Panduan Pengguna

AWS Push Pesan Pengguna Akhir

AWS Push Pesan Pengguna Akhir: Panduan Pengguna

Table of Contents

Apa itu AWS End User Messaging Push?

> **Note**
>
> Fitur pemberitahuan Push dari Amazon Pinpoint sekarang disebut AWS End User Messaging.

Dengan AWS End User Messaging Push, Anda dapat melibatkan pengguna aplikasi Anda dengan mengirimkan pemberitahuan push melalui saluran notifikasi push. Kami mendukung Layanan Pemberitahuan Push Apple (APNs), Firebase Cloud Messaging (FCM), Amazon Device Messaging (ADM), dan Baidu Push.

Topik

Apakah Anda pengguna AWS End User Messaging Push pertama kali?

Jika Anda adalah pengguna pertama kali dari AWS End User Messaging Push, kami sarankan Anda mulai dengan membaca bagian berikut:

- Menyiapkan sebuah Akun AWS
- Memulai dengan AWS End User Messaging Push
- Membuat aplikasi dan mengaktifkan saluran push

Fitur dari AWS End User Messaging Push

Anda dapat mengirim pemberitahuan push ke aplikasi Anda menggunakan saluran terpisah untuk layanan pemberitahuan push berikut:

- Firebase Cloud Messaging (FCM)
- Layanan Pemberitahuan Push Apple (APNs)

> **Note**
>
> Anda dapat menggunakannya APNs untuk mengirim pesan ke perangkat iOS seperti iPhone dan iPad, serta ke browser Safari di perangkat macOS, seperti laptop dan desktop Mac.

- Dorongan Awan Baidu
- Olahpesan Perangkat Amazon (ADM)

Mengakses AWS Push Pesan Pengguna Akhir

Jelaskan secara singkat berbagai cara untuk mendapatkan akses ke layanan, baik dengan konsol, CLI, atau API.

Anda dapat mengelola AWS End User Messaging Push menggunakan antarmuka berikut:

AWS Konsol Push Pesan Pengguna Akhir

Antarmuka web tempat Anda membuat dan mengelola sumber daya AWS End User Messaging Push. Jika Anda telah mendaftar Akun AWS, Anda dapat mengakses konsol Push Pesan Pengguna AWS Akhir dari AWS Management Console.

AWS Command Line Interface

Berinteraksi dengan AWS layanan menggunakan perintah di shell baris perintah Anda. AWS Command Line Interface Ini didukung di Windows, macOS, dan Linux. Untuk informasi selengkapnya tentang AWS CLI, lihat Panduan AWS Command Line Interface Pengguna. Anda dapat menemukan perintah AWS End User Messaging Push di AWS CLI Command Reference.

AWS SDKs

Jika Anda seorang pengembang perangkat lunak yang lebih suka membangun aplikasi menggunakan bahasa khusus APIs daripada mengirimkan permintaan melalui HTTP atau HTTPS, AWS menyediakan pustaka, kode sampel, tutorial, dan sumber daya lainnya. Pustaka ini menyediakan fungsi dasar yang mengotomatiskan tugas, seperti menandatangani permintaan Anda secara kriptografis, mencoba ulang permintaan, dan menangani respons kesalahan. Fungsi-

fungsi ini membantu membuatnya lebih efisien bagi Anda untuk memulai. Untuk informasi lebih lanjut, lihat Alat untuk Membangun di AWS.

Ketersediaan wilayah

AWS End User Messaging Push tersedia di beberapa Wilayah AWS di Amerika Utara, Eropa, Asia, dan Oseania. Di setiap Wilayah, AWS pertahankan beberapa Availability Zone. Availability Zone ini secara fisik terisolasi satu sama lain, tetapi disatukan oleh koneksi jaringan privat, latensi rendah, throughput tinggi, dan sangat redundan. Availability Zone ini digunakan untuk menyediakan tingkat ketersediaan dan redundansi yang sangat tinggi, sekaligus meminimalkan latensi.

Untuk mempelajari selengkapnya Wilayah AWS, lihat Menentukan Wilayah AWS akun mana yang dapat digunakan di Referensi Umum Amazon Web. Untuk daftar semua Wilayah tempat Push Pesan Pengguna AWS Akhir saat ini tersedia dan titik akhir untuk setiap Wilayah, lihat Titik akhir dan kuota untuk Amazon Pinpoint API dan AWS titik akhir layanan di. Referensi Umum Amazon Web Untuk mempelajari lebih lanjut tentang jumlah Availability Zone yang tersedia di setiap Wilayah, lihat infrastruktur AWS global.

Menyiapkan sebuah Akun AWS

Sebelum Anda dapat menggunakan AWS End User Messaging Push untuk mengirim pemberitahuan push ke aplikasi Anda, Anda harus terlebih dahulu mendapatkan Akun AWS izin IAM yang memadai. Ini juga Akun AWS dapat digunakan untuk layanan lain di AWS ekosistem.

Topik

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Memulai dengan AWS End User Messaging Push

Untuk menyiapkan AWS End User Messaging Push sehingga dapat mengirim pemberitahuan push ke aplikasi Anda, Anda harus terlebih dahulu memberikan kredensi yang mengizinkan Push Pesan Pengguna AWS Akhir untuk mengirim pesan ke aplikasi Anda. Kredensi yang Anda berikan bergantung pada sistem notifikasi push yang Anda gunakan:

- Untuk kredensi layanan Pemberitahuan Push Apple (APN), lihat Mendapatkan kunci enkripsi dan ID kunci dari Apple dan Mendapatkan sertifikat penyedia dari Apple dalam dokumentasi Pengembang Apple.
- Untuk kredensial Firebase Cloud Messaging (FCM) yang dapat diperoleh melalui Firebase console, lihat Firebase Cloud Messaging.
- Untuk kredensi Baidu, lihat Baidu.
- Untuk kredensial Amazon Device Messaging (ADM), lihat Mendapatkan Kredensial.

Membuat aplikasi dan mengaktifkan saluran push

Sebelum Anda dapat menggunakan AWS End User Messaging Push untuk mengirim pemberitahuan push, Anda harus terlebih dahulu membuat aplikasi dan mengaktifkan saluran pemberitahuan push.

