



ユーザーガイド

AWS エンドユーザーメッセージングプッシュ



AWS エンドユーザーメッセージングプッシュ: ユーザーガイド

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

Amazon の商標およびトレードドレスは、Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は、Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

AWS エンドユーザーメッセージングプッシュとは	1
AWS エンドユーザーメッセージングプッシュユーザーを初めてお使いですか？	1
AWS エンドユーザーメッセージングプッシュの機能	1
AWS エンドユーザーメッセージングプッシュへのアクセス	2
リージョナルな可用性	3
のセットアップ AWS アカウント	4
にサインアップする AWS アカウント	4
管理アクセスを持つユーザーを作成する	4
使用開始	7
アプリケーションの作成とプッシュチャネルの有効化	8
コンテキスト	8
前提条件	9
手順	9
プッシュチャネルの無効化	11
プッシュメッセージの送信	12
追加リソース	25
アプリケーションでのプッシュ通知の受信	26
Swift プッシュ通知の設定	26
APNs トークンの使用	26
Android プッシュ通知のセットアップ	26
Flutter プッシュ通知のセットアップ	27
React Native プッシュ通知のセットアップ	27
アプリケーションの作成	27
プッシュ通知の処理	28
アプリケーションの削除	29
コンテキスト	29
手順	29
ベストプラクティス	30
大量のプッシュ通知を送信する	30
セキュリティ	31
データ保護	32
データ暗号化	33
転送中の暗号化	33
キー管理	33

ネットワーク間トラフィックのプライバシー	34
ID およびアクセス管理	35
対象者	35
アイデンティティを使用した認証	36
ポリシーを使用したアクセスの管理	40
その方法は? AWS エンドユーザーメッセージングプッシュは と連携します IAM	42
アイデンティティベースポリシーの例	49
トラブルシューティング	53
コンプライアンス検証	55
耐障害性	57
インフラストラクチャセキュリティ	57
設定と脆弱性の分析	58
セキュリティに関するベストプラクティス	58
モニタリング	59
によるモニタリング CloudWatch	60
CloudTrail ログ	60
AWS のエンドユーザーメッセージングプッシュ情報 CloudTrail	60
AWS エンドユーザーメッセージングプッシュログファイルエントリについて	61
AWS PrivateLink	63
考慮事項	63
インターフェイスエンドポイントの作成	64
エンドポイントポリシーを作成する	64
クォータ	66
ドキュメント履歴	67
.....	lxviii

AWS エンドユーザーメッセージングプッシュとは

Note

Amazon Pinpoint のプッシュ通知機能は、AWS エンドユーザーメッセージングと呼ばれるようになりました。

AWS エンドユーザーメッセージングプッシュを使用すると、プッシュ通知チャネルを介してプッシュ通知を送信することで、アプリケーションのユーザーをエンゲージできます。Apple Push Notification Service (APNs)、Firebase Cloud Messaging (FCM)、Amazon Device Messaging (ADM)、Baidu Push をサポートしています。

トピック

- [AWS エンドユーザーメッセージングプッシュユーザーを初めてお使いですか？](#)
- [AWS エンドユーザーメッセージングプッシュの機能](#)
- [AWS エンドユーザーメッセージングプッシュへのアクセス](#)
- [リージョナルな可用性](#)

AWS エンドユーザーメッセージングプッシュユーザーを初めてお使いですか？

AWS エンドユーザーメッセージングプッシュを初めて使用する場合は、まず以下のセクションを読むことをお勧めします。

- [のセットアップ AWS アカウント](#)
- [AWS エンドユーザーメッセージングプッシュの開始方法](#)
- [アプリケーションの作成とプッシュチャネルの有効化](#)

AWS エンドユーザーメッセージングプッシュの機能

アプリケーションにプッシュ通知を送信するには、以下のプッシュ通知サービスで個別のチャネルを使用します。

- Firebase クラウドメッセージング (FCM)
- Apple プッシュ通知サービス (APNs)

Note

を使用してAPNs、 iPhones や などの iOS デバイス iPads、 および Mac ラップトップやデスクトップなどの macOS デバイスの Safari ブラウザにメッセージを送信できます。

- Baidu Cloud Push
- Amazon Device Messaging (ADM)

AWS エンドユーザーメッセージングプッシュへのアクセス

コンソール、CLIまたは のいずれかで、サービスへのアクセスを取得するさまざまな方法を簡単に説明しますAPI。

次のインターフェイスを使用して AWS 、 エンドユーザーメッセージングプッシュを管理できます。

AWS エンドユーザーメッセージングプッシュコンソール

AWS エンドユーザーメッセージングプッシュリソースを作成および管理するウェブインターフェイス。にサインアップしている場合は AWS アカウント、 から AWS エンドユーザーメッセージングプッシュコンソールにアクセスできます AWS Management Console。

AWS Command Line Interface

コマンドラインシェルのコマンドを使用して AWS サービスとやり取りします。AWS Command Line Interface は、Windows、macOSでサポートされています。の詳細については AWS CLI、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。AWS エンドユーザーメッセージングプッシュコマンドは、[AWS CLI コマンドリファレンス](#) にあります。

AWS SDKs

HTTP または 経由でリクエストを送信するAPIsのではなく、言語固有のアプリケーションを構築したいソフトウェア開発者はHTTPS、ライブラリ、サンプルコード、チュートリアル、その他のリソース AWS を提供します。これらのライブラリは、リクエストの暗号化による署名、リクエストの再試行、エラーレスポンスの処理などのタスクを自動化する基本的な機能を提供します。これらの関数は、開始をより効率的にするのに役立ちます。詳細については、「[AWSでの構築ツール](#)」を参照してください。

リージョナルな可用性

AWS エンドユーザーメッセージングプッシュは、北米、欧州、アジア、オセアニア AWS リージョンの複数のリージョンで利用できます。各リージョンで、は複数のアベイラビリティゾーン AWS を維持します。これらのアベイラビリティゾーンは物理的に相互に分離されていますが、低レイテンシーで高スループットの冗長性に優れたプライベートネットワーク接続で統合されています。これらのアベイラビリティゾーンは、レイテンシーを最小限に抑えながら、非常に高いレベルの可用性と冗長性を提供するために使用されます。

の詳細については AWS リージョン、「」の[AWS リージョン「アカウントで使用できるを指定する」](#)を参照してくださいAmazon Web Services 全般のリファレンス。AWS エンドユーザーメッセージングプッシュが現在利用可能なすべてのリージョンと各リージョンのエンドポイントのリストについては、「」の[「Amazon Pinpoint とサービスエンドポイントのエンドポイントとクォータ」](#)を参照してくださいAmazon Web Services 全般のリファレンス。Amazon Pinpoint API [AWS](#) 各リージョンで利用できるアベイラビリティゾーンの数の詳細については、「[AWS グローバルインフラストラクチャ](#)」を参照してください。

のセットアップ AWS アカウント

を使用する前に AWS エンドユーザーメッセージングプッシュでアプリにプッシュ通知を送信するには、まず を取得する必要があります AWS アカウント 十分なIAMアクセス許可を持つ。この AWS アカウント は、 の他のサービスにも使用できます。 AWS エコシステム。

トピック

- [にサインアップする AWS アカウント](#)
- [管理アクセスを持つユーザーを作成する](#)

にサインアップする AWS アカウント

をお持ちでない場合 AWS アカウントで、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. <https://portal.aws.amazon.com/billing/サインアップ> を開きます。
2. オンラインの手順に従います。

サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

にサインアップするとき AWS アカウント、 AWS アカウントのルートユーザー が作成されます。ルートユーザーはすべての にアクセスできます AWS のサービス アカウントの および リソース。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用して [ルートユーザーアクセスが必要なタスク](#) を実行してください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。 <https://aws.amazon.com/> に移動し、マイアカウント を選択すると、いつでも現在のアカウントアクティビティを表示し、アカウントを管理できます。

管理アクセスを持つユーザーを作成する

にサインアップした後 AWS アカウント、 を保護する AWS アカウントのルートユーザー、有効化 AWS IAM Identity Center、および 管理ユーザーを作成して、日常的なタスクにルートユーザーを使用しないようにします。

のセキュリティ保護 AWS アカウントのルートユーザー

1. [にサインインします。AWS Management Console](#) ルートユーザーを選択し、AWS アカウント E メールアドレス。次のページでパスワードを入力します。

ルートユーザーを使用してサインインする方法については、[「」の「ルートユーザーとしてサインインする」](#)を参照してください。AWS サインイン ユーザーガイド。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

手順については、[「の仮想MFAデバイスの有効化」](#)を参照してください。[AWS アカウントIAM ユーザーガイドのルートユーザー \(コンソール\)](#)。

管理アクセスを持つユーザーを作成する

1. IAM Identity Center を有効にします。

手順については、[「の有効化」](#)を参照してください。[AWS IAM Identity Center \(\)AWS IAM Identity Center ユーザーガイド](#)。

2. IAM Identity Center で、ユーザーに管理アクセス権を付与します。

の使用に関するチュートリアル IAM アイデンティティセンターディレクトリ ID ソースとして、[「デフォルトを使用してユーザーアクセスを設定する」](#)を参照してください。[IAM アイデンティティセンターディレクトリ \(\)AWS IAM Identity Center ユーザーガイド](#)。

管理アクセス権を持つユーザーとしてサインインする

- IAM Identity Center ユーザーでサインインするには、IAM Identity Center ユーザーの作成時に E メールアドレスに URL 送信されたサインインを使用します。

IAM Identity Center ユーザーを使用してサインインする方法については、[「へのサインイン」](#)を参照してください。[AWS の アクセスポータル](#) AWS サインイン ユーザーガイド。

追加のユーザーにアクセス権を割り当てる

1. IAM Identity Center で、最小特権のアクセス許可を適用するベストプラクティスに従うアクセス許可セットを作成します。

手順については、「」の「[アクセス許可セットの作成](#)」を参照してください。AWS IAM Identity Center ユーザーガイド。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

手順については、「」の「[グループの追加](#)」を参照してください。AWS IAM Identity Center ユーザーガイド。

AWS エンドユーザーメッセージングプッシュの開始方法

AWS エンドユーザーメッセージングプッシュをセットアップしてアプリにプッシュ通知を送信できるようにするには、まず AWS エンドユーザーメッセージングプッシュがアプリにメッセージを送信することを許可する認証情報を指定する必要があります。提供する認証情報は、使用するプッシュ通知システムによって異なります。

- Apple Push Notification Service (APN) の認証情報については、Apple [デベロッパードキュメント](#) の「[Apple から暗号化キーとキー ID を取得する](#)」および「[Apple からプロバイダー証明書を取得する](#)」を参照してください。
- Firebase コンソールから取得できる Firebase Cloud Messaging (FCM) 認証情報については、「[Firebase Cloud Messaging](#)」を参照してください。
- Baidu 認証情報については、「[Baidu](#)」を参照してください。
- Amazon Device Messaging (ADM) 認証情報については、「[認証情報の取得](#)」を参照してください。

