aws

Guía para desarrolladores

Amazon Cognito

Amazon Cognito: Guía para desarrolladores

Table of Contents

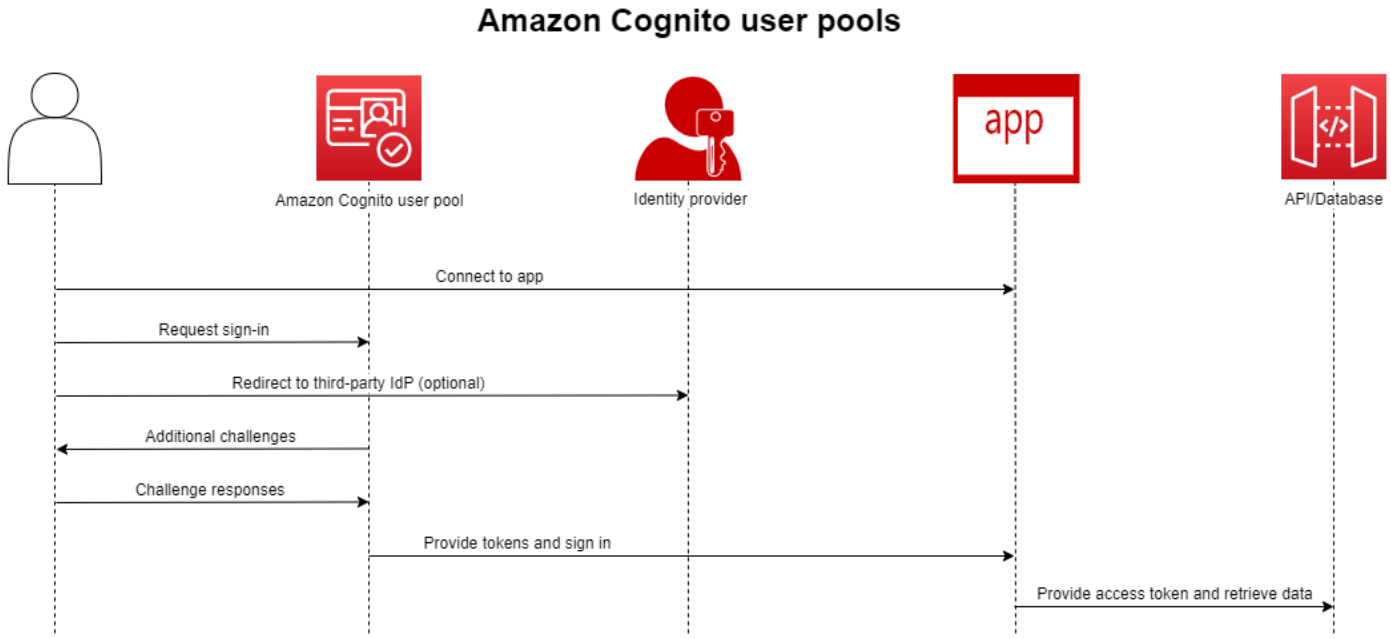
¿Qué es Amazon Cognito?

Amazon Cognito es una plataforma de identidad para aplicaciones web y móviles. Es un directorio de usuarios, un servidor de autenticación y un servicio de autorización para los tokens y credenciales de AWS de acceso de OAuth 2.0. Con Amazon Cognito, puede autenticar y autorizar a los usuarios desde el directorio de usuarios integrado, desde el directorio empresarial y desde proveedores de identidad de consumidores como Google y Facebook.

Temas

Los dos componentes siguientes componen Amazon Cognito. Funcionan de forma independiente o en conjunto, en función de las necesidades de acceso de los usuarios.

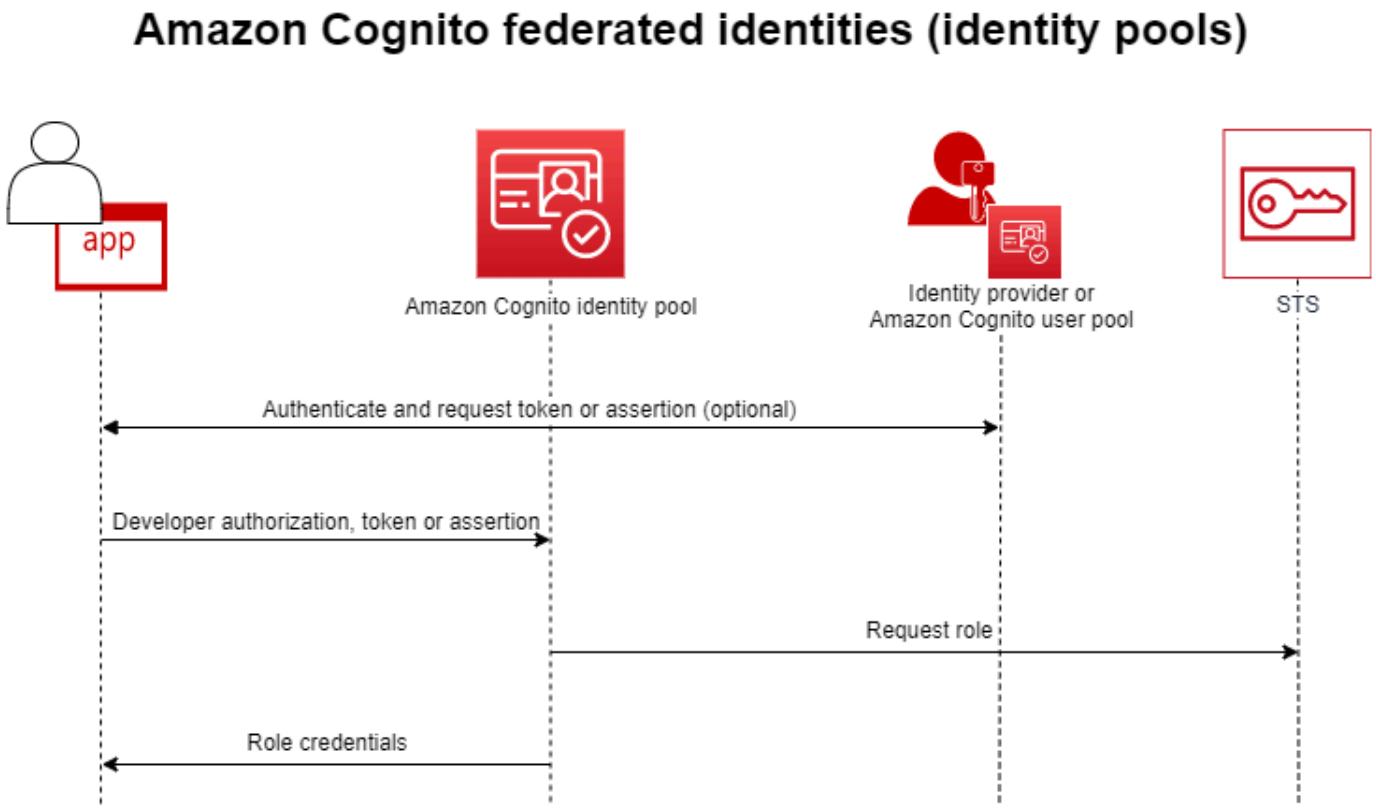
Grupos de usuarios



Cree un grupo de usuarios cuando quiera autenticar y autorizar a los usuarios a la aplicación o la API. Los grupos de usuarios son un directorio de usuarios con funciones de creación, administración y autenticación de usuarios automáticas e impulsadas por el administrador. El grupo de usuarios puede ser un directorio independiente y un proveedor de identidades de OIDC (IdP) y un proveedor de servicios intermedio (SP) para proveedores de terceros de identidades de personal y clientes. Puedes proporcionar un inicio de sesión único (SSO) en tu aplicación para las identidades de los empleados de tu organización en SAML 2.0 y IdPs OIDC con grupos de usuarios. También puede proporcionar SSO en su aplicación para las identidades de clientes de su organización en los almacenes de identidades públicos de OAuth 2.0 de Amazon, Google, Apple y Facebook. Para obtener más información acerca de la gestión de acceso e identidad de los clientes (CIAM), consulte ¿Qué es CIAM?.

Los grupos de usuarios no requieren la integración con un grupo de identidades. Desde un grupo de usuarios, puede emitir JSON Web Token (JWT) autenticados directamente a una aplicación, un servidor web o una API.

Grupos de identidades



Configure un grupo de identidades de Amazon Cognito cuando desee autorizar a usuarios autenticados o anónimos a acceder a sus recursos. AWS Un grupo de identidades emite AWS credenciales para que su aplicación sirva de recursos a los usuarios. Puede autenticar a los usuarios con un proveedor de identidades de confianza, como un grupo de usuarios o un servicio SAML 2.0. También puede emitir, opcionalmente, credenciales para los usuarios invitados. Los grupos de identidades utilizan un control de acceso basado en roles y en atributos para administrar la autorización de los usuarios para acceder a sus recursos. AWS

Los grupos de identidades no requieren la integración con un grupo de usuarios. Un grupo de identidades puede aceptar reclamaciones autenticadas directamente de los proveedores de identidad de los empleados y de los consumidores.

Un grupo de usuarios y un grupo de identidades de Amazon Cognito que se utilizan en conjunto

En el diagrama que comienza este tema, se utiliza Amazon Cognito para autenticar al usuario y, a continuación, concederle acceso a un Servicio de AWS.

1. El usuario de la aplicación inicia sesión a través de un grupo de usuarios y recibe los tokens de OAuth 2.0.
2. Tu aplicación intercambia un token de grupo de usuarios por un grupo de identidades por AWS credenciales temporales que puedes usar con las AWS API y el (). AWS Command Line Interface AWS CLI
3. La aplicación asigna la sesión de credenciales al usuario y proporciona acceso autorizado a sitios Servicios de AWS como Amazon S3 y Amazon DynamoDB.

Para ver más ejemplos que utilizan grupos de identidades y grupos de usuarios, consulte Escenarios comunes de Amazon Cognito.

En Amazon Cognito, la obligación de seguridad de la nube del modelo de responsabilidad compartida cumple con SOC 1-3, PCI DSS, ISO 27001 e HIPAA-BAA. Puede diseñar la seguridad en la nube en Amazon Cognito para que cumpla con SOC1-3, ISO 27001 e HIPAA-BAA, pero no con DSS de PCI. Para obtener más información, consulte Servicios de AWS en el ámbito. Consulte también Consideraciones de datos regionales.

Características de Amazon Cognito

Grupos de usuarios

Un grupo de usuarios de Amazon Cognito es un directorio de usuarios. Con un grupo de usuarios, los usuarios pueden iniciar sesión en su aplicación web o móvil por medio de Amazon Cognito o federarse mediante un IdP de terceros. Los usuarios federados y locales tienen un perfil de usuario en el grupo de usuarios.

Los usuarios locales son los inscritos o registrados directamente en el grupo de usuarios. Puede administrar y personalizar estos perfiles de usuario en el AWS Management Console, un AWS SDK o el AWS Command Line Interface ().AWS CLI

Los grupos de usuarios de Amazon Cognito aceptan tokens y afirmaciones de terceros IdPs y recopilan los atributos del usuario en un JWT que se envía a la aplicación. Puede estandarizar su aplicación en un conjunto de JWT mientras Amazon Cognito gestiona las interacciones con IdPs ellos y asigna sus afirmaciones a un formato de token central.

Un grupo de usuarios de Amazon Cognito puede ser un IdP independiente. Amazon Cognito se basa en el estándar OpenID Connect (OIDC) para generar JWT para la autenticación y la

autorización. Cuando inicia sesión en los usuarios locales, el grupo de usuarios tiene autoridad para esos usuarios. Tiene acceso a las funciones siguientes cuando autentica a los usuarios locales.

- Implemente su propia frontend web que llama a la API de grupos de usuarios de Amazon Cognito para autenticar, autorizar y administrar los usuarios.
- Configurar autenticación multifactor (MFA) para los usuarios. Amazon Cognito admite contraseña temporal de un solo uso (TOTP) y MFA por mensaje SMS.
- Proteja contra el acceso de cuentas de usuario que estén bajo control malintencionado.
- Cree sus propios flujos de autenticación de varios pasos personalizados.
- Busque usuarios en otro directorio y mígrelos a Amazon Cognito.

Un grupo de usuarios de Amazon Cognito también puede cumplir una doble función como proveedor de servicios (SP) para su IdPs aplicación y como IdP para su aplicación. Los grupos de usuarios de Amazon Cognito pueden conectarse con consumidores IdPs como Facebook y Google, o con empleados IdPs como Okta y Active Directory Federation Services (ADFS).

Con los tokens de OAuth 2.0 y OpenID Connect (OIDC) que emite un grupo de usuarios de Amazon Cognito, puede

- Aceptar un ID de token en la aplicación que autentica a un usuario y proporciona la información que necesita para configurar el perfil del usuario.
- Aceptar un token de acceso en la API con los ámbitos de OIDC que autorizan las llamadas a la API de los usuarios.
- Recupera AWS las credenciales de un grupo de identidades de Amazon Cognito.

Características de los grupos de usuarios de Amazon Cognito

Característica	Descripción
Proveedor de identidad OIDC	Emita tokens de identificación para autenticar a los usuarios
Servidor de autorización	Emita tokens de acceso para autorizar el acceso de los usuarios a las API
SAML 2.0 SP	Transforma las aserciones de SAML en identific adores y identificadores de acceso

OIDC SP	Transforma los tokens OIDC en tokens de identificación y acceso
OAuth 2.0 SP	Transforma los identificadores de Apple, Facebook, Amazon o Google en tus propios identificadores y tokens de acceso
Servicio frontend de autenticación	Registre, gestione y autentique a los usuarios con la interfaz de usuario alojada
Soporte de API para tu propia interfaz de usuario	Cree, gestione y autentique usuarios mediante solicitudes de API en los SDK compatibles[1] AWS
MFA	Utilice los mensajes SMS, los TOTP o el dispositivo de su usuario como factor de autenticación adicional[1]
Supervisión y respuesta de seguridad	Protéjase contra actividades maliciosas y contraseñas inseguras[1]
Personalice los flujos de autenticación	Cree su propio mecanismo de autenticación o añada pasos personalizados a los flujos existentes[1]
Grupos	Cree agrupaciones lógicas de usuarios y una jerarquía de las funciones de IAM al pasar los tokens a los grupos de identidades
Personalice los tokens de identificación	Personaliza tus fichas de identificación con reclamos nuevos, modificados y suprimidos
Personalice los atributos de usuario	Asigna valores a los atributos de usuario y añade tus propios atributos personalizados

[1] La característica solo está disponible para usuarios locales.

Para obtener más información sobre los grupos de usuarios, consulte Introducción a los grupos de usuarios y la Referencia de la API de grupos de usuarios de Amazon Cognito.

Grupos de identidades

Un grupo de identidades es un conjunto de identificadores únicos, o identidades, que usted asigna a sus usuarios o invitados y autoriza a recibir AWS credenciales temporales. Al presentar una prueba de autenticación en un grupo de identidades en forma de afirmaciones fiables de un proveedor de identidades sociales (IdP) de SAML 2.0, OpenID Connect (OIDC) u OAuth 2.0, se asocia al usuario con una identidad del grupo de identidades. El token que tu grupo de identidades crea para la identidad puede recuperar las credenciales de sesión temporales de AWS Security Token Service (AWS STS).

Para complementar las identidades autenticadas, también puede configurar un grupo de identidades para autorizar el AWS acceso sin la autenticación del IdP. Puede ofrecer su propia prueba de autenticación personalizada o no tener autenticación. Puede conceder AWS credenciales temporales a cualquier usuario de la aplicación que las solicite, con identidades no autenticadas. Los grupos de identidades también aceptan reclamaciones y emiten credenciales en función del propio esquema personalizado, con identidades autenticadas por el desarrollador.

Con los grupos de identidades de Amazon Cognito, tiene dos formas de integrarse con las políticas de IAM en la Cuenta de AWS. Puede utilizar estas dos características juntas o de forma individual.

Control de acceso con base en roles

Cuando el usuario pasa las reclamaciones al grupo de identidades, Amazon Cognito elige el rol de IAM que solicita. Para personalizar los permisos del rol según las necesidades, se aplican las políticas de IAM a cada rol. Por ejemplo, si el usuario demuestra que trabaja en el departamento de marketing, recibirá credenciales para un rol con políticas adaptadas a las necesidades de acceso del departamento de marketing. Amazon Cognito puede solicitar un rol predeterminado, un rol basado en reglas que consultan las reclamaciones del usuario o un rol basado en la suscripción al grupo del usuario en un grupo de usuarios. También puede configurar la política de confianza de roles para que IAM confíe solo en el grupo de identidades para generar sesiones temporales.

Atributos para controlar el acceso

El grupo de identidades lee los atributos de las reclamaciones de los usuarios y los asigna a las etiquetas de las entidades principales de la sesión temporal del usuario. A continuación, puede configurar las políticas basadas en recursos de IAM para permitir o denegar el acceso a los recursos en función de las entidades principales de IAM que contienen las etiquetas de sesión del grupo de identidades. Por ejemplo, si el usuario demuestra que está en el departamento de marketing, AWS STS etiqueta su sesión. `Department: marketing` Su bucket de Amazon S3 permite realizar

operaciones de lectura en función de una PrincipalTag condición aws: que requiere un valor de `marketing` para la `Department` etiqueta.

Características de los grupos de identidades de Amazon Cognito

Característica	Descripción
Grupo de usuarios de Amazon Cognito SP	Cambie un token de identificación de su grupo de usuarios por credenciales de identidad web de AWS STS
SAML 2.0 SP	Intercambie las afirmaciones de SAML para obtener credenciales de identidad web desde AWS STS
OIDC SP	Intercambie los tokens OIDC por credenciales de identidad web desde AWS STS
OAuth 2.0 SP	Intercambia los tokens de OAuth de Amazon, Facebook, Google, Apple y Twitter por credenciales de identidad web de AWS STS
SP personalizado	Con AWS las credenciales, intercambie reclamos en cualquier formato por credenciales de identidad web desde AWS STS
Acceso sin autenticar	Emita credenciales de identidad web de acceso limitado sin autenticación AWS STS
Control de acceso con base en roles	Elija una función de IAM para su usuario autenticado en función de sus afirmaciones y configure sus funciones para que solo las asuma en el contexto de su conjunto de identidades
Control de acceso basado en atributos	Convierte las notificaciones en etiquetas principales para tu sesión AWS STS temporal y utiliza las políticas de IAM para filtrar el acceso a los recursos en función de las etiquetas principales

Para obtener más información sobre los grupos de identidades, consulte Introducción a los grupos de identidades de Amazon Cognito y la Referencia de la API de grupos de identidades de Amazon Cognito.

Comparación de grupos de usuarios y grupos de identidades de Amazon Cognito

Característica	Descripción	Grupos de usuarios	Grupos de identidades
Proveedor de identidad OIDC	Emita tokens de ID de OIDC para autenticar a los usuarios de la aplicación	✓	
Servidor de autorización de API	Emita tokens de acceso para autorizar el acceso de los usuarios a las API, bases de datos y otros recursos que aceptan los ámbitos de autorización de OAuth 2.0	✓	
Servidor de autorización de identidad web de IAM	Genere fichas con las que pueda intercambiarlas AWS STS por credenciales temporales AWS		✓
IdP SAML 2.0 SP y OIDC	Emita tokens OIDC personalizados en función de las afirmaciones de un IdP de SAML 2.0	✓	

OIDC PS y OIDC IdP	Emita tokens OIDC personalizados en función de las afirmaciones de un IdP de OIDC	✓	
IdP de OAuth 2.0 PS y OIDC	Emite tokens OIDC personalizados basados en los alcances de los proveedores sociales de OAuth 2.0, como Apple y Google	✓	
SAML 2.0 SP y agente de credencia les	Emita AWS credencia les temporales en función de las afirmaciones de un IdP de SAML 2.0		✓
Agente de credencia les y SP de OIDC	Emita AWS credencia les temporales en función de las afirmaciones de un IdP de OIDC		✓
Agente de credencia les y SP de OAuth 2.0	Emite AWS credencia les temporales basadas en los alcances de los proveedores sociales de OAuth 2.0, como Apple y Google		✓

Grupo de usuarios de Amazon Cognito, SP y agente de credencia les	Emita AWS credencia les temporales en función de las solicitudes de OIDC de un grupo de usuarios de Amazon Cognito		✓
Agente de credencia les y SP personali zados	Emita AWS credencia les temporales en función de la autorizac ión de IAM del desarrollador		✓
Servicio frontend de autenticación	Registre, gestione y autentique a los usuarios con la interfaz de usuario alojada	✓	
Soporte de API para su propia interfaz de usuario de autentica ción	Cree, gestione y autentique usuarios mediante solicitudes de API en los SDK compatibles[1] AWS	✓	
MFA	Utilice los mensajes SMS, los TOTP o el dispositivo de su usuario como factor de autenticación adicional[1]	✓	
Supervisión y respuesta de seguridad	Protéjase contra actividades maliciosa s y contraseñas inseguras[1]	✓	

Personalice los flujos de autenticación	Cree su propio mecanismo de autenticación o añada pasos personalizados a los flujos existentes[1]	✓	
Grupos	Cree agrupaciones lógicas de usuarios y una jerarquía de las funciones de IAM al pasar los tokens a los grupos de identidades	✓	
Personalice los tokens de identific ación	Personaliza tus fichas de identific ación con reclamos nuevos, modificados y suprimidos	✓	
AWS WAF ACL web	Supervise y controle las solicitudes a su entorno de autentica ción con AWS WAF	✓	
Personalice los atributos del usuario	Asigna valores a los atributos de usuario y añade tus propios atributos personali zados	✓	
Acceso sin autenticar	Emita credenciales de identidad web de acceso limitado sin autenticación AWS STS		✓

Control de acceso con base en roles	Elija una función de IAM para su usuario autenticado en función de sus afirmaciones y configure sus funciones para que solo las asuma en el contexto de su conjunto de identidades	✓
Control de acceso basado en atributos	Transforma las afirmaciones de los usuarios en etiquetas principales para tu sesión AWS STS temporal y utiliza las políticas de IAM para filtrar el acceso a los recursos en función de las etiquetas principales	✓

[1] La característica solo está disponible para usuarios locales.

Introducción a Amazon Cognito

Para ver ejemplos de aplicaciones de grupos de usuarios, consulteIntroducción a los grupos de usuarios.

Para obtener una introducción a los grupos de identidades, consulteIntroducción a los grupos de identidades de Amazon Cognito.

Para obtener enlaces a experiencias de configuración guiada con grupos de usuarios y grupos de identidades, consulteOpciones de configuración guiada para Amazon Cognito.

Para ver vídeos, artículos, documentación y más aplicaciones de muestra, consulte los recursos para desarrolladores de Amazon Cognito.

Para usar Amazon Cognito necesita una Cuenta de AWS. Para obtener más información, consulte Empezando con AWS.

Disponibilidad regional

Amazon Cognito está disponible en varias AWS regiones de todo el mundo. En cada región, Amazon Cognito se distribuye en varias zonas de disponibilidad. Estas zonas de disponibilidad están físicamente aisladas entre sí, pero están unidas mediante conexiones de red privadas con un alto nivel de rendimiento y redundancia y con baja latencia. Estas zonas de disponibilidad permiten AWS proporcionar servicios, incluido Amazon Cognito, con niveles muy altos de disponibilidad y redundancia, a la vez que minimizan la latencia.

Para obtener una lista de todas las regiones en las que Amazon Cognito se encuentra actualmente disponible, consulte Regiones y puntos de conexión de AWS en la Referencia general de Amazon Web Services. Para obtener más información sobre la cantidad de zonas de disponibilidad de cada región, consulte Infraestructura global de AWS.

Precios de Amazon Cognito

Para obtener más información sobre los precios de Amazon Cognito, consulte Precios de Amazon Cognito.

Cómo funciona la autenticación con los grupos de usuarios y grupos de identidades de Amazon Cognito

Cuando su cliente inicia sesión en un grupo de usuarios de Amazon Cognito, su aplicación recibe los tokens web JSON (JWT).

Cuando su cliente inicia sesión en un grupo de identidades, ya sea con un token de grupo de usuarios u otro proveedor, su aplicación recibe credenciales temporales. AWS

Al iniciar sesión en un grupo de usuarios, puede implementar la autenticación y la autorización por completo con un AWS SDK. Si no desea crear sus propios componentes de interfaz de usuario (UI), puede invocar una interfaz de usuario web prediseñada (la interfaz de usuario alojada) o la página de inicio de sesión de su proveedor de identidad (IdP) externo.

En este tema se ofrece información general sobre algunas de las formas en que su aplicación puede interactuar con Amazon Cognito para autenticarse con tokens de identificación, autorizar con tokens de acceso y acceder Servicios de AWS con credenciales de grupo de identidades.

