Guida per l'utente

AWS Messaggistica push per l'utente finale

AWS Messaggistica push per l'utente finale: Guida per l'utente

Table of Contents

Che cos'è la messaggistica push per l'utente AWS finale?

> **Note**
>
> Le funzionalità di notifica push di Amazon Pinpoint sono ora denominate AWS End User Messaging.

Con AWS End User Messaging Push, puoi coinvolgere gli utenti delle tue app inviando notifiche push tramite un canale di notifica push. Supportiamo Apple Push Notification Service (APNs), Firebase Cloud Messaging (FCM), Amazon Device Messaging (ADM) e Baidu Push.

Argomenti

Sei un utente End User Messaging Push per la prima volta AWS ?

Se sei un utente alle prime armi di AWS End User Messaging Push, ti consigliamo di iniziare leggendo le seguenti sezioni:

- Configurare un Account AWS
- Guida introduttiva a AWS End User Messaging Push
- Creazione di un'applicazione e attivazione dei canali push

Funzionalità di AWS End User Messaging Push

È possibile inviare notifiche push ad app utilizzando canali distinti per i servizi di notifiche push seguenti:

- Firebase Cloud Messaging (FCM)
- Servizio Apple Push Notification (APNs)

> **Note**
>
> Puoi utilizzarlo APNs per inviare messaggi a dispositivi iOS come iPhone e iPad, nonché al browser Safari su dispositivi macOS, come laptop e desktop Mac.

- Baidu Cloud Push
- Amazon Device Messaging (ADM)

Accesso alla messaggistica push per l'utente finale AWS

Spiega brevemente i diversi modi per accedere al servizio, tramite console, CLI o API.

È possibile gestire AWS End User Messaging Push utilizzando le seguenti interfacce:

AWS Console End User Messaging Push

L'interfaccia web in cui è possibile creare e gestire le risorse AWS End User Messaging Push. Se ti sei registrato a Account AWS, puoi accedere alla console AWS End User Messaging Push da AWS Management Console.

AWS Command Line Interface

Interagisci con i AWS servizi utilizzando i comandi nella shell della riga di comando. AWS Command Line Interface È supportato su Windows, macOS e Linux. Per ulteriori informazioni su AWS CLI, vedere la Guida per AWS Command Line Interface l'utente. I comandi AWS End User Messaging Push sono disponibili nel AWS CLI Command Reference.

AWS SDKs

Se sei uno sviluppatore di software che preferisce creare applicazioni utilizzando specifiche lingue APIs anziché inviare una richiesta tramite HTTP o HTTPS, AWS fornisce librerie, codice di esempio, tutorial e altre risorse. Queste librerie forniscono funzioni di base che automatizzano le attività, come la firma crittografica delle richieste, il ritentativo delle richieste e la gestione delle risposte agli errori. Queste funzioni contribuiscono a rendere più efficiente l'avvio. Per ulteriori informazioni, consulta Strumenti per creare in AWS.

Disponibilità regionale

AWS End User Messaging Push è disponibile Regioni AWS in diversi paesi in Nord America, Europa, Asia e Oceania. In ogni regione, AWS mantiene più zone di disponibilità. Queste zone di disponibilità sono fisicamente isolate l'una dall'altra, ma sono unite da connessioni di rete private a bassa latenza, a velocità effettiva elevata e altamente ridondanti. Queste zone di disponibilità vengono utilizzate per fornire livelli molto elevati di disponibilità e ridondanza, riducendo al minimo la latenza.

Per ulteriori informazioni Regioni AWS, consulta Specificare quali contenuti Regioni AWS il tuo account può utilizzare in. Riferimenti generali di Amazon Web Services Per un elenco di tutte le regioni in cui è attualmente disponibile AWS End User Messaging Push e l'endpoint per ciascuna regione, consulta Endpoints and quotas for Amazon Pinpoint API and AWS service endpoint in. Riferimenti generali di Amazon Web Services Per ulteriori informazioni sul numero di zone di disponibilità presenti in ciascuna regione, consulta Infrastruttura globale AWS.

Configurare un Account AWS

Prima di poter utilizzare AWS End User Messaging Push per inviare notifiche push alla tua app, devi prima ottenere un'autorizzazione IAM Account AWS con sufficienti autorizzazioni. Questo Account AWS può essere utilizzato anche per altri servizi dell' AWS ecosistema.

Argomenti

Iscriviti per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Guida introduttiva a AWS End User Messaging Push

Per configurare AWS End User Messaging Push in modo che possa inviare notifiche push alle tue app, devi prima fornire le credenziali che autorizzano AWS End User Messaging Push a inviare messaggi alla tua app. Le credenziali fornite dipendono dal sistema di notifica push che utilizzi:

- Per le credenziali del servizio Apple Push Notification (APN), consulta Ottenere una chiave di crittografia e un ID di chiave da Apple e Ottenere un certificato di provider da Apple nella documentazione Apple per sviluppatori.
- Per le credenziali di Firebase Cloud Messaging (FCM) che possono essere ottenute tramite la console Firebase, vedi Firebase Cloud Messaging.
- Per le credenziali Baidu, vedi Baidu.
- Per le credenziali di Amazon Device Messaging (ADM), consulta Ottenere credenziali.