Kontekstual

Aplikasi

Aplikasi adalah wadah penyimpanan untuk semua pengaturan AWS End User Messaging Push Anda. Aplikasi ini juga menyimpan saluran Amazon Pinpoint, kampanye, dan pengaturan perjalanan Anda.

Kunci

Kunci penandatanganan pribadi yang digunakan oleh AWS End User Messaging Push untuk menandatangani token APNs otentikasi secara kriptografis. Anda mendapatkan kunci penandatanganan dari akun pengembang Apple Anda.

Jika Anda memberikan kunci penandatanganan,AWS End User Messaging Push menggunakan token untuk mengautentikasi setiap pemberitahuan push yang Anda kirim. APNs Dengan kunci penandatanganan, Anda dapat mengirim pemberitahuan push ke lingkungan APNs produksi dan kotak pasir.

Tidak seperti sertifikat, kunci penandatanganan Anda tidak kedaluwarsa. Anda hanya memberikan kunci Anda sekali, dan Anda tidak perlu memperbaruinya nanti. Anda dapat menggunakan kunci penandatanganan yang sama untuk beberapa aplikasi. Untuk informasi selengkapnya, lihat Berkomunikasi dengan APNs menggunakan token autentikasi di Bantuan Xcode.

Sertifikat

Sertifikat TLS yang digunakan AWS End User Messaging Push untuk mengautentikasi APNs saat Anda mengirim pemberitahuan push. APNs Sertifikat dapat mendukung lingkungan produksi dan kotak pasir, atau hanya dapat mendukung lingkungan kotak pasir. Anda mendapatkan sertifikat dari akun pengembang Apple Anda.

Sertifikat berakhir setelah satu tahun. Ketika ini terjadi, Anda harus membuat sertifikat baru, yang kemudian Anda berikan ke AWS End User Messaging Push untuk memperbarui pengiriman

pemberitahuan push. Untuk informasi selengkapnya, lihat Berkomunikasi dengan APNs menggunakan sertifikat TLS di Bantuan Xcode.

Prasyarat

Sebelum Anda dapat menggunakan saluran push apa pun, Anda memerlukan kredensyal yang valid untuk layanan push. Untuk informasi lebih lanjut tentang mendapatkan kredensyal, lihat. Memulai dengan AWS End User Messaging Push

Prosedur

Ikuti petunjuk ini untuk membuat aplikasi dan mengaktifkan salah satu saluran push. Untuk menyelesaikan prosedur ini, Anda hanya perlu memasukkan nama aplikasi. Anda dapat mengaktifkan atau menonaktifkan salah satu saluran push di lain waktu.

1. Buka konsol Push Pesan Pengguna AWS Akhir di https://console.aws.amazon.com/push-notifications/.
2. Pilih Create application (Buat aplikasi).
3. Untuk nama Aplikasi masukkan nama untuk aplikasi Anda.
4. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan layanan Pemberitahuan Push Apple (APNs).

    a. Untuk layanan Pemberitahuan Push Apple (APNs) pilih Aktifkan.

    b. Untuk jenis otentikasi Default pilih salah satu:

        i. Jika Anda memilih Kredensyal kunci, berikan informasi berikut dari akun pengembang Apple Anda. AWS End User Messaging Push memerlukan informasi ini untuk membuat token otentikasi.

            - ID Kunci — ID yang ditetapkan ke kunci penandatanganan Anda.
            - Bundle identifier — ID yang ditetapkan ke aplikasi iOS Anda.
            - Pengenal tim — ID yang ditetapkan ke tim akun pengembang Apple Anda.
            - Kunci autentikasi — File.p8 yang Anda unduh dari akun pengembang Apple saat Anda membuat kunci otentikasi.

        ii. Jika Anda memilih kredensi Sertifikat, berikan informasi berikut:

- Sertifikat SSL — File.p12 untuk sertifikat TLS Anda.
- Kata sandi sertifikat — Jika Anda menetapkan kata sandi ke sertifikat Anda, masukkan di sini.
- Jenis sertifikat — Pilih jenis sertifikat yang akan digunakan.

5. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan Firebase Cloud Messaging (FCM).

   a. Untuk Firebase Cloud Messaging (FCM) pilih Aktifkan.
   b. Untuk jenis otentikasi Default pilih salah satu:

      i. Untuk kredensyal Token (disarankan) pilih Pilih file dan kemudian pilih file JSON layanan Anda.
      ii. Untuk kredensyal Kunci, masukkan kunci Anda di kunci API.

6. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan Baidu Cloud Push.

   a. Untuk Baidu Cloud Push pilih Aktifkan.
   b. Untuk kunci API, masukkan kunci API Anda.
   c. Untuk kunci Rahasia masukkan kunci rahasia Anda.

7. (Opsional) Ikuti langkah opsional ini untuk mengaktifkan Pesan Perangkat Amazon.

   a. Untuk Amazon Device Messaging pilih Aktifkan.
   b. Untuk ID Klien, masukkan ID klien Anda.
   c. Untuk rahasia Klien masukkan rahasia klien Anda.

8. Pilih Create application (Buat aplikasi).

Menonaktifkan saluran push

Ikuti petunjuk ini untuk menonaktifkan salah satu saluran push.

1. Buka konsol Push Pesan Pengguna AWS Akhir di https://console.aws.amazon.com/push-notifications/.
2. Pilih aplikasi yang berisi kredensi push Anda.
3. (Opsional) Untuk layanan Pemberitahuan Push Apple (APNs) hapus Aktifkan.
4. (Opsional) Untuk Firebase Cloud Messaging (FCM) hapus Aktifkan.
5. (Opsional) Untuk Baidu Cloud Push clear Enable.
6. (Opsional) Untuk Amazon Device Messaging clear Enable.
7. Pilih Simpan perubahan.

Mengirim pesan

AWS End User Messaging Push API dapat mengirim pemberitahuan push transaksional ke pengidentifikasi perangkat tertentu. Bagian ini berisi contoh kode lengkap yang dapat Anda gunakan untuk mengirim pemberitahuan push melalui AWS End User Messaging Push API dengan menggunakan AWS SDK.

Anda dapat menggunakan contoh ini untuk mengirim pemberitahuan push melalui layanan pemberitahuan push apa pun yang didukung oleh AWS End User Messaging Push. Saat ini, AWS End User Messaging Push mendukung saluran berikut: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push, dan Amazon Device Messaging (ADM).