Temas

Autenticación y autorización de la API de grupo de usuarios con un SDK AWS

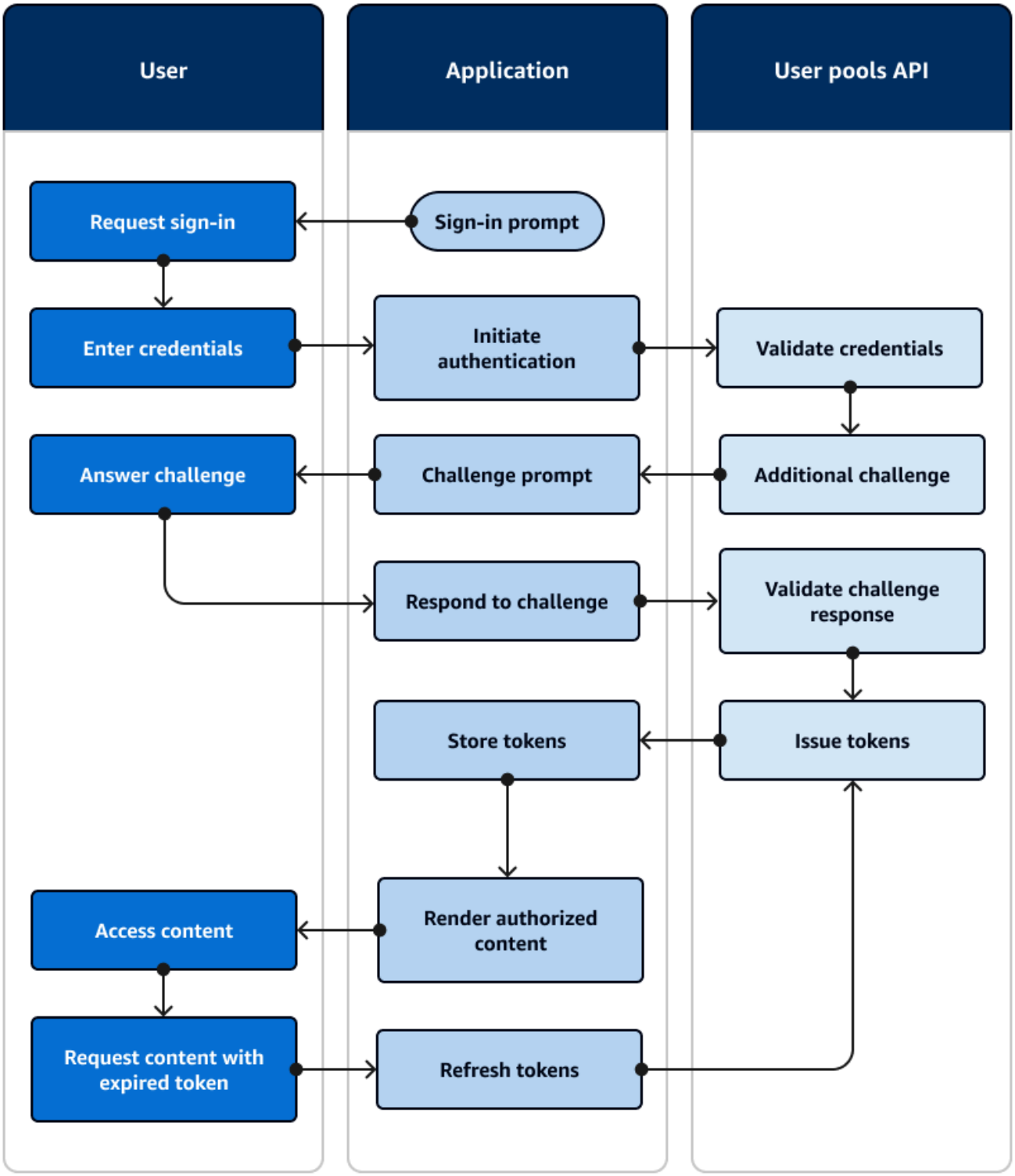
AWS ha desarrollado componentes para los grupos de usuarios de Amazon Cognito, o el proveedor de identidad de Amazon Cognito, en diversos marcos de desarrollo. Los métodos integrados en estos SDK llaman a la API de grupos de usuarios de Amazon Cognito. El mismo espacio de nombres de la API de grupos de usuarios contiene operaciones para la configuración de los grupos de usuarios y para la autenticación de los usuarios. Para obtener una descripción más detallada, consulte. Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios

La autenticación mediante API se ajusta al modelo en el que las aplicaciones tienen componentes de interfaz de usuario existentes y se basan principalmente en el grupo de usuarios como directorio de usuarios. Este diseño añade Amazon Cognito como un componente dentro de una aplicación más grande. Requiere una lógica programática para gestionar cadenas complejas de desafíos y respuestas.

Esta aplicación no necesita implementar una implementación completa de OpenID Connect (OIDC) por parte de confianza. En cambio, tiene la capacidad de decodificar y utilizar los JWT. Cuando desee acceder al conjunto completo de funciones del grupo de usuarios para los usuarios locales, cree su autenticación con el SDK de Amazon Cognito en su entorno de desarrollo.

La autenticación de la API con ámbitos de OAuth personalizados está menos orientada a la autorización de la API externa. Para añadir ámbitos personalizados a un token de acceso a partir de la autenticación de la API, modifique el token en tiempo de ejecución con un. Desencadenador de Lambda anterior a la generación del token

El siguiente diagrama ilustra una sesión de inicio de sesión típica para la autenticación de API.



Flujo de autenticación de la API

1. Un usuario accede a tu aplicación.
2. Seleccionan un enlace de «Iniciar sesión».
3. Introducen su nombre de usuario y contraseña.
4. La aplicación invoca el método que realiza una solicitud a la InitiateAuthAPI. La solicitud pasa las credenciales del usuario a un grupo de usuarios.
5. El grupo de usuarios valida las credenciales del usuario y determina que el usuario ha activado la autenticación multifactor (MFA).
6. El grupo de usuarios responde con un desafío que solicita un código MFA.
7. La aplicación genera un mensaje que recopila el código MFA del usuario.
8. La aplicación invoca el método que realiza una solicitud de RespondToAuthChallengeAPI. La solicitud pasa el código MFA del usuario.
9. El grupo de usuarios valida el código MFA del usuario.
10. El grupo de usuarios responde con los JWT del usuario.
11. La aplicación decodifica, valida y almacena o almacena en caché los JWT del usuario.
12. La aplicación muestra el componente de acceso controlado solicitado.
13. El usuario ve su contenido.
14. Más tarde, el token de acceso del usuario ha caducado y este solicita ver un componente de acceso controlado.
15. La aplicación determina que la sesión del usuario debe persistir. Vuelve a invocar el InitiateAuthmétodo con el token de actualización y recupera nuevos tokens.

Variantes y personalización

Puede aumentar este flujo con desafíos adicionales, por ejemplo, sus propios desafíos de autenticación personalizados. Puede restringir automáticamente el acceso a los usuarios cuyas contraseñas se hayan visto comprometidas o cuyas características de inicio de sesión inesperadas puedan indicar un intento de inicio de sesión malintencionado. Este flujo tiene prácticamente el mismo aspecto para las operaciones de registro, actualización de los atributos de los usuarios y restablecimiento de contraseñas. La mayoría de estos flujos tienen operaciones de API públicas (del lado del cliente) y confidenciales (del lado del servidor) duplicadas.

Recursos relacionados

- API de grupos de usuarios de Amazon Cognito
- Introducción a los grupos de usuarios
- Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles
- Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios

Autenticación de grupos de usuarios con la interfaz de usuario alojada

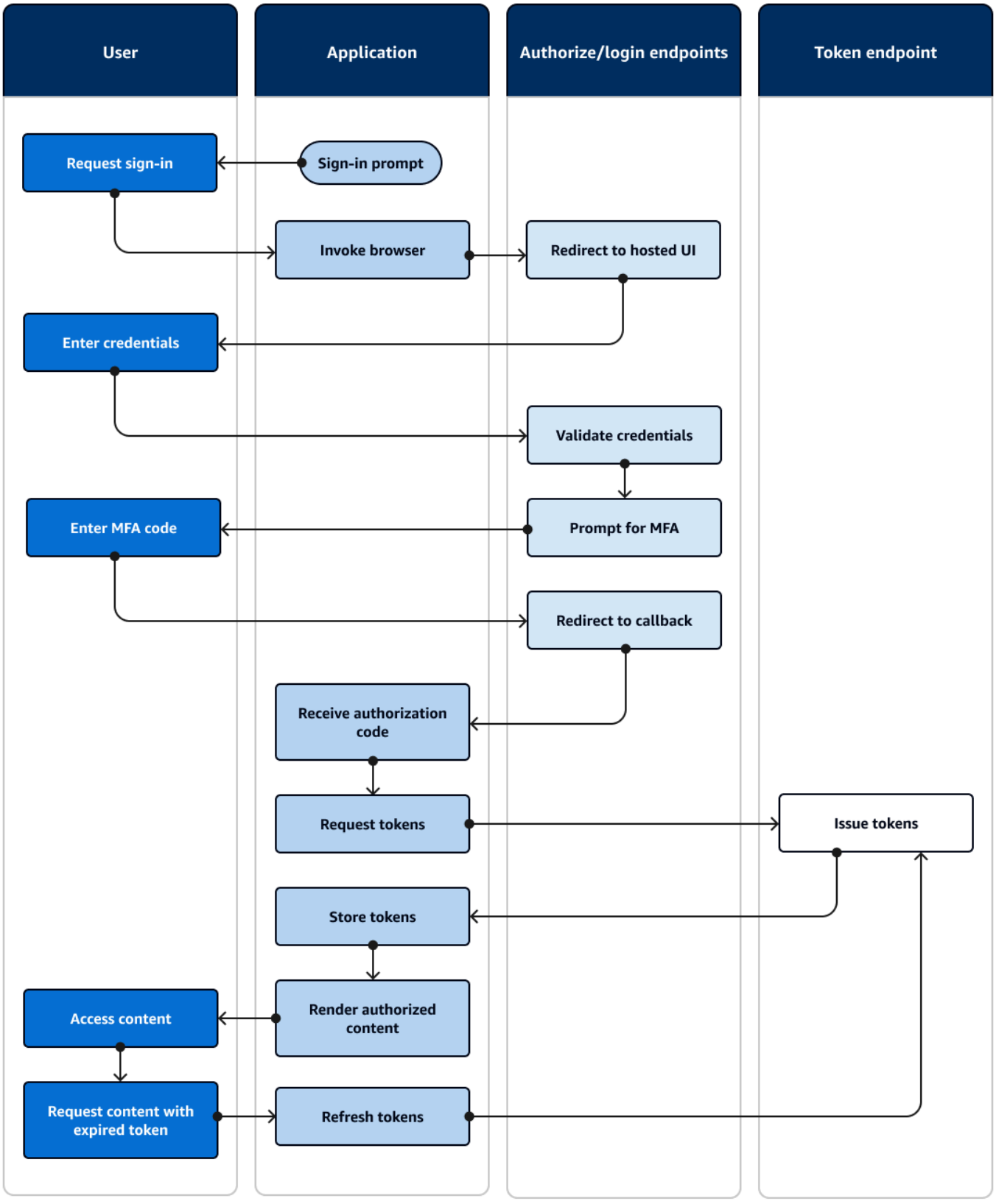
La interfaz de usuario alojada es un sitio web que está vinculado a su grupo de usuarios y al cliente de la aplicación. Puede realizar operaciones de inicio de sesión, registro y restablecimiento de contraseñas para sus usuarios. La implementación de una aplicación con un componente de interfaz de usuario alojado para la autenticación puede requerir menos esfuerzo por parte del desarrollador. Una aplicación puede omitir los componentes de la interfaz de usuario para la autenticación e invocar la interfaz de usuario alojada en el navegador del usuario.

Las aplicaciones recopilan los JWT de los usuarios con una ubicación de redireccionamiento web o de aplicación. Las aplicaciones que implementan la interfaz de usuario alojada pueden conectarse a grupos de usuarios para autenticarse como si se tratara de un IdP de OpenID Connect (OIDC).

La autenticación de la interfaz de usuario alojada se ajusta al modelo en el que las aplicaciones necesitan un servidor de autorización, pero no necesitan funciones como la autenticación personalizada, la integración de grupos de identidades o el autoservicio de atributos de usuario. Si desea utilizar algunas de estas opciones avanzadas, puede implementarlas con un componente de grupos de usuarios para un SDK.

La interfaz de usuario alojada y los modelos de autenticación de IdP de terceros, que se basan principalmente en la implementación de OIDC, son los mejores para los modelos de autorización avanzados con alcances de OAuth 2.0.

El siguiente diagrama ilustra una sesión de inicio de sesión típica para la autenticación de la interfaz de usuario alojada.


User
Application
Authorize/login endpoints
Token endpoint
Request sign-in
Sign-in prompt
Invoke browser
Redirect to hosted UI
Enter credentials
Validate credentials
Enter MFA code
Prompt for MFA
Redirect to callback
Receive authorization code
Request tokens
Issue tokens
Store tokens
Access content
Render authorized content
Request content with expired token
Refresh tokens

Flujo de autenticación de la interfaz de usuario

1. Un usuario accede a tu aplicación.
2. Seleccionan un enlace de «Iniciar sesión».
3. La aplicación dirige al usuario a una ventana de inicio de sesión alojada en la interfaz de usuario.
4. Introduce su nombre de usuario y contraseña.
5. El grupo de usuarios valida las credenciales del usuario y determina que el usuario ha activado la autenticación multifactor (MFA).
6. La interfaz de usuario alojada solicita al usuario que introduzca un código MFA.
7. El usuario introduce su código MFA.
8. La interfaz de usuario alojada redirige al usuario a la aplicación.
9. La aplicación recopila el código de autorización del parámetro de solicitud de URL que la interfaz de usuario alojada adjuntó a la URL de devolución de llamada.
10. La aplicación solicita los tokens con el código de autorización.
11. El punto final del token devuelve los JWT a la aplicación.
12. La aplicación decodifica, valida y almacena o almacena en caché los JWT del usuario.
13. La aplicación muestra el componente de acceso controlado solicitado.
14. El usuario ve su contenido.
15. Más tarde, el token de acceso del usuario ha caducado y este solicita ver un componente de acceso controlado.
16. La aplicación determina que la sesión del usuario debe persistir. Solicita nuevos tokens desde el punto final del token con el token de actualización.

Variantes y personalización

Puedes personalizar el aspecto de la interfaz de usuario alojada con CSS en cualquier cliente de aplicaciones. También puede configurar los clientes de aplicaciones con sus propios proveedores de identidad, ámbitos, acceso a los atributos de usuario y configuración de seguridad avanzada.

Recursos relacionados

- Configuración y uso de la interfaz de usuario alojada y los puntos de conexión de federación de Amazon Cognito
- Registro e inicio de sesión con la interfaz de usuario alojada

- Autorización de alcances, M2M y API con servidores de recursos
- Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada

Autenticación de grupos de usuarios con un proveedor de identidad externo

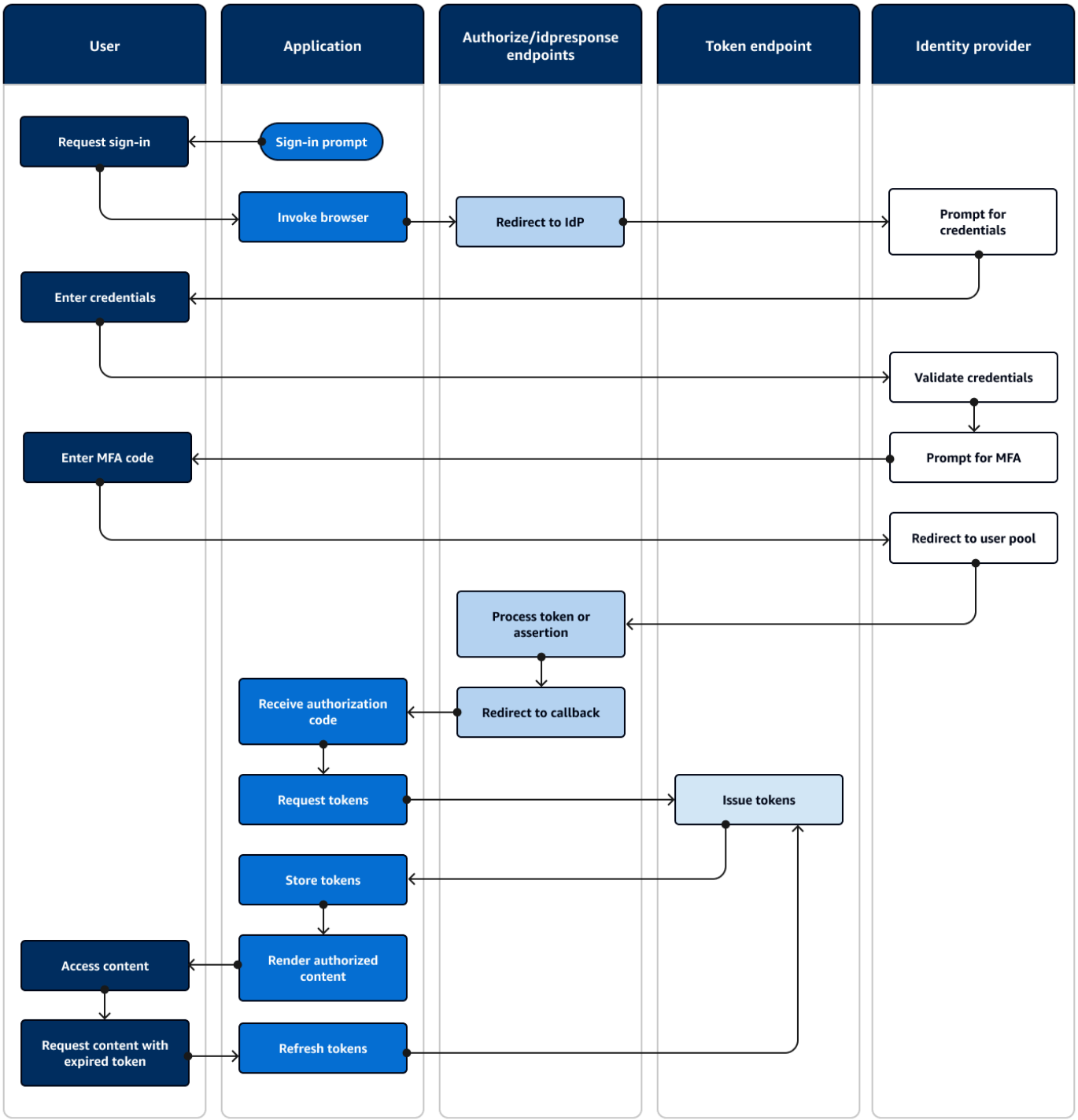
El inicio de sesión con un proveedor de identidad (IdP) externo, o autenticación federada, es un modelo similar al de la interfaz de usuario alojada. Su aplicación es una parte que depende del OIDC para su grupo de usuarios, mientras que su grupo de usuarios sirve de acceso a un IdP. El IdP puede ser un directorio de usuarios de consumidores, como Facebook o Google, o puede ser un directorio empresarial de SAML 2.0 o OIDC, como Azure.

En lugar de utilizar la interfaz de usuario alojada en el navegador del usuario, la aplicación invoca un punto final de redireccionamiento en el servidor de autorización del grupo de usuarios. Desde el punto de vista del usuario, este elige el botón de inicio de sesión de la aplicación. A continuación, su IdP les pide que inicien sesión. Al igual que ocurre con la autenticación de la interfaz de usuario alojada, una aplicación recopila los JWT en una ubicación de redireccionamiento de la aplicación.

La autenticación con un IdP de terceros se ajusta a un modelo en el que es posible que los usuarios no quieran crear una nueva contraseña cuando se registren en su aplicación. La autenticación de terceros se puede añadir con poco esfuerzo a una aplicación que haya implementado la autenticación de interfaz de usuario alojada. En efecto, la interfaz de usuario alojada y la de terceros IdPs generan un resultado de autenticación coherente a partir de pequeñas variaciones en lo que se invoca en los navegadores de los usuarios.

Al igual que la autenticación de interfaz de usuario alojada, la autenticación federada es la mejor para los modelos de autorización avanzados con alcances de OAuth 2.0.

El siguiente diagrama ilustra una sesión de inicio de sesión típica para la autenticación federada.



Flujo de autenticación federada

1. Un usuario accede a su aplicación.
2. Seleccionan un enlace de «Iniciar sesión».

3. La aplicación dirige al usuario a un mensaje de inicio de sesión con su IdP.
4. Introducen su nombre de usuario y contraseña.
5. El IdP valida las credenciales del usuario y determina que el usuario ha activado la autenticación multifactor (MFA).
6. El IdP solicita al usuario que introduzca un código MFA.
7. El usuario introduce su código MFA.
8. El IdP redirige al usuario al grupo de usuarios con una respuesta SAML o un código de autorización.
9. Si el usuario pasó un código de autorización, el grupo de usuarios intercambia silenciosamente el código por los tokens de IdP. El grupo de usuarios valida los tokens de IdP y redirige al usuario a la aplicación con un nuevo código de autorización.
10. La aplicación recopila el código de autorización del parámetro de solicitud de URL que el grupo de usuarios ha agregado a la URL de devolución de llamada.
11. La aplicación solicita los tokens con el código de autorización.
12. El punto final del token devuelve los JWT a la aplicación.
13. La aplicación decodifica, valida y almacena o almacena en caché los JWT del usuario.
14. La aplicación muestra el componente de acceso controlado solicitado.
15. El usuario ve su contenido.
16. Más tarde, el token de acceso del usuario ha caducado y este solicita ver un componente de acceso controlado.
17. La aplicación determina que la sesión del usuario debe persistir. Solicita nuevos tokens desde el punto final del token con el token de actualización.

Variantes y personalización

Puedes iniciar la autenticación federada en la interfaz de usuario alojada, donde los usuarios pueden elegir entre una lista de las IdPs que hayas asignado a tu cliente de aplicación. La interfaz de usuario alojada también puede solicitar una dirección de correo electrónico y dirigir automáticamente la solicitud de un usuario al IDP de SAML correspondiente. La autenticación con un proveedor de identidad externo no requiere la interacción del usuario con la interfaz de usuario alojada. La aplicación puede agregar un parámetro de solicitud a la solicitud del servidor de autorización de un usuario y hacer que el usuario lo redirija silenciosamente a su página de inicio de sesión de IdP.

Recursos relacionados

- Agregar inicio de sesión de grupo de usuarios a través de un tercero
- Escenario de ejemplo: marcar aplicaciones de Amazon Cognito en un panel empresarial
- Autorización de alcances, M2M y API con servidores de recursos
- Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada

Autenticación de grupos de

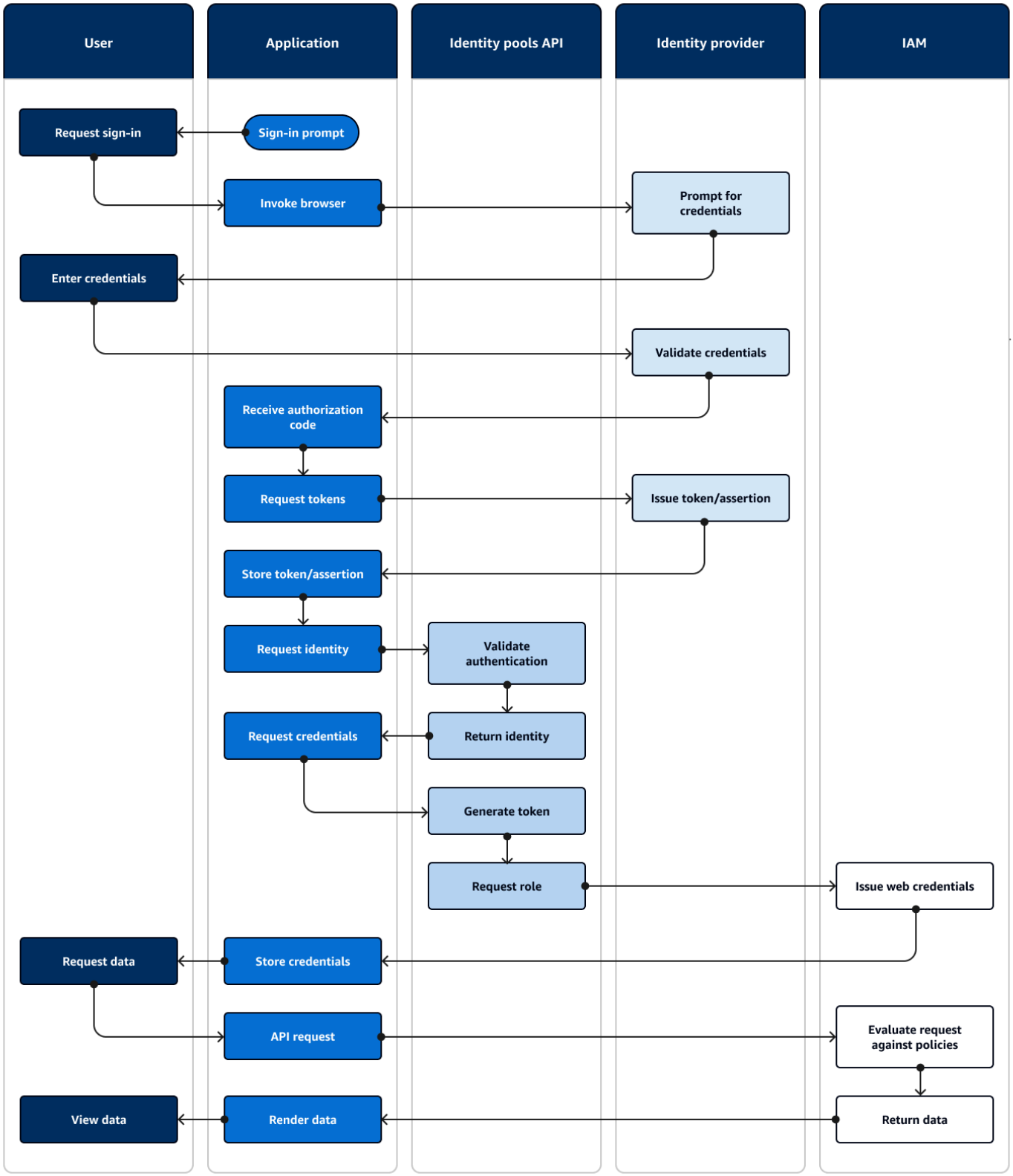
Un grupo de identidades es un componente de la aplicación que se diferencia de un grupo de usuarios en cuanto a la función, el espacio de nombres de la API y el modelo de SDK. Mientras que los grupos de usuarios ofrecen autenticación y autorización basadas en tokens, los grupos de identidades ofrecen autorización para AWS Identity and Access Management (IAM).

Puede asignar un conjunto de grupos de IdPs identidades e iniciar sesión con los usuarios con ellos. Los grupos de usuarios están estrechamente integrados como grupos de identidades IdPs y ofrecen a los grupos de identidades la mayoría de las opciones de control de acceso. Al mismo tiempo, existe una amplia selección de opciones de autenticación para los grupos de identidades. Los grupos de usuarios se unen a fuentes de identidad de SAML, OIDC, redes sociales, desarrolladores e invitados como rutas hacia las AWS credenciales temporales de los grupos de identidades.

La autenticación con un grupo de identidades es externa: sigue uno de los flujos del grupo de usuarios ilustrados anteriormente o un flujo que se desarrolla de forma independiente con otro IdP. Una vez que la aplicación realiza la autenticación inicial, pasa la prueba a un grupo de identidades y, a cambio, recibe una sesión temporal.

La autenticación con un grupo de identidades se ajusta a un modelo en el que se aplica el control de acceso a los activos y datos de las aplicaciones Servicios de AWS con la autorización de IAM. Al igual que ocurre con la autenticación mediante API en grupos de usuarios, una aplicación eficaz incluye AWS los SDK para cada uno de los servicios a los que desee acceder en beneficio de sus usuarios. AWS Los SDK aplican las credenciales de la autenticación del grupo de identidades como firmas a las solicitudes de API.

El siguiente diagrama ilustra una sesión de inicio de sesión típica para la autenticación del grupo de identidades con un IdP.