Creazione di un'applicazione e attivazione dei canali push

Prima di poter utilizzare AWS End User Messaging Push per inviare notifiche push, devi prima creare un'applicazione e abilitare il canale delle notifiche push.

Contestuale

Applicazione

Un'applicazione è un contenitore di archiviazione per tutte le impostazioni AWS End User Messaging Push. L'applicazione memorizza anche le impostazioni dei canali, delle campagne e dei percorsi di Amazon Pinpoint.

Chiave

Una chiave di firma privata utilizzata da AWS End User Messaging Push per firmare crittograficamente i token di autenticazione. APNs Si può ottenere la chiave di firma dal proprio account sviluppatore Apple.

Se fornisci una chiave di firma,AWS End User Messaging Push utilizza un token con cui autenticarsi APNs per ogni notifica push che invii. Con la chiave di firma, puoi inviare notifiche push agli ambienti di APNs produzione e sandbox.

A differenza dei certificati, la chiave di firma non scade. La chiave viene fornita una sola volta e non è necessario rinnovarla. È possibile utilizzare la stessa chiave di firma per più app. Per ulteriori informazioni, consulta Comunicare APNs utilizzando i token di autenticazione nella Guida di Xcode.

Certificate

Un certificato TLS che AWS End User Messaging Push utilizza per l'autenticazione APNs quando invii notifiche push. Un APNs certificato può supportare sia gli ambienti di produzione che quelli sandbox oppure può supportare solo l'ambiente sandbox. Si può ottenere il certificato dal proprio account sviluppatore Apple.

Un certificato scade dopo un anno. Quando ciò accade, è necessario creare un nuovo certificato, da fornire quindi a AWS End User Messaging Push per rinnovare l'invio delle notifiche push. Per ulteriori informazioni, consulta Comunicare APNs utilizzando un certificato TLS nella Guida di Xcode.

Prerequisiti

Prima di poter utilizzare qualsiasi canale push, sono necessarie credenziali valide per il servizio push. Per ulteriori informazioni sull'ottenimento delle credenziali, vedere. Guida introduttiva a AWS End User Messaging Push

Procedura

Segui queste istruzioni per creare un'applicazione e abilitare uno qualsiasi dei canali push. Per completare questa procedura è necessario solo inserire il nome di un'applicazione. È possibile abilitare o disabilitare qualsiasi canale push in un secondo momento.

1. Apri la console AWS End User Messaging Push all'indirizzo https://console.aws.amazon.com/push-notifications/.
2. Scegli Crea applicazione.
3. Per il nome dell'applicazione, inserisci il nome dell'applicazione.
4. (Facoltativo) Segui questo passaggio opzionale per abilitare il servizio Apple Push Notification (APNs).

    a. Per il servizio Apple Push Notification (APNs) seleziona Abilita.

    b. Per il tipo di autenticazione predefinito scegli una delle seguenti opzioni:

        i. Se scegli Credenziali chiave, fornisci le seguenti informazioni dal tuo account sviluppatore Apple. AWS End User Messaging Push richiede queste informazioni per creare token di autenticazione.

            - ID chiave: ID assegnato alla chiave di firma.
            - Identificatore del bundle: ID assegnato all'app iOS.
            - Identificatore del team: ID assegnato al team dell'account sviluppatore Apple.
            - Chiave di autenticazione: file .p8 scaricato dall'account sviluppatore Apple quando crei una chiave di autenticazione.

        ii. Se si sceglie Certificate credentials (Credenziali certificato), è necessario fornire le seguenti informazioni:

            - SSL certificate (Certificato SSL): il file .p12 per il certificato TLS.
            - Password certificato: se hai assegnato una password al certificato, immettila qui.

- Tipo di certificato: seleziona il tipo di certificato da utilizzare.

5. (Facoltativo) Segui questo passaggio opzionale per abilitare Firebase Cloud Messaging (FCM).

   a. Per Firebase Cloud Messaging (FCM) seleziona Abilita.
   b. Per il tipo di autenticazione predefinito scegli una delle seguenti opzioni:

      i. Per le credenziali del token (consigliato) scegli i file, quindi scegli il file JSON del servizio.
      ii. Per le credenziali chiave, inserisci la tua chiave nella chiave API.

6. (Facoltativo) Segui questo passaggio opzionale per abilitare Baidu Cloud Push.

   a. Per Baidu Cloud Push seleziona Abilita.
   b. Per la chiave API, inserisci la tua chiave API.
   c. Per chiave segreta inserisci la tua chiave segreta.

7. (Facoltativo) Segui questo passaggio facoltativo per abilitare Amazon Device Messaging.

   a. Per Amazon Device Messaging seleziona Abilita.
   b. Per Client ID inserisci il tuo ID cliente.
   c. Per Client secret, inserisci il tuo client secret.

8. Scegli Crea applicazione.

Disabilitazione dei canali push

Segui queste istruzioni per disabilitare uno qualsiasi dei canali push.

1. Apri la console AWS End User Messaging Push all'indirizzo https://console.aws.amazon.com/push-notifications/.
2. Scegliete l'applicazione che contiene le vostre credenziali push.
3. (Facoltativo) Per il servizio Apple Push Notification (APNs), deselezionate Abilita.
4. (Facoltativo) Per Firebase Cloud Messaging (FCM), seleziona Abilita.
5. (Opzionale) Per Baidu Cloud Push, seleziona Enable.
6. (Facoltativo) Per Amazon Device Messaging, seleziona Abilita.
7. Scegli Save changes (Salva modifiche).