アプリケーションの作成とプッシュチャネルの有効化

AWS エンドユーザーメッセージングプッシュを使用してプッシュ通知を送信する前に、まずアプリケーションを作成し、プッシュ通知チャネルを有効にする必要があります。

コンテキスト

アプリケーション

アプリケーションは、すべての AWS エンドユーザーメッセージングプッシュ設定のストレージコンテナです。このアプリケーションには、Amazon Pinpoint のチャネル、キャンペーン、ジャーニーの設定も保存されます。

キー

AWS エンドユーザーメッセージングプッシュが APNs 認証トークンに暗号で署名するために使用するプライベート署名キー。この署名キーは Apple 開発者アカウントから取得できます。

署名キーを指定すると、AWS エンドユーザーメッセージングプッシュはトークンを使用して、送信するプッシュ通知 APNs ごとにで認証します。署名キーを使用すると、本 APNs 番稼働用環境とサンドボックス環境にプッシュ通知を送信できます。

証明書とは異なり、署名キーが期限切れになることはありません。1 回のみキーを指定すれば、後で更新する必要はありません。複数のアプリに対して同じ署名キーを使用できます。詳細については、[「Xcode ヘルプ」の「認証トークン APNs を使用してと通信する」](#)を参照してください。

証明書

プッシュ通知の送信 APNs 時に AWS エンドユーザーメッセージングプッシュが認証に使用する TLS 証明書。APNs 証明書は、本番稼働環境とサンドボックス環境の両方をサポートすることも、サンドボックス環境のみをサポートすることもできます。証明書は Apple 開発者アカウントから取得できます。

証明書は 1 年後に期限切れになります。この場合、新しい証明書を作成し、AWS エンドユーザーメッセージングプッシュに提供してプッシュ通知配信を更新する必要があります。詳細については、「Xcode ヘルプ」の[TLS 「証明書 APNs を使用してと通信する」](#)を参照してください。

前提条件

プッシュチャネルを使用するには、プッシュサービスに有効な認証情報が必要です。認証情報の取得の詳細については、「」を参照してください[AWS エンドユーザーメッセージングプッシュの開始方法](#)。

手順

アプリケーションを作成し、プッシュチャネルのいずれかを有効にするには、次の手順に従ってください。この手順を完了するには、アプリケーション名を入力するだけで済みます。プッシュチャネルは後で有効または無効にできます。

1. で AWS エンドユーザーメッセージングプッシュコンソールを開きます <https://console.aws.amazon.com/push-notifications/>。
2. [Create application] を選択します。
3. アプリケーション名 にアプリケーションの名前を入力します。
4. (オプション) このオプションのステップに従って、Apple プッシュ通知サービス (APNs) を有効にします。
 - a. Apple Push Notification Service (APNs) で、 を有効にするを選択します。
 - b. デフォルトの認証タイプ で、次のいずれかを選択します。
 - i. キー認証情報 を選択した場合は、Apple デベロッパーアカウントから次の情報を入力します。AWS エンドユーザーメッセージングプッシュでは、認証トークンを構築するためにこの情報が必要です。
 - [Key ID] – 署名キーに割り当てられた ID。
 - [Bundle identifier] – iOS アプリケーションに割り当てられた ID。
 - [Team identifier] – Apple デベロッパーアカウントチームに割り当てられた ID。
 - [Authentication key] – 認証キーを作成するときに Apple デベロッパーアカウントからダウンロードする .p8 ファイル。
 - ii. [Certificate credentials] を選択した場合は、次の情報を入力します。
 - SSL certificate – TLS証明書の .p12 ファイル。
 - Certificate password – 証明書にパスワードを指定している場合は、そのパスワードをここに入力します。

- [証明書タイプ] – 使用する証明書の種類を選択します。
5. (オプション) このオプションのステップに従って、Firebase Cloud Messaging (FCM) を有効にします。
 - a. Firebase Cloud Messaging (FCM) で、 を有効にするを選択します。
 - b. デフォルト認証タイプでは、次のいずれかを選択します。
 - i. トークン認証情報 (推奨) では、ファイルを選択 を選択し、サービスJSONファイルを選択します。
 - ii. キー認証情報には、キー にAPIキーを入力します。
 6. (オプション) このオプションのステップに従って、Baidu Cloud Push を有効にします。
 - a. Baidu Cloud Push で、 を有効にするを選択します。
 - b. API キーには、APIキーを入力します。
 - c. シークレットキーには、シークレットキーを入力します。
 7. (オプション) このオプションのステップに従って、Amazon Device Messaging を有効にします。
 - a. Amazon Device Messaging で、 を有効にするを選択します。
 - b. クライアント ID には、クライアント ID を入力します。
 - c. クライアントシークレットには、クライアントシークレットを入力します。
 8. [Create application] を選択します。

プッシュチャネルの無効化

プッシュチャネルを無効にするには、次の手順に従ってください。

1. で AWS エンドユーザーメッセージングプッシュコンソールを開きます <https://console.aws.amazon.com/push-notifications/>。
2. プッシュ認証情報を含むアプリケーションを選択します。
3. (オプション) Apple プッシュ通知サービス (APNs) の場合は、 を有効にする をクリアします。
4. (オプション) Firebase Cloud Messaging (FCM) の場合は、 を有効にする をクリアします。
5. (オプション) Baidu Cloud Push clear の場合、 を有効にします。
6. (オプション) Amazon Device Messaging の場合は、 の有効化をクリアします。
7. [変更の保存] を選択します。

メッセージを送信する

AWS エンドユーザーメッセージングプッシュAPIは、トランザクションプッシュ通知を特定のデバイス識別子に送信できます。このセクションでは、APIを使用してAWS エンドユーザーメッセージングプッシュを介してプッシュ通知を送信するために使用できる完全なコード例を示しますAWS SDK。

これらの例を使用して、AWS エンドユーザーメッセージングプッシュがサポートするプッシュ通知サービスを介してプッシュ通知を送信できます。現在、AWS エンドユーザーメッセージングプッシュは、Firebase Cloud Messaging (FCM)、Apple Push Notification Service ()、Baidu Cloud Push、Amazon Device Messaging (APNs) の各チャネルをサポートしていますADM。

エンドポイント、セグメント、チャネルのコード例については、[「コード例」](#)を参照してください。

Note

Firebase Cloud Messaging (FCM) サービスを介してプッシュ通知を送信する場合は、AWS エンドユーザーメッセージングプッシュへの呼び出しGCMでサービス名を使用しますAPI。Google Cloud Messaging (GCM) サービスは、2018年4月10日にGoogleによって廃止されました。ただし、AWS エンドユーザーメッセージングプッシュは、GCMサービスの中止前に書き込まれたAPIコードとの互換性を維持するために、FCMサービスを介して送信されるメッセージにGCMサービス名APIを使用します。

GCM (AWS CLI)

次の例では、[send-messages](#) を使用してGCMプッシュ通知を送信しますAWS CLI。置換 *token* デバイスの一意のトークンと *611e3e3cdd47474c9c1399a50example* をアプリケーション識別子で指定します。

```
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"
  }
}
}
```

次の例では、[send-messages](#) を使用して、ですべてのレガシーキーを使用してGCM プッシュ通知を送信します AWS CLI。置換 *token* デバイスの一意のトークンと *611e3e3cdd47474c9c1399a50example* をアプリケーション識別子で指定します。

```
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 }, \n \"data\": {\"message\": \"hello in data\"} }\",
      "TimeToLive" : 309744
    }
  },
  "Addresses": {
    "token": {
      "ChannelType": "GCM"
    }
  }
}'
\ --region us-east-1
```

次の例では、[send-messages](#) を使用して、を使用してFCMv1メッセージペイロードを含むGCMプッシュ通知を送信します AWS CLI。置換 *token* デバイスの一意のトークンと *611e3e3cdd47474c9c1399a50example* をアプリケーション識別子で指定します。

```
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
```

に `ImageUrl` フィールドを使用する場合、Pinpoint GCM は フィールドをデータ通知として送信します。キーは `imageUrl` です。これにより `pinpoint.notification.imageUrl`、イメージがすぐにレンダリングされない可能性があります。アプリを と統合するなど、データキーの処理を使用する `RawContent` が、追加してください AWS Amplify。

Safari (AWS CLI)

AWS エンドユーザーメッセージングプッシュを使用して、Apple の Safari ウェブブラウザを使用する macOS コンピュータにメッセージを送信できます。Safari ブラウザにメッセージを送信するには、Raw メッセージの内容を指定し、メッセージのペイロードに特定の属性を含める必要があります。これを行うには、[raw メッセージペイロードを使用してプッシュ通知テンプレートを作成するか、Amazon Pinpoint ユーザーガイド Amazon Pinpoint のキャンペーンメッセージで raw メッセージの内容を直接指定します。](#)

Note

この特別な属性は、Safari ウェブブラウザを使用する macOS ラップトップおよびデスクトップコンピュータに送信するために必要です。iPhones や などの iOS デバイスへの送信には必要ありません iPads。

Safari ウェブブラウザにメッセージを送信するには、Raw メッセージペイロードを指定する必要があります。Raw メッセージのペイロードは、`aps` オブジェクト内に `url-args` 配列を含む必要があります。`url-args` 配列は、Safari ウェブブラウザにプッシュ通知を送信するために必要です。ただし、配列に空の要素が 1 つ含まれていてもかまいません。

次の例では、[send-messages](#) を使用して、を使用して Safari ウェブブラウザに通知を送信します AWS CLI。置換 *token* デバイスの一意のトークンと *611e3e3cdd47474c9c1399a50example* をアプリケーション識別子で指定します。

```
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
```

Safari のプッシュ通知について詳しくは、『Apple デベロッパーウェブサイト』の「[Configuring Safari Push Notifications](#)」をご覧ください。

APNS (AWS CLI)

次の例では、[send-messages](#) を使用して で APNS プッシュ通知を送信します AWS CLI。置換 *token* デバイスの一意のトークン、*611e3e3cdd47474c9c1399a50example* アプリケーション識別子、および *GAME_INVITATION* 一意の識別子を持つ。

```
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)

この例を使用して、Node.js AWS SDK JavaScript の のを使用してプッシュ通知を送信します。この例では、Node.js JavaScript で SDKの を既にインストールして設定していることを前提としています。

この例では、共有認証情報ファイルを使用して、既存の ユーザーのアクセスキーとシークレットアクセスキーを指定するものと想定しています。詳細については、「Node.js [デベロッパーガイド](#)」の「の認証情報の設定」を参照してください。AWS SDK JavaScript

```
'use strict';

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

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

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

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

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

// An object that contains the unique token of the device that you want to send
```

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

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

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

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

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

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

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

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

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

return messageRequest
}

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

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

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



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

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

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

SendMessage()
```

Python

AWS SDK for Python (Boto3)を使用してプッシュ通知を送信するには、この例を使用します。この例では、SDK for Python (Boto3) が既にインストールされ、設定されていることを前提としています。

この例では、共有認証情報ファイルを使用して、既存のユーザーのアクセスキーとシークレットアクセスキーを指定するものと想定しています。詳細については、for AWS SDKPython (Boto3) APIリファレンスの[「認証情報」](#)を参照してください。

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

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

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

# The content of the push notification.
```

```
message = ("This is a sample message sent from End User Messaging Push by using the  
"  
          "AWS SDK for Python (Boto3).")  
  