Untuk contoh kode lainnya pada titik akhir, segmen, dan saluran, lihat Contoh kode.

> **Note**
>
> Saat Anda mengirim pemberitahuan push melalui layanan Firebase Cloud Messaging (FCM), gunakan nama layanan `GCM` dalam panggilan Anda ke API Push Pesan Pengguna AWS Akhir. Layanan Google Cloud Messaging (GCM) dihentikan oleh Google pada 10 April 2018. Namun, AWS End User Messaging Push API menggunakan nama `GCM` layanan untuk pesan yang dikirim melalui layanan FCM untuk menjaga kompatibilitas dengan kode API yang ditulis sebelum penghentian layanan GCM.

GCM (AWS CLI)

Contoh berikut menggunakan send-messages untuk mengirim notifikasi GCM Push dengan. AWS CLI Ganti *`token`* dengan token unik perangkat dan *`611e3e3cdd47474c9c1399a50example`* dengan pengenal aplikasi Anda.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request file://myfile.json \
--region us-west-2

Contents of myfile.json:
{
  "Addresses": {
    "token": {
```

```
      "ChannelType" : 'GCM'
    }
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "URL",
      "Body": "This is a sample message",
      "Priority": "normal",
      "SilentPush": True,
      "Title": "My sample message",
      "TimeToLive": 30,
      "Url": "https://www.example.com"
      }
  }
}
```

Contoh berikut menggunakan send-messages untuk mengirim notifikasi GCM Push, menggunakan semua kunci lama, dengan tombol. AWS CLI Ganti *token* dengan token unik perangkat dan *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\"notification\": {\n \"title\": \"string\",\n \"body\":
 \"string\",\n \"android_channel_id\": \"string\",\n \"body_loc_args\": [\n \"string
\"\n ],\n \"body_loc_key\": \"string\",\n \"click_action\": \"string\",\n \"color\":
 \"string\",\n \"icon\": \"string\",\n \"sound\": \"string\",\n \"tag\": \"string
\",\n \"title_loc_args\": [\n \"string\"\n ],\n \"title_loc_key\": \"string\"\n },
\"data\":{\"message\":\"hello in data\"} }",
      "TimeToLive" : 309744
     }
   },
  "Addresses": {
    "token": {
      "ChannelType":"GCM"
      }
   }
}'
\ --region us-east-1
```

Contoh berikut menggunakan send-messages untuk mengirim notifikasi GCM Push dengan payload FCMv1 pesan menggunakan. AWS CLI Ganti *token* dengan token unik perangkat dan *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda.

```
aws pinpoint send-messages \
--application-id 6a2dafd84bec449ea75fb773f4c41fa1 \
--message-request
'{
  "MessageConfiguration": {
    "GCMMessage":{
      "RawContent": "{\n \"fcmV1Message\": \n {\n \"message\" :{\n  \"notification
\": {\n \"title\": \"string\",\n \"body\": \"string\"\n },\n \"android\": {\n
 \"priority\": \"high\",\n \"notification\": {\n \"title\": \"string\",\n \"body
\": \"string\",\n \"icon\": \"string\",\n \"color\": \"string\",\n \"sound\":
 \"string\",\n \"tag\": \"string\",\n \"click_action\": \"string\",\n \"body_loc_key
\": \"string\",\n \"body_loc_args\": [\n \"string\"\n ],\n \"title_loc_key
\": \"string\",\n \"title_loc_args\": [\n \"string\"\n ],\n \"channel_id\":
 \"string\",\n \"ticker\": \"string\",\n \"sticky\": true,\n \"event_time\":
 \"2024-02-06T22:11:55Z\",\n \"local_only\": true,\n \"notification_priority\":
 \"PRIORITY_UNSPECIFIED\",\n \"default_sound\": false,\n \"default_vibrate_timings
\": true,\n \"default_light_settings\": false,\n \"vibrate_timings\": [\n \"22s
\"\n ],\n \"visibility\": \"VISIBILITY_UNSPECIFIED\",\n \"notification_count\": 5,
\n \"light_settings\": {\n \"color\": {\n \"red\": 1,\n \"green\": 2,\n \"blue\":
 3,\n \"alpha\": 6\n },\n \"light_on_duration\": \"112s\",\n \"light_off_duration
\": \"1123s\"\n },\n \"image\": \"string\"\n },\n \"data\": {\n \"dataKey1\":
 \"priority message\",\n \"data_key_3\": \"priority message\",\n \"dataKey2\":
 \"priority message\",\n \"data_key_5\": \"priority message\"\n },\n \"ttl\":
 \"10023.32s\"\n },\n \"apns\": {\n \"payload\": {\n \"aps\": {\n \"alert\": {\n
 \"subtitle\": \"string\",\n \"title-loc-args\": [\n \"string\"\n ],\n \"title-loc-
key\": \"string\",\n \"launch-image\": \"string\",\n \"subtitle-loc-key\": \"string
\",\n \"subtitle-loc-args\": [\n \"string\"\n ],\n \"loc-args\": [\n \"string
\"\n ],\n \"loc-key\": \"string\",\n \"title\": \"string\",\n \"body\": \"string
\"\n },\n \"thread-id\": \"string\",\n \"category\": \"string\",\n \"content-
available\": 1,\n \"mutable-content\": 1,\n \"target-content-id\": \"string\",\n
 \"interruption-level\": \"string\",\n \"relevance-score\": 25,\n \"filter-criteria
\": \"string\",\n \"stale-date\": 6483,\n \"content-state\": {},\n \"timestamp\":
 673634,\n \"dismissal-date\": 4,\n \"attributes-type\": \"string\",\n \"attributes
\": {},\n \"sound\": \"string\",\n \"badge\": 5\n }\n }\n },\n \"webpush\": {\n
 \"notification\": {\n \"permission\": \"granted\",\n \"maxActions\": 2,\n \"actions
\": [\n \"title\"\n ],\n \"badge\": \"URL\",\n \"body\": \"Hello\",\n \"data\": {\n
 \"hello\": \"hey\"\n },\n \"dir\": \"auto\",\n \"icon\": \"icon\",\n \"image\":
 \"image\",\n \"lang\": \"string\",\n \"renotify\": false,\n \"requireInteraction\":
 true,\n \"silent\": false,\n \"tag\": \"tag\",\n \"timestamp\": 1707259524964,\n
```

```
 \"title\": \"hello\",\n \"vibrate\": [\n 100,\n 200,\n 300\n ]\n },\n \"data\": {\n
 \"data1\": \"priority message\",\n \"data2\": \"priority message\",\n \"data12\":
 \"priority message\",\n \"data3\": \"priority message\"\n }\n },\n \"data\": {\n
 \"data7\": \"priority message\",\n \"data5\": \"priority message\",\n \"data8\":
 \"priority message\",\n \"data9\": \"priority message\"\n }\n }\n \n}\n}",
      "TimeToLive" : 309744
    }
  },
  "Addresses": {
    token: {
      "ChannelType":"GCM"
    }
   }
}'
\ --region us-east-1
```

