidades para solucionar lacunas de segurança, melhorar a disponibilidade e o desempenho do sistema e economizar dinheiro. Todos os AWS clientes têm acesso a um conjunto básico de Trusted Advisor cheques. Os clientes que têm um plano de suporte Business ou Enterprise têm acesso a Trusted Advisor verificações adicionais.

Muitas dessas verificações podem ajudá-lo a avaliar a postura de segurança dos seus recursos do Amazon Pinpoint como parte da AWS sua conta geral. Por exemplo, o conjunto principal de verificações do Trusted Advisor inclui o seguinte:

- Configurações de registro para sua AWS conta, para cada AWS região suportada.
- Permissões de acesso aos buckets do Amazon Simple Storage Service (Amazon S3), que podem conter arquivos que você importa para o Amazon Pinpoint para criar segmentos.

- Uso de AWS Identity and Access Management usuários, grupos e funções para controlar o acesso aos recursos do Amazon Pinpoint.
- Configurações do IAM e definições de políticas que podem comprometer a segurança do seu AWS ambiente e dos recursos do Amazon Pinpoint.

Para obter mais informações, consulte AWS Trusted Advisor no Guia de Usuário AWS Support .

Validação de conformidade do Amazon Pinpoint

Os auditores externos avaliam a segurança e a conformidade do Amazon Pinpoint como parte de vários programas de conformidade da AWS . Isso inclui Controles de AWS Sistema e Organização (SOC), FedRAMP, HIPAA, ISO/IEC 27001:2013 para controles de gerenciamento de segurança, ISO/IEC 27017:2015 para controles específicos da nuvem, ISO/IEC 27018:2014 para proteção de dados pessoais, ISO/IEC 9001:2015 para sistemas de gerenciamento de qualidade e outros.

Para obter uma lista de AWS serviços que estão no escopo de programas de conformidade específicos, consulte AWS serviços no escopo por programa de conformidade AWS . Para obter informações gerais, consulte programas de AWS conformidade programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixando relatórios no AWS Artifact .

Sua responsabilidade de conformidade ao usar o Amazon Pinpoint é determinada pela confidencialidade dos 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 sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos com foco em segurança e conformidade em. AWS
- Documento técnico sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper descreve como as empresas podem usar para criar aplicativos compatíveis com a HIPAA. AWS
- AWS recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- Avaliação de recursos com as regras do Guia do AWS Config Desenvolvedor — O AWS Config serviço avalia se suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.

- AWS Security Hub— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

O Amazon Pinpoint é um serviço qualificado pela AWS HIPAA quando os clientes usam os canais de comunicação adequados. Se você deseja usar o Amazon Pinpoint para executar workloads que contenham informações de saúde protegidas (PHI – Protected Health Information), conforme definido pela HIPAA e pela legislação e os regulamentos associados, é necessário usar o canal de e-mail, o canal de notificações por push ou o canal de SMS para enviar mensagens que contenham PHI. Se você usa o canal SMS para enviar mensagens que contenham PHI, envie essas mensagens a partir de um código curto dedicado que você solicitou para sua AWS conta com a finalidade explícita de enviar mensagens que conterão ou poderão conter PHI. O canal de voz não está qualificado pela AWS HIPAA; não use o canal de voz para enviar mensagens que contenham PHI.

Resiliência no Amazon Pinpoint

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

Para obter mais informações, sobre arquiteturas de referência no Guia de arquitetura resiliente do Amazon Pinpoint.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte infraestrutura AWS global.

Segurança da infraestrutura no Amazon Pinpoint

Como um serviço gerenciado, o Amazon Pinpoint é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o Amazon Pinpoint pela rede. Os clientes devem ser compatíveis com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com Perfect Forward Secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, suporta esses modos.

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

Embora você possa fazer essas chamadas de API de qualquer lugar da rede, o Amazon Pinpoint oferece suporte às políticas de acesso baseadas em recursos. Essas políticas podem incluir restrições com base em endereço IP de origem. Para saber mais sobre esse tipo de política, consulte Gerenciando acesso usando políticas.

Além disso, você pode configurar e usar vários recursos de AWS segurança para controlar o acesso aos recursos do Amazon Pinpoint a partir de qualquer aplicativo móvel ou web que você integre ao Amazon Pinpoint. Isso inclui restrições às chamadas de API para tarefas como adicionar endpoints, atualizar dados de endpoint, enviar dados de eventos e relatar dados de uso.

Para usar esses recursos, recomendamos que você use os SDKs ou AWS Amplify JavaScript bibliotecas AWS móveis para integrar aplicativos móveis e web com o Amazon Pinpoint. Para aplicativos Android ou iOS, recomendamos que você use o AWS Mobile SDK for Android ou o AWS Mobile SDK for iOS, respectivamente. Para aplicativos móveis ou web JavaScript baseados, recomendamos que você use a AWS Amplify JavaScript Biblioteca para a Web ou a AWS Amplify JavaScript Biblioteca para React Native. Para saber mais sobre esses recursos, consulte Introdução aos SDKs AWS móveis, Introdução à biblioteca AWS Amplify para a web e Introdução à biblioteca AWS Amplify para react native.

Análise de configuração e vulnerabilidade no Amazon Pinpoint

Como um serviço gerenciado, o Amazon Pinpoint é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança. Isso significa que AWS gerencia e executa tarefas e procedimentos básicos de segurança para fortalecer, corrigir, atualizar e, de outra forma, manter a infraestrutura subjacente da

sua conta e dos recursos do Amazon Pinpoint. Esses procedimentos foram revisados e certificados por terceiros certificados.

Para obter mais informações, consulte os seguintes recursos do :

- Validação de conformidade do Amazon Pinpoint
- Modelo de responsabilidade compartilhada
- Amazon Web Services: visão geral dos processos de segurança (whitepaper)

Práticas recomendadas de segurança para o Amazon Pinpoint

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

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

Para obter mais informações sobre Amazon Pinpoint segurança, consulte Segurança no Amazon Pinpoint. Para obter mais informações sobre o IAM, consulte AWS Identity and Access Management. Para obter informações sobre as práticas recomendadas do IAM, acesse Melhores práticas do IAM.