Invio di un messaggio

L'API AWS End User Messaging Push può inviare notifiche push transazionali a identificatori di dispositivi specifici. Questa sezione contiene esempi di codice completi che è possibile utilizzare per inviare notifiche push tramite l'API AWS End User Messaging Push utilizzando un SDK. AWS

Puoi utilizzare questi esempi per inviare notifiche push tramite qualsiasi servizio di notifica push supportato da AWS End User Messaging Push. Attualmente, AWS End User Messaging Push supporta i seguenti canali: Firebase Cloud Messaging (FCM), Apple Push Notification Service (APNs), Baidu Cloud Push e Amazon Device Messaging (ADM).

Per ulteriori esempi di codice su endpoint, segmenti e canali, consulta Esempi di codice.

> **Note**
>
> Quando invii notifiche push tramite il servizio Firebase Cloud Messaging (FCM), usa il nome del servizio `GCM` nella chiamata all' AWS End User Messaging Push API. Il servizio Google Cloud Messaging (GCM) è stato interrotto da Google il 10 aprile 2018. Tuttavia, l'API AWS End User Messaging Push utilizza il nome del `GCM` servizio per i messaggi inviati tramite il servizio FCM per mantenere la compatibilità con il codice API scritto prima dell'interruzione del servizio GCM.

GCM (AWS CLI)

L'esempio seguente utilizza send-messages per inviare una notifica push GCM con. AWS CLI*token*Sostituiscilo con il token univoco del dispositivo e *611e3e3cdd47474c9c1399a50example* con l'identificatore dell'applicazione.

```
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"
      }
  }
}
```

L'esempio seguente utilizza send-messages per inviare una notifica push GCM, utilizzando tutte le chiavi legacy, con. AWS CLI*token*Sostituiscilo con il token univoco del dispositivo e *611e3e3cdd47474c9c1399a50example* con l'identificatore dell'applicazione.

```
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
```

L'esempio seguente utilizza send-messages per inviare una notifica push GCM con FCMv1 payload di messaggi utilizzando il. AWS CLI*token*Sostituiscilo con il token univoco del dispositivo e *611e3e3cdd47474c9c1399a50example* con l'identificatore dell'applicazione.

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

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

se si utilizza `ImageUrl` field for GCM, pinpoint invia il campo come notifica dei dati, con la chiave `pinpoint.notification.imageUrl` specificata, il che può impedire il rendering dell'immagine immediatamente. Utilizza RawContent o aggiungi la gestione delle chiavi dati, ad esempio l'integrazione della tua app con. AWS Amplify

Safari (AWS CLI)

Puoi utilizzare AWS End User Messaging Push per inviare messaggi a computer macOS che utilizzano il browser web Safari di Apple. Per inviare un messaggio al browser Safari, devi specificare il contenuto del messaggio in formato RAW e includere un attributo specifico nel payload del messaggio. Puoi farlo creando un modello di notifica push con un payload di messaggi non elaborati o specificando il contenuto del messaggio non elaborato direttamente in un messaggio della campagna, nella Amazon Pinpoint User Guide.

> **Note**
>
> Questo attributo speciale è necessario per l'invio a computer laptop e desktop macOS che utilizzano il browser Web Safari. Non è necessario per l'invio a dispositivi iOS come iPhone e iPad.

Per inviare un messaggio ai browser web Safari, devi specificare il payload del messaggio in formato RAW. Il payload dei messaggi in formato RAW deve includere un array `url-args` all'interno dell'oggetto `aps`. L'array `url-args` è necessario per inviare notifiche push al browser Web Safari. Tuttavia, è accettabile che l'array contenga un singolo elemento vuoto.

L'esempio seguente utilizza send-messages per inviare una notifica al browser Web Safari con. AWS CLI*token*Sostituiscilo con il token univoco del dispositivo e *611e3e3cdd47474c9c1399a50example* con l'identificatore dell'applicazione.

```
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
```

Per ulteriori informazioni sulle notifiche push di Safari, consulta l'argomento relativo alla configurazione delle notifiche push di Safari sul sito Web di Apple per gli sviluppatori.

APNS (AWS CLI)

L'esempio seguente utilizza send-messages per inviare una notifica push APNS con. AWS CLI*token*Sostituiscilo con il token univoco del dispositivo, *611e3e3cdd47474c9c1399a50example* con l'identificatore dell'applicazione e *GAME_INVITATION* con un identificatore univoco.

```
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)

Utilizzate questo esempio per inviare notifiche push utilizzando l' AWS SDK for JavaScript in Node.js. Questo esempio presuppone che tu abbia già installato e configurato l'SDK per JavaScript in Node.js.

Questo esempio presuppone anche che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per ulteriori informazioni, consulta Impostazione delle credenziali nell'AWS SDK for JavaScript in Node.js Developer Guide.

```
'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

Utilizza questo esempio per inviare notifiche push utilizzando AWS SDK for Python (Boto3). Questo esempio si basa sul presupposto che SDK per Python (Boto3) sia già stato installato e configurato.