# The application ID to use when you send this message.  
# Make sure that the push channel is enabled for the project or application  
# that you choose.  
application_id = "ce796be37f32f178af652b26eexample"  
  
# A dictionary that contains the unique token of the device that you want to send  
# the  
# message to, and the push service that you want to use to send the message.  
recipient = {  
    "token": "a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8t9u0v1w2x3y4z5a6b7c8d8e9f0",  
    "service": "GCM"  
}  
  
# The action that should occur when the recipient taps the message. Possible  
# values are OPEN_APP (opens the app or brings it to the foreground),  
# DEEP_LINK (opens the app to a specific page or interface), or URL (opens a  
# specific URL in the device's web browser.)  
action = "URL"  
  
# This value is only required if you use the URL action. This variable contains  
# the URL that opens in the recipient's web browser.  
url = "https://www.example.com"  
  
# The priority of the push notification. If the value is 'normal', then the  
# delivery of the message is optimized for battery usage on the recipient's  
# device, and could be delayed. If the value is 'high', then the notification is  
# sent immediately, and might wake a sleeping device.  
priority = "normal"  
  
# The amount of time, in seconds, that the push notification service provider  
# (such as FCM or APNS) should attempt to deliver the message before dropping  
# it. Not all providers allow you specify a TTL value.  
ttl = 30  
  
# Boolean that specifies whether the notification is sent as a silent  
# notification (a notification that doesn't display on the recipient's device).  
silent = False  
  
# Set the MessageType based on the values in the recipient variable.  
def create_message_request():
```

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

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

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

return message_request

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

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

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

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

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

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

send_message()
```

追加リソース

- プッシュチャネルテンプレートの詳細については、「Amazon Pinpoint [ユーザーガイド](#)」の「[プッシュ通知テンプレートの作成](#)」を参照してください。 Amazon Pinpoint

アプリケーションでのプッシュ通知の受信

以下のトピックでは、Swift、Android、React Native、または Flutter アプリを変更してプッシュ通知を受信する方法について説明します。

トピック

- [Swift プッシュ通知の設定](#)
- [Android プッシュ通知のセットアップ](#)
- [Flutter プッシュ通知のセットアップ](#)
- [React Native プッシュ通知のセットアップ](#)
- [AWS エンドユーザーメッセージングプッシュでアプリケーションを作成する](#)
- [プッシュ通知の処理](#)

Swift プッシュ通知の設定

iOS アプリのプッシュ通知は、Apple プッシュ通知サービス () を使用して送信されます APNs。iOS デバイスにプッシュ通知を送信するには、Apple 開発者ポータルでアプリ ID を作成する必要があります。必要な証明書を作成する必要があります。これらの手順の完了の詳細については、AWS Amplify ドキュメントの「[プッシュ通知サービスのセットアップ](#)」を参照してください。

APNs トークンの使用

ベストプラクティスとして、アプリケーションの再インストール時に顧客のデバイストークンが再生成されるようにアプリケーションを開発する必要があります。

受信者がデバイスを新しいメジャーバージョンの iOS (iOS 12 から iOS 13 など) にアップグレードし、後でアプリを再インストールした場合、アプリケーションにより新しいトークンが生成されます。アプリケーションによりトークンが更新されない場合、古いトークンを使用して通知が送信されます。その結果、トークンが無効になったため、Apple Push Notification Service (APNs) は通知を拒否します。通知を送信しようとする、 からメッセージ失敗通知を受け取ります APNs。

Android プッシュ通知のセットアップ

Android アプリケーションのプッシュ通知は、Google Cloud Messaging (FCM) に代わる Firebase Cloud Messaging () を使用して送信されます GCM。Android デバイスにプッシュ通知を送信する前

に、FCM 認証情報を取得する必要があります。その後それらの認証情報により、Android プロジェクトを作成し、プッシュ通知を受け取るサンプルアプリを起動することができます。これらのステップの完了の詳細については、AWS Amplify ドキュメントの「[プッシュ通知](#)」セクションを参照してください。

Flutter プッシュ通知のセットアップ

Flutter アプリケーションのプッシュ通知は、Android の場合は Firebase Cloud Messaging (FCM)、iOS APNs の場合は を使用して送信されます。これらのステップを完了する方法の詳細については、[AWS Amplify Flutter ドキュメント](#) の「Push notifications」のセクションを参照してください。

React Native プッシュ通知のセットアップ

React Native アプリケーションのプッシュ通知は、Android の場合は Firebase Cloud Messaging (FCM)、iOS APNs の場合は を使用して送信されます。これらの手順の完了の詳細については、[AWS Amplify JavaScript](#) ドキュメントの「プッシュ通知」セクションを参照してください。

AWS エンドユーザーメッセージングプッシュでアプリケーションを作成する

AWS エンドユーザーメッセージングプッシュでプッシュ通知の送信を開始するには、アプリケーションを作成する必要があります。次に、適切な認証情報を入力して、使用するプッシュ通知チャンネルを有効にする必要があります。

AWS エンドユーザーメッセージングプッシュコンソールを使用して、新しいアプリケーションを作成し、プッシュ通知チャンネルを設定できます。詳細については、「[アプリケーションの作成とプッシュチャンネルの有効化](#)」を参照してください。

、[API](#)、[AWS SDK](#)または [AWS Command Line Interface](#) () を使用してアプリケーションを作成およびセットアップすることもできますAWS CLI。アプリケーションを作成するには、Apps リソースを使用します。プッシュ通知チャンネルを設定するには、次のリソースを使用してください。

- Apple Push Notification サービスを使用して iOS デバイスのユーザーにメッセージを送信する [APNs チャンネル](#)。
- Amazon Kindle Fire デバイスのユーザーにメッセージを送信する [ADM チャンネル](#)。
- Baidu ユーザーにメッセージを送信する [Baidu チャンネル](#)。

- Firebase Cloud Messaging (FCM) を使用して Android デバイスにメッセージを送信する [GCMチャネル](#)。これは Google Cloud Messaging () を置き換えますGCM。

プッシュ通知の処理

プッシュ通知の送信に必要な認証情報を取得したら、プッシュ通知を受信できるようにアプリケーションを更新できます。詳細については、AWS Amplify ドキュメントの「[プッシュ通知 - 開始方法](#)」を参照してください。

アプリケーションの削除

この手順では、アカウントとアプリケーション内のすべてのリソースからアプリケーションを削除します。

コンテキスト

アプリケーション

アプリケーションは、すべての AWS エンドユーザーメッセージングプッシュ設定のストレージコンテナです。このアプリケーションには、Amazon Pinpoint のチャンネル、キャンペーン、ジャーニー設定も保存されます。

手順

1. で AWS エンドユーザーメッセージングプッシュコンソールを開きます <https://console.aws.amazon.com/push-notifications/>。
2. アプリケーションを選択し、削除を選択します。
3. 「アプリケーションの削除」ウィンドウで「」と入力し **delete**、「の削除」を選択します。

Important

Amazon Pinpoint のチャンネル、キャンペーン、ジャーニー、セグメントもすべて削除されます。

ベストプラクティス

お客様の利益を最優先にしておりますが、メッセージの配信性能に影響するような状況が発生する場合があります。次のセクションでは、プッシュメッセージを目的のユーザーに確実に届けるための推奨事項について説明します。

大量のプッシュ通知を送信する

大量のプッシュ通知を送信する前に、スループット要件をサポートするようにアカウントが設定されていることを確認してください。デフォルトでは、すべてのアカウントは 1 秒あたり 25,000 メッセージを送信するように設定されています。1 秒間に 25,000 通以上のメッセージを送信できるようにする必要がある場合は、クォータの増加をリクエストすることができます。詳細については、「[AWS エンドユーザーメッセージングプッシュのクォータ](#)」を参照してください。

アカウントが、FCMや など、使用する予定の各プッシュ通知プロバイダーの認証情報で正しく設定されていることを確認しますAPNs。

最後に、例外を処理する方法を検討します。プッシュ通知サービスごとに、異なる例外メッセージが用意されています。トランザクション送信の場合、メッセージ送信中に対応するプラットフォームトークン (例: FCM) または証明書 (例:) が無効APNであると判断された場合、API呼び出しのメインステータスコード 200、エンドポイントごとのステータスコード 400 の永続的失敗を受け取りません。

セキュリティ AWS エンドユーザーメッセージングプッシュ

でのクラウドセキュリティ AWS が最優先事項です。として AWS のお客様は、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、間で共有される責任です。AWS とユーザー。[責任共有モデル](#)では、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – AWS は、 が実行するインフラストラクチャを保護する責任を負います。AWS の サービス AWS クラウド. AWS は、安全に使用できる サービスも提供します。サードパーティーの監査者は、 の一環として、当社のセキュリティの有効性を定期的にテストおよび検証します。[AWS コンプライアンスプログラム](#)。に適用されるコンプライアンスプログラムについて学ぶには AWS エンドユーザーメッセージングプッシュ、「」を参照してください。[AWS コンプライアンスプログラムによる対象範囲内のサービス](#)。
- クラウドのセキュリティ — お客様の責任は によって決まります。AWS 使用する サービス。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、 の使用時に責任共有モデルを適用する方法を理解するのに役立ちます。AWS エンドユーザーメッセージングプッシュ。以下のトピックでは、 を設定する方法を示します。AWS エンドユーザーメッセージングプッシュでセキュリティとコンプライアンスの目標を達成します。また、他の の使用方法についても説明します。AWS のモニタリングと保護に役立つ のサービス AWS エンドユーザーメッセージングプッシュリソース。

トピック

- [でのデータ保護 AWS エンドユーザーメッセージングプッシュ](#)
- [の Identity and Access Management AWS エンドユーザーメッセージングプッシュ](#)
- [コンプライアンス検証 AWS エンドユーザーメッセージングプッシュ](#)
- [の耐障害性 AWS エンドユーザーメッセージングプッシュ](#)
- [のインフラストラクチャセキュリティ AWS エンドユーザーメッセージングプッシュ](#)
- [設定と脆弱性の分析](#)
- [セキュリティに関するベストプラクティス](#)