Cotas do Amazon Pinpoint

As seções a seguir listam e descrevem as cotas, anteriormente chamadas de limites, que se aplicam aos recursos e operações do Amazon Pinpoint. Algumas cotas podem ser aumentadas, enquanto outras não. Para determinar se é possível solicitar o aumento de uma cota, consulte a coluna ou a instrução Qualificada para aumento em cada seção.

Tópicos

- Cotas do projeto
- Cotas de solicitação da API
- Cotas de campanhas
- Cotas de e-mail
- Cotas de endpoint
- Cotas de importação de endpoints
- Cotas de ingestão de eventos
- Cotas de jornadas
- Cotas Lambda
- Cotas de machine learning
- Cotas de modelos de mensagem
- Cotas de notificações por push
- Cotas de mensagens no aplicativo
- Cotas de segmentos
- Cotas de SMS
- Cotas 10DLC
- Cotas de voz
- Solicitar um aumento de cota

Cotas do projeto

A tabela a seguir lista as cotas relacionadas aos projetos no Amazon Pinpoint.

Recurso	Cota padrão	Qualificada para aumento
Projetos	Em cada um Região da AWS, você pode ter até 100 projetos.	Não

Cotas de solicitação da API

O Amazon Pinpoint implementa cotas que restringem o tamanho e o número de solicitações que você pode fazer à API do Amazon Pinpoint a partir da sua conta. AWS

O tamanho máximo de uma carga de invocação (solicitação e resposta) é de 7 MB, a menos que especificado de outra forma para um determinado tipo de recurso. Para determinar se um recurso tem uma cota diferente, consulte a seção apropriada deste tópico para esse tipo de recurso.

O número máximo de solicitações varia de acordo com o tipo de cota e a operação da API. O Amazon Pinpoint implementa dois tipos de cotas para solicitações de API:

- Cotas de taxa: também conhecidas como limites de taxa, esse tipo de cota define o número máximo de solicitações que você pode fazer por segundo para uma determinada operação. Ela controla a taxa de solicitações que são enviadas ou recebidas.
- Cotas de expansão: também conhecido como limites de expansão ou capacidade de expansão, esse tipo de cota define o número máximo de solicitações que estão em andamento simultaneamente para uma conta.

A tabela a seguir lista as cotas de taxa e de expansão da API do Amazon Pinpoint.

Operation	Cota de expansão/taxa padrão (solicitações por segundo)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300

Operation	Cota de expansão/taxa padrão (solicitações por segundo)
CreatePushTemplate	10
CreateSegment	25
CreateSmsTemplate	10
CreateVoiceTemplate	10
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4.000
SendUsersMessages	6.000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25
UpdateSmsTemplate	10

Operation	Cota de expansão/taxa padrão (solicitações por segundo)
UpdateVoiceTemplate	10
Todas as outras operações	300

A tabela a seguir relaciona as cotas de importação de arquivo para `CreateImportJob`.

Operation	Cota padrão	Qualificada para aumento
Número máximo de arquivos de importação	10.000 arquivos por tarefa de importação	Não

Se você exceder uma dessas cotas, o Amazon Pinpoint controla a solicitação, ou seja, rejeita uma solicitação válida e retorna um erro `TooManyRequests`. O controle de utilização é baseado no número total de solicitações que você faz da sua conta para uma operação específica em uma Região da AWS específica. Além disso, as decisões de controle de utilização são calculadas de forma independente para cada operação. Por exemplo, se o Amazon Pinpoint controlar uma solicitação para a operação `SendMessages`, uma solicitação simultânea para a operação `UpdateEndpoint` poderá ser concluída com êxito.

Cotas de campanhas

As cotas a seguir se aplicam ao recurso Campanhas da API do Amazon Pinpoint.

As seguintes cotas se aplicam Região da AWS e algumas podem ser aumentadas. Para obter mais informações, consulte Solicitar um aumento de cota no Guia do usuário do Service Quotas..

Recurso	Cota padrão	Qualificada para aumento
Campanhas ativas	200 por conta **Note** Uma campanha ativa é uma campanha	Não

Recurso	Cota padrão	Qualificada para aumento
	que ainda não foi concluída ou apresentou falha. Campanhas ativas têm status `SCHEDULED` , `EXECUTING` ou `PENDING_NEXT_RUN` .	
Tamanho máximo do segmento	Para segmentos importados: 100.000.000 por campanha Para segmentos dinâmicos: ilimitado	Não

Recurso	Cota padrão	Qualificada para aumento
Campanhas baseadas em eventos	Cada projeto pode incluir até 25 campanhas que são enviadas quando ocorrem eventos. As campanhas que usam gatilhos baseados em eventos precisam usar segmentos dinâmicos. Eles não podem usar segmentos importados. Se você integrar seu aplicativo ao Amazon Pinpoint usando um SDK AWS móvel, as mensagens de campanhas baseadas em eventos serão enviadas somente para clientes cujos aplicativos estão executando a AWS Mobile SDK for Android versão 2.7.2 ou posterior, ou a versão 2.6.30 ou posterior. AWS Mobile SDK for iOS Se o Amazon Pinpoint não puder entregar uma mensagem de uma campanha baseada em eventos em cinco minutos, ele interromperá a mensagem e não tentará enviá-la novamente.	Não

Cotas de e-mail

As cotas das seções a seguir se aplicam ao canal de e-mail.