User
Application
Identity pools API
Identity provider
IAM
Request sign-in
Sign-in prompt
Invoke browser
Prompt for credentials
Enter credentials
Validate credentials
Receive authorization code
Request tokens
Issue token/assertion
Store token/assertion
Request identity
Validate authentication
Request credentials
Return identity
Generate token
Request role
Issue web credentials
Request data
Store credentials
API request
Evaluate request against policies
View data
Render data
Return data

Flujo de autenticación federada

1. Un usuario accede a su aplicación.
2. Seleccionan un enlace de «Iniciar sesión».
3. La aplicación dirige al usuario a un mensaje de inicio de sesión con su IdP.
4. Introducen su nombre de usuario y contraseña.
5. El IdP valida las credenciales del usuario.
6. El IdP redirige al usuario a la aplicación con una respuesta SAML o un código de autorización.
7. Si el usuario pasó un código de autorización, la aplicación lo intercambia por tokens de IdP.
8. La aplicación decodifica, valida y almacena o almacena en caché los JWT o la afirmación del usuario.
9. La aplicación invoca el método que realiza una solicitud a la API. GetId Transmite el token o la afirmación del usuario y solicita un identificador de identidad.
10. El grupo de identidades valida el token o la afirmación con respecto a los proveedores de identidad configurados.
11. El grupo de identidades devuelve un identificador de identidad.
12. La aplicación invoca el método que realiza una solicitud de GetCredentialsForIdentityAPI. Transmite el token o las afirmaciones del usuario y solicita una función de IAM.
13. El grupo de identidades genera un nuevo JWT. El nuevo JWT contiene notificaciones que solicitan una función de IAM. El grupo de identidades determina el rol en función de la solicitud del usuario y los criterios de selección de roles en la configuración del grupo de identidades para el IdP.
14. AWS Security Token Service (AWS STS) responde a la AssumeRoleWithWebIdentitysolicitud del grupo de identidades. La respuesta contiene las credenciales de API para una sesión temporal con una función de IAM.
15. La aplicación almacena las credenciales de sesión.
16. El usuario realiza una acción en la aplicación que requiere la entrada de recursos con acceso protegido. AWS
17. La aplicación aplica las credenciales temporales como firmas a las solicitudes de API en caso de que sea necesario. Servicios de AWS
18. IAM evalúa las políticas asociadas a la función en las credenciales. Las compara con la solicitud.
19. Servicio de AWS Devuelve los datos solicitados.
20. La aplicación renderiza los datos en la interfaz de usuario.

21El usuario ve los datos.

Variantes y personalización

Para visualizar la autenticación con un grupo de usuarios, inserte una de las descripciones generales del grupo de usuarios anteriores después del paso Emitir el token o la aserción. La autenticación del desarrollador reemplaza todos los pasos anteriores a la solicitud de identidad por una solicitud firmada con las credenciales del desarrollador. La autenticación de invitado también pasa directamente a Solicitar identidad, no valida la autenticación y devuelve las credenciales para una función de IAM de acceso limitado.

Recursos relacionados

- Grupos de identidades de Amazon Cognito
- Roles de IAM de usuario
- Conceptos de grupos de identidades
- Flujo de autenticación de grupos de identidades (identidades federadas)

Términos de Amazon Cognito

Amazon Cognito proporciona credenciales para aplicaciones web y móviles. Se basa en términos que son comunes en la administración de identidades y accesos y se basa en ellos. Están disponibles muchas guías sobre los términos de identidad y acceso universales. Algunos ejemplos son:

- La terminología en el acervo de conocimientos de IDPro
- AWS Servicios de identidad
- Glosario del NIST CSRC

En las siguientes listas se describen los términos que son exclusivos de Amazon Cognito o que tienen un contexto específico en Amazon Cognito.

Temas

- General
- Grupos de usuarios

- Grupos de identidades

General

Los términos de esta lista no son específicos de Amazon Cognito y son ampliamente reconocidos entre los profesionales de la gestión de identidades y accesos. La siguiente no es una lista exhaustiva de términos, sino una guía sobre su contexto específico de Amazon Cognito en esta guía.

App

Normalmente, una aplicación móvil. En esta guía, aplicación suele ser una forma abreviada de una aplicación web o móvil que se conecta a Amazon Cognito.

Control de acceso basado en atributos (ABAC)

Modelo en el que una aplicación determina el acceso a los recursos en función de las propiedades de un usuario, como su cargo o departamento. Las herramientas de Amazon Cognito para aplicar el ABAC incluyen tokens de identificación en los grupos de usuarios y etiquetas principales en los grupos de identidades.

Servidor de autorización

Un sistema basado en la web que genera tokens web JSON. Los puntos finales de federación de grupos de usuarios de Amazon Cognito son el componente del servidor de autorización de los dos métodos de autenticación y autorización de los grupos de usuarios. El otro método es la API de grupos de usuarios.

Aplicación confidencial, aplicación del lado del servidor

Una aplicación a la que los usuarios se conectan de forma remota, con el código en un servidor de aplicaciones y acceso a secretos. Suele ser una aplicación web.

Identity provider (IdP) (Proveedor de identidad (IdP))

Un servicio que almacena y verifica las identidades de los usuarios. Amazon Cognito puede solicitar la autenticación a proveedores externos y ser un IdP para las aplicaciones.

Token web JSON (JWT)

Documento con formato JSON que contiene afirmaciones sobre un usuario autenticado. Los identificadores autentican a los usuarios, los de acceso los autorizan y los de actualización actualizan las credenciales. Amazon Cognito recibe tokens de proveedores externos y los envía a aplicaciones o. AWS STS

Autenticación multifactor (MFA)

El requisito de que los usuarios proporcionen una autenticación adicional después de proporcionar su nombre de usuario y contraseña. Los grupos de usuarios de Amazon Cognito tienen funciones de MFA para los usuarios locales.

Proveedor de OAuth 2.0 (social)

Un IdP para un grupo de usuarios o un grupo de identidades que proporciona acceso a JWT y actualiza los tokens. Los grupos de usuarios de Amazon Cognito automatizan las interacciones con los proveedores de redes sociales una vez que los usuarios se autentican.

Proveedor de OpenID Connect (OIDC)

Un IdP para un grupo de usuarios o un grupo de identidades que amplía la especificación de OAuth para proporcionar tokens de identificación. Los grupos de usuarios de Amazon Cognito automatizan las interacciones con los proveedores de OIDC una vez que los usuarios se autentican.

Aplicación pública

Aplicación autónoma en un dispositivo, con el código almacenado localmente y sin acceso a datos secretos. Suele ser una aplicación móvil.

Servidor de recursos

Una API con control de acceso. Los grupos de usuarios de Amazon Cognito también utilizan el servidor de recursos para describir el componente que define la configuración para interactuar con una API.

Control de acceso basado en roles (RBAC)

Modelo que concede el acceso en función de la designación funcional de un usuario. Los grupos de identidades de Amazon Cognito implementan RBAC diferenciando las funciones de IAM.

Proveedor de servicios (SP), parte de confianza (RP)

Una aplicación que se basa en un IdP para afirmar que los usuarios son confiables. Amazon Cognito actúa como un SP para los SP externos IdPs y como un IdP para los SPs basados en aplicaciones.

Proveedor de SAML

Un IdP para un grupo de usuarios o un grupo de identidades que genera documentos de afirmación firmados digitalmente que el usuario pasa a Amazon Cognito.

Identificador único universal (UUID)

Etiqueta de 128 bits que se aplica a un objeto. Los UUID de Amazon Cognito son únicos por grupo de usuarios o grupo de identidades.

Directorio de usuarios

Conjunto de usuarios y sus atributos que envía esa información a otros sistemas. Los grupos de usuarios de Amazon Cognito son directorios de usuarios y también herramientas para consolidar usuarios de directorios de usuarios externos.

Grupos de usuarios

Si ve los términos de la siguiente lista de esta guía, se refieren a una característica o configuración específica de los grupos de usuarios.

API de grupos de usuarios de Amazon Cognito

Un conjunto de operaciones de API de autenticación y autorización que puede añadir a su aplicación con un AWS SDK. La API puede iniciar sesión tanto a usuarios locales como a usuarios vinculados.

Autenticación flexible

Función de seguridad avanzada que detecta posibles actividades maliciosas y aplica seguridad adicional a los perfiles de usuario.

Funciones de seguridad avanzadas

Un componente opcional que añade herramientas para la seguridad de los usuarios.

Cliente de aplicaciones

Componente que define la configuración de un grupo de usuarios como un IdP de una aplicación.

URL de devolución de llamada, URI de redireccionamiento

Una configuración en un cliente de aplicaciones y un parámetro en las solicitudes a los puntos finales de la federación de grupos de usuarios. La URL de devolución de llamada es el destino inicial de los usuarios autenticados de tu aplicación.

Credenciales comprometidas

Una función de seguridad avanzada que detecta las contraseñas de los usuarios que los atacantes podrían conocer y aplica medidas de seguridad adicionales a los perfiles de los usuarios.

Confirmación

Proceso que determina que se han cumplido los requisitos previos para permitir que un nuevo usuario inicie sesión. Por lo general, la confirmación se realiza mediante la verificación de la dirección de correo electrónico o el número de teléfono.

Autenticación personalizada

Una extensión de los procesos de autenticación con activadores Lambda que definen desafíos y respuestas adicionales para los usuarios.

autenticación de dispositivos

Proceso de autenticación que reemplaza la MFA por un inicio de sesión que usa el ID de un dispositivo de confianza.

Proveedor externo, proveedor externo

Un IdP que tiene una relación de confianza con un grupo de usuarios.

Usuario federado

Usuario de un grupo de usuarios autenticado por un proveedor externo.

Puntos finales de federación

Un conjunto de páginas web en el dominio de tu grupo de usuarios que alojan servicios para interactuar con aplicaciones IdPs y aplicaciones.

IU alojada

Un conjunto de páginas web interactivas en el dominio de tu grupo de usuarios que alojan servicios para la autenticación de usuarios.

Disparador de Lambda

Función AWS Lambda que un grupo de usuarios puede invocar automáticamente en puntos clave de los procesos de autenticación de usuarios. Puede usar activadores Lambda para personalizar los resultados de la autenticación.

Usuario local

Un perfil de usuario en el directorio de usuarios del grupo de usuarios que no se creó mediante la autenticación con un proveedor externo.

Usuario vinculado

Usuario de un proveedor externo cuya identidad se fusiona con la de un usuario local.

Personalización de tokens

El resultado de un activador Lambda previo a la generación del token que modifica el identificador de usuario o el token de acceso en tiempo de ejecución.

Grupo de usuarios, proveedor de identidad de Amazon Cognito**`cognito-idp`**, grupos de usuarios de Amazon Cognito

Un AWS recurso con servicios de autenticación y autorización para aplicaciones que funcionan con OIDC. IdPs

Dominio de grupo de usuarios

Un nombre de sitio web que se agrega a un grupo de usuarios. El dominio es la URL base de la interfaz de usuario alojada y los puntos finales de la federación.

Verificación

El proceso de confirmar que un usuario es propietario de una dirección de correo electrónico o un número de teléfono. Un grupo de usuarios envía un código a un usuario que ha introducido una nueva dirección de correo electrónico o número de teléfono. Cuando envían el código a Amazon Cognito, comprueban que son propietarios del destino del mensaje y pueden recibir mensajes adicionales del grupo de usuarios. Consulte también la confirmación.

Perfil de usuario, cuenta de usuario

Entrada de un usuario en el directorio de usuarios. Todos los usuarios tienen un perfil en su grupo de usuarios.

Grupos de identidades

Cuando vea los términos de la siguiente lista de esta guía, se refieren a una función o configuración específica de los grupos de identidades.

Atributos para controlar el acceso

Implementación del control de acceso basado en atributos en los grupos de identidades. Los grupos de identidades aplican los atributos de los usuarios como etiquetas a las credenciales de los usuarios.

Autenticación básica (clásica)

Un proceso de autenticación en el que puede personalizar la solicitud de credenciales de usuario.

Identidades autenticadas por el desarrollador

Un proceso de autenticación que autoriza las credenciales de usuario del grupo de identidades con las credenciales de desarrollador.

Credenciales de desarrollador

Las claves de la API de IAM de un administrador de grupos de identidades.

Autenticación mejorada

Un flujo de autenticación que selecciona un rol de IAM y aplica las etiquetas principales según la lógica que defina en su grupo de identidades.

Identidad

Un UUID que vincula a un usuario de la aplicación y sus credenciales de usuario a su perfil en un directorio de usuarios externo que tiene una relación de confianza con un grupo de identidades.

Grupo de identidades, identidades federadas de Amazon Cognito, identidad de Amazon Cognito, **`cognito-identity`**

Un AWS recurso con servicios de autenticación y autorización para aplicaciones que utilizan credenciales temporales. AWS

Identidad de no autenticada

Un usuario que no ha iniciado sesión con un IdP de grupo de identidades. Puede permitir que los usuarios generen credenciales de usuario limitadas para un único rol de IAM antes de autenticarse.

Credenciales de usuario

Claves AWS de API temporales que los usuarios reciben tras la autenticación del grupo de identidades.

Uso de este servicio con un SDK AWS

AWS Los kits de desarrollo de software (SDK) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK for C++	AWS SDK for C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK for Go	AWS SDK for Go ejemplos de código
AWS SDK for Java	AWS SDK for Java ejemplos de código
AWS SDK for JavaScript	AWS SDK for JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK for .NET	AWS SDK for .NET ejemplos de código
AWS SDK for PHP	AWS SDK for PHP ejemplos de código
AWS Tools for PowerShell	Herramientas para ejemplos PowerShell de código
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) ejemplos de código
AWS SDK for Ruby	AWS SDK for Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Ejemplo de disponibilidad

¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Empezando con AWS

Antes de empezar a trabajar con Amazon Cognito, prepárese con algunos recursos necesarios AWS .

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de registrarte en un usuario Cuenta de AWS, protege Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilita y crea un usuario administrativo para que no utilices el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Introducción a los grupos de usuarios

Puede usar las guías de esta sección para crear los recursos de su grupo de usuarios iniciales. Para ver un step-by-step tutorial, comienza con una aplicación web básica en el entorno de JavaScript desarrollador de React. A partir de ahí, puedes continuar añadiendo funciones como la interfaz de usuario alojada (interfaz de usuario alojada) y el inicio de sesión federado con proveedores de identidad externos de redes sociales o SAML 2.0 (). IdPs

A medida que vaya ampliando su conjunto de características e incorporando más componentes de Amazon Cognito, lea el capítulo sobre los grupos de usuarios de Amazon Cognito para obtener una descripción completa de todo lo que puede hacer con los grupos de usuarios.

El grupo de usuarios y la aplicación de ejemplo de esta sección muestran una integración básica de los recursos de la aplicación con los grupos de usuarios de Amazon Cognito. Más adelante, podrá ajustar su grupo de usuarios para utilizar más opciones de las que dispone. Luego, puede actualizar su aplicación para adoptar nuevas API e interactuar con la interfaz de usuario alojada y IdPs.

El tutorial de esta sección crea una aplicación con una interfaz de usuario personalizada y una autenticación basada en API con un AWS SDK. Las aplicaciones que se crean de esta manera son ideales para autenticar a los usuarios locales. Para empezar con una aplicación con una interfaz de usuario prediseñada, gestión automática de algunas funciones del grupo de usuarios y autenticación de usuarios federados, pase a. Agregue un cliente de aplicaciones con la interfaz de usuario alojada

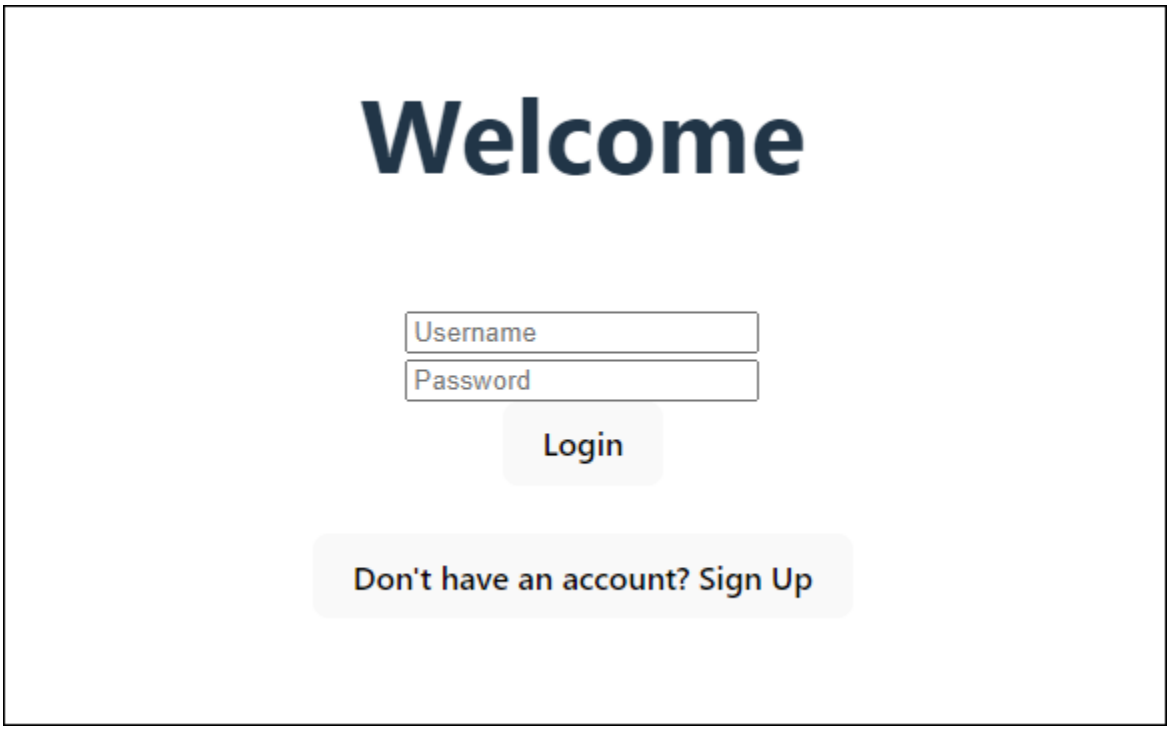
Temas

- Configura un ejemplo de aplicación React de una sola página
- Configura un ejemplo de aplicación de Android con Flutter
- Siguientes pasos

Configura un ejemplo de aplicación React de una sola página

En este tutorial, crearás una aplicación React de una sola página en la que podrás probar el registro, la confirmación y el inicio de sesión de los usuarios. React es una biblioteca JavaScript basada en aplicaciones web y móviles, que se centra en la interfaz de usuario (UI). Esta aplicación de ejemplo muestra algunas funciones básicas de los grupos de usuarios de Amazon Cognito. Si ya tiene experiencia en el desarrollo de aplicaciones web con React, descargue la aplicación de ejemplo desde GitHub.

La siguiente captura de pantalla es de la página de autenticación inicial de la aplicación que crearás.



El procedimiento de creación de un grupo de usuarios permite configurar un grupo de usuarios que funciona con la aplicación de ejemplo. Puede omitir este paso si tiene un grupo de usuarios que cumpla los siguientes requisitos:

- Los usuarios pueden iniciar sesión con su dirección de correo electrónico. Opciones de inicio de sesión del grupo de usuarios de Cognito: correo electrónico.
- Los nombres de usuario no distinguen mayúsculas de minúsculas. Requisitos de nombre de usuario: no está seleccionada la opción Hacer que el nombre de usuario distinga mayúsculas de minúsculas.
- No se requiere la autenticación multifactor (MFA). Aplicación de la MFA: MFA opcional.
- Su grupo de usuarios verifica los atributos para la confirmación del perfil de usuario mediante un mensaje de correo electrónico. Atributos a verificar: envíe un mensaje de correo electrónico, verifique la dirección de correo electrónico.
- El correo electrónico es el único atributo obligatorio. Atributos obligatorios: correo electrónico.
- Los usuarios pueden registrarse ellos mismos en su grupo de usuarios. Registro automático: está seleccionada la opción Habilitar el registro automático.
- El cliente de la aplicación inicial es un cliente público que permite iniciar sesión con un nombre de usuario y una contraseña. Tipo de aplicación: cliente público, Flujos de autenticación:`ALLOW_USER_PASSWORD_AUTH`.

Crear un grupo de usuarios

Crear un nuevo grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Pulse el botón Crear grupo de usuarios. Puede que tenga que seleccionar Grupos de usuarios en el panel de navegación izquierdo para que aparezca esta opción.
3. En la esquina superior derecha de la página, elija Create a User Pool (Crear un grupo de usuarios).
4. En Configurar la experiencia de inicio de sesión, puede elegir los proveedores de identidad (IdPs) que utilizará con este grupo de usuarios. Para obtener más información, consulte Agregar inicio de sesión de grupo de usuarios a través de un tercero.
   a. En Proveedores de autenticación, para los tipos de proveedores, asegúrese de que solo esté seleccionado el grupo de usuarios de Cognito.
   b. Para ver las opciones de inicio de sesión del grupo de usuarios de Cognito, elija Nombre de usuario. No seleccione ningún requisito de nombre de usuario adicional.
   c. Mantén todas las demás opciones como predeterminadas y selecciona Siguiente.
5. En Configurar requisitos de seguridad, puede elegir su política de contraseñas, los requisitos de autenticación multifactor (MFA) y las opciones de recuperación de cuentas de usuario. Para obtener más información, consulte Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito.
   a. Para la política de contraseñas, confirme que el modo de política de contraseñas esté establecido en los valores predeterminados de Cognito.
   b. En Autenticación multifactor, para aplicar el MFA, elija MFA opcional.
   c. Para los métodos de MFA, selecciona Aplicaciones autenticadoras y mensajes SMS.
   d. Para la recuperación de la cuenta de usuario, confirme que esté seleccionada la opción Habilitar la recuperación automática de cuentas de usuario y que el método de entrega de los mensajes de recuperación de la cuenta de usuario esté configurado como Solo correo electrónico.
   e. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.
6. En Configurar la experiencia de registro, puede determinar cómo verificarán sus identidades los nuevos usuarios al registrarse como nuevos usuarios y qué atributos deben ser obligatorios

u opcionales durante el proceso de registro de los usuarios. Para obtener más información, consulte Administración de usuarios en el grupo de usuarios.

a. Confirme que esté seleccionada la opción Habilitar el registro automático. Esta configuración permite que cualquier usuario de Internet se registre en tu grupo de usuarios. Esto está pensado para los fines de la aplicación de ejemplo, pero aplique esta configuración con precaución en los entornos de producción.

b. En Verificación y confirmación asistidas por Cognito, compruebe que la casilla Permitir que Cognito envíe mensajes automáticamente para verificar y confirmar esté seleccionada.

c. Confirme que los atributos a verificar estén configurados en Enviar mensaje de correo electrónico, verificar dirección de correo electrónico.

d. En Verificar los cambios de atributos, confirme que estén seleccionadas las opciones predeterminadas: se selecciona Conservar el valor del atributo original cuando hay una actualización pendiente y los valores de los atributos activos cuando hay una actualización pendiente se establece en Dirección de correo electrónico.

e. En Atributos obligatorios, confirme que los atributos obligatorios basados en selecciones anteriores muestren el correo electrónico.

> **Important**
>
> Para esta aplicación de ejemplo, su grupo de usuarios no debe establecer phone_number como atributo obligatorio. Si se muestra el número de teléfono como atributo obligatorio, revise y actualice las opciones anteriores:
>
> - MFA opcional, solo correo electrónico para el método de entrega de los mensajes de recuperación de cuentas de usuario
> - Envía un mensaje de correo electrónico, verifica la dirección de correo electrónico para que Attributes la verifique

f. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

7. En Configurar la entrega de mensajes, puede configurar la integración con Amazon Simple Email Service y Amazon Simple Notification Service para enviar mensajes de correo electrónico y SMS a sus usuarios para que se registren, confirmen la cuenta, MFA y recuperen la cuenta. Para obtener más información, consulte Configuración de correo electrónico para grupos de usuarios de Amazon Cognito y Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito.

a. En Proveedor de correo electrónico, elija Enviar correo electrónico con Cognito y utilice el remitente de correo electrónico predeterminado que proporciona Amazon Cognito. Esta configuración de bajo volumen de correo electrónico es suficiente para probar la aplicación. Puedes realizar la devolución después de verificar una dirección de correo electrónico con Amazon Simple Email Service (Amazon SES) y seleccionar Enviar correo electrónico con Amazon SES.

b. En el caso de SMS, selecciona Crear una nueva función de IAM e introduce el nombre de una función de IAM. Esto crea un rol que concede permisos a Amazon Cognito para enviar mensajes SMS.

c. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

8. En Integrate your app, puedes asignar un nombre a tu grupo de usuarios, configurar la interfaz de usuario alojada y crear un cliente de aplicación. Para obtener más información, consulte Agregue un cliente de aplicaciones con la interfaz de usuario alojada. Las aplicaciones de ejemplo no utilizan la interfaz de usuario alojada.

   a. En Nombre del grupo de usuarios, introduzca un nombre de grupo de usuarios.

   b. No selecciones Usar la interfaz de usuario alojada en Cognito.

   c. En Cliente de aplicación inicial, confirme que el tipo de aplicación esté configurado como Cliente público.

   d. En Secreto de cliente, confirma que esté seleccionada la opción No generar un secreto de cliente.

   e. Introduzca un nombre de cliente de aplicación.

   f. Amplíe la configuración avanzada del cliente de la aplicación. Añádalo `ALLOW_USER_PASSWORD_AUTH` a la lista de flujos de autenticación.

   g. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

9. Revise sus opciones en la pantalla Revisar y crear y modifique las selecciones según sea necesario. Cuando esté satisfecho con la configuración del grupo de usuarios, elija Crear grupo de usuarios para continuar.

10. En la página Grupos de usuarios, elija su nuevo grupo de usuarios.

11. En Descripción general del grupo de usuarios, anote su ID de grupo de usuarios. Proporcionarás esta cadena cuando crees tu aplicación de ejemplo.

12. Elija la pestaña Integración de aplicaciones y busque la sección de análisis y clientes de aplicaciones. Selecciona tu nuevo cliente de aplicaciones. Anote su ID de cliente.

Recursos relacionados

- Grupos de usuarios de Amazon Cognito
- Flujo de autenticación de los grupos de usuarios
- Uso de tokens con grupos de usuarios

Crear una aplicación

Para crear esta aplicación, debe configurar un entorno de desarrollador. Los requisitos del entorno de desarrollador son:

1. Node.js está instalado y actualizado.
2. El administrador de paquetes de nodos (npm) está instalado y actualizado al menos a la versión 10.2.3.
3. Se puede acceder al entorno en el puerto TCP 5173 de un navegador web.