でのデータ保護 AWS エンドユーザーメッセージングプッシュ

-AWS [責任共有モデル](#)、でのデータ保護に適用されます。AWS エンドユーザーメッセージングプッシュ。このモデルで説明されているように、AWS は、すべての を実行するグローバルインフラストラクチャを保護する責任があります。AWS クラウド。お客様は、このインフラストラクチャでホストされているコンテンツの制御を維持する責任があります。また、 のセキュリティ設定と管理タスクについても責任を負います。AWS のサービス 使用する。データプライバシーの詳細については、「[データプライバシーFAQ](#)」を参照してください。欧州でのデータ保護の詳細については、「」を参照してください。[AWS の責任共有モデルとGDPR](#) ブログ記事 AWS セキュリティブログ。

データ保護の目的で、 を保護することをお勧めします。AWS アカウント 認証情報と を使用して個々のユーザーをセットアップする AWS IAM Identity Center または AWS Identity and Access Management (IAM)。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して と通信する AWS リソースの使用料金を見積もることができます。1TLS.2 が 必要で、1.3 TLS をお勧めします。
- で APIとユーザーアクティビティのログ記録を設定する AWS CloudTrail。CloudTrail 証跡を使用してキャプチャする方法については、「」を参照してください。AWS アクティビティ、「」の「[証 CloudTrail 跡の使用](#)」を参照してください。AWS CloudTrail ユーザーガイド。
- 使用アイテム AWS 暗号化ソリューションと 内のすべてのデフォルトのセキュリティコントロール AWS のサービス。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- アクセス時に FIPS 140-3 検証済みの暗号化モジュールが必要な場合 AWS コマンドラインインターフェイスまたは を介してAPI、FIPSエンドポイントを使用します。利用可能なFIPSエンドポイントの詳細については、「[連邦情報処理規格 \(FIPS\) 140-3](#)」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これには、 で作業する場合も含まれます。AWS エンドユーザーメッセージングプッシュまたはその他の AWS のサービス コンソール、API、AWS CLI、または AWS SDKs。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。URL を外部サーバーに提供する

場合は、そのサーバーへのリクエストを検証URLするために認証情報を に含めないことを強くお勧めします。

データ暗号化

AWS エンドユーザーメッセージングプッシュデータは、転送中および保管中に暗号化されます。にデータを送信する場合 AWS エンドユーザーメッセージングプッシュは、データを受信して保存するときにデータを暗号化します。からデータを取得する場合 AWS エンドユーザーメッセージングプッシュでは、現在のセキュリティプロトコルを使用してデータを送信します。

保管中の暗号化

AWS エンドユーザーメッセージングプッシュは、保存されているすべてのデータを暗号化します。これには、設定データ、ユーザーおよびエンドポイントデータ、分析データ、および に追加またはインポートするデータが含まれます。AWS エンドユーザーメッセージングプッシュ。データを暗号化するには、AWS エンドユーザーメッセージングプッシュは内部 を使用します AWS Key Management Service (AWS KMS) サービスがユーザーに代わって所有および維持するキー。これらのキーは定期的に更新されます。参考情報 AWS KMS、「」を参照してください。 [AWS Key Management Service デベロッパーガイド](#)。

転送中の暗号化

AWS エンドユーザーメッセージングプッシュは、HTTPSおよび Transport Layer Security (TLS) 1.2 以降を使用して、クライアントおよびアプリケーションと通信します。他の と通信するには AWS サービス、AWS エンドユーザーメッセージングプッシュは HTTPSおよび 1.2 TLS を使用します。さらに、 を作成して管理する場合 AWS エンドユーザーメッセージング コンソール、 を使用してリソースをプッシュする AWS SDK、または AWS Command Line Interface、すべての通信は HTTPS および 1.2 TLS を使用して保護されます。

キー管理

を暗号化するには AWS エンドユーザーメッセージングプッシュデータ、AWS エンドユーザーメッセージングプッシュは内部 を使用します AWS KMS ユーザーに代わってサービスが所有および維持する キー。これらのキーは定期的に更新されます。独自の をプロビジョニングして使用することはできません AWS KMS または に保存したデータを暗号化するためのその他のキー AWS エンドユーザーメッセージングプッシュ。

ネットワーク間トラフィックのプライバシー

ネットワークトラフィックのプライバシーとは、間の接続とトラフィックを保護することです。AWS エンドユーザーメッセージングプッシュとオンプレミスのクライアントとアプリケーション、および間の AWS エンドユーザーメッセージングプッシュおよびその他 AWS 同じ内のリソース AWS リージョン。以下の機能とプラクティスは、のネットワークトラフィックのプライバシーを確保するのに役立ちます。AWS エンドユーザーメッセージングプッシュ。

間のトラフィック AWS エンドユーザーメッセージングプッシュとオンプレミスのクライアントとアプリケーション

間のプライベート接続を確立するには AWS エンドユーザーメッセージングプッシュ、オンプレミスネットワーク上のクライアントとアプリケーション、を使用できます。AWS Direct Connect。これにより、ネットワークをにリンクできます。AWS Direct Connect 標準の光ファイバーサネットケーブルを使用してを配置します。ケーブルの一端はユーザーのルーターに接続します。もう一方の端はに接続されています。AWS Direct Connect ルーター。詳細については、[「とは」を参照してください](#)。AWS Direct Connectの ? AWS Direct Connect ユーザーガイド。

へのアクセスをセキュア化するには AWS エンドユーザーメッセージング が公開した を通じてプッシュするには APIs、に準拠することをお勧めします。AWS API エンドユーザーメッセージングのプッシュ要件。AWS エンドユーザーメッセージングプッシュでは、クライアントは Transport Layer Security (TLS) 1.2 以降を使用する必要があります。クライアントは、エフェメラル Diffie-Hellman (PFS) や楕円曲線 Diffie-Hellman Ephemeral (DHE) など、完全な前方秘匿性 () を持つ暗号スイートもサポートする必要があります ECDHE。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

さらに、リクエストは、に関連付けられたアクセスキー ID とシークレットアクセスキーを使用して署名する必要があります。AWS Identity and Access Management の (IAM) プリンシパル AWS アカウント。または、[AWS Security Token Service](#) (AWS STS) リクエストに署名するための一時的なセキュリティ認証情報を生成します。

間のトラフィック AWS エンドユーザーメッセージングプッシュおよびその他 AWS リソース

間の通信を保護するには AWS エンドユーザーメッセージングプッシュおよびその他 AWS 同じ内のリソース AWS リージョン、AWS エンドユーザーメッセージングプッシュは、デフォルトで HTTPS および TLS 1.2 を使用します。

の Identity and Access Management AWS エンドユーザーメッセージングプッシュ

AWS Identity and Access Management (IAM) は AWS のサービス 管理者は へのアクセスを安全に制御できます。AWS リソースの使用料金を見積もることができます。IAM 管理者は、誰を認証 (サインイン) し、誰に使用を承認する (アクセス許可を付与する) かを制御します。AWS エンドユーザーメッセージングプッシュリソース。IAM は AWS のサービス 追加料金なしで使用できます。

トピック

- [対象者](#)
- [アイデンティティを使用した認証](#)
- [ポリシーを使用したアクセスの管理](#)
- [その方法は? AWS エンドユーザーメッセージングプッシュは と連携します IAM](#)
- [のアイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ](#)
- [トラブルシューティング AWS エンドユーザーメッセージングプッシュアイデンティティとアクセス](#)

対象者

の使用方法 AWS Identity and Access Management (IAM) は、 で行う作業によって異なります。AWS エンドユーザーメッセージングプッシュ。

サービスユーザー – を使用する場合 AWS エンドユーザーメッセージングプッシュサービスでジョブを実行すると、管理者から必要な認証情報とアクセス許可が与えられます。より多くの を使用する AWS エンドユーザーメッセージングプッシュ機能を使用して作業を行うには、追加のアクセス許可が必要になる場合があります。アクセスの管理方法を理解しておく、管理者に適切な許可をリクエストするうえで役立ちます。で機能にアクセスできない場合 AWS エンドユーザーメッセージングプッシュについては、「」を参照してください [トラブルシューティング AWS エンドユーザーメッセージングプッシュアイデンティティとアクセス](#)。

サービス管理者 – を担当している場合 AWS エンドユーザーメッセージング 社内の リソースをプッシュします。通常、 へのフルアクセスがあります。AWS エンドユーザーメッセージングプッシュ。どの を決定するのは管理者の仕事です。AWS エンドユーザーメッセージング サービスユーザーがアクセスする必要のある機能とリソースをプッシュします。次に、サービスユーザーのアクセス許可を変更するリクエストをIAM管理者に送信する必要があります。このページの情報を確認し

て、の基本概念を理解してくださいIAM。会社で を と使用方法の詳細については、IAM「」を参照してください。AWS エンドユーザーメッセージングプッシュについては、「」を参照してください[その方法は? AWS エンドユーザーメッセージングプッシュは と連携します IAM。](#)

IAM 管理者 - IAM管理者は、へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。AWS エンドユーザーメッセージングプッシュ。例を表示するにはAWS エンドユーザーメッセージングで利用できるアイデンティティベースのポリシーをプッシュするにはIAM、「」を参照してください[のアイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ。](#)

アイデンティティを使用した認証

認証は、にサインインする方法です。AWS ID 認証情報を使用する。認証されている必要があります(にサインインしますAWSとして)AWS アカウントのルートユーザー、IAM ユーザーとして、またはIAMロールを引き受ける方法。

にサインインできます。AWS ID ソースを通じて提供される認証情報を使用して、フェデレーテッドIDとしてを指定します。AWS IAM Identity Center (IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーテッドIDの例です。フェデレーテッドIDとしてサインインすると、管理者は以前にIAMロールを使用してIDフェデレーションをセットアップしていました。にアクセスする場合AWS フェデレーションを使用すると、間接的にロールを引き受けることとなります。

ユーザーのタイプに応じて、にサインインできます。AWS Management Console または AWS アクセスポータル。へのサインインの詳細については、「」を参照してください。AWS「[にサインインする方法](#)」を参照してください。[AWS アカウント](#) ()AWS サインイン ユーザーガイド。

アクセスする場合AWS プログラムにより、AWS は、認証情報を使用してリクエストに暗号で署名するためのソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) を提供します。を使用しない場合AWS ツールの場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストに自分で署名する方法の詳細については、「[署名](#)」を参照してください。[AWS API IAMユーザーガイドのリクエスト。](#)

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、などですAWS では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用することをお勧めします。詳細については、「」の「[多要素認証](#)」を参照してください。AWS IAM Identity Center ユーザーガイドと [での多要素認証 \(MFA\) の使用 AWS 「](#)」(IAM ユーザーガイド) を参照してください。