Cotas de mensagens de e-mail

Recurso	Cota padrão	Qualificada para aumento
Tamanho máximo da mensagem, incluindo anexos	10 MB por mensagem	Não
Número de identidades verificadas	10.000 identidades **Note** As identidades referem-se a endereços de e-mail ou domínios ou a qualquer combinação de ambos. Todos os e-mails enviados usando o Amazon Pinpoint devem ser enviados de uma identidade verificada.	Não

Cotas de remetentes e destinatários de e-mail

Recurso	Cota padrão	Qualificada para aumento
Endereço do remetente	Todos os endereços ou domínios de envio devem ser verificados.	Não
Endereço do destinatário	Se sua conta ainda estiver na sandbox, todos os endereços ou domínios de e-mail de destinatários devem ser verificados.	Sim

Recurso	Cota padrão	Qualificada para aumento
	Se sua conta não estiver na sandbox, você pode enviar a qualquer endereço válido.	
Número de destinatários por mensagem	50 destinatários por mensagem	Não
Número de identidades que podem ser verificadas	10.000 identidades por região AWS **Note** As identidades referem-se a endereços de e-mail ou domínios ou a qualquer combinação de ambos. Todos os e-mails enviados usando o Amazon Pinpoint devem ser enviados de uma identidade verificada.	Não

Cotas de envio de e-mail

A cota de envio, a taxa de envio e os limites da sandbox são compartilhados entre os dois serviços na mesma Região. Se você usa o Amazon SES em us-east-1, foi removido da sandbox e teve sua cota/taxa de envio aumentada, todas essas alterações se aplicam à sua conta do Pinpoint em us-east-1.

Recurso	Cota padrão	Qualificada para aumento
Número de e-mails que podem ser enviados em um período de 24 horas (cota de envio)	Se sua conta estiver na sandbox, 200 e-mails por período de 24 horas. Se sua conta sair da sandbox, a cota variará de acordo com seu caso de uso específico. **Note** Essa cota baseia-se no número de destinatários, e não no número de mensagens exclusivas enviadas. Um destinatário é qualquer endereço de e-mail na linha "Para:".	Sim
Número de e-mails que podem ser enviados por segundo (taxa de envio)	Se sua conta estiver na sandbox, 1 e-mail por segundo. Se sua conta sair da sandbox, a taxa variará de acordo com seu caso de uso específico. **Note** Essa taxa baseia-se no número de destinatários, e não no número de mensagens exclusiva	Sim

Recurso	Cota padrão	Qualificada para aumento
	s enviadas. Um destinatário é qualquer endereço de e-mail na linha "Para:".	

Cotas de endpoint

As cotas a seguir se aplicam ao recurso Endpoints da API do Amazon Pinpoint.

O número máximo de atributos compatíveis por endpoint é 250 e o tamanho máximo do endpoint é 15 KB. No entanto, esse número de atributos pode ser limitado pelo tamanho total de um endpoint, que inclui todos os atributos. Se você encontrar algum erro ao adicionar atributos ao seu modelo, considere diminuir a quantidade de dados em cada atributo ou diminuir o número de atributos.

Recurso	Cota padrão	Qualificada para aumento
Tamanho do endpoint	Tamanho máximo: 15 KB	Não
Atributos conferidos coletivamente aos parâmetros `Attributes` , `Metrics` e `UserAttributes`	250 para todos os parâmetros de atributos por aplicativo	Não
Atributos conferidos ao parâmetro `Attributes`	250 para todos os parâmetros de atributos por aplicativo	Não
Atributos conferidos ao parâmetro `Metrics`	250 para todos os parâmetros de atributos por aplicativo	Não
Atributos conferidos ao parâmetro `UserAttributes`	250 para todos os parâmetros de atributos por aplicativo	Não
Extensão do nome do atributo	50 caracteres	Não

Recurso	Cota padrão	Qualificada para aumento
Extensão do valor do atributo	100 caracteres	Não
Objetos `EndpointBatchItem` em uma carga `EndpointBatchRequest`	100 por carga. O tamanho da carga não deve ser superior a 7 MB.	Não
Endpoints com o mesmo ID de usuário	15 endpoints exclusivos por ID de usuário	Não
Valores conferidos aos atributos de parâmetro `Attributes`	50 por atributo	Não
Valores conferidos aos atributos de parâmetro `UserAttributes`	50 por atributo	Não

Cotas de importação de endpoints

As cotas a seguir se aplicam à importação de endpoints para o Amazon Pinpoint.

Recurso	Cota padrão	Qualificada para aumento
Tarefas de importação ativas	10 por conta As tarefas de importação só contam nessa cota se estiverem em execução. Depois que a tarefa de importação for concluída, ela não será mais contabilizada nessa cota.	Não
Tamanho da importação	1 GB por tarefa de importação	Não

Recurso	Cota padrão	Qualificada para aumento
	Por exemplo, se cada endpoint tiver 4 KB ou menos, você poderá importar 250.000 endpoints.	

Cotas de ingestão de eventos

As cotas a seguir se aplicam à ingestão de eventos usando os SDKs AWS móveis e o recurso de eventos da API do Amazon Pinpoint.

Recurso	Cota padrão	Qualificada para aumento
Número máximo de tipos de eventos personalizados	1,500 por aplicativo	Não
Número máximo de chaves de atributo personalizado	500 por aplicativo	Não
Número máximo de valores de atributos personalizados por chave de atributo	100.000. Qualquer número que exceda 100.000 ainda está registrado, mas não estará disponível no console de análise do Amazon Pinpoint.	Não
Número máximo de caracteres por chave de atributo	50	Não
Número máximo de caracteres por valor de atributo	200. Se o número de caracteres exceder 200, o evento será descartado.	Não
Número máximo de chaves de métricas personalizadas	500 por aplicativo	Não

Recurso	Cota padrão	Qualificada para aumento
Número máximo de eventos em uma solicitação	100 por solicitação	Não
Tamanho máximo de uma solicitação	4 MB	Não
Tamanho máximo de um evento individual	1,000 KB	Não
Número máximo de chaves de atributos e de chaves de métricas para cada evento	40 por solicitação	Não

Cotas de jornadas

As cotas a seguir aplicam-se a jornadas.

As seguintes cotas se aplicam Região da AWS e algumas podem ser aumentadas. Para obter mais informações, consulte Solicitar um aumento de cota no Guia do usuário do Service Quotas..