jika menggunakan `ImageUrl` bidang untuk GCM, pinpoint mengirimkan bidang sebagai pemberitahuan data, dengan kuncinya`pinpoint.notification.imageUrl`, yang dapat mencegah gambar dirender keluar dari kotak. Harap gunakan RawContent atau tambahkan penanganan kunci data seperti mengintegrasikan aplikasi Anda. AWS Amplify

Safari (AWS CLI)

Anda dapat menggunakan AWS End User Messaging Push untuk mengirim pesan ke komputer macOS yang menggunakan browser web Safari Apple. Untuk mengirim pesan ke browser Safari, Anda harus menentukan konten pesan mentah, dan Anda harus menyertakan atribut tertentu dalam payload pesan. Anda dapat melakukannya dengan membuat templat pemberitahuan push dengan payload pesan mentah, atau dengan menentukan konten pesan mentah secara langsung dalam pesan kampanye, di Panduan Pengguna Amazon Pinpoint.

> **Note**
>
> Atribut khusus ini diperlukan untuk mengirim ke laptop macOS dan komputer desktop yang menggunakan browser web Safari. Tidak diperlukan untuk mengirim ke perangkat iOS seperti iPhone dan iPad.

Untuk mengirim pesan ke browser web Safari, Anda harus menentukan payload pesan mentah. Payload pesan mentah harus menyertakan `url-args` array dalam `aps` objek. `url-args`Array diperlukan untuk mengirim pemberitahuan push ke browser web Safari. Namun, array dapat diterima untuk berisi satu elemen kosong.

Contoh berikut menggunakan kirim-pesan untuk mengirim pemberitahuan ke browser web Safari dengan. AWS CLI Ganti *token* dengan token unik perangkat dan *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
  "Addresses": {
    "token":
    {
      "ChannelType":"APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
        "RawContent":
          "{\"aps\": {\"alert\": { \"title\": \"Title of my message\", \"body\":
 \"This is a push notification for the Safari web browser.\"},\"content-available\":
 1,\"url-args\": [\"\"]}}"
     }
  }
}'
\ --region us-east-1
```

Untuk informasi selengkapnya tentang pemberitahuan push Safari, lihat Mengonfigurasi Pemberitahuan Push Safari di situs web Pengembang Apple.

APNS (AWS CLI)

Contoh berikut menggunakan kirim-pesan untuk mengirim pemberitahuan APNS Push dengan. AWS CLI Ganti *token* dengan token unik perangkat, *611e3e3cdd47474c9c1399a50example* dengan pengenal aplikasi Anda, dan *GAME_INVITATION* dengan pengenal unik.

```
aws pinpoint send-messages \
--application-id 611e3e3cdd47474c9c1399a50example \
--message-request
'{
    "Addresses": {
     "token":
    {
      "ChannelType":"APNS"
    }
```

```
  },
  "MessageConfiguration": {
    "APNSMessage": {
      "RawContent": "{\"aps\" : {\"alert\" : {\"title\" : \"Game Request\",
\"subtitle\" : \"Five Card Draw\",\"body\" : \"Bob wants to play poker\"},\"category
\" : \"GAME_INVITATION\"},\"gameID\" : \"12345678\"}"
      }
    }
}'
\ --region us-east-1
```

JavaScript (Node.js)

Gunakan contoh ini untuk mengirim pemberitahuan push dengan menggunakan AWS SDK untuk JavaScript di Node.js. Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengonfigurasi SDK untuk JavaScript di Node.js.

Contoh ini juga mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat Menyetel kredensional di AWS SDK untuk JavaScript di Panduan Pengembang Node.js.

```
'use strict';

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

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

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

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

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

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

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

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

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

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

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

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

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

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

  return messageRequest
}

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

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

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

```
  AWS.config.credentials = credentials;

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

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

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

SendMessage()
```

Python

Gunakan contoh ini untuk mengirim pemberitahuan push dengan menggunakan AWS SDK untuk Python (Boto3). Contoh ini mengasumsikan bahwa Anda telah menginstal dan mengkonfigurasi SDK for Python (Boto3).

Contoh ini juga mengasumsikan bahwa Anda menggunakan file kredensial bersama untuk menentukan Kunci Akses dan Kunci Akses Rahasia untuk pengguna yang sudah ada. Untuk informasi selengkapnya, lihat Kredensial di Referensi API AWS SDK for Python (Boto3).

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

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

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

# The content of the push notification.
```

```
message = ("This is a sample message sent from End User Messaging Push by using the
 "
           "AWS SDK untuk Python (Boto3).")