AWS アカウント ルートユーザー

を作成する場合 AWS アカウントでは、すべてのへの完全なアクセス権を持つ1つのサインイン ID から始めます。AWS のサービス アカウントの および リソース。この ID は と呼ばれます。AWS アカウント root ユーザー とには、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAMユーザーガイド」の「[ルートユーザーの認証情報を必要とするタスク](#)」を参照してください。

フェデレーテッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、ID プロバイダーとのフェデレーションを使用して にアクセスすることを要求します。AWS のサービス一時的な認証情報を使用する。

フェデレーテッド ID は、エンタープライズユーザーディレクトリのユーザー、ウェブ ID プロバイダー、AWS Directory Service、Identity Center ディレクトリ、または にアクセスする任意のユーザー AWS のサービス ID ソースを通じて提供された認証情報を使用する。フェデレーテッド ID が にアクセスする場合 AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

一元的なアクセス管理を行うには、 を使用することをお勧めします。AWS IAM Identity Center。Identity Center でユーザーとグループを作成するか、独自の IAM ID ソース内のユーザーとグループのセットに接続して同期し、すべての で使用できます。AWS アカウント および アプリケーション。IAM Identity Center の詳細については、「」の [IAM 「Identity Center とは」](#) を参照してください。AWS IAM Identity Center ユーザーガイド。

IAM ユーザーとグループ

[IAM ユーザー](#) は 内のアイデンティティです。AWS アカウント 1 人のユーザーまたはアプリケーションに対して特定のアクセス許可を持つ。可能な場合は、パスワードやアクセスキーなどの長期的な認証情報を持つIAMユーザーを作成するのではなく、一時的な認証情報を使用することをお勧めします。ただし、IAMユーザーとの長期的な認証情報を必要とする特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、「[ユーザーガイド](#)」の「[長期的な認証情報を必要とするユースケースでアクセスキーを定期的にローテーションするIAM](#)」を参照してください。

[IAM グループ](#)は、IAMユーザーのコレクションを指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、 という名前のグループを作成しIAMAdmins、そのグループにIAMリソースを管理するアクセス許可を付与できます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時認証情報が提供されます。詳細については、「[ユーザーガイド](#)」の[IAM「\(ロールではなく\)ユーザーを作成する場合IAM」](#)を参照してください。

IAM ロール

[IAM ロール](#)は 内のアイデンティティです。AWS アカウント 特定のアクセス許可を持つ。これは IAM ユーザーと似ていますが、特定のユーザーに関連付けられていません。で一時的に IAMロールを引き受けることができます。AWS Management Console [ロールを切り替えます](#)。を呼び出すことでロールを引き受けることができます。AWS CLI または AWS API オペレーション、またはカスタムの使用URL。ロールの使用の詳細については、「[ユーザーガイド](#)」の[IAM「ロールの使用IAM」](#)を参照してください。

IAM 一時的な認証情報を持つ ロールは、以下の状況で役立ちます。

- フェデレーションユーザーアクセス – フェデレーテッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーテッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールの詳細については、「[ユーザーガイド](#)」の「[サードパーティー ID プロバイダーのロールの作成IAM](#)」を参照してください。IAM Identity Center を使用する場合は、アクセス許可セットを設定します。ID が認証後にアクセスできる内容を制御するために、IAM Identity Center はアクセス許可セットを のロールに関連付けますIAM。アクセス許可セットの詳細については、「[」](#)の「[アクセス許可セット](#)」を参照してください。AWS IAM Identity Center ユーザーガイド。
- 一時的なIAMユーザーアクセス許可 – IAM ユーザーまたはロールは、IAMロールを引き受けて、特定のタスクに対して異なるアクセス許可を一時的に引き受けることができます。
- クロスアカウントアクセス – IAMロールを使用して、別のアカウントのユーザー (信頼されたプリンシパル) が自分のアカウントのリソースにアクセスすることを許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、AWS のサービスでは、(ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロ

スアカウントアクセスのロールとリソースベースのポリシーの違いについては、「ユーザーガイド」の「[でのクロスアカウントリソースアクセスIAMIAM](#)」を参照してください。

- クロスサービスアクセス – 一部 AWS のサービス 他 の機能を使用する AWS のサービス。例えば、サービスで呼び出しを行うと、そのサービスが Amazon でアプリケーションを実行EC2したり、Amazon S3 にオブジェクトを保存したりするのが一般的です。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を使用します。AWS のサービス、リクエスト元との組み合わせ AWS のサービス ダウンストリームサービスにリクエストを行う。FAS リクエストは、他の とのやり取りを必要とするリクエストをサービスが受信した場合にのみ行われます。AWS のサービス または完了するリソース。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。
- サービスロール – サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受ける [IAMロール](#)です。IAM 管理者は、 内からサービスロールを作成、変更、削除できますIAM。詳細については、「[にアクセス許可を委任するロールの作成](#)」を参照してください。[AWS のサービス「」](#) (IAM ユーザーガイド) を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です。AWS のサービス。このサービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールが に表示されます。AWS アカウント とは サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。
- Amazon で実行されているアプリケーション EC2 – IAMロールを使用して、EC2インスタンスで実行され、 を作成しているアプリケーションの一時的な認証情報を管理できます。AWS CLI または AWS API リクエスト。これは、EC2インスタンス内にアクセスキーを保存するよりも望ましいです。 を割り当てるには AWS ロールをEC2インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルには ロールが含まれており、EC2インスタンスで実行されているプログラムが一時的な認証情報を取得できるようにします。詳細については、「ユーザーガイド」の「[IAMロールを使用して Amazon EC2インスタンスで実行されているアプリケーションにアクセス許可を付与するIAM](#)」を参照してください。

IAM ロールとIAMユーザーのどちらを使用するかについては、「[ユーザーガイド](#)」の「[IAMロールを作成するタイミング \(ユーザーではなく \) IAM](#)」を参照してください。

ポリシーを使用したアクセスの管理

でアクセスをコントロールする AWS ポリシーを作成して にアタッチする AWS ID またはリソース。ポリシーは のオブジェクトです。AWS ID またはリソースに関連付けられている場合、そのアクセス許可を定義します。AWS は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシーを評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは に保存されます。AWS JSON ドキュメントとして。JSON ポリシードキュメントの構造と内容の詳細については、「[ユーザーガイド](#)」の[JSON「ポリシーの概要IAM](#)」を参照してください。

管理者は を使用できます AWS JSON ポリシー。誰が何にアクセスできるかを指定します。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するために、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行に使用するメソッドに関係なく、アクションのアクセス許可を定義します。例えば、iam:GetRoleアクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 からロール情報を取得できます。AWS Management Console、AWS CLI、または AWS API。

アイデンティティベースのポリシー

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできる JSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、「[ユーザーガイド](#)」の[IAM「ポリシーの作成IAM](#)」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです。AWS アカウント。管理ポリシーには以下が含まれます。AWS 管理ポリシーとカスタマー管理ポリシー。管理ポリシーとインラインポリシーのどちらかを選択する方法については、「[IAMユーザーガイド](#)」の「[管理ポリシーとインラインポリシーの選択](#)」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースにアタッチするJSONポリシードキュメントです。リソースベースのポリシーの例としては、IAMロールの信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、またはAWSのサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。は使用できませんAWS リソースベースのポリシーIAMのからの マネージドポリシー。

アクセスコントロールリスト (ACLs)

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLs はリソースベースのポリシーに似ていますが、JSONポリシードキュメント形式を使用しません。

Amazon S3、AWS WAF、および Amazon VPCは、をサポートするのサービスの例ですACLs。の詳細についてはACLs、Amazon Simple Storage Service デベロッパーガイドの「[アクセスコントロールリスト \(ACL\) の概要](#)」を参照してください。

その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- **アクセス許可の境界** – アクセス許可の境界は、アイデンティティベースのポリシーがIAMエンティティ (IAMユーザーまたはロール) に付与できるアクセス許可の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。Principal フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAMユーザーガイド」の「[IAMエンティティのアクセス許可の境界](#)」を参照してください。
- **サービスコントロールポリシー (SCPs)** – SCPsは、の組織または組織単位 (OU) の最大アクセス許可を指定するJSONポリシーです。AWS Organizations. AWS Organizations は、複数のを

グループ化して一元管理するためのサービスです。AWS アカウント お客様のビジネスが所有する。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCPs) をアカウントの一部またはすべてに適用できます。は、各 を含むメンバーアカウントのエンティティのアクセス許可SCPを制限します。AWS アカウントのルートユーザー。Organizations との詳細についてはSCPs、「」の「[サービスコントロールポリシー](#)」を参照してください。AWS Organizations ユーザーガイド。

- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「ユーザーガイド」の「[セッションポリシーIAM](#)」を参照してください。

複数のポリシータイプ

1つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。方法を学ぶには AWS は、複数のポリシータイプが関与する場合にリクエストを許可するかどうかを決定します。「ユーザーガイド」の「[ポリシー評価ロジックIAM](#)」を参照してください。

その方法は? AWS エンドユーザーメッセージングプッシュは と連携します IAM

IAM を使用して へのアクセスを管理する前に AWS エンドユーザーメッセージングプッシュ、 で使用できるIAM機能について説明します AWS エンドユーザーメッセージングプッシュ。

IAM で使用できる の機能 AWS エンドユーザーメッセージングプッシュ

IAM 機能	AWS エンドユーザーメッセージングプッシュのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	あり
ポリシーアクション	あり

IAM 機能	AWS エンドユーザーメッセージングプッシュのサポート
ポリシーリソース	Yes
ポリシー条件キー	あり
ACLs	なし
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	あり
プリンシパル権限	あり
サービスロール	あり
サービスリンクロール	なし

方法の概要を把握するには AWS エンドユーザーメッセージングプッシュおよびその他 AWS サービスはほとんどのIAM機能で動作します。「」を参照してください。[AWS ユーザーガイドIAM](#)の「と連携する IAM のサービス」。

のアイデンティティベースのポリシー AWS エンドユーザーメッセージングプッシュ

アイデンティティベースのポリシーのサポート: あり

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできる JSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、「ユーザーガイド」の[IAM「ポリシーの作成IAM」](#)を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否されたアクションとリソース、およびアクションが許可または拒否される条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「ユーザーガイド」の「[IAMJSONポリシー要素のリファレンスIAM](#)」を参照してください。

のアイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ

の例を表示するには AWS エンドユーザーメッセージングプッシュアイデンティティベースのポリシーについては、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ](#)。

内のリソースベースのポリシー AWS エンドユーザーメッセージングプッシュ

リソースベースのポリシーのサポート: はい

リソースベースのポリシーは、リソースにアタッチするJSONポリシードキュメントです。リソースベースのポリシーの例としては、IAMロールの信頼ポリシー や Amazon S3 バケットポリシー などがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーテッドユーザー、または AWS のサービス。