Recurso	Cota padrão	Qualificada para aumento
Número máximo de jornadas ativas	50 por conta da	Não
Número máximo de ativos EventTriggeredJourneys	20 por conta	Não
Número máximo de atividades de jornada	40 por jornada	Não
Tamanho máximo do segmento	Para segmentos importados: 100.000.000 por jornada. Para segmentos dinâmicos: ilimitado	Não

Recurso	Cota padrão	Qualificada para aumento
Máximo de atividades da central de atendimento	3 por jornada	Não
Número máximo de regras de dias fechados	Vinte por canal	Não
Tamanho máximo do nome da regra de dias fechados	150 caracteres	Não
Número máximo de dias entre o horário de início e término para uma regra de dias fechados	7 dias	Não
Número máximo de regras de horário de funcionamento	Quatro por dia	Não

Cotas Lambda

As cotas a seguir se aplicam às configurações do Amazon Pinpoint para recuperar e processar dados do Lambda

Recurso	Cota padrão	Qualificada para aumento
Tamanho máximo de uma carga de invocação (solicitação e resposta) de uma função do Lambda	6 MB	Não
Tempo máximo de espera para que uma função do Lambda processe dados	15 segundos	Não
Número máximo de atributos de eventos por endpoint	5	Não

Recurso	Cota padrão	Qualificada para aumento
Número máximo de caracteres para um nome de atributo de evento	128 caracteres	Não
Número máximo de caracteres por valor de atributo de evento	128 caracteres	Não
Quantidade máxima de dias que uma jornada pode durar	540 dias	Não

Cotas de machine learning

As cotas a seguir se aplicam às configurações do Amazon Pinpoint para recuperar e processar dados de modelos de machine learning (ML).

Recurso	Cota padrão	Qualificada para aumento
Número máximo de configurações do modelo	1 por modelo de mensagem 100 por conta	Não
Número máximo de recomendações	5 por endpoint ou usuário	Não
Número máximo de atributos recomendados por endpoint ou usuário	1, se os valores dos atributos não forem processados por uma função do AWS Lambda 10, se os valores dos atributos forem processados por uma função do AWS Lambda	Não
Comprimento máximo de um nome de atributo recomendado	50 caracteres para um nome de atributo	Não

Recurso	Cota padrão	Qualificada para aumento
	25 caracteres para um nome de exibição de atributo (o nome que aparece no Attribute finder (Localizador de atributos) no console)	
Comprimento máximo de um valor de atributo recomendado recuperado do Amazon Personalize	100 caracteres	Não
Tamanho máximo de uma carga de invocação (solicitação e resposta) de uma função do Lambda	6 MB	Não
Tempo máximo de espera para que uma função do Lambda processe dados	15 segundos	Não
Número máximo de tentativas de invocação de uma função do Lambda	3 tentativas	Não

Dependendo de como você configurar o Amazon Pinpoint para usar um modelo de ML, cotas adicionais podem ser aplicadas. Para saber mais sobre as cotas do Amazon Personalize, consulte Cotas no Guia do desenvolvedor do Amazon Personalize. Para saber mais sobre as cotas do AWS Lambda , consulte Cotas no Guia do desenvolvedor do AWS Lambda .

Cotas de modelos de mensagem

As cotas a seguir se aplicam aos modelos de mensagens de sua conta do Amazon Pinpoint.

Recurso	Cota padrão	Qualificada para aumento
Número máximo de modelos de mensagem	20.000 por conta	Não
Número máximo de versões	5.000 por modelo	Não
Número máximo de caracteres em um modelo de e-mail	600 mil caracteres	Não
Número máximo de caracteres em um modelo dentro do aplicativo	200 mil caracteres	Não
Número máximo de caracteres nas partes de modelo padrão de um modelo de notificação por push	4.000 caracteres	Não
Número máximo de caracteres em partes de modelo específicas do ADM de um modelo de notificação por push	6.000 caracteres	Não
Número máximo de caracteres em partes de modelo específicas de APNs de um modelo de notificação por push	4.000 caracteres	Não
Número máximo de caracteres em partes de modelo específicas do Baidu de um modelo de notificação por push	4.000 caracteres	Não
Número máximo de caracteres em partes de modelo específicas do FCM de um modelo de notificação por push	4.000 caracteres	Não

Recurso	Cota padrão	Qualificada para aumento
Número máximo de caracteres em um modelo SMS	1.600 caracteres	Não
Número máximo de caracteres em um modelo de voz	10 mil caracteres	Não

Cotas de notificações por push

As cotas a seguir se aplicam a mensagens que o Amazon Pinpoint envia por meio dos canais de notificação por 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
Tamanho da carga da mensagem no Amazon Device Messaging (ADM)	6 KB por mensagem	Não
Tamanho da carga da mensagem do serviço Apple Push Notification (APNs)	4 KB por mensagem	Não
Tamanho da carga da mensagem no 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 da mensagem do Firebase Cloud Messaging (FCM)	4 KB por mensagem	Não

Cotas de mensagens no aplicativo

A cota a seguir se aplica às mensagens no aplicativo que você gerencia com o Amazon Pinpoint.

As seguintes cotas se aplicam Região da AWS e algumas podem ser aumentadas. Para obter mais informações, consulte Solicitar um aumento de cota no Guia do usuário do Service Quotas..

Recurso	Cota padrão	Qualificada para aumento
Número máximo de vezes que você pode chamar a `GetInAppMessages` API por segundo.	5.000 solicitações por segundo	Sim
Campanhas de mensagens no aplicativo	Cada projeto pode incluir até 25 campanhas que usam o canal de mensagens no aplicativo.	Para obter ajuda, consulte Solicitar um aumento de cota no Guia do usuário do Service Quotas

Cotas de segmentos

As cotas a seguir se aplicam ao recurso Segmentos da API do Amazon Pinpoint.