Questo esempio presuppone anche che tu stia utilizzando un file di credenziali condivise per specificare la chiave di accesso e la chiave di accesso segreta per un utente esistente. Per informazioni dettagliate, consulta Credentials nella documentazione di riferimento delle API AWS SDK per 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 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()
```

Risorse aggiuntive

- Per ulteriori informazioni sui modelli di canali Push, consulta Creazione di modelli di notifica push nella Guida per l'utente di Amazon Pinpoint.

Ricezione di notifiche push nell'applicazione

I seguenti argomenti descrivono come modificare l'app Swift, Android, React Native o Flutter in modo che riceva notifiche push.

Argomenti

Configurazione delle notifiche push Swift

Le notifiche push per le app iOS vengono inviate utilizzando il servizio Apple Push Notification (APNs). Per poter inviare notifiche push ai dispositivi iOS, è necessario creare un ID app nel portale Apple Developer e creare i certificati richiesti. Puoi trovare ulteriori informazioni sul completamento di questi passaggi in Configurazione dei servizi di notifica push nella documentazione di AWS Amplify.

Lavorare con i token APNs

Come best practice, è consigliabile sviluppare l'app in modo che i token di dispositivo dei clienti vengano rigenerati quando l'app viene reinstallata.

Se un destinatario aggiorna il dispositivo a una nuova versione principale di iOS (ad esempio, da iOS 12 a iOS 13) e successivamente reinstalla l'app, questa genera un nuovo token. Se l'app non aggiorna il token, per inviare la notifica viene utilizzato il token precedente. Di conseguenza, il servizio Apple Push Notification (APNs) rifiuta la notifica, poiché il token ora non è valido. Quando tenti di inviare la notifica, ricevi un messaggio di notifica di errore da. APNs

Configurazione delle notifiche push per Android

Le notifiche push per le app Android vengono inviate utilizzando Firebase Cloud Messaging (FCM), che sostituisce Google Cloud Messaging (GCM). Prima di poter inviare notifiche push ai dispositivi

Android, è necessario ottenere le credenziali FCM. È quindi possibile utilizzare quelle credenziali per creare un progetto Android e avviare un'app di esempio in grado di ricevere notifiche push. Puoi trovare ulteriori informazioni sul completamento di questi passaggi nella sezione Notifiche push della documentazione di AWS Amplify.

Configurazione delle notifiche push di Flutter

Le notifiche push per le app Flutter vengono inviate utilizzando Firebase Cloud Messaging (FCM) per Android e per iOS. APNs Per ulteriori informazioni sull'esecuzione di questa procedura, consulta la sezione relativa alle notifiche push nella documentazione di AWS Amplify Flutter.

Configurazione delle notifiche push per React Native

Le notifiche push per le app React Native vengono inviate utilizzando Firebase Cloud Messaging (FCM) per Android e per APNs iOS. Puoi trovare ulteriori informazioni sul completamento di questi passaggi nella sezione Notifiche push della documentazione di AWS Amplify. JavaScript

Crea un'applicazione in AWS End User Messaging Push

Per iniziare a inviare notifiche push in AWS End User Messaging Push, devi creare un'applicazione. Quindi, è necessario abilitare i canali delle notifiche push da utilizzare fornendo le credenziali appropriate.

È possibile creare nuove applicazioni e configurare canali di notifica push utilizzando la console AWS End User Messaging Push. Per ulteriori informazioni, consulta Creazione di un'applicazione e attivazione dei canali push.

Puoi anche creare e configurare un'applicazione utilizzando l'API, un AWS SDK o AWS Command Line Interface(AWS CLI). Per creare un'applicazione, utilizzate la `Apps` risorsa. Per configurare i canali delle notifiche push, usa le risorse seguenti:

- APNs canale per inviare messaggi agli utenti di dispositivi iOS utilizzando il servizio Apple Push Notification.
- Canale ADM per l'invio di messaggi agli utenti di Amazon Kindle Fire.
- Canale Baidu per l'invio di messaggi agli utenti di Baidu.
- Canale GCM per l'invio di messaggi a dispositivi Android mediante Firebase Cloud Messaging (FCM), che sostituisce Google Cloud Messaging (GCM).

Gestione delle notifiche push

Dopo aver ottenuto le credenziali necessarie per inviare notifiche push, puoi aggiornare l'applicazione in modo che sia in grado di ricevere notifiche push. Per ulteriori informazioni, consulta Notifiche push: Guida introduttiva nella documentazione. AWS Amplify

Eliminazione di un'applicazione

Questa procedura rimuove l'applicazione dall'account e tutte le risorse dell'applicazione.

Contestuale

Applicazione

Un'applicazione è un contenitore di archiviazione per tutte le impostazioni AWS End User Messaging Push. L'applicazione memorizza anche le impostazioni dei canali, delle campagne e dei percorsi di Amazon Pinpoint.

Procedura

1. Apri la console AWS End User Messaging Push all'indirizzo. https://console.aws.amazon.com/push-notifications/
2. Scegliete un'applicazione, quindi scegliete Elimina.
3. Nella finestra Elimina applicazione, inserisci **`delete`** e quindi scegli Elimina.

> **Important**
>
> Vengono eliminati anche tutti i canali, le campagne, i percorsi o i segmenti di Amazon Pinpoint.

Best practice

Anche quando operi nell'interesse dei clienti è possibile che si verifichino situazioni che impattano sull'efficienza del recapito dei tuoi messaggi. Le seguenti sezioni contengono raccomandazioni utili ad garantire che le comunicazioni e-mail raggiungano i destinatari previsti.

Invio di un volume elevato di notifiche push

Prima di inviare un volume elevato di notifiche push, assicurati che il tuo account sia configurato per supportare i tuoi requisiti di throughput. Per impostazione predefinita, tutti gli account sono configurati per inviare 25.000 messaggi al secondo. Se è necessario inviare più di 25.000 messaggi in un secondo, puoi richiedere un aumento della quota. Per ulteriori informazioni, consulta Quote per la messaggistica AWS push per l'utente finale.

Assicurati che il tuo account sia configurato correttamente con le credenziali di ciascuno dei provider di notifiche push che intendi utilizzare, come FCM o. APNs

Infine, elabora un modo per gestire le eccezioni. Ogni servizio di notifica push fornisce diversi messaggi di eccezione. Per gli invii transazionali, viene restituito un codice di stato principale di 200 per la chiamata API, con un codice di stato per endpoint di 400 (errore permanente) se il token della piattaforma (ad esempio, FCM) o il certificato (ad esempio, APN) viene considerato non valido durante l'invio dei messaggi.

Sicurezza nella messaggistica push per l'utente AWS finale

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità che si applicano alla messaggistica push per gli utenti AWS finali, consulta AWS Servizi nell'ambito del programma di conformitàAWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza AWS End User Messaging Push. I seguenti argomenti mostrano come configurare AWS End User Messaging Push per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse AWS End User Messaging Push.

Argomenti

Protezione dei dati in AWS End User Messaging Push

Il modello di responsabilità AWS condivisa modello si applica alla protezione dei dati in AWS End User Messaging Push. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con AWS End User Messaging Push o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia dei dati

AWS I dati End User Messaging Push sono crittografati in transito e a riposo. Quando invii dati a AWS End User Messaging Push, i dati vengono crittografati non appena vengono ricevuti e archiviati. Quando recuperi i dati da AWS End User Messaging Push, questi ti trasmette i dati utilizzando i protocolli di sicurezza correnti.