クロスアカウントアクセスを有効にするには、リソースベースのポリシーのプリンシパルとして、アカウント全体または別のアカウントのIAMエンティティを指定できます。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる場合 AWS アカウント、信頼されたアカウントのIAM管理者は、リソースにアクセスするためのアクセス許可をプリンシパルエンティティ (ユーザーまたはロール) に付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、「[ユーザーガイド](#)」の「[でのクロスアカウントリソースアクセスIAMIAM](#)」を参照してください。

のポリシーアクション AWS エンドユーザーメッセージングプッシュ

ポリシーアクションのサポート: あり

管理者は を使用できます AWS JSON ポリシー。誰が何にアクセスできるかを指定します。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action要素は、ポリシーでアクセスを許可または拒否するために使用できるアクションを記述します。ポリシーアクションは通常、関連付けられていると同じ名前です。AWS API オペレーション。一致するAPIオペレーションを持たないアクセス許可のみのアクションなど、いく

つかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

のリストを表示するには AWS エンドユーザーメッセージングプッシュアクション、「」で定義される [アクション](#) を参照してください。AWS「サービス認証リファレンス」の「エンドユーザーメッセージングプッシュ」。

でのポリシーアクション AWS エンドユーザーメッセージングプッシュは、アクションの前に次のプレフィックスを使用します。

```
mobiletargeting
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

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

の例を表示するには AWS エンドユーザーメッセージングプッシュアイデンティティベースのポリシーについては、「」を参照してくださいの [アイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ](#)。

のポリシーリソース AWS エンドユーザーメッセージングプッシュ

ポリシーリソースのサポート: あり

管理者は を使用できます AWS JSON ポリシー。誰が何にアクセスできるかを指定します。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Policy ResourceJSON要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、Resource または NotResource 要素を含める必要があります。ベストプラクティスとし

て、[Amazon リソースネーム \(ARN\) を使用してリソース](#)を指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

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

のリストを表示するには AWS エンドユーザーメッセージングプッシュリソースタイプとその ARNs、「」で定義される[リソース](#)を参照してください。AWS「[サービス認証リファレンス](#)」の「[エンドユーザーメッセージングプッシュ](#)」。各リソースARNの を指定できるアクションについては、「」で定義される[アクション](#)を参照してください。[AWS エンドユーザーメッセージングプッシュ](#)。

の例を表示するには AWS エンドユーザーメッセージングプッシュアイデンティティベースのポリシーについては、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ](#)。

のポリシー条件キー AWS エンドユーザーメッセージングプッシュ

サービス固有のポリシー条件キーのサポート: あり

管理者は を使用できます AWS JSON ポリシー。誰が何にアクセスできるかを指定します。つまり、どのプリンシパルが、どのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの [条件演算子](#) を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

ステートメントで複数のCondition要素を指定するか、単一のCondition要素で複数のキーを指定する場合は、AWS は論理ANDオペレーションを使用してそれら进行评估します。1つの条件キーに複数の値を指定する場合は、AWS は論理ORオペレーションを使用して条件进行评估します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば、リソースにIAMユーザー名でタグ付けされている場合にのみ、リソースへのアクセス許可をIAMユーザーに付与できます。詳細については、「[ユーザーガイド](#)」の[IAM「ポリシー要素: 変数とタグIAM」](#)を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべてを表示するには AWS グローバル条件キー、「」を参照してください。[AWSIAM ユーザーガイドのグローバル条件コンテキストキー](#)。

のリストを表示するには AWS エンドユーザーメッセージングプッシュ条件キー、「[の条件キー](#)」を参照してください。[AWS「サービス認証リファレンス」の「エンドユーザーメッセージングプッシュ」](#)。条件キーを使用できるアクションとリソースについては、「で定義される[アクション](#)」を参照してください。[AWS エンドユーザーメッセージングプッシュ](#)。

の例を表示するには AWS エンドユーザーメッセージングプッシュアイデンティティベースのポリシーについては、「」を参照してください。[のアイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ](#)。

ACLs の AWS エンドユーザーメッセージングプッシュ

をサポートACLs：いいえ

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLs はリソースベースのポリシーに似ていますが、JSONポリシードキュメント形式を使用しません。

ABAC で AWS エンドユーザーメッセージングプッシュ

サポート ABAC (ポリシー内のタグ): 部分的

属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。In AWSでは、これらの属性はタグと呼ばれます。IAM エンティティ (ユーザーまたはロール) と多くの にタグをアタッチできます。AWS リソースの使用料金を見積もることができます。エンティティとリソースのタグ付けは、の最初のステップですABAC。次に、プリンシパルのタグが、アクセスしようとしているリソースのタグと一致する場合に、オペレーションを許可するABACポリシーを設計します。

ABAC は、急速に成長している環境や、ポリシー管理が煩雑になる状況に役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの [条件要素](#) でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

の詳細についてはABAC、「[ユーザーガイド](#)」の「[とはABACIAM](#)」を参照してください。のセットアップ手順を含むチュートリアルを表示するにはABAC、「[ユーザーガイド](#)」の「[属性ベースのアクセスコントロール \(ABAC\)](#)」を使用するIAM」を参照してください。

での一時的な認証情報の使用 AWS エンドユーザーメッセージングプッシュ

一時的な認証情報のサポート: あり

ある程度 AWS のサービス 一時的な認証情報を使用してサインインすると、は機能しません。以下を含む追加情報 AWS のサービス 一時的な認証情報の使用については、「」を参照してください。
[AWS のサービス ユーザーガイドの IAM](#)で動作する IAM 。

にサインインする場合、一時的な認証情報を使用している AWS Management Console ユーザー名とパスワード以外の方法を使用する。例えば、にアクセスする場合 AWS 会社のシングルサインオン (SSO) リンクを使用すると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えの詳細については、「[IAMユーザーガイド](#)」の「[ロールへの切り替え \(コンソール\)](#)」を参照してください。

を使用して、一時的な認証情報を手動で作成できます。AWS CLI または AWS API。その後、これらの一時的な認証情報を使用してにアクセスできます。AWS. AWS では、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「」の「[一時的なセキュリティ認証情報IAM](#)」を参照してください。

のクロスサービスプリンシパル許可 AWS エンドユーザーメッセージングプッシュ

転送アクセスセッションをサポート (FAS): はい

IAM ユーザーまたはロールを使用してでアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、を呼び出すプリンシパルのアクセス許可を使用します。AWS のサービス、リクエスト元との組み合わせ AWS のサービス ダウンストリームサービスにリクエストを行う。FAS リクエストは、他のとのやり取りを必要とするリクエストをサービスが受信した場合にのみ行われます。AWS のサービス または完了するリソース。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。

のサービスロール AWS エンドユーザーメッセージングプッシュ

サービスロールのサポート: あり

サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受ける [IAM ロール](#)です。IAM 管理者は、内からサービスロールを作成、変更、削除できますIAM。詳細については、「[にアクセス許可を委任するロールの作成](#)」を参照してください。AWS のサービス「[IAM ユーザーガイド](#)」を参照してください。

Warning

サービスロールのアクセス許可を変更すると、AWS エンドユーザーメッセージングプッシュ機能。サービスロールを編集するのは、次の場合のみです。AWS エンドユーザーメッセージングプッシュは、そのためのガイダンスを提供します。

のサービスにリンクされたロール AWS エンドユーザーメッセージングプッシュ

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、にリンクされたサービスロールの一種です。AWS のサービス。このサービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールがに表示されます。AWS アカウントとはサービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「」を参照してください。[AWS と連携するのサービスIAM](#)。表の中から、[Service-linked role] (サービスにリンクされたロール) 列に Yes と記載されたサービスを見つけます。サービスリンクロールに関するドキュメントをサービスで表示するには、はい リンクを選択します。

のアイデンティティベースのポリシーの例 AWS エンドユーザーメッセージングプッシュ

デフォルトでは、ユーザーとロールには作成または変更するアクセス許可がありません AWS エンドユーザーメッセージングプッシュリソース。また、を使用してタスクを実行することはできません。AWS Management Console, AWS Command Line Interface (AWS CLI)、または AWS API。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するために、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、ユーザーはロールを引き受けることができます。

これらのポリシードキュメント例を使用してIAMアイデンティティベースのJSONポリシーを作成する方法については、「ユーザーガイド」の[IAM 「ポリシーの作成IAM」](#)を参照してください。

で定義されるアクションとリソースタイプの詳細については、「」を参照してください。AWS ARNs 各リソースタイプの の形式を含むエンドユーザーメッセージングプッシュについては、「[のアクション、リソース、および条件キー](#)」を参照してください。AWS「[サービス認証リファレンス](#)」の「[エンドユーザーメッセージングプッシュ](#)」。

トピック

- [ポリシーのベストプラクティス](#)
- [以下を使用 AWS エンドユーザーメッセージングプッシュコンソール](#)
- [自分の権限の表示をユーザーに許可する](#)

ポリシーのベストプラクティス

ID ベースのポリシーは、誰かが作成、アクセス、または削除できるかどうかを決定します。AWS エンドユーザーメッセージング アカウントのリソースをプッシュします。これらのアクションでは、のコストが発生する可能性があります。AWS アカウント。アイデンティティベースのポリシーを作成または編集するときは、以下のガイドラインと推奨事項に従ってください。

- の使用を開始する AWS 管理ポリシーと最小特権のアクセス許可への移行 – ユーザーとワークロードへのアクセス許可の付与を開始するには、を使用します。AWS 多くの一般的なユースケースにアクセス許可を付与する マネージドポリシー。これらは で利用できます。AWS アカウント。を定義してアクセス許可をさらに減らすことをお勧めします。AWS ユースケースに固有の カスタマー管理ポリシー。詳細については、「」を参照してください[AWS マネージドポリシー](#)または[AWS ユーザーガイドの ジョブ機能の IAM マネージドポリシー](#)。
- 最小特権のアクセス許可を適用する – IAMポリシーでアクセス許可を設定する場合は、タスクの実行に必要なアクセス許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用してアクセス許可を適用する方法の詳細については、「[ユーザーガイド](#)」の「[のポリシーとアクセス許可IAMIAM](#)」を参照してください。
- IAM ポリシーの条件を使用してアクセスをさらに制限する – ポリシーに条件を追加して、アクションとリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを を使用して送信する必要があることを指定できますSSL。特定の を通じてサービスアクションが使用されている場合、条件を使用してサービスアクションへのアクセスを許可することもできます。AWS のサービスまたは AWS CloudFormation。詳細については、「[ユーザーガイド](#)」の[IAMJSON「ポリシー要素: 条件IAM](#)」を参照してください。
- IAM Access Analyzer を使用してIAMポリシーを検証し、安全で機能的なアクセス許可を確保する – IAM Access Analyzer は、ポリシーがポリシー言語 (JSON) とIAMベストプラクティスに準拠す

るように、新規および既存のIAMポリシーを検証します。IAM Access Analyzer には、安全で機能的なポリシーの作成に役立つ 100 を超えるポリシーチェックと実用的な推奨事項が用意されています。詳細については、「ユーザーガイド」の [IAM 「Access Analyzer ポリシーの検証IAM」](#) を参照してください。

- 多要素認証を要求する (MFA) — でIAMユーザーまたはルートユーザーを必要とするシナリオがある場合 AWS アカウントのセキュリティを強化するMFAには、 をオンにします。API オペレーションが呼び出されるMFAタイミングを要求するには、ポリシーにMFA条件を追加します。詳細については、「IAMユーザーガイド」の [MFA 「で保護されたAPIアクセスの設定」](#) を参照してください。

のベストプラクティスの詳細についてはIAM、「ユーザーガイド」の [「のセキュリティのベストプラクティスIAMIAM」](#) を参照してください。

以下を使用 AWS エンドユーザーメッセージングプッシュコンソール

にアクセスするには AWS エンドユーザーメッセージングプッシュコンソールでは、最小限のアクセス許可のセットが必要です。これらのアクセス許可により、 の詳細を一覧表示および表示できます。AWS のエンドユーザーメッセージングプッシュリソース AWS アカウント。最小限必要なアクセス許可よりも制限されたアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。AWS CLI または AWS API。代わりに、実行しようとしているAPIオペレーションに一致するアクションのみへのアクセスを許可します。

ユーザーとロールが引き続き を使用できるようにするには AWS エンドユーザーメッセージングプッシュコンソール、 もアタッチします AWSEndUserMessaging AWS エンティティへの マネージドポリシー。詳細については、「ユーザーガイド」の [「ユーザーへのアクセス許可の追加IAM」](#) を参照してください。

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