Recurso	Cota padrão	Qualificada para aumento
Número máximo de dimensões que podem ser usadas para criar um segmento	100 por segmento	Não

Recurso	Cota padrão	Qualificada para aumento
Número máximo de grupos de segmentos por segmento	5	Não
Número máximo de segmentos de origem por segmento	5	Não
Profundidade máxima dos segmentos de origem. Por exemplo, se um segmento tem um segmento de origem que também tem um segmento de origem, a cadeia de profundidade não é maior que esse limite.	5	Não

Cotas de SMS

As cotas a seguir se aplicam ao canal de SMS.

Para obter mais informações sobre o custo do SMS, consulte Entender os relatórios de uso e faturamento de SMS no Amazon Pinpoint no Guia do usuário do Amazon Pinpoint.

Recurso	Cota padrão	Qualificada para aumento
Limite de gastos	USD 1.00 por conta	Sim, mas os limites de gastos variam de acordo com a região. Você deve especific ar as regiões em que você precisa de um aumento.
Número de mensagens SMS que podem ser enviadas por segundo (taxa de envio)	Varia de acordo com o país de destino e o número de telefone de origem. Para obter mais informações, consulte	Sim, no entanto, talvez seja necessário obter um número de telefone que ofereça suporte a um throughput mais

Recurso	Cota padrão	Qualificada para aumento
	Partes de mensagem por limites de segundo no Guia do usuário do Amazon Pinpoint.	alto. Se você não tiver certeza de qual tipo de número usar, entre em contato com AWS Support nosso gerente de AWS conta para obter mais informações Se você usar um ID de remetente alfanumérico para enviar mensagens, talvez consiga aumentar a taxa de throughput. Para descobrir se um aumento de throughput está disponível para seu ID de remetente, abra uma solicitação de ID de remetente no console do Support Center. Em sua solicitação, inclua o ID de remetente existente, o país em que você usa esse ID de remetente e a taxa de throughput que você deseja solicitar.
Número de mensagens SMS que podem ser enviadas a um único destinatário a cada segundo	1 mensagem por segundo	Não
Número de tópicos do Amazon SNS para SMS bidirecional	100.000 por conta	Sim
Número de palavras-chave para SMS bidirecional	30 palavras-chave por número	Sim

Recurso	Cota padrão	Qualificada para aumento
Número de SMS e números de voz	25 por conta e região	Sim
Quantidade de números de telefone dedicados	25 por conta	Sim
Número de listas de exclusão Observação: a lista de exclusão padrão necessária é contabilizada nessa cota.	25 por conta	Sim
Número de conjuntos de configuração	25 por conta	Sim
Número de destinos de eventos	Cinco por conjunto de configuração	Não
Quantidade de números de telefone de destino verificados enquanto no sandbox de SMS	10 por conta	Sim
Quantidade de grupos de números de telefone	25 por conta	Sim
Número de identidades de origem que podem ser associadas a um grupo de números de telefone	100 por grupo de telefone	Sim

Cotas 10DLC

As cotas a seguir se aplicam a mensagens SMS enviadas usando números de telefone 10DLC. Números 10DLC só podem ser usados para enviar mensagens para destinatários nos Estados Unidos.

Recurso	Cota padrão	Qualificada para aumento
Máximo de 10 empresas DLC por Conta da AWS	25	Sim
Máximo de campanhas 10DLC por empresa 10DLC	10	Sim
Máximo de números 10DLC por campanha de 10DLC	49	Não

Cotas de voz

As cotas a seguir se aplicam ao canal de voz.

> **Note**
>
> Quando a conta é removida da sandbox, você se qualifica automaticamente para as cotas máximas mostradas na tabela a seguir.

Recurso	Cota padrão	Qualificada para aumento
O número máximo de mensagens de voz que podem ser enviadas em um período de 24 horas	Se sua conta estiver na sandbox: 20 mensagens	Não
O número de mensagens de voz que podem ser enviadas a um único destinatário em um período de 24 horas	5 mensagens	Não
Número de mensagens de voz que podem ser enviadas por minuto	Se sua conta estiver na sandbox: 5 chamadas por minuto	Não

Recurso	Cota padrão	Qualificada para aumento
	Se sua conta estiver fora da sandbox: 20 chamadas por minuto	
Número de mensagens de voz que podem ser enviadas a partir de um único número de telefone de origem por segundo	1 mensagem por segundo	Não
Duração da mensagem de voz	Se sua conta estiver na sandbox: 30 segundos Se sua conta estiver fora da sandbox: 5 minutos	Não

Recurso	Cota padrão	Qualificada para aumento
Capacidade de enviar mensagens de voz para números de telefone internacional	Se sua conta estiver na sandbox, você poderá enviar mensagens para destinatários somente nos seguintes países: • Austrália • Canadá • Alemanha • Hong Kong • Israel • Japão • México • Cingapura • Suécia • Estados Unidos • Reino Unido Se sua conta não estiver na sandbox, você poderá enviar mensagens para destinatários em qualquer país. **Note** As chamadas internacionais estão sujeitas a taxas adicionais, que variam de acordo com o país ou região de destino.	Não

Recurso	Cota padrão	Qualificada para aumento
Número de caracteres em uma mensagem de voz	3.000 caracteres faturáveis, em palavras que são faladas Total de 6.000 caracteres, incluindo caracteres faturáveis e tags SSML	Não
Número de conjuntos de configuração	10.000 conjuntos de configurações de voz por AWS região	Não

Solicitar um aumento de cota

Se o valor na coluna Eligible for Increase (Qualificada para aumento) em qualquer uma das tabelas anteriores for Yes (Sim), você poderá solicitar um aumento dessa cota.