Crittografia a riposo

AWS End User Messaging Push crittografa tutti i dati che archivia per te. Ciò include i dati di configurazione, i dati degli utenti e degli endpoint, i dati di analisi e tutti i dati aggiunti o importati in AWS End User Messaging Push. Per crittografare i dati, AWS End User Messaging Push utilizza chiavi interne AWS Key Management Service (AWS KMS) che il servizio possiede e gestisce per conto dell'utente. Queste chiavi vengono ruotate su base regolare. Per informazioni in merito AWS KMS, consulta la Guida per gli AWS Key Management Service sviluppatori.

Crittografia in transito

AWS End User Messaging Push utilizza HTTPS e Transport Layer Security (TLS) 1.2 o versione successiva per comunicare con client e applicazioni. Per comunicare con altri AWS servizi, AWS End User Messaging Push utilizza HTTPS e TLS 1.2. Inoltre, quando si creano e gestiscono risorse AWS End User Messaging Push utilizzando la console, un AWS SDK o il AWS Command Line Interface, tutte le comunicazioni sono protette tramite HTTPS e TLS 1.2.

Gestione delle chiavi

Per crittografare i dati AWS End User Messaging Push, AWS End User Messaging Push utilizza AWS KMS chiavi interne che il servizio possiede e gestisce per conto dell'utente. Queste chiavi vengono ruotate su base regolare. Non puoi fornire e utilizzare le tue AWS KMS o altre chiavi per crittografare i dati archiviati in AWS End User Messaging Push.

Riservatezza del traffico Internet

La privacy del traffico internetwork si riferisce alla protezione delle connessioni e del traffico tra AWS End User Messaging Push e i client e le applicazioni locali e tra AWS End User Messaging Push e altre AWS risorse nella stessa regione. AWS Le seguenti funzionalità e pratiche possono aiutarti a garantire la privacy del traffico di rete per AWS End User Messaging Push.

Traffico tra AWS End User Messaging Push e client e applicazioni locali

Per stabilire una connessione privata tra AWS End User Messaging Push e client e applicazioni sulla rete locale, puoi usare. AWS Direct Connect Consente di collegare la rete a una posizione AWS Direct Connect utilizzando un cavo Ethernet standard in fibra ottica. Un'estremità del cavo è collegata al router. L'altra estremità è connessa a un AWS Direct Connect router. Per ulteriori informazioni, consulta Che cos'è AWS Direct Connect? nella Guida per l'utente di AWS Direct Connect .

Per garantire un accesso sicuro a AWS End User Messaging Push tramite publiced APIs, ti consigliamo di rispettare i requisiti AWS End User Messaging Push per le chiamate API. AWS End User Messaging Push richiede ai client di utilizzare Transport Layer Security (TLS) 1.2 o versione successiva. I client devono inoltre supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID di chiave di accesso e una chiave di accesso segreta associata a un principale AWS Identity and Access Management (IAM) per l' AWS account. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.

Traffico tra AWS End User Messaging Push e altre AWS risorse

Per proteggere le comunicazioni tra AWS End User Messaging Push e altre AWS risorse nella stessa AWS regione, AWS End User Messaging Push utilizza HTTPS e TLS 1.2 per impostazione predefinita.

Gestione delle identità e degli accessi per AWS End User Messaging Push

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse AWS End User Messaging Push. IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità

- Gestione dell'accesso con policy
- Come funziona AWS End User Messaging Push con IAM
- Esempi di policy basate sull'identità per End User Messaging Push AWS
- Risoluzione dei problemi relativi AWS alla messaggistica con l'utente finale, all'identità e all'accesso push

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in AWS End User Messaging Push.

Utente del servizio: se utilizzi il servizio AWS End User Messaging Push per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità Push di messaggistica con l'utente AWS finale per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di AWS End User Messaging Push, consultaRisoluzione dei problemi relativi AWS alla messaggistica con l'utente finale, all'identità e all'accesso push.

Amministratore del servizio: se sei responsabile delle risorse AWS End User Messaging Push della tua azienda, probabilmente hai pieno accesso a AWS End User Messaging Push. È tuo compito determinare a quali funzionalità e risorse AWS End User Messaging Push devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con AWS End User Messaging Push, consultaCome funziona AWS End User Messaging Push con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso a AWS End User Messaging Push. Per visualizzare esempi di policy basate sull'identità di AWS End User Messaging Push che puoi utilizzare in IAM, consulta. Esempi di policy basate sull'identità per End User Messaging Push AWS

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta Signature Version 4 AWS per le richieste API nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA)AWS in IAM nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi passare da un ruolo utente a un ruolo IAM (console). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o

utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Create a role for a third-party identity provider (federation) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso inoltrato (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. Questa soluzione è preferibile alla memorizzazione delle chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella IAM User Guide.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di proprietà dell' Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.
- Politiche di controllo delle risorse (RCPs): RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di Servizi AWS tale supporto RCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire

da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la logica di valutazione delle policy nella IAM User Guide.

Come funziona AWS End User Messaging Push con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS End User Messaging Push, scopri quali funzionalità IAM sono disponibili per l'uso con AWS End User Messaging Push.

Funzionalità IAM che puoi utilizzare con AWS End User Messaging Push

Funzionalità IAM	AWS Supporto End User Messaging Push
Policy basate su identità	Sì
Policy basate su risorse	Sì
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
ACLs	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una visione di alto livello di come AWS End User Messaging Push e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per AWS End User Messaging Push

Supporta le policy basate su identità: sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate sull'identità per End User Messaging Push AWS

Per visualizzare esempi di politiche basate sull'identità di AWS End User Messaging Push, vedere. Esempi di policy basate sull'identità per End User Messaging Push AWS

Politiche basate sulle risorse all'interno di End User Messaging Push AWS

Supporta le policy basate sulle risorse: sì

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account

a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Azioni politiche per AWS End User Messaging Push

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni push di messaggistica con l'utente AWS finale, consulta Azioni definite da AWS End User Messaging Push nel riferimento di autorizzazione del servizio.

Le azioni politiche in AWS End User Messaging Push utilizzano il seguente prefisso prima dell'azione:

```
mobiletargeting
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

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

Per visualizzare esempi di politiche basate sull'identità di AWS End User Messaging Push, consulta. Esempi di policy basate sull'identità per End User Messaging Push AWS

Risorse politiche per AWS End User Messaging Push

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le operazioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di risorse AWS End User Messaging Push e relativi tipi di risorse ARNs, consulta Resources Defined by AWS End User Messaging Push nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, consulta Azioni definite da AWS End User Messaging Push.

Per visualizzare esempi di politiche basate sull'identità di AWS End User Messaging Push, consulta. Esempi di policy basate sull'identità per End User Messaging Push AWS

Chiavi delle condizioni delle policy per AWS End User Messaging Push

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni

condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di AWS End User Messaging Push, consulta Condition Keys for AWS End User Messaging Push nel Service Authorization Reference. Per sapere con quali azioni e risorse è possibile utilizzare una chiave di condizione, consulta Azioni definite da AWS End User Messaging Push.

Per visualizzare esempi di politiche basate sull'identità di AWS End User Messaging Push, consulta. Esempi di policy basate sull'identità per End User Messaging Push AWS

ACLs in AWS End User Messaging Push

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con End User Messaging Push AWS

Supporta ABAC (tag nelle policy): parzialmente

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è

il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con AWS End User Messaging Push

Supporta le credenziali temporanee: sì

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Passaggio da un ruolo utente a un ruolo IAM (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni principali multiservizio per AWS End User Messaging Push

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per AWS End User Messaging Push

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità AWS End User Messaging Push. Modifica i ruoli di servizio solo quando AWS End User Messaging Push fornisce indicazioni in tal senso.

Ruoli collegati ai servizi per AWS End User Messaging Push

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta Servizi AWS supportati da IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per End User Messaging Push AWS

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare le risorse AWS End User Messaging Push. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da AWS End User Messaging Push, incluso il formato di ARNs per ogni tipo di risorsa, consulta Actions, Resources and Condition Keys for AWS End User Messaging Push nel Service Authorization Reference.