Para crear un ejemplo de aplicación web de React

1. Inicia sesión en tu entorno de desarrollador y navega hasta el directorio principal de tu aplicación.

```
cd ~/path/to/project/folder/
```

2. Crea un nuevo servicio de React.

```
npm create vite@latest frontend-client -- --template react-ts
```

3. Clona la carpeta del `cognito-developer-guide-react-example` proyecto desde el repositorio AWS de ejemplos de código en GitHub.

```
cd ~/some/other/path
```

```
git clone https://github.com/awsdocs/aws-doc-sdk-examples.git
```

```
cp -r ./aws-doc-sdk-examples/javascriptv3/example_code/cognito-identity-provider/
scenarios/cognito-developer-guide-react-example/frontend-client ~/path/to/project/
folder/frontend-client
```

4. Navegue hasta el `src` directorio de su proyecto.

```
cd ~/path/to/project/folder/frontend-client/src
```

5. Edita `config.ts` y reemplaza los siguientes valores:

   a. `YOUR_AWS_REGION`Sustitúyalos por un Región de AWS código. Por ejemplo: `us-east-1`.

   b. `YOUR_COGNITO_USER_POOL_ID`Sustitúyalo por el ID del grupo de usuarios que designaste para las pruebas. Por ejemplo: `us-east-1_EXAMPLE`. El grupo de usuarios debe estar en el Región de AWS que ingresó en el paso anterior.

   c. `YOUR_COGNITO_APP_CLIENT_ID`Sustitúyalo por el ID del cliente de la aplicación que designaste para la prueba. Por ejemplo: `1example23456789`. El cliente de la aplicación debe estar en el grupo de usuarios del paso anterior.

6. Si quiere acceder a la aplicación de ejemplo desde una IP distinta `alocalhost`, edite `package.json` y cambie la línea `"dev": "vite",` a`"dev": "vite --host 0.0.0.0",`.

7. Instala tu aplicación.

```
npm install
```

8. Inicie la aplicación.

```
npm run dev
```

9. Acceda a la aplicación en un navegador web en `http://localhost:5173` o`http://[IP address]:5173`.

10. Registre un nuevo usuario con una dirección de correo electrónico válida.

11. Recupera el código de confirmación de tu mensaje de correo electrónico. Introduzca el código de confirmación en la aplicación.

12. Inicie sesión con su nombre de usuario y contraseña.

Creación de un entorno para desarrolladores de React con Amazon Lightsail

Una forma rápida de empezar a utilizar esta aplicación es crear un servidor virtual en la nube con Amazon Lightsail.

Con Lightsail, puede crear rápidamente una pequeña instancia de servidor que venga preconfigurada con los requisitos previos para esta aplicación de ejemplo. Puede conectarse mediante SSH a su instancia con un cliente basado en un navegador y conectarse al servidor web desde una dirección IP pública o privada.

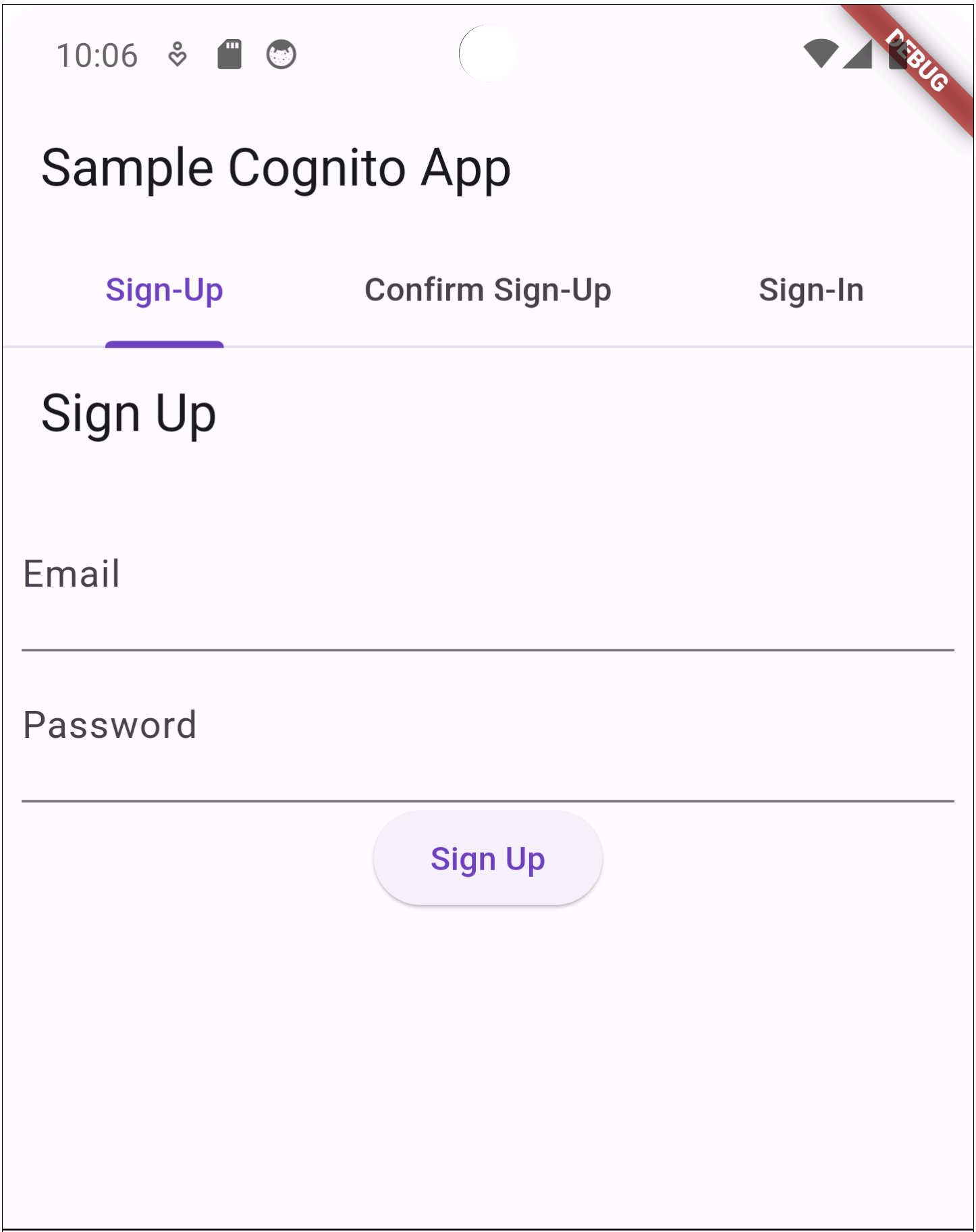
Para crear una instancia de Lightsail para esta aplicación de ejemplo

1. Vaya a la consola Lightsail. Si se le solicita, introduzca sus credenciales. AWS
2. Elija Crear instancia.
3. En Seleccione una plataforma, elija Linux/Unix.
4. En Seleccione un blueprint, elija Node.js.
5. En Identifique su instancia, asigne un nombre descriptivo a su entorno de desarrollo.
6. Elija Crear instancia.
7. Una vez que Lightsail haya creado la instancia, selecciónela y, en la pestaña Connect, elija Connect using SSH.
8. Se abre una sesión SSH en una ventana del navegador. Ejecuta `node -v` y confirma `npm -v` que tu instancia se aprovisionó con Node.js y la versión npm mínima de 10.2.3.
9. Continúe con la configuración de la aplicación React.

Configura un ejemplo de aplicación de Android con Flutter

En este tutorial, crearás una aplicación móvil en Android Studio en la que podrás emular un dispositivo y probar el registro, la confirmación y el inicio de sesión de los usuarios. Esta aplicación de ejemplo crea un cliente móvil básico de grupos de usuarios de Amazon Cognito para Android en Flutter. Si ya tiene experiencia en el desarrollo de aplicaciones móviles con Flutter, descargue la aplicación de ejemplo de GitHub.

La siguiente captura de pantalla muestra la aplicación ejecutándose en un dispositivo Android virtual.


10:06
DEBUG
Sample Cognito App
Sign-Up
Confirm Sign-Up
Sign-In
Sign Up
Email
Password
Sign Up

El procedimiento Crear un grupo de usuarios permite configurar un grupo de usuarios que funciona con la aplicación de ejemplo. Puede omitir este paso si tiene un grupo de usuarios que cumpla los siguientes requisitos:

- Los usuarios pueden iniciar sesión con su dirección de correo electrónico. Opciones de inicio de sesión del grupo de usuarios de Cognito: correo electrónico.
- Los nombres de usuario no distinguen mayúsculas de minúsculas. Requisitos de nombre de usuario: no está seleccionada la opción Hacer que el nombre de usuario distinga mayúsculas de minúsculas.
- No se requiere la autenticación multifactor (MFA). Aplicación de la MFA: MFA opcional.
- Su grupo de usuarios verifica los atributos para la confirmación del perfil de usuario mediante un mensaje de correo electrónico. Atributos a verificar: envíe un mensaje de correo electrónico, verifique la dirección de correo electrónico.
- El correo electrónico es el único atributo obligatorio. Atributos obligatorios: correo electrónico.
- Los usuarios pueden registrarse ellos mismos en su grupo de usuarios. Registro automático: está seleccionada la opción Habilitar el registro automático.
- El cliente de la aplicación inicial es un cliente público que permite iniciar sesión con un nombre de usuario y una contraseña. Tipo de aplicación: cliente público, Flujos de autenticación:`ALLOW_USER_PASSWORD_AUTH`.

Crear un grupo de usuarios

Crear un nuevo grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Pulse el botón Crear grupo de usuarios. Puede que tenga que seleccionar Grupos de usuarios en el panel de navegación izquierdo para que aparezca esta opción.
3. En la esquina superior derecha de la página, elija Create a User Pool (Crear un grupo de usuarios).
4. En Configurar la experiencia de inicio de sesión, puede elegir los proveedores de identidad (IdPs) que utilizará con este grupo de usuarios. Para obtener más información, consulte Agregar inicio de sesión de grupo de usuarios a través de un tercero.

    a. En Proveedores de autenticación, para los tipos de proveedores, asegúrese de que solo esté seleccionado el grupo de usuarios de Cognito.

b. Para ver las opciones de inicio de sesión del grupo de usuarios de Cognito, elija Nombre de usuario. No seleccione ningún requisito de nombre de usuario adicional.

c. Mantén todas las demás opciones como predeterminadas y selecciona Siguiente.

5. En Configurar requisitos de seguridad, puede elegir su política de contraseñas, los requisitos de autenticación multifactor (MFA) y las opciones de recuperación de cuentas de usuario. Para obtener más información, consulte Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito.

   a. Para la política de contraseñas, confirme que el modo de política de contraseñas esté establecido en los valores predeterminados de Cognito.

   b. En Autenticación multifactor, para aplicar el MFA, elija MFA opcional.

   c. Para los métodos de MFA, selecciona Aplicaciones autenticadoras y mensajes SMS.

   d. Para la recuperación de la cuenta de usuario, confirme que esté seleccionada la opción Habilitar la recuperación automática de cuentas de usuario y que el método de entrega de los mensajes de recuperación de la cuenta de usuario esté configurado como Solo correo electrónico.

   e. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

6. En Configurar la experiencia de registro, puede determinar cómo verificarán sus identidades los nuevos usuarios al registrarse como nuevos usuarios y qué atributos deben ser obligatorios u opcionales durante el proceso de registro de los usuarios. Para obtener más información, consulte Administración de usuarios en el grupo de usuarios.

   a. Confirme que esté seleccionada la opción Habilitar el registro automático. Esta configuración permite que cualquier usuario de Internet se registre en tu grupo de usuarios. Esto está pensado para los fines de la aplicación de ejemplo, pero aplique esta configuración con precaución en los entornos de producción.

   b. En Verificación y confirmación asistidas por Cognito, compruebe que la casilla Permitir que Cognito envíe mensajes automáticamente para verificar y confirmar esté seleccionada.

   c. Confirme que los atributos a verificar estén configurados en Enviar mensaje de correo electrónico, verificar dirección de correo electrónico.

   d. En Verificar los cambios de atributos, confirme que estén seleccionadas las opciones predeterminadas: se selecciona Conservar el valor del atributo original cuando hay una actualización pendiente y los valores de los atributos activos cuando hay una actualización pendiente se establece en Dirección de correo electrónico.

e. En Atributos obligatorios, confirme que los atributos obligatorios basados en selecciones anteriores muestren el correo electrónico.

> **Important**
>
> Para esta aplicación de ejemplo, su grupo de usuarios no debe establecer phone_number como atributo obligatorio. Si se muestra el número de teléfono como atributo obligatorio, revise y actualice las opciones anteriores:
>
> - MFA opcional, solo correo electrónico para el método de entrega de los mensajes de recuperación de cuentas de usuario
> - Envía un mensaje de correo electrónico, verifica la dirección de correo electrónico para que Attributes la verifique

f. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

7. En Configurar la entrega de mensajes, puede configurar la integración con Amazon Simple Email Service y Amazon Simple Notification Service para enviar mensajes de correo electrónico y SMS a sus usuarios para que se registren, confirmen la cuenta, MFA y recuperen la cuenta. Para obtener más información, consulte Configuración de correo electrónico para grupos de usuarios de Amazon Cognito y Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito.

   a. En Proveedor de correo electrónico, elija Enviar correo electrónico con Cognito y utilice el remitente de correo electrónico predeterminado que proporciona Amazon Cognito. Esta configuración de bajo volumen de correo electrónico es suficiente para probar la aplicación. Puedes realizar la devolución después de verificar una dirección de correo electrónico con Amazon Simple Email Service (Amazon SES) y seleccionar Enviar correo electrónico con Amazon SES.

   b. En el caso de SMS, selecciona Crear una nueva función de IAM e introduce el nombre de una función de IAM. Esto crea un rol que concede permisos a Amazon Cognito para enviar mensajes SMS.

   c. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

8. En Integrate your app, puedes asignar un nombre a tu grupo de usuarios, configurar la interfaz de usuario alojada y crear un cliente de aplicación. Para obtener más información, consulte Agregue un cliente de aplicaciones con la interfaz de usuario alojada. Las aplicaciones de ejemplo no utilizan la interfaz de usuario alojada.

a. En Nombre del grupo de usuarios, introduzca un nombre de grupo de usuarios.
b. No selecciones Usar la interfaz de usuario alojada en Cognito.
c. En Cliente de aplicación inicial, confirme que el tipo de aplicación esté configurado como Cliente público.
d. En Secreto de cliente, confirma que esté seleccionada la opción No generar un secreto de cliente.
e. Introduzca un nombre de cliente de aplicación.
f. Amplíe la configuración avanzada del cliente de la aplicación. Añádalo `ALLOW_USER_PASSWORD_AUTH` a la lista de flujos de autenticación.
g. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

9. Revise sus opciones en la pantalla Revisar y crear y modifique las selecciones según sea necesario. Cuando esté satisfecho con la configuración del grupo de usuarios, elija Crear grupo de usuarios para continuar.
10. En la página Grupos de usuarios, elija su nuevo grupo de usuarios.
11. En Descripción general del grupo de usuarios, anote su ID de grupo de usuarios. Proporcionarás esta cadena cuando crees tu aplicación de ejemplo.
12. Elija la pestaña Integración de aplicaciones y busque la sección de análisis y clientes de aplicaciones. Selecciona tu nuevo cliente de aplicaciones. Anote su ID de cliente.

Recursos relacionados

- Grupos de usuarios de Amazon Cognito
- Flujo de autenticación de los grupos de usuarios
- Uso de tokens con grupos de usuarios

Crear una aplicación

Para crear una aplicación de Android de ejemplo

1. Instala Android Studio y las herramientas de línea de comandos.
2. En Android Studio, instala el complemento Flutter.
3. Crea un nuevo proyecto de Android Studio a partir del contenido del `cognito_flutter_mobile_app` directorio de esta aplicación de ejemplo.

- Edita `assets/config.json` y `<< YOUR CLIENT ID>>` reemplaza `<<YOUR USER POOL ID>>` y por los ID del grupo de usuarios y el cliente de la aplicación que creaste anteriormente.

4. Instala Flutter.

   a. Agrega Flutter a tu variable PATH.

   b. Acepte las licencias con el siguiente comando.

   ```
   flutter doctor --android-licenses
   ```

   c. Verifica tu entorno de Flutter e instala los componentes que falten.

   ```
   flutter doctor
   ```

   - Si falta algún componente, ejecútelo `flutter doctor -v` para saber cómo solucionar el problema.

   d. Cambia al directorio de tu nuevo proyecto de Flutter e instala las dependencias.

   - Ejecute `flutter pub add amazon_cognito_identity_dart_2`.

   e. Ejecute `flutter pub add flutter_secure_storage`.

5. Crea un dispositivo Android virtual.

   1. En la GUI de Android Studio, crea un dispositivo nuevo con el administrador de dispositivos.
   2. En la CLI, ejecute`flutter emulators --create --name android-device`.

6. Inicie su dispositivo Android virtual.

   1. En la GUI de Android Studio, selecciona el

      icono de inicio situado junto al dispositivo virtual.
   2. En la CLI, ejecute`flutter emulators --launch android-device`.

7. Inicie la aplicación en el dispositivo virtual.

   1. En la GUI de Android Studio, selecciona el

      icono de implementación.
   2. En la CLI, ejecute`flutter run`.

8. Navega hasta tu dispositivo virtual en ejecución en Android Studio.
9. Registra un nuevo usuario con una dirección de correo electrónico válida.
10. Recupera el código de confirmación de tu mensaje de correo electrónico. Introduzca el código de confirmación en la aplicación.
11. Inicie sesión con su nombre de usuario y contraseña.

Siguientes pasos

Una vez que haya seguido los tutoriales para completar las aplicaciones de ejemplo, puede ampliar el alcance de la implementación de su grupo de usuarios. Puede crear grupos de usuarios adicionales, personalizar las funciones del grupo de usuarios para otras aplicaciones o añadir proveedores de identidad externos. Cuando planifique su transición para incluir grupos de usuarios de Amazon Cognito en aplicaciones de producción, puede evaluar ejemplos y tutoriales adicionales.

A continuación se muestran algunas funciones adicionales de los grupos de usuarios de Amazon Cognito:

- Personalizar las páginas web integradas de registro e inicio de sesión
- Adición de MFA a un grupo de usuarios.
- Adición de seguridad avanzada a un grupo de usuarios
- Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda
- Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito

Para obtener información general sobre los modelos de autenticación y autorización de Amazon Cognito, consulte. Cómo funciona la autenticación con los grupos de usuarios y grupos de identidades de Amazon Cognito

Para acceder a otros Servicios de AWS tras una autenticación correcta de un grupo de usuarios, consulteAcceder Servicios de AWS mediante un grupo de identidades después de iniciar sesión.

Además de usar los SDK AWS Management Console y el grupo de usuarios, también puede administrar sus grupos de usuarios mediante el AWS Command Line Interface.

Temas

- Cree un nuevo grupo de usuarios
- Agregue un cliente de aplicaciones con la interfaz de usuario alojada

- Añadir inicio de sesión de redes sociales a un grupo de usuarios (opcional)
- Añadir inicio de sesión con un proveedor de identidad SAML a un grupo de usuarios (opcional)

Cree un nuevo grupo de usuarios

Con un grupo de usuarios, los usuarios pueden iniciar sesión en su aplicación web o móvil mediante Amazon Cognito.

Cree un nuevo grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Pulse el botón Crear grupo de usuarios. Puede que tenga que seleccionar Grupos de usuarios en el panel de navegación izquierdo para que aparezca esta opción.
3. En la esquina superior derecha de la página, elija Create a User Pool (Crear un grupo de usuarios).
4. En Configurar la experiencia de inicio de sesión, puede elegir los proveedores de identidad (IdPs) que utilizará con este grupo de usuarios. Para obtener más información, consulte Agregar inicio de sesión de grupo de usuarios a través de un tercero.
    a. En Proveedores de autenticación, para los tipos de proveedores, asegúrese de que solo esté seleccionado el grupo de usuarios de Cognito.
    b. Para ver las opciones de inicio de sesión del grupo de usuarios de Cognito, elija Nombre de usuario. No seleccione ningún requisito de nombre de usuario adicional.
    c. Mantén todas las demás opciones como predeterminadas y selecciona Siguiente.
5. En Configurar requisitos de seguridad, puede elegir su política de contraseñas, los requisitos de autenticación multifactor (MFA) y las opciones de recuperación de cuentas de usuario. Para obtener más información, consulte Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito.
    a. Para la política de contraseñas, confirme que el modo de política de contraseñas esté establecido en los valores predeterminados de Cognito.
    b. En Autenticación multifactor, para aplicar el MFA, elija MFA opcional.
    c. Para los métodos de MFA, selecciona Aplicaciones autenticadoras y mensajes SMS.
    d. Para la recuperación de la cuenta de usuario, confirme que esté seleccionada la opción Habilitar la recuperación automática de cuentas de usuario y que el método de entrega de

los mensajes de recuperación de la cuenta de usuario esté configurado como Solo correo electrónico.

e. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

6. En Configurar la experiencia de registro, puede determinar cómo verificarán sus identidades los nuevos usuarios al registrarse como nuevos usuarios y qué atributos deben ser obligatorios u opcionales durante el proceso de registro de los usuarios. Para obtener más información, consulte Administración de usuarios en el grupo de usuarios.

   a. Confirme que esté seleccionada la opción Habilitar el registro automático. Esta configuración permite que cualquier usuario de Internet se registre en tu grupo de usuarios. Esto está pensado para los fines de la aplicación de ejemplo, pero aplique esta configuración con precaución en los entornos de producción.

   b. En Verificación y confirmación asistidas por Cognito, compruebe que la casilla Permitir que Cognito envíe mensajes automáticamente para verificar y confirmar esté seleccionada.

   c. Confirme que los atributos a verificar estén configurados en Enviar mensaje de correo electrónico, verificar dirección de correo electrónico.

   d. En Verificar los cambios de atributos, confirme que estén seleccionadas las opciones predeterminadas: se selecciona Conservar el valor del atributo original cuando hay una actualización pendiente y los valores de los atributos activos cuando hay una actualización pendiente se establece en Dirección de correo electrónico.

   e. En Atributos obligatorios, confirme que los atributos obligatorios basados en selecciones anteriores muestren el correo electrónico.

      > **Important**
      >
      > Para esta aplicación de ejemplo, su grupo de usuarios no debe establecer phone_number como atributo obligatorio. Si se muestra el número de teléfono como atributo obligatorio, revise y actualice las opciones anteriores:
      >
      > - MFA opcional, solo correo electrónico para el método de entrega de los mensajes de recuperación de cuentas de usuario
      > - Envía un mensaje de correo electrónico, verifica la dirección de correo electrónico para que Attributes la verifique

   f. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

7. En Configurar la entrega de mensajes, puede configurar la integración con Amazon Simple Email Service y Amazon Simple Notification Service para enviar mensajes de correo electrónico y SMS a sus usuarios para que se registren, confirmen la cuenta, MFA y recuperen la cuenta. Para obtener más información, consulte Configuración de correo electrónico para grupos de usuarios de Amazon Cognito y Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito.

   a. En Proveedor de correo electrónico, elija Enviar correo electrónico con Cognito y utilice el remitente de correo electrónico predeterminado que proporciona Amazon Cognito. Esta configuración de bajo volumen de correo electrónico es suficiente para probar la aplicación. Puedes realizar la devolución después de verificar una dirección de correo electrónico con Amazon Simple Email Service (Amazon SES) y seleccionar Enviar correo electrónico con Amazon SES.

   b. En el caso de SMS, selecciona Crear una nueva función de IAM e introduce el nombre de una función de IAM. Esto crea un rol que concede permisos a Amazon Cognito para enviar mensajes SMS.

   c. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

8. En Integrate your app, puedes asignar un nombre a tu grupo de usuarios, configurar la interfaz de usuario alojada y crear un cliente de aplicación. Para obtener más información, consulte Agregue un cliente de aplicaciones con la interfaz de usuario alojada. Las aplicaciones de ejemplo no utilizan la interfaz de usuario alojada.

   a. En Nombre del grupo de usuarios, introduzca un nombre de grupo de usuarios.

   b. No selecciones Usar la interfaz de usuario alojada en Cognito.

   c. En Cliente de aplicación inicial, confirme que el tipo de aplicación esté configurado como Cliente público.

   d. En Secreto de cliente, confirma que esté seleccionada la opción No generar un secreto de cliente.

   e. Introduzca un nombre de cliente de aplicación.

   f. Amplíe la configuración avanzada del cliente de la aplicación. Añádalo `ALLOW_USER_PASSWORD_AUTH` a la lista de flujos de autenticación.

   g. Mantenga todas las demás opciones como predeterminadas y seleccione Siguiente.

9. Revise sus opciones en la pantalla Revisar y crear y modifique las selecciones según sea necesario. Cuando esté satisfecho con la configuración del grupo de usuarios, elija Crear grupo de usuarios para continuar.

10. En la página Grupos de usuarios, elija su nuevo grupo de usuarios.
11. En Descripción general del grupo de usuarios, anote su ID de grupo de usuarios. Proporcionarás esta cadena cuando crees tu aplicación de ejemplo.
12. Elija la pestaña Integración de aplicaciones y busque la sección de análisis y clientes de aplicaciones. Selecciona tu nuevo cliente de aplicaciones. Anote su ID de cliente.

Para crear un grupo de usuarios

1. Diríjase a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. ElegirGrupos de usuarios de.
3. En la esquina superior derecha de la página, elija Create a User Pool (Crear un grupo de usuarios).
4. EnConfigurar la experiencia de inicio de sesión, elija los proveedores federados que utilizará con este grupo de usuarios. Para obtener más información, consulte Agregar inicio de sesión de grupo de usuarios a través de un tercero.
5. En Configure security requirements (Configuración de requisitos de seguridad), elija la política de contraseñas, los requisitos de autenticación multifactor (MFA) y las opciones de recuperación de cuentas de usuario. Para obtener más información, consulte Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito.
6. EnConfigure la experiencia de registro, determine cómo los nuevos usuarios verificarán sus identidades al registrarse y qué atributos deben ser obligatorios u opcionales durante el flujo de registro de usuarios. Para obtener más información, consulte Administración de usuarios en el grupo de usuarios.

> **Important**
>
> Si activa el registro de usuarios en el grupo de usuarios, cualquier usuario de Internet podrá crear una cuenta e iniciar sesión en las aplicaciones. No habilite el registro automático en el grupo de usuarios a menos que quiera abrir la aplicación para que el público se registre. Para cambiar esta configuración, actualiza el registro de autoservicio en la pestaña Experiencia de registro de la consola del grupo de usuarios o actualiza el valor de una solicitud AllowAdminCreateUserOnlyde API CreateUserPool. UpdateUserPool

Para obtener información sobre las características de seguridad que puede configurar en los grupos de usuarios, consulte Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito.