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

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

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

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

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

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

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

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

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

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

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

    return message_request

# Show a success or failure message, and provide the response from the API.
def show_output(response):
    if response['MessageResponse']['Result'][recipient["token"]]['DeliveryStatus']
 == "SUCCESSFUL":
        status = "Message sent! Response information:\n"
    else:
```

```
        status = "The message wasn't sent. Response information:\n"
    print(status, json.dumps(response,indent=4))

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

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

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

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

send_message()
```

Sumber daya tambahan

- Untuk informasi selengkapnya tentang templat saluran Push, lihat Membuat templat pemberitahuan push di Panduan Pengguna Amazon Pinpoint.

Menerima pemberitahuan push di aplikasi Anda

Topik berikut menjelaskan cara memodifikasi aplikasi Swift, Android, React Native, atau Flutter Anda sehingga menerima pemberitahuan push.

Topik

Menyiapkan Pemberitahuan Push Swift

Pemberitahuan push untuk aplikasi iOS dikirim menggunakan layanan Pemberitahuan Push Apple (APNs). Sebelum dapat mengirim pemberitahuan push ke perangkat iOS, Anda harus membuat ID aplikasi di portal Pengembang Apple, dan Anda harus membuat sertifikat yang diperlukan. Anda dapat menemukan informasi selengkapnya tentang menyelesaikan langkah-langkah ini di Menyiapkan layanan pemberitahuan push di dokumentasi AWS Amplify.

Bekerja dengan APNs token

Sebagai praktik terbaik, Anda harus mengembangkan aplikasi sehingga token perangkat pelanggan dibuat ulang saat aplikasi diinstal ulang.

Jika penerima memutakhirkan perangkatnya ke iOS versi utama yang baru (misalnya, dari iOS 12 ke iOS 13), dan kemudian menginstal ulang aplikasi Anda, aplikasi akan menghasilkan token baru. Jika aplikasi Anda tidak me-refresh token, token lama akan digunakan untuk mengirim notifikasi. Akibatnya, layanan Pemberitahuan Push Apple (APNs) menolak notifikasi, karena token sekarang tidak valid. Ketika Anda mencoba mengirim pemberitahuan, Anda menerima pemberitahuan kegagalan pesan dari APNs.

Menyiapkan pemberitahuan push Android

Pemberitahuan push untuk aplikasi Android dikirim menggunakan Firebase Cloud Messaging (FCM), yang menggantikan Google Cloud Messaging (GCM). Sebelum Anda dapat mengirim pemberitahuan push ke perangkat Android, Anda harus mendapatkan kredensi FCM. Anda kemudian dapat menggunakan kredensi tersebut untuk membuat proyek Android dan meluncurkan aplikasi contoh yang dapat menerima pemberitahuan push. Anda dapat menemukan informasi selengkapnya tentang menyelesaikan langkah-langkah ini di bagian Pemberitahuan push di dokumentasi AWS Amplify.

Menyiapkan Pemberitahuan Flutter Push

Notifikasi push untuk aplikasi Flutter dikirim menggunakan Firebase Cloud Messaging (FCM) untuk Android, dan untuk iOS. APNs Anda dapat menemukan informasi selengkapnya tentang menyelesaikan langkah-langkah ini di bagian Pemberitahuan push pada dokumentasi AWS Amplify Flutter.

Menyiapkan Pemberitahuan Push React Native

Notifikasi push untuk aplikasi React Native dikirim menggunakan Firebase Cloud Messaging (FCM) untuk Android, dan untuk APNs iOS. Anda dapat menemukan informasi selengkapnya tentang menyelesaikan langkah-langkah ini di bagian Pemberitahuan push pada dokumentasi AWS Amplify JavaScript.

Buat aplikasi di AWS End User Messaging Push

Untuk mulai mengirim pemberitahuan push di AWS End User Messaging Push, Anda harus membuat aplikasi. Selanjutnya, Anda harus mengaktifkan saluran pemberitahuan push yang ingin Anda gunakan dengan memberikan kredensi yang sesuai.

Anda dapat membuat aplikasi baru dan mengatur saluran pemberitahuan push dengan menggunakan konsol AWS End User Messaging Push. Untuk informasi selengkapnya, lihat Membuat aplikasi dan mengaktifkan saluran push.

Anda juga dapat membuat dan menyiapkan aplikasi dengan menggunakan API, AWS SDK, atau AWS Command Line Interface(AWS CLI). Untuk membuat aplikasi, gunakan `Apps` sumber daya. Untuk mengonfigurasi saluran pemberitahuan push, gunakan sumber daya berikut:

- APNs saluran untuk mengirim pesan ke pengguna perangkat iOS dengan menggunakan layanan Pemberitahuan Push Apple.
- Saluran ADM untuk mengirim pesan ke pengguna perangkat Amazon Kindle Fire.
- Saluran Baidu untuk mengirim pesan ke pengguna Baidu.
- Saluran GCM untuk mengirim pesan ke perangkat Android menggunakan Firebase Cloud Messaging (FCM), yang menggantikan Google Cloud Messaging (GCM).

Menangani pemberitahuan push

Setelah Anda mendapatkan kredensi yang diperlukan untuk mengirim pemberitahuan push, Anda dapat memperbarui aplikasi Anda sehingga mereka dapat menerima pemberitahuan push. Untuk informasi selengkapnya, lihat Pemberitahuan push—Memulai dokumentasi. AWS Amplify

Menghapus aplikasi

Prosedur ini menghapus aplikasi dari akun Anda dan semua sumber daya dalam aplikasi.

Kontekstual

Aplikasi

> Aplikasi adalah wadah penyimpanan untuk semua pengaturan AWS End User Messaging Push Anda. Aplikasi ini juga menyimpan saluran Amazon Pinpoint, kampanye, dan pengaturan perjalanan Anda.

Prosedur

1. Buka konsol Push Pesan Pengguna AWS Akhir di https://console.aws.amazon.com/push-notifications/.
2. Pilih aplikasi dan kemudian pilih Hapus.
3. Di jendela Hapus aplikasi masukkan **`delete`** dan kemudian pilih Hapus.

> **Important**
>
> Setiap saluran, kampanye, perjalanan, atau segmen Amazon Pinpoint juga akan dihapus.

Praktik terbaik

Bahkan ketika Anda memiliki minat terbaik pelanggan, Anda mungkin masih menghadapi situasi yang berdampak pada kemampuan pengiriman pesan Anda. Bagian berikut berisi rekomendasi untuk membantu memastikan bahwa komunikasi push Anda menjangkau audiens yang dituju.

Mengirim notifikasi push volume tinggi

Sebelum Anda mengirim notifikasi push volume tinggi, pastikan akun Anda dikonfigurasi untuk mendukung persyaratan throughput Anda. Secara default, semua akun dikonfigurasi untuk mengirim 25.000 pesan per detik. Jika Anda harus dapat mengirim lebih dari 25.000 pesan dalam satu detik, Anda dapat meminta peningkatan kuota. Untuk informasi selengkapnya, lihat Kuota untuk Push Pesan Pengguna AWS Akhir.

Pastikan akun Anda dikonfigurasi dengan benar dengan kredensi untuk setiap penyedia pemberitahuan push yang akan Anda gunakan, seperti FCM atau. APNs

Akhirnya, rancang cara untuk menangani pengecualian. Setiap layanan pemberitahuan push menyediakan pesan pengecualian yang berbeda. Untuk pengiriman transaksional, Anda menerima kode status utama 200 untuk panggilan API, dengan kode status per titik akhir 400 kegagalan permanen jika token platform yang sesuai (misalnya, FCM) atau sertifikat (misalnya, APN) ditentukan tidak valid selama pengiriman pesan.

Keamanan di AWS End User Messaging Push

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS End User Messaging Push, lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS End User Messaging Push. Topik berikut menunjukkan cara mengonfigurasi AWS End User Messaging Push untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Push Pesan Pengguna AWS Akhir Anda.