Argomenti

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse AWS End User Messaging Push nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come

autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.

- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console AWS End User Messaging Push

Per accedere alla console AWS End User Messaging Push, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse AWS End User Messaging Push presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la console AWS End User Messaging Push, allega anche la policy `AWSEndUserMessaging` AWS gestita alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente 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": [
    "*"
   ]
  }
 ]
}
```

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa

politica include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l' AWS CLI API o. 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": "*"
        }
    ]
}
```

Risoluzione dei problemi relativi AWS alla messaggistica con l'utente finale, all'identità e all'accesso push

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS End User Messaging Push e IAM.

Argomenti

Non sono autorizzato a eseguire alcuna azione in AWS End User Messaging Push

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `mobiletargeting:`*`GetWidget`* fittizie.

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

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `mobiletargeting:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a AWS End User Messaging Push.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in AWS End User Messaging Push. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

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

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse AWS End User Messaging Push

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS End User Messaging Push supporta queste funzionalità, consulta. Come funziona AWS End User Messaging Push con IAM
- Per scoprire come fornire l'accesso alle risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM di un altro Account AWS utente di tua proprietà nella IAM User Guide. Account AWS
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Convalida della conformità per AWS End User Messaging Push

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Governance e conformità per la sicurezza: queste guide all'implementazione di soluzioni illustrano considerazioni relative all'architettura e i passaggi per implementare le funzionalità di sicurezza e conformità.
- Riferimenti sui servizi conformi ai requisiti HIPAA: elenca i servizi HIPAA idonei. Non tutti Servizi AWS sono idonei alla normativa HIPAA.
- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- Valutazione delle risorse con regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty

può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.

- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Resilienza nella messaggistica AWS push per l'utente finale

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle zone di disponibilità, vedere Global Regioni AWS Infrastructure.AWS

Oltre all'infrastruttura AWS globale, AWS End User Messaging Push offre diverse funzionalità per supportare le esigenze di resilienza e backup dei dati.

Sicurezza dell'infrastruttura nella messaggistica AWS push per l'utente finale

In quanto servizio gestito, AWS End User Messaging Push è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper Amazon Web Services: Overview of Security Processes.

Utilizzi chiamate API AWS pubblicate per accedere a AWS End User Messaging Push attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Analisi della configurazione e delle vulnerabilità

In quanto servizio gestito, AWS End User Messaging Push è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper Amazon Web Services: panoramica dei processi di sicurezza. Ciò significa che AWS gestisce ed esegue attività e procedure di sicurezza di base per rafforzare, applicare patch, aggiornare e mantenere in altro modo l'infrastruttura sottostante per il tuo account e le tue risorse. Queste procedure sono state riviste e certificate dalle terze parti appropriate.

Best practice di sicurezza

Utilizza gli account AWS Identity and Access Management (IAM) per controllare l'accesso alle operazioni API, in particolare alle operazioni che creano, modificano o eliminano risorse. Per quanto riguarda l'API , tali risorse includono progetti, campagne e percorsi.

- Crea un utente IAM per ogni persona che gestisce le risorse , incluso l'utente stesso. Non utilizzare le credenziali AWS root per gestire le risorse.
- Assegna a ciascun utente un set minimo di autorizzazioni richieste per eseguire le proprie mansioni.
- Utilizza gruppi IAM per gestire in modo efficace le autorizzazioni per più utenti.
- Ruota periodicamente le credenziali IAM.

Per ulteriori informazioni sulla sicurezza, consulta Sicurezza nella messaggistica push per l'utente AWS finale. Per ulteriori informazioni su IAM, consulta AWS Identity and Access Management. Per informazioni sulle best practice di IAM, consulta Best practice di IAM.

Monitoraggio dei messaggi push per gli utenti AWS finali

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di AWS End User Messaging Push e delle altre soluzioni AWS. AWS fornisce i seguenti strumenti di monitoraggio per monitorare i messaggi push degli utenti AWS finali, segnalare quando qualcosa non va e intraprendere azioni automatiche se necessario:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue EC2 istanze Amazon e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di registro da EC2 istanze Amazon e altre fonti. CloudTrail CloudWatch I log possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.
- Amazon EventBridge può essere utilizzato per automatizzare i AWS servizi e rispondere automaticamente agli eventi di sistema, come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi AWS relativi ai servizi vengono forniti quasi EventBridge in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali operazioni automatizzate intraprendere quando un evento corrisponde a una regola. Per ulteriori informazioni, consulta Amazon EventBridge User Guide.
- AWS CloudTrailacquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di log a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account hanno chiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudTrail.

Monitoraggio dei messaggi push per gli utenti AWS finali con Amazon CloudWatch

È possibile monitorare AWS End User Messaging Push utilizzando CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. Queste statistiche vengono conservate

per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Per un elenco di metriche e dimensioni, consulta Monitoring Amazon Pinpoint CloudWatch with nella Amazon Pinpoint User Guide.