7. EnConfiguración de entrega de mensajes, configure la integración con Amazon Simple Email Service y Amazon Simple Notification Service para enviar mensajes de correo electrónico y SMS a sus usuarios para su registro, confirmación de cuenta, MFA y recuperación de cuentas. Para obtener más información, consulte Configuración de correo electrónico para grupos de usuarios de Amazon Cognito y Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito.
8. En Integrar la aplicación, asigne un nombre al grupo de usuarios, configure la interfaz de usuario alojada y cree un cliente de aplicaciones. Para obtener más información, consulte Agregue un cliente de aplicaciones con la interfaz de usuario alojada.
9. Revise sus opciones en la pantalla Revisar y crear y modifique las selecciones según sea necesario. Cuando esté satisfecho con la configuración del grupo de usuarios, seleccione Crear grupo de usuarios para continuar.

Recursos relacionados

Para obtener más información acerca de los grupos de usuarios, consulte Grupos de usuarios de Amazon Cognito.

Consulte también: Flujo de autenticación de los grupos de usuarios yUso de tokens con grupos de usuarios.

Agregue un cliente de aplicaciones con la interfaz de usuario alojada

Tras crear un grupo de usuarios, puede crear un cliente de aplicaciones para una aplicación que muestre las páginas web integradas en la interfaz de usuario alojada. En la interfaz de usuario alojada, los usuarios pueden:

- Regístrese para obtener un perfil de usuario.
- Inicia sesión con un proveedor de identidad externo.
- Inicie sesión con o sin autenticación multifactorial.
- Restablezca su contraseña.

Para crear un cliente de aplicación para el inicio de sesión en la interfaz de usuario alojada

1. Vaya a la consola de Amazon Cognito. Si se te solicita, introduce tus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios. Si crea un nuevo grupo de usuarios, se le pedirá que configure un cliente de aplicación y configure la IU alojada durante el asistente.
4. Elija el iconoIntegración de aplicacionesde su grupo de usuarios.
5. Junto a Dominio, elija Acciones y, a continuación, seleccione Crear dominio personalizado o Crear dominio de Amazon Cognito. Si ya ha configurado un dominio de grupo de usuarios, elija Eliminar dominio de Amazon Cognito o Eliminar dominio personalizado antes de crear el nuevo dominio personalizado.
6. Ingrese un prefijo de dominio disponible para utilizarlo con un dominio de Amazon Cognito. Para obtener información sobre cómo configurar un dominio personalizado, consulte Uso de un dominio propio con la interfaz de usuario alojada
7. Seleccione Create (Crear).
8. Vuelva a laIntegración de aplicacionespara el mismo grupo de usuarios y localizarClientes de aplicaciones. Elija Create app client.
9. Elija un Application type (Tipo de aplicación). Se proporcionarán algunos ajustes recomendados en función de tu selección. Una aplicación que utiliza la interfaz de usuario alojada es un Cliente público.
10. Ingrese un nombre de cliente de aplicación.
11. Para este ejercicio, elijaNo generar secreto de cliente. El secreto de cliente lo utilizan las aplicaciones confidenciales que autentican a los usuarios desde una aplicación centralizada. En este ejercicio, presentará una página de inicio de sesión de IU alojada a los usuarios y no requerirá ningún secreto de cliente.
12. Elija los flujos de autenticación que permitirá con su aplicación. Asegúrese de que`USER_SRP_AUTH`se ha seleccionado.
13. Personalice token expiration (Vencimiento de token), Advanced security configuration (Configuración avanzada de seguridad) y Attribute read and write permissions (Permisos de lectura y escritura de atributos) según sea necesario. Para obtener más información, consulte Configuración de un cliente de aplicación para grupos de usuarios.
14. Agregar una URL de devolución de llamada para el cliente de aplicación. Aquí es donde se le dirigirá después de la autenticación de la interfaz de usuario alojada. No necesitas añadir

una URL de cierre de sesión permitida hasta que puedas implementar el cierre de sesión en tu aplicación.

En el caso de aplicaciones de iOS o Android, puede utilizar una URL de devolución de llamada como `myapp://`.

15. Seleccione Identity providers (Proveedores de identidades) para el cliente de aplicación. Como mínimo, habilite Grupo de usuarios de Amazon Cognito como proveedor.

> **Note**
>
> Para iniciar sesión con proveedores de identidad externos (IdPs) como Facebook, Amazon, Google y Apple, así como a través de OpenID Connect (OIDC) o SAML IdPs, primero configúrelos como se muestra en Añadir el inicio de sesión en un grupo de usuarios a través de un tercero. A continuación, vuelve a la página de configuración del cliente de la aplicación para activarlos.

16. Elija OAuth 2.0 Grant Types (Tipos de subvenciones OAuth 2.0). Seleccione Authorization code grant (Concesión de código de autorización) para devolver un código de autorización que se intercambie por tokens de grupos de usuarios. Debido a que los tokens nunca se exponen directamente a un usuario final, es menos probable que se vean comprometidos. Sin embargo, se requiere una aplicación personalizada en el backend para intercambiar el código de autorización para tokens de grupos de usuarios. Por motivos de seguridad, le recomendamos utilizar el flujo de concesión de código de autorización junto con PKCE (Proof Key for Code Exchange, clave de prueba para intercambio de código) para las aplicaciones móviles.

    Seleccione Implicit grant (Concesión implícita) para que se devuelvan JSON Web Tokens (JWT) del grupo de usuarios desde Amazon Cognito. Puede utilizar este flujo cuando no hay un backend disponible para intercambiar un código de autorización para tokens. También es útil para depurar tokens.

> **Note**
>
> Puede habilitar tanto Authorization code grant (Concesión de código de autorización) como Implicit code grant (Concesión de código implícita) y, a continuación, utilizar cada concesión según sea necesario.
> Seleccione Client credentials (Credenciales del cliente) solo si la aplicación debe solicitar tokens de acceso en su propio nombre y no en nombre de un usuario.

17. A menos que desee excluir específicamente alguno, seleccione todos los ámbitos de OpenID Connect.
18. Seleccione los ámbitos personalizados que haya configurado. Los ámbitos personalizados se utilizan normalmente con clientes confidenciales.
19. Seleccione Crear.

Para ver su página de inicio de sesión

En la página del cliente de la aplicación, selecciona Ver la interfaz de usuario alojada para abrir una nueva pestaña del navegador y abrir una página de inicio de sesión que viene rellenada previamente con los parámetros del ID del cliente de la aplicación, el ámbito, la concesión y la URL de devolución de llamada.

Puede ver la página web de inicio de sesión de la interfaz de usuario alojada con la siguiente URL. Anote el `response_type`. En este caso, response_type=code para la concesión de código de autorización.

```
https://your_domain/login?
response_type=code&client_id=your_app_client_id&redirect_uri=your_callback_url
```

Puede ver la página web de inicio de sesión de la interfaz de usuario alojada con la siguiente dirección URL para la concesión de código implícita, donde response_type = token. Tras un inicio de sesión correcto, Amazon Cognito devuelve tokens de grupo de usuarios a su barra de direcciones de navegador web.

```
https://your_domain/login?
response_type=token&client_id=your_app_client_id&redirect_uri=your_callback_url
```

Puede encontrar el token de identidad de JSON Web Token (JWT) a continuación del parámetro `#idtoken=` en la respuesta.

A continuación, le mostramos una respuesta de ejemplo de una solicitud de concesión implícita. La cadena del token de identidad será mucho más larga.

```
https://www.example.com/
#id_token=123456789tokens123456789&expires_in=3600&token_type=Bearer
```

Los tokens de grupos de usuarios de Amazon Cognito se firman con un algoritmo RS256. Puedes decodificar y verificar los tokens del grupo de usuarios mediante. AWS Lambda Para obtener más información, consulte Decodificar y verificar los tokens JWT de Amazon Cognito en el sitio web. AWS GitHub

Su dominio aparece en la página Domain Name (Nombre de dominio). Su ID de cliente de aplicación y URL de devolución de llamada se muestran en la página General settings (Configuración general). Si los cambios que ha realizado en la consola no aparecen inmediatamente, espere unos minutos y, a continuación, actualice el navegador.

Añadir inicio de sesión de redes sociales a un grupo de usuarios (opcional)

Puede habilitar a los usuarios de la aplicación para que inicien sesión a través de un proveedor de identidad social (IdP) como, por ejemplo, Facebook, Google, Amazon y Apple. Tanto si los usuarios inician sesión directamente o a través de un tercero, todos los usuarios tienen un perfil en el grupo de usuarios. Omita este paso si no desea agregar inicio de sesión a través de un proveedor de identidad de inicio de sesión de redes sociales.

Registrarse en un proveedor de identidad social

Antes de crear un proveedor de identidad social con Amazon Cognito, debe registrar su aplicación en él para recibir un ID y un secreto del cliente.

Para registrar una aplicación en Facebook

1. Cree una cuenta de desarrollador con Facebook.
2. Inicie sesión con sus credenciales de Facebook.
3. En el menú My Apps (Mis aplicaciones), elija Create New App (Crear nueva aplicación).

   Si no tienes una aplicación de Facebook existente, verás una opción diferente. Seleccione Crear una aplicación.
4. En la página Create an app, elija un caso de uso para la aplicación y, a continuación, elija Next.
5. Ingrese un nombre para la aplicación de Facebook y elija Create App.
6. En la barra de navegación de la izquierda, elija App Settings y luego Basic.

7. Tome nota del valor de App ID (ID de aplicación) y de App Secret (Secreto de la aplicación). Los usará en la sección siguiente.
8. Elija + Add platform en la parte inferior de la página.
9. En la pantalla de selección de plataforma, selecciona tus plataformas y, a continuación, selecciona Siguiente.
10. Elija Guardar cambios.
11. ParaDominios de aplicación, introduzca el dominio del grupo de usuarios.

    ```
    https://your_user_pool_domain
    ```

12. Elija Guardar cambios.
13. En la barra de navegación, selecciona Productos y, a continuación, selecciona Configurar desde el inicio de sesión de Facebook.
14. En el menú Facebook Login Configure, elija Settings.

    Escriba su URL de redirección en Valid OAuth Redirect URIs (URI de redireccionamiento de OAuth válidos). La URL de redireccionamiento consiste en el dominio del grupo de usuarios con el `/oauth2/idpresponse` punto final.

    ```
    https://your_user_pool_domain/oauth2/idpresponse
    ```

15. Elija Guardar cambios.

Para registrar una aplicación en Amazon

1. Cree una cuenta de desarrollador con Amazon.
2. Inicie sesión con las credenciales de Amazon.
3. Debe crear un perfil de seguridad de Amazon para recibir un ID y un secreto de cliente de Amazon.

   Selecciona Aplicaciones y servicios en la barra de navegación situada en la parte superior de la página y, a continuación, selecciona Login with Amazon.
4. Elija Create a Security Profile (Crear un perfil de seguridad).
5. Escriba un valor en Security Profile Name (Nombre del perfil de seguridad), en Security Profile Description (Descripción del perfil de seguridad) y en Consent Privacy Notice URL (URL del aviso sobre consentimiento de confidencialidad).

6. Seleccione Save (Guardar).
7. Elija Client ID (ID de cliente) y Client Secret (Secreto de cliente) para mostrar el ID de cliente y el secreto. Los usará en la sección siguiente.
8. Coloque el cursor sobre el engranaje, elija Web Settings (Configuración de web) y, a continuación, elija Edit (Editar).
9. Escriba el dominio del grupo de usuarios en Allowed Origins (Orígenes permitidos).

   ```
   https://<your-user-pool-domain>
   ```

10. Escriba el dominio del grupo de usuarios con el punto de conexión `/oauth2/idpresponse` en URL de devolución permitidas.

    ```
    https://<your-user-pool-domain>/oauth2/idpresponse
    ```

11. Seleccione Save (Guardar).

Para registrar una aplicación en Google

Para obtener más información sobre OAuth 2.0 en la plataforma de Google Cloud, consulte Más información sobre la autenticación y la autorización en la documentación de Google Workspace for Developers.

1. Cree una cuenta de desarrollador con Google.
2. Inicie sesión en la consola de Google Cloud Platform.
3. En la barra de navegación superior, elija Select a project (Seleccionar un proyecto). Si ya tiene un proyecto en la plataforma de Google, este menú muestra tu proyecto predeterminado.
4. Seleccione NEW PROJECT (NUEVO PROYECTO).
5. Escriba un nombre para su proyecto y, a continuación, elija CREATE (CREAR).
6. En la barra de navegación izquierda, selecciona API y servicios y, a continuación, selecciona la pantalla de consentimiento de Oauth.
7. Introduce la información de la aplicación, el dominio de la aplicación, los dominios autorizados y la información de contacto del desarrollador. Sus dominios autorizados deben incluir `amazoncognito.com` y la raíz de su dominio personalizado. Por ejemplo: `example.com`. Elija SAVE AND CONTINUE (GUARDAR Y CONTINUAR).
8. 1. En Ámbitos, selecciona Añadir o eliminar ámbitos y, a continuación, elige, como mínimo, los siguientes ámbitos de OAuth.

1. `.../auth/userinfo.email`
2. `.../auth/userinfo.profile`
3. openid

9. En Test Users (Usuarios de prueba), elija Add Users (Añadir usuarios). Introduce tu dirección de correo electrónico y la de cualquier otro usuario de prueba autorizado y, a continuación, selecciona GUARDAR Y CONTINUAR.
10. Vuelva a expandir la barra de navegación izquierda, elija API y servicios y, a continuación, elija Credenciales.
11. Selecciona CREAR CREDENCIALES y, a continuación, selecciona el ID de cliente de OAuth.
12. Seleccione un tipo de aplicacióny asigne un nombre al cliente.
13. En JavaScript Orígenes autorizados, selecciona AÑADIR URI. Introduzca el dominio del grupo de usuarios.

```
https://<your-user-pool-domain>
```

14. En Authorized redirect URIs (URI de redirección autorizadas), elija ADD URI (AÑADIR URI). Introduzca la al punto de conexión `/oauth2/idpresponse` de su dominio de grupo de usuarios.

```
https://<your-user-pool-domain>/oauth2/idpresponse
```

15. Seleccione CREATE (Crear).
16. Almacene de forma segura los valores que muestra Google en ID del cliente y Secreto del cliente. Proporcione estos valores a Amazon Cognito cuando agregue un proveedor de IdP Google.

Para registrar una aplicación con Apple, siga estos pasos:

Para obtener más información sobre la configuración de inicio de sesión con Apple, consulte Configuring Your Environment for Sign in with Apple en la documentación del desarrollador de Apple.

1. Cree una cuenta de desarrollador en Apple.
2. Inicie sesión con las credenciales de Apple.
3. En la barra de navegación de la izquierda, elija Certificates, Identifiers & Profiles (Certificados, identificadores y perfiles).

4. En la barra de navegación de la izquierda, elija Identifiers (Identificadores).
5. En la página Identifiers (Identificadores), elija el icono +.
6. En la página Register a New Identifier (Registrar un nuevo identificador), elija App IDs (ID de aplicaciones) y, a continuación, Continue (Continuar).
7. En la página Seleccione un tipo, elija Aplicación y, a continuación, elija Continuar.
8. En la página Register an App ID (Registrar un ID de aplicación), haga lo siguiente:

    1. En Description (Descripción), introduzca una descripción.
    2. En App ID Prefix (Prefijo de ID de aplicación), introduzca un ID del paquete. Anote el valor de laPrefijo de ID de aplicación. Utilizarás este valor después de elegir Apple como proveedor de identidad enPaso 2: Añadir un proveedor de identidad social al grupo de usuarios.
    3. En Capabilities (Funcionalidades), elija SignInWithApple y, a continuación, elija Edit (Editar).
    4. En la página Sign in with Apple: App ID Configuration (Inicio de sesión con Apple: Configuración del ID de aplicación), elija configurar la aplicación como principal o agrupada con otros ID de aplicación y, a continuación, elija Save (Guardar).
    5. Elija Continue (Continuar).
9. En la página Confirm your App ID (Confirmar ID de Apple), elija Register (Registrarse).
10. En la página Identifiers (Identificadores), elija el icono +.
11. En la página Register a New Identifier (Registrar un nuevo identificador), elija Services IDs (ID de servicios) y, a continuación, Continue (Continuar).
12. En la página Register a Services ID (Registrar un ID de servicio), haga lo siguiente:

    1. En Description (Descripción), introduzca una descripción.
    2. En Identificador (Identifier), ingrese un identificador. Anota este identificador de servicios, ya que necesitarás este valor después de elegir a Apple como tu proveedor de identidad enPaso 2: Añadir un proveedor de identidad social al grupo de usuarios.
    3. Seleccione Continue (Continuar) y, a continuación, Register (Registrarse).
13. Elige el ID de servicios que acabas de crear en la página de identificadores.

    1. Seleccione SignInWithApple y, a continuación, elija Configure (Configurar).
    2. En la página Web Authentication Configuration (Configuración de autenticación web), seleccione el ID de aplicación creado anteriormente comoPrimary App ID (ID de aplicación principal).
    3. Elija el icono + situado al lado de Website URLs (URL de sitio web).

4. En Domains and subdomains (Dominios y subdominios), introduzca el dominio del grupo de usuarios sin un prefijo `https://`.

   ```
   <your-user-pool-domain>
   ```

5. En Return URLs (URL de devolución), introduzca la ruta al punto de conexión `/oauth2/idpresponse` del dominio del grupo de usuarios.

   ```
   https://<your-user-pool-domain>/oauth2/idpresponse
   ```

6. Selecciona Siguiente y, a continuación, selecciona Listo. No es necesario verificar el dominio.
7. Elija Continue (Continuar) y, a continuación, elija Save (Guardar).

14. En la barra de navegación de la izquierda, elija Keys (Claves).
15. En la página Keys (Claves), elija el icono +.
16. En la página Register a New Key (Registrar una nueva clave), haga lo siguiente:

    1. En Key Name (Nombre de clave), escriba un nombre de clave.
    2. Elija SignInWithApple y, a continuación, Configure (Configurar).
    3. En la página Configurar clave, selecciona el ID de aplicación que creaste anteriormente como ID de aplicación principal. Seleccione Guardar.
    4. Seleccione Continue (Continuar) y, a continuación, Register (Registrarse).

17. En la página Descarga tu clave, selecciona Descargar para descargar la clave privada, anota el ID de clave que se muestra y, a continuación, selecciona Listo. Necesitará esta clave privada y el valor de ID de clave que se muestra en esta página después de elegir Apple como proveedor de identidad en Paso 2: Añadir un proveedor de identidad social al grupo de usuarios.

Añadir un proveedor de identidad social al grupo de usuarios

En esta sección configurará un proveedor de identidad social en el grupo de usuarios utilizando el ID y el secreto de cliente de la sección anterior.

Para configurar un proveedor de identidad social para un grupo de usuarios con AWS Management Console

1. Vaya a la consola de Amazon Cognito. Es posible que se le pidan sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).

3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). LocalizarInicio de sesión federadoy seleccioneAñadir un proveedor de identidad.
5. Elija un proveedor de identidad de red social: Facebook, Google, Login with Amazon o Apple.
6. Elija uno de los siguientes pasos, según su elección de proveedor de identidad social:

   - Google y Login with Amazon: introduce el ID de cliente de la aplicación y el secreto del cliente de la aplicación que se generaron en la sección anterior.
   - Facebook: introduce el ID de cliente de la aplicación y el secreto del cliente de la aplicación que se generaron en la sección anterior y, a continuación, elige una versión de la API (por ejemplo, la versión 2.12). Te recomendamos elegir la versión más reciente posible: cada API de Facebook tiene un ciclo de vida y una fecha de caducidad. Los ámbitos y atributos de Facebook pueden variar según las versiones de la API. Te recomendamos que pruebes tu inicio de sesión de identidad social con Facebook para asegurarte de que la federación funcione según lo previsto.
   - Inicia sesión con Apple: introduce el ID de servicio, el ID del equipo, el ID de clave y la clave privada que se generaron en la sección anterior.
7. Introduce los nombres de los ámbitos autorizados que quieres usar. Los ámbitos definen a qué atributos de usuario (como `name` y `email`) desea acceder con su aplicación. En el caso de Facebook, deben separarse con comas. En el caso de Google y Login with Amazon, deben separarse con espacios. Para SignInWithApple, marque las casillas de verificación de los ámbitos a los que desee acceder.

Proveedor de identidad social	Ámbitos de ejemplo
Facebook	`public_profile, email`
Google	`profile email openid`
Login with Amazon	`profile postal_code`
Inicio de sesión con Apple	`email name`

Al usuario de la aplicación se le pedirá que esté de acuerdo con proporcionar estos atributos a su aplicación. Para obtener más información acerca de sus ámbitos, consulte la documentación de Google, Facebook, Login with Amazon o Inicio de sesión con Apple.

En el caso de Sign in with Apple (Inicio de sesión con Apple), estos son escenarios de usuario en los que es posible que no se devuelvan los ámbitos.

- Un usuario final encuentra errores al salir de la página de inicio de sesión de Apple (pueden deberse a errores internos en Amazon Cognito o a cualquier error escrito por el desarrollador).
- El identificador del ID de servicio se utiliza en los grupos de usuarios u otros servicios de autenticación.
- Un desarrollador agrega ámbitos adicionales después de que el usuario inicie sesión. Los usuarios solo recuperan información nueva cuando se autentican y cuando actualizan sus tokens.
- Un desarrollador elimina al usuario y, a continuación, el usuario vuelve a iniciar sesión sin eliminar la aplicación de su perfil de ID de Apple.

8. Asigne atributos de su proveedor de identidad a su grupo de usuarios. Para obtener más información, consulte Cuestiones que debe saber acerca de los mapeos.
9. Seleccione Crear.
10. DelIntegración de clientes de aplicaciones, elija uno de los Clientes de aplicaciones en la lista y Edit hosted UI settings (Modificar la configuración de IU). Agregue el nuevo proveedor de identidad social al cliente de la aplicación enProveedores de identidad.
11. Elija Guardar cambios.

Probar la configuración del proveedor de identidad social

Puede crear una URL de inicio de sesión con los elementos de las dos secciones anteriores. Úselo para probar la configuración del proveedor de identidad social.

```
https://mydomain.us-east-1.amazoncognito.com/login?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Puede encontrar el dominio en la página de la consola Domain name (Nombre de dominio) del grupo de usuarios. El valor de client_id se encuentra en la página App client settings (Configuración del cliente de aplicación). Use la URL de devolución de llamada para el parámetro redirect_uri. Esta es la URL de la página a la que se redirigirá al usuario después de una autenticación correcta.

> **Note**
>
> Amazon Cognito cancela las solicitudes de autenticación que no se completan en 5 minutos y redirige al usuario a la IU alojada. La página muestra un mensaje de error `Something went wrong`.

Añadir inicio de sesión con un proveedor de identidad SAML a un grupo de usuarios (opcional)

Puede habilitar que los usuarios de la aplicación inicien sesión a través de un proveedor de identidades (IdP) SAML. Tanto si los usuarios inician sesión directamente o a través de un tercero, todos los usuarios tienen un perfil en el grupo de usuarios. Omita este paso si no desea agregar inicio de sesión a través de un proveedor de identidad SAML.

Para obtener más información, consulte Uso de proveedores de identidad SAML con un grupo de usuarios.

Debes actualizar tu proveedor de identidades SAML y configurar tu grupo de usuarios. Para obtener información sobre cómo añadir tu grupo de usuarios como parte de confianza o aplicación para tu proveedor de identidades de SAML 2.0, consulta la documentación de tu proveedor de identidades de SAML.

También debes proporcionar un punto de enlace del servicio de confirmación al consumidor (ACS) a tu proveedor de identidad de SAML. Configure el siguiente punto de conexión en el dominio de su grupo de usuarios para enlace POST de SAML 2.0 en su proveedor de identidades SAML. Para obtener más información sobre los dominios de grupos de usuarios, consulte. Configuración de un dominio del grupo de usuarios

```
https://Your user pool domain/saml2/idpresponse
With an Amazon Cognito domain:
https://<yourDomainPrefix>.auth.<region>.amazoncognito.com/saml2/idpresponse
With a custom domain:
https://Your custom domain/saml2/idpresponse
```

Puede encontrar el prefijo de dominio y el valor de región de su grupo de usuarios en la pestaña Nombre de dominio de la consola de Amazon Cognito.

En el caso de algunos proveedores de identidad de SAML, también debe proporcionar el proveedor de servicios (SP)`urn`, también denominado URI de audiencia o ID de entidad del SP, en el formato:

```
urn:amazon:cognito:sp:<yourUserPoolID>
```

El ID del grupo de usuarios se encuentra en la pestaña General settings (Configuración general) de la consola de Amazon Cognito.

Asimismo, debe configurar el proveedor de identidad SAML para que proporcione los valores de todos los atributos necesarios en su grupo de usuarios. Normalmente, `email` es un atributo obligatorio para grupos de usuarios. En ese caso, el proveedor de identidad SAML debe proporcionar un valor `email` (notificación) en la aserción SAML.

Los grupos de usuarios de Amazon Cognito admiten la federación SAML 2.0 con puntos de enlace post-binding". Esto elimina la necesidad de que tu aplicación recupere o analice las respuestas a las aserciones de SAML, ya que el grupo de usuarios recibe directamente la respuesta de SAML de tu proveedor de identidad a través de un agente de usuario.

Para configurar un proveedor de identidad SAML 2.0 en su grupo de usuarios

1. Diríjase a la consola de Amazon Cognito. Si se le solicita, introduzca sus credenciales. AWS
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). LocalizarInicio de sesión federadoy seleccioneAñadir un proveedor de identidad.
5. Elija unSAMLProveedor de identidad social.
6. Introduzca Identificadores separados por comas. Un identificador indica a Amazon Cognito que debe comprobar la dirección de correo electrónico que el usuario introduce al iniciar sesión. A continuación, los dirige al proveedor correspondiente a su dominio.
7. Elija Add sign-out flow (Añadir flujo de cierre de sesión) si desea que Amazon Cognito envíe solicitudes de cierre de sesión firmadas a su proveedor cuando un usuario cierra la sesión. Debe configurar el proveedor de identidad SAML 2.0 para enviar respuestas de cierre de sesión al punto de enlace de `https://<your Amazon Cognito domain>/saml2/logout` que se crea al configurar la IU alojada. El `saml2/logout` punto final utiliza el enlace POST.

> **Note**
>
> Si selecciona esta opción y su proveedor de identidad de SAML espera que se firme una solicitud de cierre de sesión, también tendrá que configurar el certificado de firma que proporciona Amazon Cognito con su IDP de SAML.
> El IDP de SAML procesará la solicitud de cierre de sesión firmada y cerrará la sesión del usuario de Amazon Cognito.

8. Seleccione un Origen de documentos de metadatos. Si su proveedor de identidad ofrece metadatos SAML en una URL pública, puede elegir Metadata document URL (URL del documento de metadatos) e introducir esa URL pública. En caso contrario, elija Upload metadata document (Cargar documento de metadatos) y seleccione un archivo de metadatos que haya descargado anteriormente de su proveedor.

   > **Note**
   >
   > Le recomendamos que introduzca la URL de un documento de metadatos si su proveedor tiene un punto final público, en lugar de cargar un archivo. Esto permite a Amazon Cognito actualizar los metadatos automáticamente. Normalmente, los metadatos se actualizan cada seis horas o antes de que caduquen, lo que ocurra primero.

9. SelectAsignar atributos entre el proveedor de SAML y la aplicaciónpara asignar atributos de proveedor SAML al perfil de usuario de su grupo de usuarios. Incluya los atributos requeridos del grupo de usuarios en el mapa de atributos.

   Por ejemplo, cuando eliges laAtributo grupo de usuarios `email`, introduzca el nombre de atributo SAML tal como aparece en la aserción SAML del proveedor de identidad. Es posible que su proveedor de identidades ofrezca afirmaciones SAML de ejemplo como referencia. Algunos proveedores de identidad utilizan nombres sencillos, como`email`, mientras que otros utilizan nombres de atributos con formato URL, como el siguiente ejemplo:

   ```
   http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
   ```

10. Seleccione Crear.

Introducción a los grupos de identidades de Amazon Cognito

Los grupos de identidades de Amazon Cognito permiten crear identidades únicas y asignar permisos a los usuarios. El grupo de identidades puede incluir:

- Usuarios de un grupo de usuarios de Amazon Cognito
- Usuarios que se autentican con proveedores de identidad externos como Facebook, Google, Apple o un OIDC o un proveedor de identidad basado en SAML.
- Usuarios que se autentican de acuerdo con el proceso de autenticación existente.

Con un grupo de identidades, puede obtener AWS credenciales temporales con permisos que usted defina para acceder directamente a otros recursos Servicios de AWS o a través de Amazon API Gateway.

Temas

Creación de un grupo de identidades en Amazon Cognito

Puede crear un grupo de identidades mediante la consola de Amazon Cognito o utilizar la AWS Command Line Interface (CLI) o las API de Amazon Cognito.

Para crear un grupo de identidades nuevo en la consola

1. Inicie sesión en la consola de Amazon Cognito y seleccione Grupos de identidades.
2. Elija Crear grupo de identidades.
3. En Configurar confianza de grupo de identidades, elija configurar el grupo de identidades para el acceso autenticado, el acceso de invitado o ambos.

    - Si elige Acceso autenticado, seleccione uno o más tipos de identidades que desee establecer como origen de identidades autenticadas en el grupo de identidades. Si

configura un Proveedor de desarrolladores personalizado, no podrá modificarlo ni eliminarlo después de crear el grupo de identidades.

4. En Configurar permisos, elija un rol de IAM predeterminado para los usuarios autenticados o invitados del grupo de identidades.

   a. Elija Crear un nuevo rol de IAM si desea que Amazon Cognito cree uno nuevo para usted con permisos básicos y una relación de confianza con el grupo de identidades. Ingrese un Nombre de rol de IAM para identificar el nuevo rol, por ejemplo `myidentitypool_authenticatedrole`. Seleccione Ver documento de política para revisar los permisos que Amazon Cognito asignará al nuevo rol de IAM.

   b. Puede optar por utilizar una función de IAM existente si ya tiene una función Cuenta de AWS que desee utilizar. Debe configurar la política de confianza del rol de IAM para incluir `cognito-identity.amazonaws.com`. Configure la política de confianza del rol para que solo permita que Amazon Cognito asuma el rol cuando presente pruebas de que la solicitud proviene de un usuario autenticado del grupo de identidades específico. Para obtener más información, consulte Confianza y permisos de rol.

5. En Connect Identity Providers, introduzca los detalles de los proveedores de identidad (IdPs) que eligió en Configurar la confianza del grupo de identidades. Es posible que se le pida que proporcione información del cliente de la aplicación OAuth, que elija un grupo de usuarios de Amazon Cognito, que elija un IdP de IAM o que ingrese un identificador personalizado para un proveedor de desarrolladores.

   a. Elija la Configuración del rol para cada IdP. Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas. Con un IdP del grupo de usuarios de Amazon Cognito, también puede Elegir un rol con preferred_role en los tokens. Para obtener más información acerca de la reclamación de `cognito:preferred_role`, consulte Asignación de valores de prioridad a los grupos.

      i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

      ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

b. Configure Atributos para el control de acceso para cada IdP. Los atributos del control de acceso asignan las reclamaciones de los usuarios a las Etiquetas de las entidades principales que Amazon Cognito aplica a la sesión temporal. Puede crear políticas de IAM para filtrar el acceso de los usuarios en función de las etiquetas que aplique a la sesión.

   i. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

   ii. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

   iii. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

6. En Configurar propiedades, ingrese un Nombre en Nombre de grupo de identidades.
7. En Autenticación básica (clásica), elija si desea Activar el flujo básico. Con el flujo básico activo, puede omitir las funciones que ha seleccionado para usted IdPs y llamar AssumeRoleWithWebIdentitydirectamente. Para obtener más información, consulte Flujo de autenticación de grupos de identidades (identidades federadas).
8. En Etiquetas, elija Agregar etiqueta si quiere aplicar etiquetas al grupo de identidades.
9. En Revisar y crear, confirme las selecciones que realizó para el nuevo grupo de identidades. Seleccione Editar para volver al asistente y cambiar cualquier configuración. Cuando haya acabado, seleccione Crear grupo de identidades.

Configurar un SDK

Para usar los grupos de identidades de Amazon Cognito, configure AWS Amplify AWS SDK for Java, el o el. AWS SDK for .NET Para obtener más información, consulte los siguientes temas.

- Cómo configurar el SDK de JavaScript la Guía para AWS SDK for Java desarrolladores
- Documentación de Amplify en el Amplify Dev Center
- Proveedor de credenciales de Amazon Cognito en la Guía para desarrolladores de AWS SDK for .NET

Integración de los proveedores de identidad

Los grupos de identidades de Amazon Cognito (identidades federadas) admiten la autenticación de usuarios mediante grupos de usuarios de Amazon Cognito, proveedores de identidad federadas (como Amazon, Facebook, Google, Apple y proveedores de identidad SAML) e identidades sin autenticar. Esta característica también es compatible con Identidades autenticadas por el desarrollador (grupos de identidades), que le permite registrar y autenticar usuarios siguiendo su propio proceso de autenticación de backend.

Si desea obtener más información sobre el uso de un grupo de usuarios de Amazon Cognito para crear su propio directorio de usuarios, consulte Grupos de usuarios de Amazon Cognito y Acceder Servicios de AWS mediante un grupo de identidades después de iniciar sesión.

Para obtener más información acerca del uso de proveedores de identidad externos, consulte Proveedores de identidad externos de grupos de identidades.

Para obtener más información acerca de la integración de su propio proceso de autenticación de backend, consulte Identidades autenticadas por el desarrollador (grupos de identidades).

Obtención de credenciales

Los grupos de identidades de Amazon Cognito proporcionan AWS credenciales temporales para los usuarios que son invitados (sin autenticar) y para los usuarios que se han autenticado y recibido un token. Con esas AWS credenciales, su aplicación puede acceder de forma segura a un backend interno AWS o externo AWS a través de Amazon API Gateway. Consulte Obtención de credenciales.

Opciones de configuración guiada para Amazon Cognito

Es posible que desee evaluar las características de Amazon Cognito en una experiencia guiada y estructurada. Estos son algunos recursos externos que proporcionan experiencias personalizadas con grupos de usuarios y grupos de identidades.

Complete un taller

AWS Workshop Studio organiza un taller en el que se explica la configuración de la mayoría de las funciones de Amazon Cognito. Estas características incluyen la API de grupos de usuarios, la interfaz de usuario alojada en los grupos de usuarios, los grupos de identidades y la configuración de seguridad.

Agregue el código de la aplicación a partir de ejemplos

El capítulo de ejemplos de código de esta guía contiene código de aplicación que puede usar con grupos de usuarios y grupos de identidades. La sección de grupos de usuarios del capítulo de ejemplos de código contiene fragmentos breves que describen operaciones individuales y ejemplos más extensos, por end-to-end ejemplo, aplicaciones en diversos lenguajes de programación.

Cree una aplicación completa con AWS Amplify

AWS Amplifyes Servicio de AWS para desarrolladores que desean desarrollar y alojar una aplicación y una interfaz de usuario. Amazon Cognito es el componente de autenticación de Amplify. Al añadir la autenticación a la aplicación, Amplify puede automatizar la implementación de los recursos del grupo de usuarios y del grupo de identidades de Amazon Cognito. Consulte también Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles.

Más recursos de la aplicación Amazon Cognito en GitHub

- Ejemplos de flujo de autenticación con .NET para Amazon Cognito
- Autenticación inalámbrica de Amazon Cognito
- PetStoreejemplo con Amazon Verified Permissions
- Ejemplo de una aplicación React que utiliza ABAC + Identity Pools para acceder a los recursos AWS

- Autorización de máquina a máquina basada en Amazon Cognito y API Gateway mediante CDK AWS
- Creación de autorizaciones detalladas mediante Amazon Cognito, API Gateway e IAM
- CloudFrontautorización @edge

Más talleres

- Implemente la autenticación sin contraseña con Amazon Cognito y WebAuthn
- Identidad de SaaS multiusuario con grupos de usuarios de Amazon Cognito
- Análisis profundo de Amazon Cognito JWT

Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles

Al integrar la aplicación con un cliente de aplicaciones de Amazon Cognito, puede invocar las operaciones de la API para la autenticación y la autorización de los usuarios. Le recomendamos que lo utilice AWS Amplifypara integrar Amazon Cognito con sus aplicaciones web y móviles. AWS Amplify es una solución completa que permite a los desarrolladores web y móviles de interfaz crear, conectar y alojar fácilmente aplicaciones completas AWS, con la flexibilidad de aprovechar la variedad de aplicaciones a Servicios de AWS medida que evolucionan sus casos de uso. Amplify Auth utiliza principalmente Amazon Cognito para crear características de autenticación.

Temas

- Autenticación con AWS Amplify
- Autenticación con SDK de AWS
- Autorización con Amazon Verified Permissions

Una implementación típica de Amazon Cognito utiliza una combinación de herramientas visuales y API. La consola de Amazon Cognito es la interfaz visual para configurar y administrar los grupos de usuarios y grupos de identidades de Amazon Cognito. La interfaz de usuario alojada es una aplicación de inicio de sesión ready-to-use basada en la web que permite probar e implementar rápidamente los grupos de usuarios de Amazon Cognito. Además, en la mayoría de las implementaciones de Amazon Cognito, debe agregar código en las aplicaciones para interactuar con los grupos de usuarios y grupos de identidades. Por ejemplo, es posible que la aplicación invoque la interfaz de usuario alojada para iniciar sesión como usuario y, a continuación, llame al punto de conexión del token desde el código de la aplicación para intercambiar el código de autorización de usuario por tokens. A continuación, la aplicación debe interpretar y almacenar los tokens de usuario y presentarlos en el contexto adecuado para la autenticación y la autorización. Amplify agrega herramientas de integración guiadas con funciones integradas para estos procesos.

También puede crear los recursos de Amazon Cognito completamente en código. Para empezar con el propio código de la aplicación personalizado, consulte ejemplos de código de Amazon Cognito para AWS SDK. Para la integración con Amazon Cognito como proveedor de identidades de OpenID Connect, utilice Herramientas para desarrolladores de OpenID Connect.

Antes de utilizar la autenticación y autorización de Amazon Cognito, elija una plataforma de aplicaciones y prepare el código para integrarlo con el servicio. Para ver las plataformas disponibles, consulte Autenticación con SDK de AWS. AWS CLI Es un SDK de línea de comandos para Amazon Cognito y Servicios de AWS otros, y es un lugar valioso para empezar a familiarizarse con la API de Amazon Cognito.

> **Note**
>
> Algunos componentes de Amazon Cognito solo se pueden configurar con la API. Por ejemplo, solo puede configurar un activador Lambda de remitente de SMS o correo electrónico personalizado para un grupo de usuarios con una solicitud que actualice la `LambdaConfig` propiedad de la UserPoolclase en una solicitud de UpdateUserPoolAPI CreateUserPoolo SMS.

La API de los grupos de usuarios de Amazon Cognito comparte el espacio de nombres con varias clases de operaciones de la API. Una clase configura los grupos de usuarios y los procesos, proveedores de identidades y usuarios. Otra incluye operaciones no autenticadas para que los usuarios de un cliente público inicien sesión, cierren sesión y administren los perfiles. La última clase de operaciones de API realiza operaciones de usuario que usted autoriza con sus propias AWS credenciales en un cliente confidencial del lado del servidor. Debe conocer la arquitectura de la aplicación prevista antes de empezar a implementar el código de la aplicación. Para obtener más información, consulte Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios.

Autenticación con AWS Amplify

AWS Amplify es una solución completa para crear aplicaciones web y móviles. Con Amplify, puede conectarse a los recursos existentes con las bibliotecas de Amplify o puede crear y configurar nuevos recursos con la interfaz de línea de comandos (CLI) de Amplify. Amplify también tiene componentes de interfaz de usuario conectados, como Autenticador para configurar y personalizar la experiencia de inicio y registro en la aplicación.

Para usar las características de autenticación de Amplify en la aplicación de frontend, consulte la siguiente documentación por plataforma.

- Amplify la autenticación para JavaScript
- Autenticación Amplify para iOS

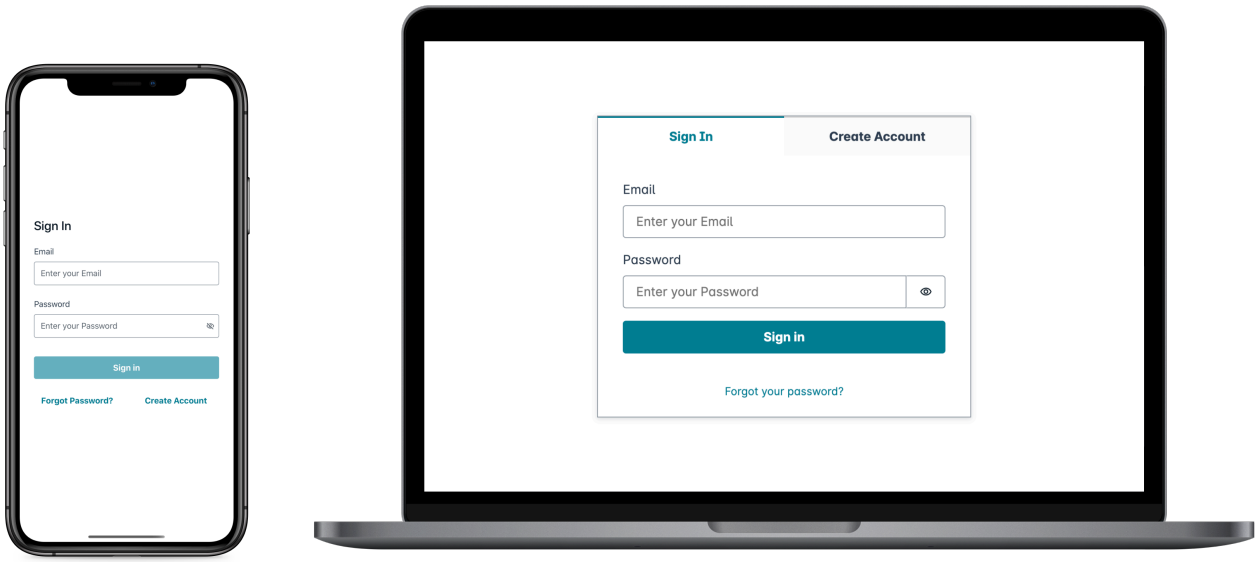
- Autenticación Amplify para Android
- Amplificar la autenticación para Flutter

Las bibliotecas Amplify son de código abierto y están disponibles en. GitHub Para obtener más información sobre cómo Amplify Auth implementa la autenticación de Amazon Cognito, consulte las siguientes bibliotecas.

- amplify-js
- amplify-swift
- amplify-flutter
- amplify-android

Creación de una interfaz de usuario (IU) con Amplify

La Interfaz de usuario alojada de grupos de usuarios de Amazon Cognito puede satisfacer las necesidades esenciales de una frontend de autenticación para una aplicación web o móvil. Para personalizar la interfaz de usuario (IU) más allá de los parámetros que admite la interfaz de usuario alojada, cree una aplicación personalizada. La interfaz de usuario de Amplify es una recopilación personalizable de componentes de frontend en varios idiomas.



Para empezar con el componente de autenticación personalizado, consulte la siguiente documentación del componente del autenticador.

- Autenticador para Android
- Autenticador para Angular
- Autenticador para Flutter
- Autenticador para React
- Autenticador para React Native
- Autenticador para Swift
- Autenticador para Vue

Autenticación con SDK de AWS

Si desea utilizar un backend seguro para crear su propio microservicio de identidad que interactúe con Amazon Cognito, conéctese a los grupos de usuarios de Amazon Cognito y a la API de grupos de identidades de Amazon Cognito con AWS un SDK en el idioma que prefiera.

Para obtener más información sobre cada operación de la API, consulte la referencia de las API de grupos de usuarios de Amazon Cognito y la referencia de las API de Amazon Cognito. Estos documentos contienen secciones Vea también con recursos para utilizar diversos SDK en plataformas compatibles.

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- AWS SDK para JavaScript
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Autorización con Amazon Verified Permissions

Amazon Verified Permissions es un servicio de autorización para las aplicaciones que crea. Al agregar un grupo de usuarios de Amazon Cognito como origen de identidad, la aplicación puede pasar tokens de acceso o identidad (ID) de grupo de usuarios a Verified Permissions para que tomen una decisión de permitir o denegar. Los permisos verificados consideran las propiedades del usuario y el contexto de la solicitud en función de las políticas que escriba en Lenguaje de política de Cedar. El contexto de la solicitud puede incluir un identificador del documento, la imagen u otro recurso que solicitaron y la acción que el usuario desea realizar en el recurso.

Tu aplicación puede proporcionar la identidad de tu usuario o los tokens de acceso a los permisos verificados IsAuthorizedWithTokeno a las solicitudes de BatchIsAuthorizedWithTokenAPI. Estas operaciones de la API aceptan a tus usuarios como usuarios `Principal` y toman decisiones de `Action` autorización para aquellos a los `Resource` que desean acceder. La personalización adicional `Context` puede contribuir a una decisión de acceso detallada.

Cuando la aplicación presenta un token en una solicitud de API `IsAuthorizedWithToken`, Verified Permissions realiza las siguientes validaciones.

1. El grupo de usuarios es un origen de identidad de Verified Permissions configurado para el almacén de políticas solicitado.
2. La reclamación `client_id` o `aud`, en el token de acceso o identidad, respectivamente, coincide con el ID de cliente de la aplicación de un grupo de usuarios que proporcionó a Verified Permissions. Para verificar esta reclamación, debe configurar la validación del ID de cliente en el origen de identidad de Verified Permissions.
3. El token no ha caducado.
4. El valor de la `token_use` reclamación que figura en tu token coincide con los parámetros que has introducido`IsAuthorizedWithToken`. La `token_use` reclamación debe ser `access` si la pasaste al `accessToken` parámetro y `id` si la pasaste al `identityToken` parámetro.
5. La firma del token proviene de las claves web JSON (JWK) publicadas del grupo de usuarios. Puede consultar JWK en `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/.well-known/jwks.json`.

Tokens revocados y usuarios eliminados

Los permisos verificados solo validan la información que conoce del origen de identidad y de la fecha de caducidad del token del usuario. Los permisos verificados no comprueban la revocación del token

ni la existencia del usuario. Si revocó el token del usuario o eliminó el perfil de usuario del grupo de usuarios, Verified Permissions seguirá considerando que el token es válido hasta que caduque.

Evaluación de políticas

Configure el grupo de usuarios como origen de identidad para el almacén de políticas. Configure la aplicación para enviar los tokens de los usuarios en las solicitudes de permisos verificados. Para cada solicitud, Verified Permissions compara las reclamaciones del token con una política. Una política de Verified Permissions es como una política de IAM en AWS. Declara un entidad principal, un recurso y una acción. Verified Permissions responde a tu solicitud `Allow` si coincide con una acción permitida y no coincide con una `Deny` acción explícita; de lo contrario, responde con`Deny`. Para obtener más información, consulte las políticas de Amazon Verified Permissions en la Guía del usuario de Amazon Verified Permissions.

Personalización de tokens

Para cambiar, añadir y eliminar las reclamaciones de los usuarios que deseas presentar a Verified Permissions, personaliza el contenido de tus identificadores de acceso e identidad con unDesencadenador de Lambda anterior a la generación del token. Con un desencadenador previo a la generación del token, puede agregar y modificar reclamaciones en los tokens. Por ejemplo, puede consultar una base de datos para atributos de usuario adicionales y codificarlos en el token de ID.

> **Note**
>
> Debido a la forma en que Verified Permissions procesa las reclamaciones, no agregue las reclamaciones con nombres `cognito`, `dev` y `custom` en la función de generación previa al token. Si presenta estos prefijos de reclamación reservados no en formato delimitado por dos puntos, como `cognito:username` sino como nombres de reclamación completos, las solicitudes de autorización producen un error.

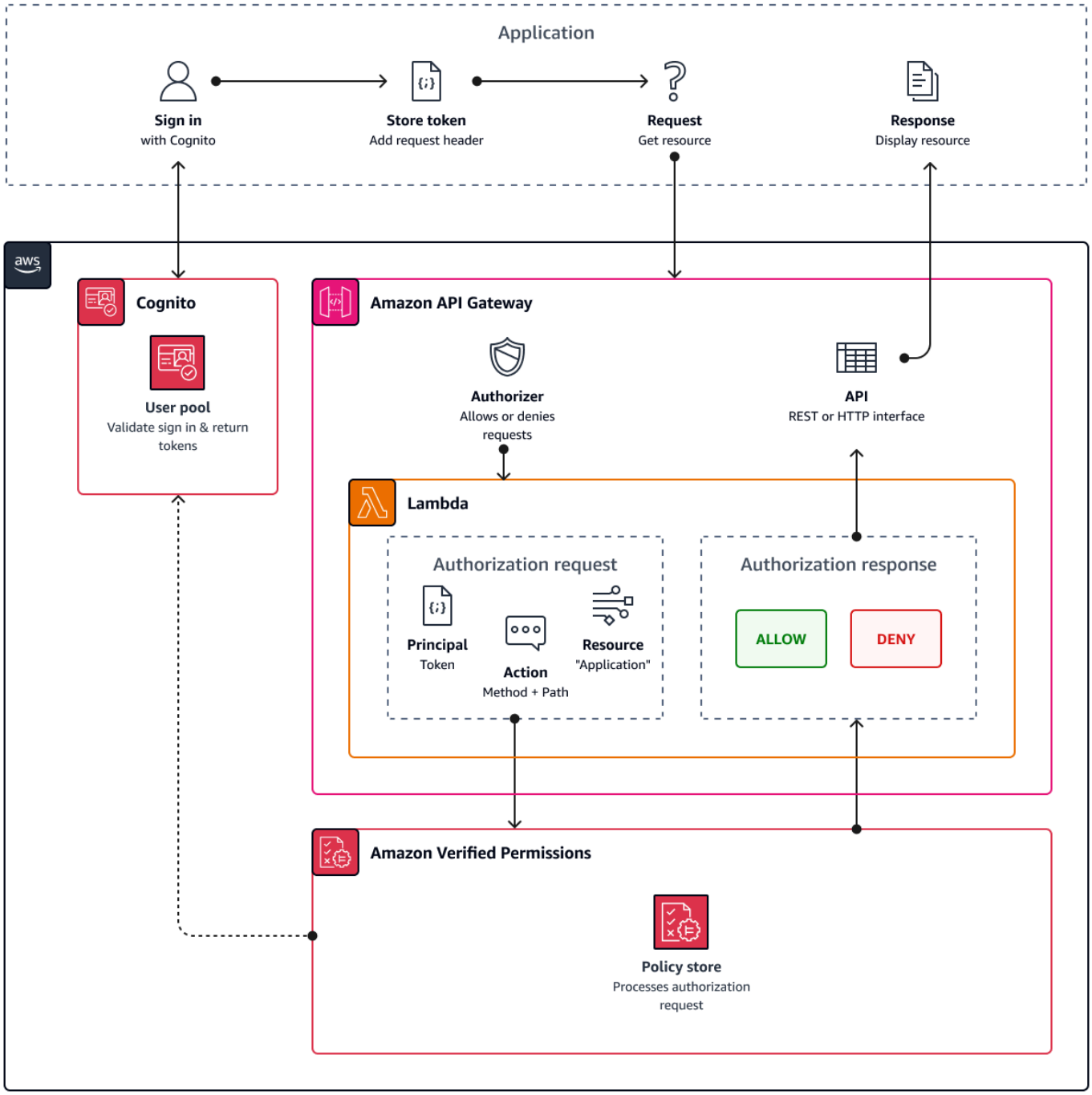
Para obtener más información sobre cómo los permisos verificados mapean las reclamaciones de los tokens de Amazon Cognito a las políticas de autorización, consulte el esquema de mapeo de los tokens de Amazon Cognito a permisos verificados.

Autorización de API con permisos verificados

Su ID o sus tokens de acceso pueden autorizar las solicitudes a las API REST de Amazon API Gateway de backend con permisos verificados. Puede crear un almacén de políticas con enlaces

inmediatos a su grupo de usuarios y a su API. Con la opción de inicio Configurar con Cognito y API Gateway, Verified Permissions añade una fuente de identidad del grupo de usuarios al almacén de políticas y un autorizador Lambda a la API. Cuando la aplicación pasa un token portador del grupo de usuarios a la API, el autorizador de Lambda invoca los permisos verificados. El autorizador transfiere el token como principal y la ruta y el método de solicitud como acción.

El siguiente diagrama ilustra el flujo de autorización de una API API Gateway con permisos verificados. Para obtener un desglose detallado, consulta los almacenes de políticas vinculados a API en la Guía del usuario de permisos verificados de Amazon.



Verified Permissions estructura la autorización de la API en torno a grupos de usuarios. Como tanto el identificador como el token de acceso incluyen una `cognito:groups` reclamación, su almacén de políticas puede gestionar el control de acceso basado en roles (RBAC) para sus API en una variedad de contextos de aplicación.

Elegir la configuración del almacén de políticas

Al configurar una fuente de identidad en un almacén de políticas, debe elegir si desea procesar los tokens de acceso o de identificación. Esta decisión es importante para el funcionamiento de su motor de políticas. Los tokens de identificación contienen atributos de usuario. Los tokens de acceso contienen información sobre el control de acceso de los usuarios: ámbitos de OAuth. Si bien ambos tipos de token contienen información sobre la pertenencia a un grupo, generalmente recomendamos el token de acceso para RBAC con un almacén de políticas de permisos verificados. El token de acceso aumenta la pertenencia a un grupo con alcances que pueden contribuir a la decisión de autorización. Las afirmaciones de un token de acceso pasan a formar parte del contexto de la solicitud de autorización.

También debe configurar los tipos de entidades de usuario y grupo al configurar un grupo de usuarios como fuente de identidad. Los tipos de entidad son identificadores principales, de acciones y de recursos a los que puede hacer referencia en las políticas de permisos verificados. Las entidades de los almacenes de políticas pueden tener una relación de pertenencia, en la que una entidad puede ser miembro de una entidad principal. Con la pertenencia, puede hacer referencia a grupos principales, grupos de acción y grupos de recursos. En el caso de los grupos de usuarios, el tipo de entidad de usuario que especifique debe ser miembro del tipo de entidad del grupo. Al configurar un almacén de políticas vinculado a una API o seguir la configuración guiada en la consola de permisos verificados, el almacén de políticas tiene automáticamente esta relación padre-miembro.

El token de identificación puede combinar el RBAC con el control de acceso basado en atributos (ABAC). Tras crear un almacén de políticas vinculado a una API, puede mejorarlas con los atributos de los usuarios y la pertenencia a grupos. Las afirmaciones de atributos de un token de ID se convierten en los atributos principales de la solicitud de autorización. Sus políticas pueden tomar decisiones de autorización en función de los atributos principales.

También puede configurar un almacén de políticas para que acepte tokens con una `aud` o una `client_id` afirmación que coincida con una lista de clientes de aplicaciones aceptables que usted proporcione.

Ejemplo de política para la autorización de API basada en roles

El siguiente ejemplo de política se creó mediante la configuración de un almacén de políticas de permisos verificados para una API REST de PetStoreejemplo.