Para solicitar um aumento da cota

1. Faça login AWS Management Console no https://console.aws.amazon.com/.
2. Crie um novo caso de AWS Support em https://console.aws.amazon.com/support/home#/case/create.
3. No painel Seus casos de suporte, escolha Criar caso.
4. Clique no link Deseja aumentar o limite de serviço?.
5. Em Aumento da cota de serviço, em Serviço, escolha uma das seguintes opções:
   - Para solicitar um aumento de cota relacionado ao canal de e-mail, escolha E-mail do Pinpoint.
   - Para solicitar um aumento de cota para limites de gastos de SMS ou taxas de envio de SMS, escolha Pinpoint SMS. Para todos os outros aumentos de cota de SMS, escolha Pinpoint
   - Para solicitar um aumento de cota relacionado ao canal de voz, escolha Voz do Pinpoint.
   - Para solicitar um aumento de cota relacionado a qualquer outro atributo do Amazon Pinpoint, escolha Pinpoint.
6. Dependendo do Serviço que você escolher, você pode ser solicitado a inserir o seguinte:
   - (Opcional) Em Fornecer um link ao site ou à aplicação que enviará mensagens SMS, forneça informações sobre o site, a aplicação ou o serviço que enviará mensagens SMS.

- (Opcional) Em Qual tipo de mensagem você pretende enviar, escolha o tipo de mensagem que pretende enviar utilizando o código longo:
  - Senhas de uso único: mensagens que fornecem senhas que seus clientes usam para se autenticarem em seu site ou aplicação.
  - Promocional: mensagens não críticas que promovem o seu negócio ou serviço, como ofertas especiais ou anúncios.
  - Transacional: mensagens informativas importantes que oferecem suporte para transações do cliente, como confirmações de pedidos ou alertas de contas. As mensagens transacionais não devem conter conteúdo promocional nem de marketing.
- (Opcional) Para qual AWS região você enviará mensagens, escolha a região de onde você enviará mensagens.
- (Opcional) Em Para quais países você planeja enviar mensagens, insira o país ou a região em que deseja comprar códigos simplificados.
- (Opcional) Em Como seus clientes optam por receber suas mensagens, forneça detalhes sobre seu processo de consentimento.
- (Opcional) No campo Forneça o modelo de mensagem que você planeja usar para enviar mensagens aos seus clientes, inclua o modelo que será usado.

7. Em Requests (Solicitações), faça o seguinte:
   - Para Região, escolha seu Região da AWS.
   - Em Tipo de recurso, escolha Limites gerais. O campo Tipo de recurso está presente somente para alguns serviços.
   - Em Cota, escolha a cota a ser alterada.
   - Em Novo valor da cota, insira um novo valor para a cota.
   - Para solicitar um aumento da mesma cota em uma solicitação adicional Região da AWS, escolha Adicionar outra solicitação e, em seguida, escolha a adicional Região da AWS e preencha a nova solicitação.
8. Escolha a cota que deseja aumentar e insira o novo valor desejado para a cota.
9. Em Descrição do caso, explique por que você está solicitando o aumento da cota.
10. Em Opções de contato, em Idioma de contato preferencial, escolha o idioma que você prefere usar ao se comunicar com a equipe de AWS Support.
11. Em Método de contato, escolha seu método preferido de comunicação com a equipe de AWS Support.

12. Selecione Enviar.

A equipe de AWS Support fornece uma resposta inicial à sua solicitação em 24 horas.

Para evitar que nossos sistemas sejam usados para enviar conteúdo indesejado ou malicioso, consideramos cuidadosamente cada solicitação. Se for possível, atenderemos à sua solicitação dentro desse período de 24 horas. No entanto, se precisarmos obter informações adicionais sobre você, o tempo de resolução poderá ser mais longo.

Talvez não possamos atender à sua solicitação se o seu caso de uso não estiver alinhado com nossas políticas.

Histórico da documentação do Amazon Pinpoint

A tabela a seguir descreve alterações importantes em cada versão do Guia do desenvolvedor do Amazon Pinpoint após dezembro de 2018. Para receber notificações sobre atualizações dessa documentação, você poderá se inscrever em um feed RSS.

- Última atualização da documentação: 16 de novembro de 2023

Alteração	Descrição	Data
Cabeçalhos de e-mail	Você pode adicionar cabeçalhos de e-mail às suas mensagens de e-mail. Para obter mais informações, consulte Envio de um e-mail com cabeçalhos de cancelamento de assinatura.	7 de maio de 2024
Orquestração de e-mail	O Amazon Pinpoint atualizou a forma como usa seus recursos do Amazon SES para enviar e-mails. Para obter mais informações, consulte Função do IAM para enviar e-mails com o Amazon SES.	30 de abril de 2024
O Guia do usuário do Amazon Pinpoint foi atualizado.	Os tópicos de gerenciamento de recursos de SMS e voz agora são redirecionados para o Guia do usuário de SMS do Amazon Pinpoint. Para obter mais informações, consulte o Guia do usuário do Amazon Pinpoint SMS.	8 de fevereiro de 2024
Cotas do Amazon Pinpoint	Foram adicionadas cotas para regras de número	19 de dezembro de 2023

	máximo de dias fechados, tamanho máximo do nome da regra de dias fechados, número máximo de dias entre o horário de início e término para uma regra de dias fechados e número máximo de regras de horário de funcionamento. Para obter mais informações, consulte cotas do Amazon Pinpoint.	
O Guia do usuário do Amazon Pinpoint foi atualizado.	Para obter as informações mais recentes sobre como criar, configurar e gerenciar os recursos de voz e SMS do Amazon Pinpoint, consulte o novo Guia do usuário de SMS do Amazon Pinpoint.	16 de novembro de 2023
Cotas do Amazon Pinpoint	Atualizou as cotas para UpdateEndpointsBatch UpdateEndpoint, PutEvents, DeleteEndpoint, e. GetEndpoint Para obter mais informações, consulte Cotas do Amazon Pinpoint.	22 de setembro de 2023

Cotas do Amazon Pinpoint	Atualizou as cotas para CreateEmailTemplate CreateSmsTemplate, CreatePushTemplate , CreateInAppTemplat e, CreateVoiceTemplat e, UpdateEmailTemplat e, UpdateSmsTemplate, UpdatePushTemplate, UpdateInAppTemplate, UpdateVoiceTemplate e. CreateImportJob Para obter mais informações, consulte Cotas do Amazon Pinpoint.	12 de setembro de 2023
Métricas de jornada e execução de campanhas	Novas métricas analítica s foram adicionadas para jornadas e campanhas. Para obter mais informações, consulte Métricas de execução de campanha e jornada.	25 de abril de 2023
Criar um endpoint da VPC de interface para o Amazon Pinpoint	O Amazon Pinpoint agora oferece suporte a endpoints de interface VPC. Para obter mais informações, consulte Criar um endpoint da VPC de interface para o Amazon Pinpoint.	11 de abril de 2023