Registrazione delle chiamate all'API push di messaggistica per l'utente AWS finale utilizzando AWS CloudTrail

AWS End User Messaging Push è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in AWS End User Messaging Push. CloudTrail acquisisce tutte le chiamate API per AWS End User Messaging Push come eventi. Le chiamate acquisite includono chiamate dalla console AWS End User Messaging Push e chiamate in codice alle operazioni dell'API AWS End User Messaging Push. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per AWS End User Messaging Push. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a AWS End User Messaging Push, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

AWS Messaggistica con l'utente finale Informazioni push in CloudTrail

CloudTrail è abilitato sul tuo Account AWS quando crei l'account. Quando si verifica un'attività in AWS End User Messaging Push, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi in tuo Account AWS, inclusi gli eventi per AWS End User Messaging Push, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per

analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni push di messaggistica con l'utente AWS finale vengono registrate CloudTrail e sono documentate nell'AWS End User Messaging Push API Reference. Ad esempio, le chiamate a `UpdateApnsChannel` e `GetAdmChannel` le `GetApnsVoipChannel` azioni generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Comprensione delle voci dei file di registro push di AWS End User Messaging

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

Accedere a AWS End User Messaging Push utilizzando un'interfaccia endpoint ()AWS PrivateLink

Puoi utilizzarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e AWS End User Messaging Push. Puoi accedere a AWS End User Messaging Push come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per AWS accedere a End User Messaging Push.

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Si tratta di interfacce di rete gestite dai richiedenti che fungono da punto di ingresso per il traffico destinato all'End User Messaging Push. AWS

Per ulteriori informazioni, consulta Access Servizi AWS through AWS PrivateLink nella Guida.AWS PrivateLink

Considerazioni sulla messaggistica push per l'utente AWS finale

Prima di configurare un endpoint di interfaccia per AWS End User Messaging Push, consulta le considerazioni nella Guida.AWS PrivateLink

AWS End User Messaging Push supporta l'esecuzione di chiamate a tutte le sue azioni API tramite l'endpoint dell'interfaccia.

Le policy degli endpoint VPC non sono supportate per AWS End User Messaging Push. Per impostazione predefinita, l'accesso completo a AWS End User Messaging Push è consentito tramite l'interfaccia endpoint. In alternativa, è possibile associare un gruppo di sicurezza alle interfacce di rete dell'endpoint per controllare il traffico verso l'utente AWS finale di messaggistica Push attraverso l'endpoint dell'interfaccia.

Crea un endpoint di interfaccia per AWS End User Messaging Push

Puoi creare un endpoint di interfaccia per AWS End User Messaging Push utilizzando la console Amazon VPC o AWS Command Line Interface il AWS CLI(). Per ulteriori informazioni, consulta la sezione Creazione di un endpoint di interfaccia nella Guida per l'utente di AWS PrivateLink .

Crea un endpoint di interfaccia per AWS End User Messaging Push utilizzando il seguente nome di servizio:

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

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API a AWS End User Messaging Push utilizzando il nome DNS regionale predefinito. Ad esempio `com.amazonaws.us-east-1.pinpoint`.

Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia

Una policy dell'endpoint è una risorsa IAM che è possibile allegare all'endpoint dell'interfaccia. La policy predefinita per gli endpoint consente l'accesso completo all' AWS End User Messaging Push tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito a AWS End User Messaging Push dal tuo VPC, allega una policy endpoint personalizzata all'endpoint dell'interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (Account AWS, utenti IAM e ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai servizi con policy di endpoint nella Guida di AWS PrivateLink .

Esempio: policy degli endpoint VPC per le azioni push di messaggistica con l'utente AWS finale

Di seguito è riportato l'esempio di una policy dell'endpoint personalizzata. Quando alleghi questa policy all'endpoint dell'interfaccia, concede l'accesso alle azioni push di messaggistica per l'utente AWS finale elencate per tutti i principali utenti su tutte le risorse.

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

```
         ],
         "Resource":"*"
      }
   ]
}
```

Quote per la messaggistica AWS push per l'utente finale

Your Account AWS ha delle quote predefinite, precedentemente denominate limiti, per ogni servizio. AWS Salvo diversa indicazione, ogni quota si applica a una regione specifica. Se per alcune quote è possibile richiedere aumenti, altre quote non possono essere modificate.

Per visualizzare le quote per AWS End User Messaging Push, apri la console Service Quotas. Nel riquadro di navigazione, scegli i servizi AWS e seleziona Amazon Pinpoint.

Il tuo account AWS ha le seguenti quote relative a AWS End User Messaging Push.

Risorsa	Quota predefinita	Idoneità all'incremento
Numero massimo di notifiche push che possono essere inviate al secondo in una campagna	25.000 notifiche al secondo	Sì, usa la console Service Quotas
Dimensione del payload dei messaggi di Amazon Device Messaging (ADM)	6 KB per messaggio	No
Dimensioni del payload dei messaggi del servizio Apple Push Notification (APNs)	4 KB per messaggio	No
APNs dimensione del payload dei messaggi sandbox	4 KB per messaggio	No
Dimensione del payload dei messaggi di Baidu Cloud Push	4 KB per messaggio	No
Dimensione del payload dei messaggi Firebase Cloud Messaging (FCM)	4 KB per messaggio	No

Cronologia dei documenti per la AWS End User Messaging Push User Guide

La tabella seguente descrive le versioni della documentazione per AWS End User Messaging Push.

Modifica	Descrizione	Data
Versione iniziale	Versione iniziale della Guida per l'utente di AWS End User Messaging Push	24 luglio 2024

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.