```
permit(
  principal in PetStore::UserGroup::"us-east-1_EXAMPLE|MyGroup",
```

```
  action in [ PetStore::Action::"get /pets", PetStore::Action::"get /pets/{petId}" ],
  resource
  );
```

Verified Permissions devuelve una `Allow` decisión a la solicitud de autorización de tu aplicación cuando:

1. Tu aplicación pasó un identificador o un token de acceso en un `Authorization` encabezado como token portador.
2. Tu aplicación pasó un token con una `cognito:groups` afirmación que contiene la cadena`MyGroup`.
3. Su solicitud hizo una `HTTP GET` solicitud a, por ejemplo, `https://myapi.example.com/pets` o`https://myapi.example.com/pets/scrappy`.

Política de ejemplo para un usuario de Amazon Cognito

Su grupo de usuarios también puede generar solicitudes de autorización para permisos verificados en condiciones distintas de las solicitudes de API. Puede enviar cualquier decisión de control de acceso de su aplicación a su almacén de políticas. Por ejemplo, puede complementar la seguridad de Amazon DynamoDB o Amazon S3 con un control de acceso basado en atributos antes de que las solicitudes transiten por la red, lo que reduce el uso de la cuota.

El siguiente ejemplo utiliza el Lenguaje de políticas de Cedar para permitir que los usuarios del departamento de Finanzas que se autentiquen con un cliente de aplicación de grupo de usuarios puedan leer y escribir `example_image.png`. John, un usuario de la aplicación, recibe un token de ID del cliente de la aplicación y lo pasa en una solicitud GET a una URL que requiere autorización, `https://example.com/images/example_image.png`. El token de ID de John tiene una reclamación `aud` del ID de cliente de la aplicación de grupo de usuarios `1234567890example`. La función de Lambda previa a la generación del token también insertó una nueva reclamación `costCenter` con un valor, para John, de `Finance1234`.

```
permit (
   principal,
   actions in [ExampleCorp::Action::"readFile", "writeFile"],
   resource == ExampleCorp::Photo::"example_image.png"
)
when {
   principal.aud == "1234567890example" &&
```

```
   principal.custom.costCenter like "Finance*"
};
```

El siguiente cuerpo de la solicitud da como resultado una respuesta `Allow`.

```
{
   "accesstoken": "[John's ID token]",
   "action": {
      "actionId": "readFile",
      "actionType": "Action"
   },
   "resource": {
      "entityId": "example_image.png",
      "entityType": "Photo"
   }
}
```

Cuando desee especificar una entidad principal en una política de Verified Permissions, utilice el siguiente formato:

```
permit (
   principal == [Namespace]::[Entity]::"[user pool ID]"|"[user sub]",
   action,
   resource
);
```

A continuación, se muestra un ejemplo principal para un usuario de un grupo de usuarios con un identificador `us-east-1_Example` con un subidentificador o identificador de usuario. `973db890-092c-49e4-a9d0-912a4c0a20c7`

```
principal == ExampleCorp::User::"us-east-1_Example|973db890-092c-49e4-
a9d0-912a4c0a20c7",
```

Cuando desee especificar un grupo de usuarios en una política de permisos verificados, utilice el siguiente formato:

```
permit (
   principal in [Namespace]::[Group Entity]::"[Group name]",
   action,
   resource
```

```
);
```

A continuación se muestra un ejemplo

Control de acceso basado en atributos

La autorización con permisos verificados para sus aplicaciones y los atributos para la función de control de acceso de los grupos de identidades de Amazon Cognito para AWS las credenciales son dos formas de control de acceso basado en atributos (ABAC). A continuación, se muestra una comparación de las características de Verified Permissions y Amazon Cognito ABAC. En ABAC, un sistema examina los atributos de una entidad y toma una decisión de autorización a partir de las condiciones que usted defina.

Servicio	Proceso	Resultado
Amazon Verified Permissions	Devuelve una `Deny` decisión `Allow` o una decisión obtenida a partir del análisis de un grupo de usuarios (JWT).	El acceso a los recursos de la aplicación se realiza correctamente o fracasa según la evaluación de las políticas de Cedar.
Grupos de identidades de Amazon Cognito (atributos para el control de acceso)	Asigna etiquetas de sesión a su usuario en función de sus atributos . Las condiciones de la política de IAM pueden comprobar las etiquetas `Allow` o el acceso `Deny` de los usuarios. Servicios de AWS	Una sesión etiquetada con AWS credenciales temporales para un rol de IAM.

Ejemplos de código de Amazon Cognito con AWS SDK

En los siguientes ejemplos de código, se muestra cómo utilizar Amazon Cognito con un kit de desarrollo de software (SDK) de AWS.

Para obtener una lista completa de las guías para desarrolladores del SDK de AWS y ejemplos de código, consulte Uso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Ejemplos de código para Amazon Cognito Identity mediante SDK AWS
  - Acciones para Amazon Cognito Identity mediante SDK AWS
    - Úselo CreateIdentityPool con un AWS SDK o CLI
    - Úselo DeleteIdentityPool con un AWS SDK o CLI
    - Úselo DescribeIdentityPool con un AWS SDK o CLI
    - Úselo GetCredentialsForIdentity con un AWS SDK o CLI
    - Úselo GetIdentityPoolRoles con un AWS SDK o CLI
    - Úselo ListIdentityPools con un AWS SDK o CLI
    - Úselo SetIdentityPoolRoles con un AWS SDK o CLI
    - Úselo UpdateIdentityPool con un AWS SDK o CLI
  - Ejemplos de servicios cruzados para Amazon Cognito Identity mediante SDK AWS
    - Cree una aplicación Amazon Transcribe
    - Creación de una aplicación de exploración de Amazon Textract
- Ejemplos de código para Amazon Cognito Identity Provider mediante SDK AWS
  - Acciones para el proveedor de identidad de Amazon Cognito mediante SDK AWS
    - Úselo AdminCreateUser con un AWS SDK o CLI
    - Úselo AdminGetUser con un AWS SDK o CLI
    - Úselo AdminInitiateAuth con un AWS SDK o CLI
    - Úselo AdminRespondToAuthChallenge con un AWS SDK o CLI
    - Úselo AdminSetUserPassword con un AWS SDK o CLI
    - Úselo AssociateSoftwareToken con un AWS SDK o CLI
    - Úselo ConfirmDevice con un AWS SDK o CLI

Ejemplos de código para Amazon Cognito Identity mediante SDK AWS

Los siguientes ejemplos de código muestran cómo utilizar Amazon Cognito Identity con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los ejemplos con varios servicios son aplicaciones de muestra que funcionan con varios Servicios de AWS.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Acciones para Amazon Cognito Identity mediante SDK AWS
  - Úselo CreateIdentityPool con un AWS SDK o CLI
  - Úselo DeleteIdentityPool con un AWS SDK o CLI
  - Úselo DescribeIdentityPool con un AWS SDK o CLI
  - Úselo GetCredentialsForIdentity con un AWS SDK o CLI
  - Úselo GetIdentityPoolRoles con un AWS SDK o CLI
  - Úselo ListIdentityPools con un AWS SDK o CLI
  - Úselo SetIdentityPoolRoles con un AWS SDK o CLI
  - Úselo UpdateIdentityPool con un AWS SDK o CLI
- Ejemplos de servicios cruzados para Amazon Cognito Identity mediante SDK AWS
  - Cree una aplicación Amazon Transcribe
  - Creación de una aplicación de exploración de Amazon Textract

Acciones para Amazon Cognito Identity mediante SDK AWS

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Cognito Identity con AWS los SDK. Estos fragmentos llaman a la API de identidades de Amazon Cognito

y son fragmentos de código de programas más grandes que se deben ejecutar en contexto. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para consultar la lista completa, vea la referencia de la API de Amazon Cognito Identity.

Ejemplos

- Úselo CreateIdentityPool con un AWS SDK o CLI
- Úselo DeleteIdentityPool con un AWS SDK o CLI
- Úselo DescribeIdentityPool con un AWS SDK o CLI
- Úselo GetCredentialsForIdentity con un AWS SDK o CLI
- Úselo GetIdentityPoolRoles con un AWS SDK o CLI
- Úselo ListIdentityPools con un AWS SDK o CLI
- Úselo SetIdentityPoolRoles con un AWS SDK o CLI
- Úselo UpdateIdentityPool con un AWS SDK o CLI

Úselo **`CreateIdentityPool`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `CreateIdentityPool`.

CLI

AWS CLI

Para crear un grupo de identidades con el proveedor de grupos de identidades de Cognito

En este ejemplo se crea un grupo de identidades denominado MyIdentityPool. Tiene un proveedor de grupo de identidades de Cognito. No se permiten identidades no autenticadas.

Comando:

```
aws cognito-identity create-identity-pool --identity-pool-name
 MyIdentityPool --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Salida:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Para obtener más información sobre la API, consulte CreateIdentityPoolla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.CreateIdentityPoolRequest;
import
 software.amazon.awssdk.services.cognitoidentity.model.CreateIdentityPoolResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
```

```
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateIdentityPool {
    public static void main(String[] args) {
        final String usage = """
                Usage:
                    <identityPoolName>\s

                Where:
                    identityPoolName - The name to give your identity pool.
                """;

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

        String identityPoolName = args[0];
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String identityPoolId = createIdPool(cognitoClient, identityPoolName);
        System.out.println("Unity pool ID " + identityPoolId);
        cognitoClient.close();
    }

    public static String createIdPool(CognitoIdentityClient cognitoClient, String
 identityPoolName) {
        try {
            CreateIdentityPoolRequest poolRequest =
 CreateIdentityPoolRequest.builder()
                    .allowUnauthenticatedIdentities(false)
                    .identityPoolName(identityPoolName)
                    .build();

            CreateIdentityPoolResponse response =
 cognitoClient.createIdentityPool(poolRequest);
            return response.identityPoolId();

        } catch (CognitoIdentityProviderException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
        return "";
    }
}
```

- Para obtener más información sobre la API, consulta CreateIdentityPoolla Referencia AWS SDK for Java 2.x de la API.

PowerShell

Herramientas para PowerShell

Ejemplo 1: Crea un nuevo grupo de identidades que permite identidades no autenticadas.

```
New-CGIIdentityPool -AllowUnauthenticatedIdentities $true -IdentityPoolName
 CommonTests13
```

Salida:

```
LoggedAt                        : 8/12/2015 4:56:07 PM
AllowUnauthenticatedIdentities  : True
DeveloperProviderName           :
IdentityPoolId                  : us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3
IdentityPoolName                : CommonTests13
OpenIdConnectProviderARNs       : {}
SupportedLoginProviders         : {}
ResponseMetadata                : Amazon.Runtime.ResponseMetadata
ContentLength                   : 136
HttpStatusCode                  : OK
```

- Para obtener más información sobre la API, consulte la referencia de CreateIdentityPool AWS Tools for PowerShellcmdlets.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree un nuevo grupo de identidades.

```
    /// Create a new identity pool and return its ID.
    ///
    /// - Parameters:
    ///     - name: The name to give the new identity pool.
    ///
    /// - Returns: A string containing the newly created pool's ID, or `nil`
    ///   if an error occurred.
    ///
    func createIdentityPool(name: String) async throws -> String? {
        let cognitoInputCall = CreateIdentityPoolInput(developerProviderName:
"com.exampleco.CognitoIdentityDemo",
                                                       identityPoolName: name)

        let result = try await cognitoIdentityClient.createIdentityPool(input:
cognitoInputCall)
        guard let poolId = result.identityPoolId else {
            return nil
        }

        return poolId
    }
```

- Para obtener información, consulte Guía para desarrolladores del SDK de AWS SDK para Swift.
- Para obtener más información sobre la API, consulta CreateIdentityPoolla referencia sobre la API de AWS SDK for Swift.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DeleteIdentityPool`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DeleteIdentityPool`.

CLI

AWS CLI

Para eliminar un grupo de identidades

En el siguiente ejemplo de `delete-identity-pool` se elimina el grupo de identidades especificado.

Comando:

```
aws cognito-identity delete-identity-pool \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Este comando no genera ninguna salida.

- Para obtener más información sobre la API, consulte DeleteIdentityPoolla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.awscore.exception.AwsServiceException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.DeleteIdentityPoolRequest;

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

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

                Usage:
                    <identityPoolId>\s

                Where:
                    identityPoolId - The Id value of your identity pool.
                """;

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

        String identityPoold = args[0];
        CognitoIdentityClient cognitoIdClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        deleteIdPool(cognitoIdClient, identityPoold);
        cognitoIdClient.close();
    }
```

```
    public static void deleteIdPool(CognitoIdentityClient cognitoIdClient, String
 identityPoold) {
        try {

            DeleteIdentityPoolRequest identityPoolRequest =
 DeleteIdentityPoolRequest.builder()
                    .identityPoolId(identityPoold)
                    .build();

            cognitoIdClient.deleteIdentityPool(identityPoolRequest);
            System.out.println("Done");

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

- Para obtener más información sobre la API, consulta DeleteIdentityPoolla Referencia AWS SDK for Java 2.x de la API.

PowerShell

Herramientas para PowerShell

Ejemplo 1: Elimina un grupo de identidades específico.

```
Remove-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

- Para obtener más información sobre la API, consulte la referencia DeleteIdentityPoolde AWS Tools for PowerShell cmdlets.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine el grupo de identidades especificado.

```
/// Delete the specified identity pool.
///
/// - Parameters:
///   - id: The ID of the identity pool to delete.
///
func deleteIdentityPool(id: String) async throws {
    let input = DeleteIdentityPoolInput(
        identityPoolId: id
    )

    _ = try await cognitoIdentityClient.deleteIdentityPool(input: input)
}
```

- Para obtener información, consulte Guía para desarrolladores del SDK de AWS SDK para Swift.
- Para obtener más información sobre la API, consulta DeleteIdentityPoolla referencia sobre la API de AWS SDK for Swift.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DescribeIdentityPool`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DescribeIdentityPool`.

CLI

AWS CLI

Para describir un grupo de identidades

En este ejemplo se describe un grupo de identidades.

Comando:

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

Salida:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Para obtener más información sobre la API, consulte DescribeIdentityPoolla Referencia de AWS CLI comandos.

PowerShell

Herramientas para PowerShell

Ejemplo 1: recupera información sobre un grupo de identidades específico por su identificador.

```
Get-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

Salida:

```
LoggedAt                        : 8/12/2015 4:29:40 PM
AllowUnauthenticatedIdentities  : True
DeveloperProviderName           :
IdentityPoolId                  : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
IdentityPoolName                : CommonTests1
OpenIdConnectProviderARNs       : {}
SupportedLoginProviders         : {}
ResponseMetadata                : Amazon.Runtime.ResponseMetadata
ContentLength                   : 142
HttpStatusCode                  : OK
```

- Para obtener más información sobre la API, consulte la referencia DescribeIdentityPoolde AWS Tools for PowerShell cmdlets.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetCredentialsForIdentity`** con un AWS SDK o CLI

En el siguiente ejemplo de código, se muestra cómo usar `GetCredentialsForIdentity`.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.GetCredentialsForIdentityRequest;
import
 software.amazon.awssdk.services.cognitoidentity.model.GetCredentialsForIdentityResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc

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

        final String usage = """

                Usage:
                    <identityId>\s

            Where:
                identityId - The Id of an existing identity in the format
 REGION:GUID.
            """;

        if (args.length != 1) {
```

```
            System.out.println(usage);
            System.exit(1);
        }

        String identityId = args[0];
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .build();

        getCredsForIdentity(cognitoClient, identityId);
        cognitoClient.close();
    }

    public static void getCredsForIdentity(CognitoIdentityClient cognitoClient,
 String identityId) {
        try {
            GetCredentialsForIdentityRequest getCredentialsForIdentityRequest =
 GetCredentialsForIdentityRequest
                    .builder()
                    .identityId(identityId)
                    .build();

            GetCredentialsForIdentityResponse response = cognitoClient
                    .getCredentialsForIdentity(getCredentialsForIdentityRequest);
            System.out.println(
                    "Identity ID " + response.identityId() + ", Access key ID " +
 response.credentials().accessKeyId());

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

- Para obtener más información sobre la API, consulta GetCredentialsForIdentityla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetIdentityPoolRoles`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `GetIdentityPoolRoles`.

CLI

AWS CLI

Para obtener los roles del grupo de identidades

En este ejemplo se obtienen los roles del grupo de identidades.

Comando:

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

Salida:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "Roles": {
      "authenticated": "arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolAuth_Role",
      "unauthenticated": "arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolUnauth_Role"
  }
}
```

- Para obtener más información sobre la API, consulte GetIdentityPoolRolesla Referencia de AWS CLI comandos.

PowerShell

Herramientas para PowerShell

Ejemplo 1: Obtiene la información sobre las funciones de un grupo de identidades específico.

```
Get-CGIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1
```

Salida:

```
LoggedAt         : 8/12/2015 4:33:51 PM
IdentityPoolId   : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
Roles            : {[unauthenticated, arn:aws:iam::123456789012:role/
CommonTests1Role]}
ResponseMetadata : Amazon.Runtime.ResponseMetadata
ContentLength    : 165
HttpStatusCode   : OK
```

- Para obtener más información sobre la API, consulte GetIdentityPoolRoleslа referencia de AWS Tools for PowerShell cmdlets.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListIdentityPools`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ListIdentityPools`.

CLI

AWS CLI

Para mostrar grupos de identidades

En este ejemplo, se muestran los grupos de identidades. Hay un máximo de 20 identidades en la lista.

Comando:

```
aws cognito-identity list-identity-pools --max-results 20
```

Salida:

```
{
  "IdentityPools": [
      {
          "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
          "IdentityPoolName": "MyIdentityPool"
      },
      {
```

```
            "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
            "IdentityPoolName": "AnotherIdentityPool"
        },
        {
            "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
            "IdentityPoolName": "IdentityPoolRegionA"
        }
    ]
}
```

- Para obtener más información sobre la API, consulte ListIdentityPoolsla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cognitoidentity.CognitoIdentityClient;
import
 software.amazon.awssdk.services.cognitoidentity.model.ListIdentityPoolsRequest;
import
 software.amazon.awssdk.services.cognitoidentity.model.ListIdentityPoolsResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc

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

```
public class ListIdentityPools {
    public static void main(String[] args) {
        CognitoIdentityClient cognitoClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listIdPools(cognitoClient);
        cognitoClient.close();
    }

    public static void listIdPools(CognitoIdentityClient cognitoClient) {
        try {
            ListIdentityPoolsRequest poolsRequest =
 ListIdentityPoolsRequest.builder()
                    .maxResults(15)
                    .build();

            ListIdentityPoolsResponse response =
 cognitoClient.listIdentityPools(poolsRequest);
            response.identityPools().forEach(pool -> {
                System.out.println("Pool ID: " + pool.identityPoolId());
                System.out.println("Pool name: " + pool.identityPoolName());
            });

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

- Para obtener más información sobre la API, consulta ListIdentityPoolsla Referencia AWS SDK for Java 2.x de la API.

PowerShell

Herramientas para PowerShell

Ejemplo 1: recupera una lista de grupos de identidades existentes.

```
Get-CGIIdentityPoolList
```

Salida:

```
IdentityPoolId
 IdentityPoolName
--------------
 ----------------
us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1                    CommonTests1
us-east-1:118d242d-204e-4b88-b803-EXAMPLEGUID2                    Tests2
us-east-1:15d49393-ab16-431a-b26e-EXAMPLEGUID3                    CommonTests13
```

- Para obtener más información sobre la API, consulte la referencia ListIdentityPoolsde AWS Tools for PowerShell cmdlets.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Busque el ID de un grupo de identidades con su nombre.

```
    /// Return the ID of the identity pool with the specified name.
    ///
    /// - Parameters:
    ///   - name: The name of the identity pool whose ID should be returned.
    ///
    /// - Returns: A string containing the ID of the specified identity pool
    ///   or `nil` on error or if not found.
    ///
    func getIdentityPoolID(name: String) async throws -> String? {
```

```
        var token: String? = nil

        // Iterate over the identity pools until a match is found.

        repeat {
            /// `token` is a value returned by `ListIdentityPools()` if the
            /// returned list of identity pools is only a partial list. You
            /// use the `token` to tell Amazon Cognito that you want to
            /// continue where you left off previously. If you specify `nil`
            /// or you don't provide the token, Amazon Cognito will start at
            /// the beginning.

            let listPoolsInput = ListIdentityPoolsInput(maxResults: 25,
nextToken: token)

            /// Read pages of identity pools from Cognito until one is found
            /// whose name matches the one specified in the `name` parameter.
            /// Return the matching pool's ID. Each time we ask for the next
            /// page of identity pools, we pass in the token given by the
            /// previous page.

            let output = try await cognitoIdentityClient.listIdentityPools(input:
listPoolsInput)

            if let identityPools = output.identityPools {
                for pool in identityPools {
                    if pool.identityPoolName == name {
                        return pool.identityPoolId!
                    }
                }
            }

            token = output.nextToken
        } while token != nil

        return nil
    }
```

Obtenga el ID de un grupo de identidades existente o créelo si aún no existe.

```
    /// Return the ID of the identity pool with the specified name.
    ///
```

```
    /// - Parameters:
    ///   - name: The name of the identity pool whose ID should be returned
    ///
    /// - Returns: A string containing the ID of the specified identity pool.
    ///   Returns `nil` if there's an error or if the pool isn't found.
    ///
    public func getOrCreateIdentityPoolID(name: String) async throws -> String? {
        // See if the pool already exists. If it doesn't, create it.

        guard let poolId = try await self.getIdentityPoolID(name: name) else {
            return try await self.createIdentityPool(name: name)
        }

        return poolId
    }
```

- Para obtener información, consulte Guía para desarrolladores del SDK de AWS SDK para Swift.
- Para obtener más información sobre la API, consulta ListIdentityPoolsla referencia sobre la API de AWS SDK for Swift.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`SetIdentityPoolRoles`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `SetIdentityPoolRoles`.

CLI

AWS CLI

Para establecer las funciones del grupo de identidades

El siguiente `set-identity-pool-roles` ejemplo establece un rol de grupo de identidades.

```
aws cognito-identity set-identity-pool-roles \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" \
```

```
    --roles authenticated="arn:aws:iam::111111111111:role/
Cognito_MyIdentityPoolAuth_Role"
```

- Para obtener más información sobre la API, consulte SetIdentityPoolRoleslа Referencia de AWS CLI comandos.

PowerShell

Herramientas para PowerShell

Ejemplo 1: Configura el grupo de identidades específico para que tenga una función de IAM no autenticada.

```
Set-CGIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -Role @{ "unauthenticated" = "arn:aws:iam::123456789012:role/
CommonTests1Role" }
```

- Para obtener más información sobre la API, consulte SetIdentityPoolRoleslа referencia de cmdlets.AWS Tools for PowerShell

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`UpdateIdentityPool`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `UpdateIdentityPool`.

CLI

AWS CLI

Para actualizar un grupo de identidades

En este ejemplo se actualiza un grupo de identidades. Establece el nombre en MyIdentityPool. Añade Cognito como proveedor de identidad. No permite las identidades no autenticadas.

Comando:

```
aws cognito-identity update-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111" --identity-pool-name
```

```
 "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Salida:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Para obtener más información sobre la API, consulte la Referencia de UpdateIdentityPoolcomandos AWS CLI .

PowerShell

Herramientas para PowerShell

Ejemplo 1: actualiza algunas de las propiedades del grupo de identidades, en este caso el nombre del grupo de identidades.