```

自分の権限の表示をユーザーに許可する

この例では、IAMユーザーがユーザー ID にアタッチされているインラインポリシーと管理ポリシーを表示できるようにするポリシーを作成する方法を示します。このポリシーには、コンソールで、またはを使用してプログラムでこのアクションを実行するアクセス許可が含まれています。AWS CLI または AWS API。

```

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

```



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

トラブルシューティング AWS エンドユーザーメッセージングプッシュアイデンティティとアクセス

次の情報は、の使用時に発生する可能性がある一般的な問題の診断と修正に役立ちます。AWS エンドユーザーメッセージングプッシュとIAM。

トピック

- [でアクションを実行する権限がない AWS エンドユーザーメッセージングプッシュ](#)
- [iam を実行する権限がありません。PassRole](#)
- [自分の 以外のユーザーに許可したい AWS アカウント にアクセスして AWS エンドユーザーメッセージングプッシュリソース](#)

でアクションを実行する権限がない AWS エンドユーザーメッセージングプッシュ

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次の例のエラーは、mateojacksonIAMユーザーがコンソールを使用して架空の*my-example-widget*リソースの詳細を表示しようとしているが、架空のmobiletargeting:*GetWidget*アクセス許可がない場合に発生します。

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

この場合、mobiletargeting:*GetWidget* アクションを使用して *my-example-widget* リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、お問い合わせください。AWS 管理者。サインイン認証情報を提供した担当者が管理者です。

iam を実行する権限がありません。PassRole

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して iam:PassRole を渡すことができるようにする必要があります。AWS エンドユーザーメッセージングプッシュ。

ある程度 AWS のサービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

次の例のエラーは、というIAMユーザーがコンソールを使用して marymajor でアクションを実行しようする場合に発生します。AWS エンドユーザーメッセージングプッシュ。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

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

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、お問い合わせください。AWS 管理者。サインイン認証情報を提供した担当者が管理者です。

自分の 以外のユーザーに許可したい AWS アカウント にアクセスして AWS エンドユーザーメッセージングプッシュリソース

他のアカウントのユーザーや組織外の人、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACLs) をサポートするサービスでは、これらのポリシーを使用して、ユーザーにリソースへのアクセスを許可できます。

詳細については、以下を参照してください。

- 以下について確認するには AWS エンドユーザーメッセージングプッシュはこれらの機能をサポートしています。「」を参照してください[その方法は? AWS エンドユーザーメッセージングプッシュは と連携します IAM](#)。
- 全体で リソースへのアクセスを提供する方法を学ぶには AWS アカウント 所有している。「別の [IAMユーザーへのアクセスの許可](#)」を参照してください。AWS アカウント ユーザーガイドで [所有している IAM](#)。
- リソースへのアクセスをサードパーティーに提供する方法を学ぶには AWS アカウント、「[へのアクセスの提供](#)」を参照してください。AWS アカウント ユーザーガイドの「[第三者が所有していますIAM](#)」。
- ID フェデレーションを通じてアクセスを提供する方法については、IAMユーザーガイドの「[外部認証されたユーザーへのアクセスの提供 \(ID フェデレーション\)](#)」を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いについては、ユーザーガイドの「[でのクロスアカウントリソースアクセスIAMIAM](#)」を参照してください。


コンプライアンス検証 AWS エンドユーザーメッセージングプッシュ

が AWS のサービス は特定のコンプライアンスプログラムの範囲内にあります。「」を参照してください。[AWS のサービス コンプライアンスプログラムによる対象範囲内](#)による対象範囲内、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、「」を参照してください。[AWS コンプライアンスプログラム](#)。

サードパーティーの監査レポートは、を使用してダウンロードできます。AWS Artifact。詳細については、「[でのレポートのダウンロード](#)」を参照してください。[AWS Artifact](#)。

を使用する際のお客様のコンプライアンス責任 AWS のサービスは、データの機密性、企業のコンプライアンス目的、適用可能な法律および規制によって決まります。AWS では、コンプライアンスに役立つ以下のリソースを提供しています。

- [セキュリティとコンプライアンスのクイックスタートガイド](#) – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、にベースライン環境をデプロイする手順について説明します。AWS セキュリティとコンプライアンスに重点を置いた。
- [アマゾン ウェブ サービスHIPAAのセキュリティとコンプライアンスのためのアーキテクチャ](#) – このホワイトペーパーでは、企業が を使用する方法について説明します。AWS は、HIPAA対象アプリケーションを作成します。

 Note

すべてではない AWS のサービス がHIPAA対象です。詳細については、[HIPAA「対象サービスリファレンス」](#)を参照してください。

- [AWS コンプライアンスリソース](#) – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- [AWS カスタマーコンプライアンスガイド](#) – コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、 を保護するためのベストプラクティスをまとめています。AWS のサービスとは、ガイダンスを複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールにマッピングします。
- [のルールによるリソースの評価](#) AWS Config デベロッパーガイド – AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。
- [AWS Security Hub](#) – これは AWS のサービスは、内のセキュリティ状態を包括的に表示します。AWS Security Hub は、セキュリティコントロールを使用して を評価します。AWS リソースとを使用して、セキュリティ業界標準とベストプラクティスに照らしてコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、[Security Hub のコントロールリファレンス](#)を参照してください。
- [Amazon GuardDuty](#) – これは AWS のサービス が に対する潜在的な脅威を検出する AWS アカウント、ワークロード、コンテナ、およびデータをモニタリングして、疑わしいアクティビティや悪意のあるアクティビティがないかを確認します。PCI GuardDuty は、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことでDSS、などのさまざまなコンプライアンス要件への対応に役立ちます。

- [AWS Audit Manager](#) – これは AWS のサービスは、の継続的な監査に役立ちます。AWS を使用して、リスクの管理方法と規制や業界標準への準拠を簡素化します。

の耐障害性 AWS エンドユーザーメッセージングプッシュ

- AWS グローバルインフラストラクチャは AWS リージョン およびアベイラビリティゾーン。AWS リージョン は、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離および隔離された複数のアベイラビリティゾーンを提供します。アベイラビリティゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケラブルです。

の詳細については、「」を参照してください。AWS リージョン およびアベイラビリティゾーンについては、「」を参照してください。 [AWS グローバルインフラストラクチャ](#)。

に加えて AWS グローバルインフラストラクチャ、AWS エンドユーザーメッセージングプッシュには、データの耐障害性とバックアップのニーズをサポートするのに役立ついくつかの機能があります。

のインフラストラクチャセキュリティ AWS エンドユーザーメッセージングプッシュ

マネージドサービスとして、AWS エンドユーザーメッセージングプッシュは によって保護されています AWS ホワイトペーパー「[Amazon Web Services: セキュリティプロセスの概要](#)」に記載されている グローバルネットワークセキュリティの手順。

を使用する AWS が にアクセスするためのAPI呼び出しを公開 AWS エンドユーザーメッセージングネットワーク経由でプッシュします。クライアントは Transport Layer Security (TLS) 1.2 以降をサポートしている必要があります。クライアントは、(Ephemeral Diffie-HellmanPFS) や DHE (Elliptic Curve Ephemeral Diffie-Hellman) などの完全前方秘匿性 ECDHE () を持つ暗号スイートもサポートする必要があります。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

さらに、リクエストは、IAMプリンシパルに関連付けられたアクセスキー ID とシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service](#) (AWS STS) リクエストに署名するための一時的なセキュリティ認証情報を生成します。

設定と脆弱性の分析

マネージドサービスとして、AWS エンドユーザーメッセージングプッシュは によって保護されています。AWS ホワイトペーパー「[Amazon Web Services: セキュリティプロセスの概要](#)」に記載されている グローバルネットワークセキュリティの手順。つまり、AWS は、基本的なセキュリティタスクと手順を管理および実行して、アカウントとリソースの基盤となるインフラストラクチャを強化、パッチ適用、更新、その他の方法で維持します。これらの手順は適切なサードパーティーによって確認され、認証されています。

セキュリティに関するベストプラクティス

使用アイテム AWS Identity and Access Management (IAM) アカウントは、API オペレーション、特に リソースを作成、変更、削除するオペレーションへのアクセスを制御します。の場合API、このようなリソースにはプロジェクト、キャンペーン、ジャーニーが含まれます。

- リソースを管理するユーザー (本人を含む) ごとに個別のユーザーを作成します。を使用しない AWS リソースを管理するためのルート認証情報。
- それぞれの職務の実行に最低限必要になる一連のアクセス許可を各ユーザーに付与します。
- IAM グループを使用して、複数のユーザーのアクセス許可を効果的に管理します。
- IAM 認証情報のローテーションを定期的に行います。

セキュリティの詳細については、「[セキュリティ AWS エンドユーザーメッセージングプッシュ](#)」を参照してください。の詳細については、IAM「」を参照してください。[AWS Identity and Access Management](#) IAM ベストプラクティスの詳細については、[IAM「のベストプラクティス](#)」を参照してください。