Criptografia em trânsito	A partir de 22/03/2023, o Amazon Pinpoint não oferecerá mais suporte à versão TLS 1.0, mas você ainda poderá usar a TLS 1.2 ou posterior. Para obter mais informações, consulte Criptografia em trânsito.	20 de março de 2023
Cotas do Amazon Pinpoint	Atualizado o processo para solicitar um aumento de cota para campanhas, jornadas e mensagens no aplicativo. Para obter mais informações, consulte Cotas do Amazon Pinpoint.	16 de dezembro de 2022
Disponibilidade regional	O Amazon Pinpoint já está disponível na seguinte Região: Leste dos EUA (Ohio).	5 de outubro de 2022
Atualizações de exemplo de perfil do IAM	Vários exemplos de perfil do IAM foram atualizados em todo o documento para melhor se alinhar às práticas recomendadas de segurança.	27 de maio de 2022

API de SMS e voz, versão 2	O Amazon Pinpoint agora inclui uma API dedicada para enviar mensagens SMS e de voz. Essa API inclui novos atributos, como conjuntos de configurações, grupos e listas de cancelamento de assinatura, que são úteis para clientes que enviam SMS e mensagens de voz de forma transacional. Para obter mais informações, consulte Uso da API de SMS e de voz do Amazon Pinpoint.	1.º de abril de 2022
Criação e validação de senha única	O Amazon Pinpoint agora inclui um atributo que gera senhas de uso único (OTPs) e as envia aos seus usuários como mensagens SMS. Também inclui uma API para validar os códigos OTP quando seus usuários os inserem no aplicativo ou site. Para obter mais informações, consulte Envio e validação de senhas de uso único (OTPs).	26 de novembro de 2021
Mensagens no aplicativo	Foram adicionadas informaçõ es sobre a integração da capacidade de mensagens no aplicativo do Amazon Pinpoint com seus aplicativos.	10 de novembro de 2021

Exemplos de código	Foi adicionada uma biblioteca de exemplos de código para operações comuns do Amazon Pinpoint.	3 de novembro de 2021
Cotas do projeto	O número máximo de projetos do Amazon Pinpoint permanece em 100, mas essa cota agora pode ser aumentada abrindo uma solicitação de aumento do limite de serviço com. AWS Support	11 de outubro de 2021
Atualizações da política do Lambda.	Certas políticas de permissão do Lambda agora devem incluir uma condição `AWS:SourceAccount` . Atualizamos os exemplos de políticas nos tópicos Criação de canais personalizados no Amazon Pinpoint e Personalizar segmentos com o AWS Lambda para atender a esse requisito.	7 de outubro de 2021
UpdateEndpoint	A UpdateEndpoint API do Amazon Pinpoint agora está registrada por. CloudTrail	16 de novembro de 2020
Atributos personalizados	O Amazon Pinpoint agora oferece suporte a 250 atributos em modelos de mensagens de e-mail. Consulte Cotas.	18 de setembro de 2020

Disponibilidade regional	O Amazon Pinpoint já está disponível nas seguintes Regiões: Ásia-Pacífico (Tóquio), Europa (Londres) e Canadá (Central). A API de SMS e voz do Amazon Pinpoint não está disponível nessas Regiões.	10 de setembro de 2020
Disponibilidade regional	O Amazon Pinpoint já está disponível também na Região Ásia-Pacífico (Tóquio). A API de SMS e voz do Amazon Pinpoint não é compatível com voz nessa Região.	2 de setembro de 2020
Eventos de campanha	Foram adicionadas informaçõ es sobre um novo parâmetro `delivery_type` de evento a Eventos de campanha.	2 de agosto de 2020
Disponibilidade regional	O Amazon Pinpoint já está disponível na Região Ásia-Pacífico (Seul). A API do Amazon Pinpoint não é compatível com voz ou SMS nessa Região.	31 de julho de 2020
Disponibilidade regional	O Amazon Pinpoint agora está disponível na região AWS GovCloud (US) .	30 de abril de 2020
Canais personalizados	Informações atualizadas sobre como criar canais personali zados usando funções do Lambda ou webhooks.	23 de abril de 2020

Machine learning	Foram adicionadas informações sobre como recuperar recomendações personalizadas de modelos de recomendação e, opcionalmente, aprimorar essas recomendações usando funções. AWS Lambda	4 de março de 2020
Segurança	Capítulo de segurança adicionado, com informações sobre diversos controles de segurança e atributos do Amazon Pinpoint.	4 de fevereiro de 2020
Jornadas	Adição de informações sobre como usar jornadas do Amazon Pinpoint para desenvolver fluxos de trabalho automatizados que executam atividades de mensagens para projetos. Também foram adicionadas informações sobre como consultar dados de análise para um subconjunto de métricas que se aplicam a jornadas.	31 de outubro de 2019
Análise	Foram adicionados procedimentos que explicam como consultar dados analíticos para campanhas e mensagens transacionais, além de informações adicionadas sobre o uso dos resultados da consulta.	17 de outubro de 2019

Análise	Adição de informações sobre como consultar dados de análise para um subconjunto de métricas que se aplicam a e-mails transacionais e mensagens SMS.	6 de setembro de 2019
Exemplos de código	Adição de exemplos de código que podem ser usados para enviar notificações por push transacionais usando todos os serviços compatíveis com o Amazon Pinpoint.	30 de julho de 2019
Análise	Informações adicionadas sobre como consultar dados de análise para um subconjunto de métricas que se aplicam a projetos (aplicativos) e campanhas.	24 de julho de 2019
Segmentos	Foi adicionado um tutorial que descreve uma solução para importar dados de clientes para o Amazon Pinpoint de sistemas externos, como o Salesforce ou o Marketo.	14 de maio de 2019
Disponibilidade regional	O Amazon Pinpoint agora está disponível nas regiões AWS Ásia-Pacífico (Mumbai) e Ásia-Pacífico (Sydney).	25 de abril de 2019
Uso do Postman com o Amazon Pinpoint	Foi adicionado um tutorial que descreve como usar o Postman para interagir com a API do Amazon Pinpoint.	8 de abril de 2019