Topik

Perlindungan data di AWS End User Messaging Push

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di AWS End User Messaging Push. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS End User Messaging Push atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan

untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi data

AWS Data End User Messaging Push dienkripsi saat transit dan saat istirahat. Ketika Anda mengirimkan data ke AWS End User Messaging Push, itu mengenkripsi data saat menerima dan menyimpannya. Saat Anda mengambil data dari AWS End User Messaging Push, data akan ditransmisikan kepada Anda dengan menggunakan protokol keamanan saat ini.

Enkripsi diam

AWS End User Messaging Push mengenkripsi semua data yang disimpan untuk Anda. Ini termasuk data konfigurasi, data pengguna dan titik akhir, data analitik, dan data apa pun yang Anda tambahkan atau impor ke AWS End User Messaging Push. Untuk mengenkripsi data Anda, AWS End User Messaging Push menggunakan kunci internal AWS Key Management Service (AWS KMS) yang dimiliki dan dikelola oleh layanan atas nama Anda. Kami memutar tombol-tombol ini secara teratur. Untuk selengkapnya AWS KMS, lihat Panduan AWS Key Management Service Pengembang.

Enkripsi bergerak

AWS End User Messaging Push menggunakan HTTPS dan Transport Layer Security (TLS) 1.2 atau yang lebih baru untuk berkomunikasi dengan klien dan aplikasi Anda. Untuk berkomunikasi dengan AWS layanan lain, AWS End User Messaging Push menggunakan HTTPS dan TLS 1.2. Selain itu, saat Anda membuat dan mengelola sumber daya AWS End User Messaging Push dengan menggunakan konsol, AWS SDK, atau AWS Command Line Interface, semua komunikasi diamankan menggunakan HTTPS dan TLS 1.2.

Manajemen kunci

Untuk mengenkripsi data Push Pesan Pengguna AWS Akhir Anda, AWS End User Messaging Push menggunakan AWS KMS kunci internal yang dimiliki dan dikelola oleh layanan atas nama Anda. Kami memutar tombol-tombol ini secara teratur. Anda tidak dapat menyediakan dan menggunakan kunci Anda sendiri AWS KMS atau lainnya untuk mengenkripsi data yang Anda simpan di AWS End User Messaging Push.

Privasi lalu lintas antar jaringan

Privasi lalu lintas internetwork mengacu pada pengamanan koneksi dan lalu lintas antara AWS End User Messaging Push dan klien dan aplikasi lokal Anda, dan antara AWS End User Messaging Push dan AWS sumber daya lain di Wilayah yang sama. AWS Fitur dan praktik berikut dapat membantu Anda memastikan privasi lalu lintas internetwork untuk AWS End User Messaging Push.

Lalu lintas antara AWS End User Messaging Push dan klien dan aplikasi lokal

Untuk membuat koneksi pribadi antara AWS End User Messaging Push dan klien dan aplikasi di jaringan lokal, Anda dapat menggunakannya AWS Direct Connect. Ini memungkinkan Anda untuk menghubungkan jaringan Anda ke suatu AWS Direct Connect lokasi dengan menggunakan kabel Ethernet serat optik standar. Salah satu ujung kabel terhubung ke router Anda. Ujung lainnya terhubung ke AWS Direct Connect router. Untuk informasi selengkapnya, lihat Apa itu AWS Direct Connect? dalam Panduan Pengguna AWS Direct Connect .

Untuk membantu mengamankan akses ke AWS End User Messaging Push melalui publikasi APIs, sebaiknya Anda mematuhi persyaratan Push Pesan Pengguna AWS Akhir untuk panggilan API. AWS End User Messaging Push mengharuskan klien untuk menggunakan Transport Layer Security (TLS) 1.2 atau yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS), seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Sebagian besar sistem modern seperti Java 7 dan versi yang lebih baru support dengan mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal AWS Identity and Access Management (IAM) untuk AWS akun Anda. Atau, Anda dapat menggunakan AWS Security Token Service(AWS STS) untuk menghasilkan kredensyal keamanan sementara untuk menandatangani permintaan.

Lalu lintas antara AWS End User Messaging Push dan AWS sumber daya lainnya

Untuk mengamankan komunikasi antara AWS End User Messaging Push dan AWS sumber daya lainnya di AWS Wilayah yang sama, AWS End User Messaging Push menggunakan HTTPS dan TLS 1.2 secara default.

Manajemen identitas dan akses untuk AWS End User Messaging Push

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Push Pesan Pengguna AWS Akhir. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di AWS End User Messaging Push.

Pengguna layanan — Jika Anda menggunakan layanan AWS End User Messaging Push untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensyal dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur AWS End User Messaging Push untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS End User Messaging Push, lihatPemecahan Masalah AWS End User Messaging Push identitas dan akses.

Administrator layanan — Jika Anda bertanggung jawab atas sumber daya AWS End User Messaging Push di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS End User Messaging Push. Tugas Anda adalah menentukan fitur dan sumber daya AWS End User Messaging Push mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini

untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan AWS End User Messaging Push, lihatCara Kerja AWS End User Messaging Push dengan IAM.

Administrator IAM — Jika Anda seorang administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke AWS End User Messaging Push. Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir yang dapat Anda gunakan di IAM, lihat. Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensyal yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensyal Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensyal Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensyal sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensyal yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensyal sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai

proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.
  - Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensyal sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensyal sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat dilampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations

adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.

- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Cara Kerja AWS End User Messaging Push dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke AWS End User Messaging Push, pelajari fitur IAM apa saja yang tersedia untuk digunakan dengan AWS End User Messaging Push.

Fitur IAM yang dapat Anda gunakan dengan AWS End User Messaging Push

Fitur IAM	AWS Dukungan Push Pesan Pengguna Akhir
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja AWS End User Messaging Push dan AWS layanan lainnya dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk AWS End User Messaging Push

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Kebijakan berbasis sumber daya dalam AWS End User Messaging Push

Mendukung kebijakan berbasis sumber daya: Ya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke principal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Push Pesan Pengguna AWS Akhir

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Push Pesan Pengguna AWS Akhir, lihat Tindakan yang Ditentukan oleh Push Pesan Pengguna AWS Akhir di Referensi Otorisasi Layanan.

Tindakan kebijakan di AWS End User Messaging Push menggunakan awalan berikut sebelum tindakan:

```
mobiletargeting
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Sumber daya kebijakan untuk Push Pesan Pengguna AWS Akhir

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya Push Pesan Pengguna AWS Akhir dan jenisnya ARNs, lihat Sumber Daya yang Ditentukan oleh Push Pesan Pengguna AWS Akhir di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang Ditentukan oleh Push Pesan Pengguna AWS Akhir.

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Kunci kondisi kebijakan untuk AWS End User Messaging Push

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar tombol kondisi Push Pesan Pengguna AWS Akhir, lihat Kunci Kondisi untuk Push Pesan Pengguna AWS Akhir di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang Ditentukan oleh Push Pesan Pengguna AWS Akhir.

Untuk melihat contoh kebijakan berbasis identitas Push Pesan Pengguna AWS Akhir, lihat. Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

ACLs di Push Pesan Pengguna AWS Akhir

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Push Pesan Pengguna AWS Akhir

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensil sementara dengan AWS End User Messaging Push

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensil sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensyal sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensyal sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensyal sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensyal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensil sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda menghasilkan kredensyal sementara secara dinamis alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk AWS End User Messaging Push

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima

permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

Peran layanan untuk AWS End User Messaging Push

Mendukung peran layanan: Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas Push Pesan Pengguna AWS Akhir. Edit peran layanan hanya jika AWS End User Messaging Push memberikan panduan untuk melakukannya.

Peran terkait layanan untuk AWS End User Messaging Push

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk AWS End User Messaging Push

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Push Pesan Pengguna AWS Akhir. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang

mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Push Pesan Pengguna AWS Akhir, termasuk format ARNs untuk setiap jenis sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk Push Pesan Pengguna AWS Akhir di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol Push Pesan Pengguna AWS Akhir
- Mengizinkan pengguna melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Push Pesan Pengguna AWS Akhir di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber

daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.

- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan konsol Push Pesan Pengguna AWS Akhir

Untuk mengakses konsol AWS End User Messaging Push, Anda harus memiliki seperangkat izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Push Pesan Pengguna AWS Akhir di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Push Pesan Pengguna AWS Akhir, lampirkan juga kebijakan `AWSEndUserMessaging` AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat Menambah izin untuk pengguna dalam Panduan Pengguna IAM.

```
{
```