AWS エンドユーザーメッセージングプッシュのモニタリング

モニタリングは、AWS エンドユーザーメッセージングプッシュやその他の AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。AWS には、AWS エンドユーザーメッセージングプッシュを監視し、問題が発生したときに報告し、必要に応じて自動アクションを実行するための以下のモニタリングツールが用意されています。

- Amazon CloudWatch は、AWS リソースと、で実行しているアプリケーションを AWS リアルタイムでモニタリングします。メトリクスの収集と追跡、カスタマイズしたダッシュボードの作成、および指定したメトリクスが指定したしきい値に達したときに通知またはアクションを実行するアラームの設定を行うことができます。例えば、で Amazon EC2 インスタンスの CPU 使用状況やその他のメトリクス CloudWatch を追跡し、必要に応じて新しいインスタンスを自動的に起動できます。詳細については、[「Amazon ユーザーガイド CloudWatch」](#) を参照してください。
- Amazon CloudWatch Logs を使用すると、Amazon EC2 インスタンスやその他のソースからログファイルをモニタリング、保存 CloudTrail、およびアクセスできます。CloudWatch Logs はログファイル内の情報をモニタリングし、特定のしきい値に達したときに通知できます。高い耐久性を備えたストレージにログデータをアーカイブすることもできます。詳細については、[「Amazon CloudWatch Logs ユーザーガイド」](#) を参照してください。
- Amazon EventBridge を使用すると、AWS サービスを自動化し、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。AWS サービスからのイベントは、ほぼリアルタイムで EventBridge に配信されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。詳細については、[「Amazon ユーザーガイド EventBridge」](#) を参照してください。
- AWS CloudTrail は、AWS アカウントによって、またはアカウントに代わって行われた API 呼び出しおよび関連イベントをキャプチャし、指定した Amazon S3 バケットにログファイルを配信します。を呼び出したユーザーとアカウント AWS、呼び出し元の IP アドレス、呼び出しが発生した日時を特定できます。詳細については、[「AWS CloudTrail ユーザーガイド」](#) を参照してください。

Amazon による AWS エンドユーザーメッセージングプッシュのモニタリング CloudWatch

を使用して AWS エンドユーザーメッセージングプッシュをモニタリングできます。これにより CloudWatch、raw データを収集し、読み取り可能なほぼリアルタイムのメトリクスに処理します。これらの統計は 15 か月間保持されるため、履歴情報にアクセスし、ウェブアプリケーションまたはサービスの動作をよりの確に把握できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、「[Amazon ユーザーガイド CloudWatch](#)」を参照してください。

メトリクスとディメンションのリストについては、「[Amazon Pinpoint ユーザーガイド](#)」の「[による Amazon Pinpoint のモニタリング CloudWatch Amazon Pinpoint](#)」を参照してください。

を使用した AWS エンドユーザーメッセージングプッシュ API コールのログ記録 AWS CloudTrail

AWS エンドユーザーメッセージングプッシュは と統合されています。これは AWS CloudTrail、エンドユーザーメッセージングプッシュのすべての API 呼び出しをイベントとして CloudTrail キャプチャする AWS ユーザー、ロール、または AWS AWS のサービスによって実行されたアクションを記録するサービスです。キャプチャされた呼び出しには、AWS エンドユーザーメッセージングプッシュコンソールからの呼び出しと AWS、エンドユーザーメッセージングプッシュ API オペレーションへのコード呼び出しが含まれます。証跡を作成する場合は、AWS エンドユーザーメッセージングプッシュの CloudTrail イベントなど、Amazon S3 バケットへのイベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールのイベント履歴で最新のイベントを表示できます。によって収集された情報を使用して CloudTrail、AWS エンドユーザーメッセージングプッシュに対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

の詳細については CloudTrail、「[AWS CloudTrail ユーザーガイド](#)」を参照してください。

AWS のエンドユーザーメッセージングプッシュ情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウントと、で が有効になります。AWS エンドユーザーメッセージングプッシュでアクティビティが発生すると、そのアクティビティは CloudTrail イベント履歴の他の AWS サービスイベントとともにイベントに記録されます。で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。詳細については、「[イベント履歴を使用した CloudTrail イベントの表示](#)」を参照してください。

AWS エンドユーザーメッセージングプッシュのイベントなど AWS アカウント、 のイベントの継続的な記録については、証跡を作成します。証跡により CloudTrail、 はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをより詳細に分析し、それに基づいて行動するように他の AWS サービスを設定できます。詳細については、次を参照してください:

- [追跡を作成するための概要](#)
- [CloudTrail がサポートするサービスと統合](#)
- [の Amazon SNS通知の設定 CloudTrail](#)
- [複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信](#)

すべての AWS エンドユーザーメッセージングプッシュアクションは によってログに記録 CloudTrail され、[AWS 「エンドユーザーメッセージングプッシュAPIリファレンス」](#) に記載されています。例えば、、、 GetApnsVoipChannelアクションを呼び出す UpdateApnsChannelと GetAdmChannel、 CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して行われたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、[CloudTrail userIdentity 「」要素](#)を参照してください。

AWS エンドユーザーメッセージングプッシュログファイルエントリについて

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。CloudTrail ログファイルには 1 つ以上のログエントリが含まれます。イベントは任意のソー

スからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルはパブリックAPIコールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

インターフェイスエンドポイント (AWS PrivateLink) を使用して AWS エンドユーザーメッセージングプッシュにアクセスする

を使用して AWS PrivateLink、VPCと AWS エンドユーザーメッセージングプッシュの間にプライベート接続を作成できます。インターネットゲートウェイ、NATデバイスVPC、VPN接続、または AWS Direct Connect 接続を使用せずに、[こちら](#)にあるかのように AWS エンドユーザーメッセージングプッシュにアクセスできます。このインスタンスは、AWS エンドユーザーメッセージングプッシュにアクセスするためにパブリック IP アドレスを必要としません。

このプライベート接続を確立するには、AWS PrivateLinkを利用したインターフェイスエンドポイントを作成します。インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。これらは、AWS エンドユーザーメッセージングプッシュ宛てのトラフィックのエントリポイントとして機能するリクエストマネージドネットワークインターフェイスです。

詳細については、「AWS PrivateLink ガイド」の「[AWS のサービスによるアクセス AWS PrivateLink](#)」を参照してください。

AWS エンドユーザーメッセージングプッシュに関する考慮事項

AWS エンドユーザーメッセージングプッシュのインターフェイスエンドポイントを設定する前に、「AWS PrivateLink ガイド」の「[考慮事項](#)」を確認してください。

AWS エンドユーザーメッセージングプッシュは、インターフェイスエンドポイントを介したすべての API アクションの呼び出しをサポートします。

VPC エンドポイントポリシーは AWS、エンドユーザーメッセージングプッシュではサポートされていません。デフォルトでは、インターフェイスエンドポイントを介して AWS エンドユーザーメッセージングプッシュへのフルアクセスが許可されます。または、セキュリティグループをエンドポイントネットワークインターフェイスに関連付けて、インターフェイスエンドポイント経由で AWS エンドユーザーメッセージングプッシュへのトラフィックを制御することもできます。

AWS エンドユーザーメッセージングプッシュ用のインターフェイスエンドポイントを作成する

Amazon VPCコンソールまたは AWS Command Line Interface () を使用して、AWS エンドユーザーメッセージングプッシュのインターフェイスエンドポイントを作成できますAWS CLI。詳細については、「AWS PrivateLink ガイド」の「[インターフェイスエンドポイントを作成](#)」を参照してください。

次のサービス名を使用して、AWS エンドユーザーメッセージングプッシュのインターフェイスエンドポイントを作成します。

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

インターフェイスエンドポイントDNSのプライベートを有効にすると、デフォルトのリージョンDNS名を使用して AWS エンドユーザーメッセージングプッシュにAPIリクエストを行うことができます。例えば com.amazonaws.us-east-1.pinpoint です。

インターフェイスエンドポイントのエンドポイントポリシーを作成する

エンドポイントポリシーは、インターフェイスエンドポイントにアタッチできる IAMリソースです。デフォルトのエンドポイントポリシーでは、インターフェイスエンドポイントを介した AWS エンドユーザーメッセージングプッシュへのフルアクセスが許可されます。から AWS エンドユーザーメッセージングプッシュに許可されるアクセスを制御するにはVPC、カスタムエンドポイントポリシーをインターフェイスエンドポイントにアタッチします。

エンドポイントポリシーは、以下の情報を指定します。

- アクションを実行できるプリンシパル (AWS アカウント、IAMユーザー、IAMロール)。
- 実行可能なアクション。
- このアクションを実行できるリソース。

詳細については、AWS PrivateLink ガイドの[Control access to services using endpoint policies \(エンドポイントポリシーを使用してサービスへのアクセスをコントロールする\)](#)を参照してください。

例: AWS エンドユーザーメッセージングプッシュアクションのVPCエンドポイントポリシー

以下は、カスタムエンドポイントポリシーの例です。このポリシーをインターフェイスエンドポイントにアタッチすると、すべてのリソースのすべてのプリンシパルに対して、リストされている AWS エンドユーザーメッセージングプッシュアクションへのアクセスが許可されます。

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

AWS エンドユーザーメッセージングプッシュのクォータ

には、サービスごとに、以前 AWS アカウント は制限と呼ばれていたデフォルトのクォータがあります AWS 。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。

AWS エンドユーザーメッセージングプッシュのクォータを表示するには、[Service Quotas コンソール](#)を開きます。ナビゲーションペインで、AWS サービスを選択し、Amazon Pinpointを選択します。

AWS アカウントには、AWS エンドユーザーメッセージングプッシュに関連する次のクォータがあります。

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
キャンペーンで 1 秒あたりに送信できるプッシュ通知の最大数	25000 通知 / 秒	はい、 Service Quotas コンソール を使用します
Amazon Device Messaging (ADM) メッセージペイロードサイズ	メッセージごとに 6 KB	なし
Apple Push Notification Service (APNs) メッセージペイロードサイズ	メッセージごとに 4 KB	なし
APNs サンドボックスメッセージのペイロードサイズ	メッセージごとに 4 KB	なし
Baidu Cloud Push メッセージペイロードサイズ	メッセージごとに 4 KB	なし
Firebase Cloud Messaging (FCM) メッセージペイロードサイズ	メッセージごとに 4 KB	なし

AWS 「エンドユーザーメッセージングプッシュユーザーガイド」のドキュメント履歴

次の表に、AWS エンドユーザーメッセージングプッシュのドキュメントリリースを示します。

変更	説明	日付
初回リリース	AWS エンドユーザーメッセージングプッシュユーザーガイドの初回リリース	2024 年 7 月 24 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。