Tags	Adicionadas informações sobre marcação de recursos do Amazon Pinpoint.	27 de fevereiro de 2019
Registro de SMS	Foram adicionados um capítulo Tutorials (Tutoriais) e um tutorial que descreve como criar uma solução que lida com o registro do usuário do SMS.	27 de fevereiro de 2019
Exemplos de código	Foram adicionados exemplos de código em várias linguagens de programação que mostram como enviar e-mail, SMS e mensagens de voz programaticamente.	6 de fevereiro de 2019

Atualizações anteriores

A tabela a seguir descreve alterações importantes em cada versão do Guia do desenvolvedor do Amazon Pinpoint até dezembro de 2018.

Alteração	Descrição	Data
Disponibilidade regional	O Amazon Pinpoint agora está disponível nas regiões Oeste dos AWS EUA (Oregon) e Europa (Frankfurt).	21 de dezembro de 2018
Canal de voz	Você pode usar o novo canal de voz do Amazon Pinpoint para criar mensagens de voz e fornecê-las para seus clientes por telefone. Atualmente, você só pode enviar mensagens de voz	15 de novembro de 2018

Alteração	Descrição	Data
	usando a API de SMS e voz do Amazon Pinpoint.	
Disponibilidade na Europa (Irlanda)	O Amazon Pinpoint agora está disponível na região AWS Europa (Irlanda).	25 de outubro de 2018
API de eventos	Use a API do Amazon Pinpoint para registrar eventos e associá-los a endpoints.	7 de agosto de 2018
Códigos de exemplo para definir e pesquisar endpoints	São adicionados códigos de exemplo que mostram como definir, atualizar, excluir e pesquisar endpoints. Exemplos são fornecidos para o AWS CLI, AWS SDK for Java, e para a API do Amazon Pinpoint. Para obter mais informações, consulte Definir o público-alvo para o Amazon Pinpoint e Acessar os dados do público-alvo no Amazon Pinpoint.	7 de agosto de 2018
Permissões de exportação de endpoint	Configurar uma política do IAM que permita exportar endpoints do Amazon Pinpoint para um bucket do Amazon S3.	1º de maio de 2018

Alteração	Descrição	Data
Verificação de número de telefone para SMS	Use a API do Amazon Pinpoint para verificar um número de telefone a fim de determinar se ele é um destino válido para mensagens SMS.	23 de abril de 2018
Tópicos atualizados da integração do Amazon Pinpoint	Integre o Amazon Pinpoint com seu Android, iOS ou JavaScript aplicativo usando AWS SDKs ou bibliotecas.	23 de março de 2018
AWS CloudTrail registro	Foram adicionadas informaçõ es sobre o registro de chamadas de API do Amazon Pinpoint com. CloudTrail	6 de fevereiro de 2018
Service Quotas atualizadas	Atualização de Cotas com informações adicionais sobre cotas de e-mail.	19 de janeiro de 2018
Versão beta pública das extensões do Amazon Pinpoint	Use AWS Lambda funções para personalizar segmentos ou criar canais de mensagens personalizados.	28 de novembro de 2017
Cotas de carga de notificação por push	As cotas incluem tamanhos de carga para mensagens por push para dispositivos móveis.	25 de outubro de 2017
Service Quotas atualizadas	Informações sobre canal de SMS e e-mail adicionados a Cotas.	9 de outubro de 2017

Alteração	Descrição	Data
Push para dispositivos móveis do ADM e Baidu	Atualize o código de seu aplicativo para lidar com notificações por push dos canais de push para dispositivos móveis do Baidu e ADM.	27 de setembro de 2017
IDs de usuário e eventos de autenticação com grupos de usuários do Amazon Cognito.	Se usar grupos de usuários do Amazon Cognito para gerenciar o acesso de usuários a seus aplicativos móveis, o Amazon Cognito atribuirá IDs de usuário aos endpoints e relatará eventos de autenticação para o Amazon Pinpoint.	26 de setembro de 2017
IDs de usuário	Atribua IDs de usuário a endpoints para monitorar o uso de aplicativo por usuários específicos. São fornecidos exemplos para AWS Mobile SDKs e SDK para Java.	31 de agosto de 2017
Eventos de autenticação	Relate eventos de autenticação para saber com que frequência os usuários são autenticados com seu aplicativo. São fornecidos exemplos em Relatar eventos no aplicativo.	31 de agosto de 2017
Atualização de eventos de exemplo	Os eventos de exemplo incluem eventos que o Amazon Pinpoint envia para atividades de e-mail e SMS.	08 de junho de 2017

Alteração	Descrição	Data
Gerenciamento da sessão do Android	Gerencie as sessões dos aplicativos Android usando uma classe informada pelo aplicativo de amostra AWS Mobile Hub .	20 de abril de 2017
Amostras de eventos de monetização atualizadas	O código de exemplo está atualizado para relatar eventos de monetização.	31 de março de 2017
Streams de eventos	É possível configurar o Amazon Pinpoint para enviar seus eventos de aplicativo e campanha para um fluxo do Kinesis.	24 de março de 2017
Permissões	Consulte Como o Amazon Pinpoint funciona com o IAM para obter informações sobre como conceder acesso ao Amazon Pinpoint AWS para usuários em sua conta e usuários de seu aplicativo móvel.	12 de janeiro de 2017
Disponibilidade geral do Amazon Pinpoint	Esta versão apresenta o Amazon Pinpoint.	1° de dezembro de 2016

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.