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

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI 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": "*"
        }
    ]
}
```

Pemecahan Masalah AWS End User Messaging Push identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS End User Messaging Push dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di AWS End User Messaging Push
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Push Pesan Pengguna AWS Akhir saya

Saya tidak berwenang untuk melakukan tindakan di AWS End User Messaging Push

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `mobiletargeting:`*`GetWidget`* rekaan.

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

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `mobiletargeting:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke AWS End User Messaging Push.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di AWS End User Messaging Push. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

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

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Push Pesan Pengguna AWS Akhir saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah AWS End User Messaging Push mendukung fitur-fitur ini, lihatCara Kerja AWS End User Messaging Push dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Validasi kepatuhan untuk AWS End User Messaging Push

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Kepatuhan dan Tata Kelola Keamanan – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- Referensi Layanan yang Memenuhi Syarat HIPAA — Daftar layanan yang memenuhi syarat HIPAA. Tidak semua memenuhi Layanan AWS syarat HIPAA.
- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan dalam Push AWS Pesan Pengguna Akhir

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Selain infrastruktur AWS global, AWS End User Messaging Push menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

Keamanan Infrastruktur dalam Push Pesan Pengguna AWS Akhir

Sebagai layanan terkelola, AWS End User Messaging Push dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper Amazon Web Services: Overview of Security Processes.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS End User Messaging Push melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.2 atau versi yang lebih baru. Klien juga harus mendukung suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Konfigurasi dan analisis kerentanan

Sebagai layanan terkelola, AWS End User Messaging Push dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper Amazon Web Services: Tinjauan proses

keamanan. Ini berarti bahwa AWS mengelola dan melakukan tugas dan prosedur keamanan dasar untuk mengeras, menambal, memperbarui, dan memelihara infrastruktur dasar untuk akun dan sumber daya Anda. Prosedur ini telah ditinjau dan disertifikasi oleh pihak ketiga yang sesuai.

Praktik terbaik keamanan

Gunakan akun AWS Identity and Access Management (IAM) untuk mengontrol akses ke operasi API, terutama operasi yang membuat, memodifikasi, atau menghapus sumber daya. Untuk API, sumber daya tersebut mencakup proyek, kampanye, dan perjalanan.

- Buat pengguna individu untuk setiap orang yang mengelola sumber daya, termasuk Anda sendiri. Jangan gunakan kredensi AWS root untuk mengelola sumber daya.
- Beri setiap pengguna set izin minimum yang diperlukan untuk melakukan tugas-tugasnya.
- Gunakan grup IAM untuk mengelola izin secara efektif bagi beberapa pengguna.
- Putar kredensial IAM Anda secara rutin.

Untuk informasi lebih lanjut tentang keamanan, lihatKeamanan di AWS End User Messaging Push. Untuk informasi selengkapnya tentang IAM, lihat AWS Identity and Access Management. Untuk informasi tentang praktik terbaik IAM, lihat Praktik terbaik IAM.

Memantau Push Pesan Pengguna AWS Akhir

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja AWS End User Messaging Push dan solusi AWS Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton AWS End User Messaging Push, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari EC2 instans Amazon Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari EC2 instans Amazon CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- Amazon EventBridge dapat digunakan untuk mengotomatiskan AWS layanan Anda dan merespons secara otomatis peristiwa sistem, seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Acara dari AWS layanan dikirimkan ke EventBridge dalam waktu dekat. Anda dapat menuliskan aturan sederhana untuk menunjukkan peristiwa mana yang sesuai kepentingan Anda, dan tindakan otomatis mana yang diambil ketika suatu peristiwa sesuai dengan suatu aturan. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.
- AWS CloudTrailmenangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.

Memantau Push Pesan Pengguna AWS Akhir dengan Amazon CloudWatch

Anda dapat memantau AWS End User Messaging Push menggunakan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati real-time. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

Untuk daftar metrik dan dimensi, lihat Memantau Amazon Pinpoint CloudWatch dengan di Panduan Pengguna Amazon Pinpoint.

Mencatat panggilan API Push Pesan Pengguna AWS Akhir menggunakan AWS CloudTrail

AWS End User Messaging Push terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS End User Messaging Push. CloudTrail menangkap semua panggilan API untuk AWS End User Messaging Push sebagai event. Panggilan yang diambil termasuk panggilan dari konsol AWS End User Messaging Push dan panggilan kode ke operasi AWS End User Messaging Push API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk Push Pesan Pengguna AWS Akhir. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk AWS End User Messaging Push, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

AWS Informasi Push Pesan Pengguna Akhir di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di AWS End User Messaging Push, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh

acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan peristiwa yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk AWS End User Messaging Push, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan Push Pesan Pengguna AWS Akhir dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API Push Pesan Pengguna AWS Akhir. Misalnya, panggilan ke`GetAdmChannel`, `UpdateApnsChannel` dan `GetApnsVoipChannel` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

AWS Memahami entri file log Push Pesan Pengguna Akhir

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili

permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Akses AWS End User Messaging Push menggunakan endpoint antarmuka ()AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC dan AWS End User Messaging Push. Anda dapat mengakses AWS End User Messaging Push seolah-olah berada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk AWS mengakses End User Messaging Push.

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditujukan untuk AWS End User Messaging Push.

Untuk informasi selengkapnya, lihat Akses Layanan AWS melalui AWS PrivateLink di AWS PrivateLink Panduan.

Pertimbangan untuk Push Pesan Pengguna AWS Akhir

Sebelum Anda menyiapkan titik akhir antarmuka untuk AWS End User Messaging Push, tinjau Pertimbangan dalam Panduan.AWS PrivateLink

AWS End User Messaging Push mendukung panggilan ke semua tindakan API-nya melalui titik akhir antarmuka.

Kebijakan titik akhir VPC tidak didukung untuk Push Pesan Pengguna AWS Akhir. Secara default, akses penuh ke AWS End User Messaging Push diizinkan melalui titik akhir antarmuka. Atau, Anda dapat mengaitkan grup keamanan dengan antarmuka jaringan titik akhir untuk mengontrol lalu lintas ke AWS End User Messaging Push melalui titik akhir antarmuka.

Buat titik akhir antarmuka untuk AWS End User Messaging Push

Anda dapat membuat titik AWS akhir antarmuka untuk End User Messaging Push menggunakan konsol Amazon VPC atau AWS Command Line Interface ().AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di AWS PrivateLink Panduan.

Buat endpoint antarmuka untuk AWS End User Messaging Push menggunakan nama layanan berikut:

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

Jika Anda mengaktifkan DNS pribadi untuk titik akhir antarmuka, Anda dapat membuat permintaan API ke AWS End User Messaging Push menggunakan nama DNS Regional default. Misalnya, `com.amazonaws.us-east-1.pinpoint`.

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah sumber daya IAM yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh ke AWS End User Messaging Push melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan ke AWS End User Messaging Push dari VPC Anda, lampirkan kebijakan endpoint khusus ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (Akun AWS, pengguna IAM, dan peran IAM).
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

Untuk informasi selengkapnya, lihat Mengontrol akses ke layanan menggunakan kebijakan titik akhir di Panduan AWS PrivateLink .

Contoh: Kebijakan titik akhir VPC untuk tindakan Push Pesan Pengguna AWS Akhir

Berikut ini adalah contoh kebijakan endpoint kustom. Saat Anda melampirkan kebijakan ini ke titik akhir antarmuka Anda, kebijakan ini akan memberikan akses ke tindakan Push Pesan Pengguna AWS Akhir yang terdaftar untuk semua prinsip di semua sumber daya.

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

```
    ]
}
```

Kuota untuk Push Pesan Pengguna AWS Akhir

Anda Akun AWS memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah. Anda dapat meminta peningkatan untuk beberapa kuota dan kuota lainnya tidak dapat ditingkatkan.

Untuk melihat kuota untuk AWS End User Messaging Push, buka konsol Service Quotas. Di panel navigasi, pilih layanan AWS dan pilih Amazon Pinpoint.

Akun AWS Anda memiliki kuota berikut yang terkait dengan AWS End User Messaging Push.

Sumber Daya	Kuota bawaan	Memenuhi syarat untuk kenaikan
Jumlah maksimum notifikasi push yang dapat dikirim per detik dalam kampanye	25.000 notifikasi per detik	Ya, gunakan konsol Service Quotas
Ukuran payload pesan Amazon Device Messaging (ADM)	6 KB per pesan	Tidak
Layanan Pemberitahuan Push Apple (APNs) ukuran payload pesan	4 KB per pesan	Tidak
APNs ukuran payload pesan kotak pasir	4 KB per pesan	Tidak
Ukuran payload pesan Baidu Cloud Push	4 KB per pesan	Tidak
Ukuran payload pesan Firebase Cloud Messaging (FCM)	4 KB per pesan	Tidak

Riwayat dokumen untuk Panduan Pengguna Push Pesan Pengguna AWS Akhir

Tabel berikut menjelaskan rilis dokumentasi untuk AWS End User Messaging Push.

Perubahan	Deskripsi	Tanggal
Rilis awal	Rilis awal Panduan Pengguna Push Pesan Pengguna AWS Akhir	Juli 24, 2024

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.