```
Update-CGIIdentityPool -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -IdentityPoolName NewPoolName
```

Salida:

```
LoggedAt                      : 8/12/2015 4:53:33 PM
AllowUnauthenticatedIdentities : False
DeveloperProviderName         :
IdentityPoolId                : us-east-1:0de2af35-2988-4d0b-b22d-EXAMPLEGUID1
IdentityPoolName              : NewPoolName
OpenIdConnectProviderARNs     : {}
```

```
SupportedLoginProviders        : {}
ResponseMetadata               : Amazon.Runtime.ResponseMetadata
ContentLength                  : 135
HttpStatusCode                 : OK
```

- Para obtener más información sobre la API, consulte UpdateIdentityPoolla referencia de AWS Tools for PowerShell cmdlets.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de servicios cruzados para Amazon Cognito Identity mediante SDK AWS

Las siguientes aplicaciones de ejemplo utilizan AWS los SDK para combinar Amazon Cognito Identity con otros. Servicios de AWS Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar la aplicación.

Ejemplos

- Cree una aplicación Amazon Transcribe
- Creación de una aplicación de exploración de Amazon Textract

Cree una aplicación Amazon Transcribe

En el siguiente ejemplo de código, se muestra cómo utilizar Amazon Transcribe para transcribir y mostrar grabaciones de voz en el navegador.

JavaScript

SDK para JavaScript (v3)

Cree una aplicación que utilice Amazon Transcribe para transcribir y mostrar grabaciones de voz en el navegador. La aplicación utiliza dos buckets de Amazon Simple Storage Service (Amazon S3), uno para alojar el código de la aplicación y otro para almacenar transcripciones. La aplicación utiliza un grupo de usuarios de Amazon Cognito para autenticar a los usuarios. Los usuarios autenticados tienen permisos AWS Identity and Access Management (IAM) para acceder a los servicios necesarios. AWS

Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en. GitHub

Este ejemplo también está disponible en la guía para desarrolladores de AWS SDK for JavaScript v3.

Servicios utilizados en este ejemplo

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Creación de una aplicación de exploración de Amazon Textract

Los siguientes ejemplos de código indican cómo explorar la salida de Amazon Textract mediante una aplicación interactiva.

JavaScript

SDK para JavaScript (v3)

Muestra cómo utilizarla AWS SDK for JavaScript para crear una aplicación de React que utilice Amazon Textract para extraer datos de la imagen de un documento y mostrarlos en una página web interactiva. Este ejemplo se ejecuta en un navegador web y requiere una identidad autenticada de Amazon Cognito para las credenciales. Para el almacenamiento utiliza Amazon Simple Storage Service (Amazon S3) y para las notificaciones consulta una cola de Amazon Simple Queue Service (Amazon SQS) que está suscrita a un tema de Amazon Simple Notification Service (Amazon SNS).

Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Cognito Identity
- Amazon S3

- Amazon SNS
- Amazon SQS
- Amazon Textract

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código para Amazon Cognito Identity Provider mediante SDK AWS

Los siguientes ejemplos de código muestran cómo utilizar Amazon Cognito Identity Provider con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Introducción

Introducción a Amazon Cognito

En los siguientes ejemplos de código se muestra cómo empezar a utilizar Amazon Cognito.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Código para el MakeLists archivo CMake C.txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS cognito-idp)

# Set this project's name.
project("hello_cognito")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
     # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

     # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
 may need to uncomment this
                                 # and set the proper subdirectory to the
 executables' location.

     AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_cognito.cpp)
```

```
target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Código del archivo de origen hello_cognito.cpp.

```
#include <aws/core/Aws.h>
#include <aws/cognito-idp/CognitoIdentityProviderClient.h>
#include <aws/cognito-idp/model/ListUserPoolsRequest.h>
#include <iostream>

/*
 *  A "Hello Cognito" starter application which initializes an Amazon Cognito
 client and lists the Amazon Cognito
 *  user pools.
 *
 *  main function
 *
 *  Usage: 'hello_cognito'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
 cognitoClient(clientConfig);

        Aws::String nextToken; // Used for pagination.
        std::vector<Aws::String> userPools;

        do {
            Aws::CognitoIdentityProvider::Model::ListUserPoolsRequest
 listUserPoolsRequest;
            if (!nextToken.empty()) {
```

```
                listUserPoolsRequest.SetNextToken(nextToken);
            }

            Aws::CognitoIdentityProvider::Model::ListUserPoolsOutcome
 listUserPoolsOutcome =
                    cognitoClient.ListUserPools(listUserPoolsRequest);

            if (listUserPoolsOutcome.IsSuccess()) {
                for (auto &userPool:
 listUserPoolsOutcome.GetResult().GetUserPools()) {

                    userPools.push_back(userPool.GetName());
                }

                nextToken = listUserPoolsOutcome.GetResult().GetNextToken();
            } else {
                std::cerr << "ListUserPools error: " <<
 listUserPoolsOutcome.GetError().GetMessage() << std::endl;
                result = 1;
                break;
            }


        } while (!nextToken.empty());
        std::cout << userPools.size() << " user pools found." << std::endl;
        for (auto &userPool: userPools) {
            std::cout << "   user pool: " << userPool << std::endl;
        }
    }

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Para obtener más información sobre la API, consulte la Referencia de ListUserPoolsla AWS SDK for C++ API.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
package main

import (
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Notification
 Service
// (Amazon SNS) client and list the topics in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 cognitoClient := cognitoidentityprovider.NewFromConfig(sdkConfig)
 fmt.Println("Let's list the user pools for your account.")
 var pools []types.UserPoolDescriptionType
 paginator := cognitoidentityprovider.NewListUserPoolsPaginator(
```

```
  cognitoClient, &cognitoidentityprovider.ListUserPoolsInput{MaxResults:
 aws.Int32(10)})
 for paginator.HasMorePages() {
  output, err := paginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get user pools. Here's why: %v\n", err)
  } else {
   pools = append(pools, output.UserPools...)
  }
 }
 if len(pools) == 0 {
  fmt.Println("You don't have any user pools!")
 } else {
  for _, pool := range pools {
   fmt.Printf("\t%v: %v\n", *pool.Name, *pool.Id)
  }
 }
}
```

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia AWS SDK for Go de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsResponse;
```

```
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListUserPools {
    public static void main(String[] args) {
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllUserPools(cognitoClient);
        cognitoClient.close();
    }

    public static void listAllUserPools(CognitoIdentityProviderClient
 cognitoClient) {
        try {
            ListUserPoolsRequest request = ListUserPoolsRequest.builder()
                    .maxResults(10)
                    .build();

            ListUserPoolsResponse response =
 cognitoClient.listUserPools(request);
            response.userPools().forEach(userpool -> {
                System.out.println("User pool " + userpool.name() + ", User ID "
 + userpool.id());
            });

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

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import {
  paginateListUserPools,
  CognitoIdentityProviderClient,
} from "@aws-sdk/client-cognito-identity-provider";

const client = new CognitoIdentityProviderClient({});

export const helloCognito = async () => {
  const paginator = paginateListUserPools({ client }, {});

  const userPoolNames = [];

  for await (const page of paginator) {
    const names = page.UserPools.map((pool) => pool.Name);
    userPoolNames.push(...names);
  }

  console.log("User pool names: ");
  console.log(userPoolNames.join("\n"));
  return userPoolNames;
};
```

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia AWS SDK for JavaScript de la API.

Ejemplos de código

- Acciones para el proveedor de identidad de Amazon Cognito mediante SDK AWS
  - Úselo AdminCreateUser con un AWS SDK o CLI
  - Úselo AdminGetUser con un AWS SDK o CLI
  - Úselo AdminInitiateAuth con un AWS SDK o CLI
  - Úselo AdminRespondToAuthChallenge con un AWS SDK o CLI
  - Úselo AdminSetUserPassword con un AWS SDK o CLI
  - Úselo AssociateSoftwareToken con un AWS SDK o CLI
  - Úselo ConfirmDevice con un AWS SDK o CLI
  - Úselo ConfirmForgotPassword con un AWS SDK o CLI
  - Úselo ConfirmSignUp con un AWS SDK o CLI
  - Úselo CreateUserPool con un AWS SDK o CLI
  - Úselo CreateUserPoolClient con un AWS SDK o CLI
  - Úselo DeleteUser con un AWS SDK o CLI
  - Úselo ForgotPassword con un AWS SDK o CLI
  - Úselo InitiateAuth con un AWS SDK o CLI
  - Úselo ListUserPools con un AWS SDK o CLI
  - Úselo ListUsers con un AWS SDK o CLI
  - Úselo ResendConfirmationCode con un AWS SDK o CLI
  - Úselo RespondToAuthChallenge con un AWS SDK o CLI
  - Úselo SignUp con un AWS SDK o CLI
  - Úselo UpdateUserPool con un AWS SDK o CLI
  - Úselo VerifySoftwareToken con un AWS SDK o CLI
- Escenarios para el uso de SDK por parte del proveedor de identidad de Amazon Cognito AWS
  - Confirme automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un SDK AWS
  - Migre automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un SDK AWS
  - Registrar un usuario con un grupo de usuarios de Amazon Cognito que requiera MFA mediante un SDK AWS

- Escriba datos de actividad personalizados con una función Lambda tras la autenticación de usuarios de Amazon Cognito mediante un SDK AWS

Acciones para el proveedor de identidad de Amazon Cognito mediante SDK AWS

Los siguientes ejemplos de código muestran cómo realizar acciones individuales del proveedor de identidad de Amazon Cognito con AWS los SDK. Estos fragmentos llaman a la API de proveedor de identidades de Amazon Cognito y son fragmentos de código de programas más grandes que se deben ejecutar en contexto. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para consultar la lista completa, consulte Amazon Cognito Identity Provider API Reference (Referencia de la API de Amazon Cognito Identity Provider).

Ejemplos

- Úselo AdminCreateUser con un AWS SDK o CLI
- Úselo AdminGetUser con un AWS SDK o CLI
- Úselo AdminInitiateAuth con un AWS SDK o CLI
- Úselo AdminRespondToAuthChallenge con un AWS SDK o CLI
- Úselo AdminSetUserPassword con un AWS SDK o CLI
- Úselo AssociateSoftwareToken con un AWS SDK o CLI
- Úselo ConfirmDevice con un AWS SDK o CLI
- Úselo ConfirmForgotPassword con un AWS SDK o CLI
- Úselo ConfirmSignUp con un AWS SDK o CLI
- Úselo CreateUserPool con un AWS SDK o CLI
- Úselo CreateUserPoolClient con un AWS SDK o CLI
- Úselo DeleteUser con un AWS SDK o CLI
- Úselo ForgotPassword con un AWS SDK o CLI
- Úselo InitiateAuth con un AWS SDK o CLI
- Úselo ListUserPools con un AWS SDK o CLI

- Úselo ListUsers con un AWS SDK o CLI
- Úselo ResendConfirmationCode con un AWS SDK o CLI
- Úselo RespondToAuthChallenge con un AWS SDK o CLI
- Úselo SignUp con un AWS SDK o CLI
- Úselo UpdateUserPool con un AWS SDK o CLI
- Úselo VerifySoftwareToken con un AWS SDK o CLI

Úselo **`AdminCreateUser`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `AdminCreateUser`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Escriba datos de actividad personalizados con una función de Lambda tras la autenticación de usuario de Amazon Cognito

CLI

AWS CLI

Para crear un usuario

En el siguiente `admin-create-user` ejemplo, se crea un usuario con la dirección de correo electrónico y el número de teléfono especificados.

```
aws cognito-idp admin-create-user \
    --user-pool-id us-west-2_aaaaaaaaa \
    --username diego \
    --user-attributes Name=email,Value=diego@example.com
 Name=phone_number,Value="+15555551212" \
    --message-action SUPPRESS
```

Salida:

```
{
    "User": {
        "Username": "diego",
```

```
        "Attributes": [
            {
                "Name": "sub",
                "Value": "7325c1de-b05b-4f84-b321-9adc6e61f4a2"
            },
            {
                "Name": "phone_number",
                "Value": "+15555551212"
            },
            {
                "Name": "email",
                "Value": "diego@example.com"
            }
        ],
        "UserCreateDate": 1548099495.428,
        "UserLastModifiedDate": 1548099495.428,
        "Enabled": true,
        "UserStatus": "FORCE_CHANGE_PASSWORD"
    }
}
```

- Para obtener más información sobre la API, consulte AdminCreateUserla Referencia de AWS CLI comandos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}
```

```
// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
 userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
 })
 if err != nil {
  var userExists *types.UsernameExistsException
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
 }
 return err
}
```

- Para obtener más información sobre la API, consulta AdminCreateUserla Referencia AWS SDK for Go de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`AdminGetUser`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `AdminGetUser`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// Get the specified user from an Amazon Cognito user pool with
administrator access.
   /// </summary>
   /// <param name="userName">The name of the user.</param>
   /// <param name="poolId">The Id of the Amazon Cognito user pool.</param>
   /// <returns>Async task.</returns>
   public async Task<UserStatusType> GetAdminUserAsync(string userName, string
poolId)
   {
       AdminGetUserRequest userRequest = new AdminGetUserRequest
       {
           Username = userName,
           UserPoolId = poolId,
       };

       var response = await _cognitoService.AdminGetUserAsync(userRequest);

       Console.WriteLine($"User status {response.UserStatus}");
       return response.UserStatus;
   }
```

- Para obtener más información sobre la API, consulta AdminGetUserla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

    Aws::CognitoIdentityProvider::Model::AdminGetUserRequest request;
    request.SetUsername(userName);
    request.SetUserPoolId(userPoolID);

    Aws::CognitoIdentityProvider::Model::AdminGetUserOutcome outcome =
            client.AdminGetUser(request);

    if (outcome.IsSuccess()) {
        std::cout << "The status for " << userName << " is " <<

Aws::CognitoIdentityProvider::Model::UserStatusTypeMapper::GetNameForUserStatusType(
                          outcome.GetResult().GetUserStatus()) << std::endl;
        std::cout << "Enabled is " << outcome.GetResult().GetEnabled() <<
std::endl;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminGetUser. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
```

- Para obtener más información sobre la API, consulta AdminGetUserla Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para obtener un usuario

En este ejemplo se obtiene información sobre el nombre de usuario jane@example.com.

Comando:

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --username
 jane@example.com
```

Salida:

```
{
  "Username": "4320de44-2322-4620-999b-5e2e1c8df013",
  "Enabled": true,
  "UserStatus": "FORCE_CHANGE_PASSWORD",
  "UserCreateDate": 1548108509.537,
  "UserAttributes": [
      {
          "Name": "sub",
          "Value": "4320de44-2322-4620-999b-5e2e1c8df013"
      },
      {
          "Name": "email_verified",
          "Value": "true"
      },
      {
          "Name": "phone_number_verified",
          "Value": "true"
      },
      {
          "Name": "phone_number",
          "Value": "+01115551212"
      },
      {
          "Name": "email",
          "Value": "jane@example.com"
      }
  ],
  "UserLastModifiedDate": 1548108509.537
```

```
}
```

- Para obtener más información sobre la API, consulta AdminGetUserla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void getAdminUser(CognitoIdentityProviderClient
identityProviderClient, String userName,
           String poolId) {
       try {
           AdminGetUserRequest userRequest = AdminGetUserRequest.builder()
                   .username(userName)
                   .userPoolId(poolId)
                   .build();

           AdminGetUserResponse response =
identityProviderClient.adminGetUser(userRequest);
           System.out.println("User status " + response.userStatusAsString());

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

- Para obtener más información sobre la API, consulta AdminGetUserla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const adminGetUser = ({ userPoolId, username }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminGetUserCommand({
    UserPoolId: userPoolId,
    Username: username,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta AdminGetUserla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun getAdminUser(userNameVal: String?, poolIdVal: String?) {
    val userRequest = AdminGetUserRequest {
        username = userNameVal
        userPoolId = poolIdVal
    }
```

```
    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminGetUser(userRequest)
        println("User status ${response.userStatus}")
    }
}
```

- Para obtener más información sobre la API, consulta AdminGetUserla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret
```

```
   def sign_up_user(self, user_name, password, user_email):
       """
       Signs up a new user with Amazon Cognito. This action prompts Amazon
Cognito
       to send an email to the specified email address. The email contains a
code that
       can be used to confirm the user.

       When the user already exists, the user status is checked to determine
whether
       the user has been confirmed.

       :param user_name: The user name that identifies the new user.
       :param password: The password for the new user.
       :param user_email: The email address for the new user.
       :return: True when the user is already confirmed with Amazon Cognito.
                Otherwise, false.
       """
       try:
           kwargs = {
               "ClientId": self.client_id,
               "Username": user_name,
               "Password": password,
               "UserAttributes": [{"Name": "email", "Value": user_email}],
           }
           if self.client_secret is not None:
               kwargs["SecretHash"] = self._secret_hash(user_name)
           response = self.cognito_idp_client.sign_up(**kwargs)
           confirmed = response["UserConfirmed"]
       except ClientError as err:
           if err.response["Error"]["Code"] == "UsernameExistsException":
               response = self.cognito_idp_client.admin_get_user(
                   UserPoolId=self.user_pool_id, Username=user_name
               )
               logger.warning(
                   "User %s exists and is %s.", user_name,
response["UserStatus"]
               )
               confirmed = response["UserStatus"] == "CONFIRMED"
           else:
               logger.error(
                   "Couldn't sign up %s. Here's why: %s: %s",
                   user_name,
                   err.response["Error"]["Code"],
```

```
                err.response["Error"]["Message"],
            )
            raise
    return confirmed
```

- Para obtener más información sobre la API, consulta AdminGetUserla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`AdminInitiateAuth`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `AdminInitiateAuth`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/// <summary>
/// Initiate an admin auth request.
/// </summary>
/// <param name="clientId">The client ID to use.</param>
/// <param name="userPoolId">The ID of the user pool.</param>
/// <param name="userName">The username to authenticate.</param>
```

```
    /// <param name="password">The user's password.</param>
    /// <returns>The session to use in challenge-response.</returns>
    public async Task<string> AdminInitiateAuthAsync(string clientId, string
userPoolId, string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var request = new AdminInitiateAuthRequest
        {
            ClientId = clientId,
            UserPoolId = userPoolId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.AdminInitiateAuthAsync(request);
        return response.Session;
    }
```

- Para obtener más información sobre la API, consulta AdminInitiateAuthla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);
```

```
    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthRequest request;
    request.SetClientId(clientID);
    request.SetUserPoolId(userPoolID);
    request.AddAuthParameters("USERNAME", userName);
    request.AddAuthParameters("PASSWORD", password);
    request.SetAuthFlow(

Aws::CognitoIdentityProvider::Model::AuthFlowType::ADMIN_USER_PASSWORD_AUTH);


    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthOutcome outcome =
            client.AdminInitiateAuth(request);

    if (outcome.IsSuccess()) {
        std::cout << "Call to AdminInitiateAuth was successful." << std::endl;
        sessionResult = outcome.GetResult().GetSession();
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminInitiateAuth. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
```

- Para obtener más información sobre la API, consulta AdminInitiateAuthla Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para iniciar la autorización

En este ejemplo, se inicia la autorización mediante el flujo ADMIN_NO_SRP_AUTH para el nombre de usuario jane@example.com

El cliente debe tener habilitada la API de inicio de sesión para la autenticación basada en servidor (ADMIN_NO_SRP_AUTH).

Usa la información de la sesión en el valor devuelto para llamar a admin-respond-to-auth -challenge.

Comando:

```
aws cognito-idp admin-initiate-auth --user-pool-id us-west-2_aaaaaaaaa --client-
id 3n4b5urk1ft4fl3mg5e62d9ado --auth-flow ADMIN_NO_SRP_AUTH --auth-parameters
 USERNAME=jane@example.com,PASSWORD=password
```

Salida:

```
{
  "ChallengeName": "NEW_PASSWORD_REQUIRED",
  "Session": "SESSION",
  "ChallengeParameters": {
      "USER_ID_FOR_SRP": "84514837-dcbc-4af1-abff-f3c109334894",
      "requiredAttributes": "[]",
      "userAttributes": "{\"email_verified\":\"true\",\"phone_number_verified\":
\"true\",\"phone_number\":\"+01xxx5550100\",\"email\":\"jane@example.com\"}"
  }
}
```

- Para obtener más información sobre la API, consulte AdminInitiateAuthla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static AdminInitiateAuthResponse
initiateAuth(CognitoIdentityProviderClient identityProviderClient,
           String clientId, String userName, String password, String userPoolId)
{
       try {
           Map<String, String> authParameters = new HashMap<>();
           authParameters.put("USERNAME", userName);
           authParameters.put("PASSWORD", password);
```

```
            AdminInitiateAuthRequest authRequest =
 AdminInitiateAuthRequest.builder()
                    .clientId(clientId)
                    .userPoolId(userPoolId)
                    .authParameters(authParameters)
                    .authFlow(AuthFlowType.ADMIN_USER_PASSWORD_AUTH)
                    .build();

            AdminInitiateAuthResponse response =
 identityProviderClient.adminInitiateAuth(authRequest);
            System.out.println("Result Challenge is : " +
 response.challengeName());
            return response;

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

        return null;
    }
```

- Para obtener más información sobre la API, consulta AdminInitiateAuthla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const adminInitiateAuth = ({ clientId, userPoolId, username, password }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminInitiateAuthCommand({
    ClientId: clientId,
```

```
    UserPoolId: userPoolId,
    AuthFlow: AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
    AuthParameters: { USERNAME: username, PASSWORD: password },
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta AdminInitiateAuthla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun checkAuthMethod(clientIdVal: String, userNameVal: String,
 passwordVal: String, userPoolIdVal: String): AdminInitiateAuthResponse {
    val authParas = mutableMapOf<String, String>()
    authParas["USERNAME"] = userNameVal
    authParas["PASSWORD"] = passwordVal

    val authRequest = AdminInitiateAuthRequest {
        clientId = clientIdVal
        userPoolId = userPoolIdVal
        authParameters = authParas
        authFlow = AuthFlowType.AdminUserPasswordAuth
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminInitiateAuth(authRequest)
        println("Result Challenge is ${response.challengeName}")
        return response
    }
```

```
}
```

- Para obtener más información sobre la API, consulta AdminInitiateAuthla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def start_sign_in(self, user_name, password):
        """
        Starts the sign-in process for a user by using administrator credentials.
        This method of signing in is appropriate for code running on a secure
 server.
```

```
        If the user pool is configured to require MFA and this is the first sign-
in
        for the user, Amazon Cognito returns a challenge response to set up an
        MFA application. When this occurs, this function gets an MFA secret from
        Amazon Cognito and returns it to the caller.

        :param user_name: The name of the user to sign in.
        :param password: The user's password.
        :return: The result of the sign-in attempt. When sign-in is successful,
 this
                 returns an access token that can be used to get AWS credentials.
 Otherwise,
                 Amazon Cognito returns a challenge to set up an MFA application,
                 or a challenge to enter an MFA code from a registered MFA
 application.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
                "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
                "AuthParameters": {"USERNAME": user_name, "PASSWORD": password},
            }
            if self.client_secret is not None:
                kwargs["AuthParameters"]["SECRET_HASH"] =
 self._secret_hash(user_name)
            response = self.cognito_idp_client.admin_initiate_auth(**kwargs)
            challenge_name = response.get("ChallengeName", None)
            if challenge_name == "MFA_SETUP":
                if (
                    "SOFTWARE_TOKEN_MFA"
                    in response["ChallengeParameters"]["MFAS_CAN_SETUP"]
                ):
                    response.update(self.get_mfa_secret(response["Session"]))
                else:
                    raise RuntimeError(
                        "The user pool requires MFA setup, but the user pool is
 not "
                        "configured for TOTP MFA. This example requires TOTP
 MFA."
                    )
        except ClientError as err:
            logger.error(
                "Couldn't start sign in for %s. Here's why: %s: %s",
```

```
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response
```

- Para obtener más información sobre la API, consulta AdminInitiateAuthla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`AdminRespondToAuthChallenge`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `AdminRespondToAuthChallenge`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/// <summary>
/// Respond to an admin authentication challenge.
```

```
    /// </summary>
    /// <param name="userName">The name of the user.</param>
    /// <param name="clientId">The client ID.</param>
    /// <param name="mfaCode">The multi-factor authentication code.</param>
    /// <param name="session">The current application session.</param>
    /// <param name="clientId">The user pool ID.</param>
    /// <returns>The result of the authentication response.</returns>
    public async Task<AuthenticationResultType> AdminRespondToAuthChallengeAsync(
        string userName,
        string clientId,
        string mfaCode,
        string session,
        string userPoolId)
    {
        Console.WriteLine("SOFTWARE_TOKEN_MFA challenge is generated");

        var challengeResponses = new Dictionary<string, string>();
        challengeResponses.Add("USERNAME", userName);
        challengeResponses.Add("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        var respondToAuthChallengeRequest = new
AdminRespondToAuthChallengeRequest
        {
            ChallengeName = ChallengeNameType.SOFTWARE_TOKEN_MFA,
            ClientId = clientId,
            ChallengeResponses = challengeResponses,
            Session = session,
            UserPoolId = userPoolId,
        };

        var response = await
_cognitoService.AdminRespondToAuthChallengeAsync(respondToAuthChallengeRequest);
        Console.WriteLine($"Response to Authentication
{response.AuthenticationResult.TokenType}");
        return response.AuthenticationResult;
    }
```

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region (overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

       Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeRequest
request;
       request.AddChallengeResponses("USERNAME", userName);
       request.AddChallengeResponses("SOFTWARE_TOKEN_MFA_CODE", mfaCode);
       request.SetChallengeName(

Aws::CognitoIdentityProvider::Model::ChallengeNameType::SOFTWARE_TOKEN_MFA);
       request.SetClientId(clientID);
       request.SetUserPoolId(userPoolID);
       request.SetSession(session);

       Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeOutcome
outcome =
               client.AdminRespondToAuthChallenge(request);

       if (outcome.IsSuccess()) {
           std::cout << "Here is the response to the challenge.\n" <<

outcome.GetResult().GetAuthenticationResult().Jsonize().View().WriteReadable()
                     << std::endl;

           accessToken =
outcome.GetResult().GetAuthenticationResult().GetAccessToken();
       }
       else {
```

```
            std::cerr << "Error with
CognitoIdentityProvider::AdminRespondToAuthChallenge. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela Referencia AWS SDK for C++ de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   // Respond to an authentication challenge.
   public static void adminRespondToAuthChallenge(CognitoIdentityProviderClient
identityProviderClient,
            String userName, String clientId, String mfaCode, String session) {
        System.out.println("SOFTWARE_TOKEN_MFA challenge is generated");
        Map<String, String> challengeResponses = new HashMap<>();

        challengeResponses.put("USERNAME", userName);
        challengeResponses.put("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        AdminRespondToAuthChallengeRequest respondToAuthChallengeRequest =
AdminRespondToAuthChallengeRequest.builder()
                .challengeName(ChallengeNameType.SOFTWARE_TOKEN_MFA)
                .clientId(clientId)
                .challengeResponses(challengeResponses)
                .session(session)
                .build();

        AdminRespondToAuthChallengeResponse respondToAuthChallengeResult =
identityProviderClient
```

```
                .adminRespondToAuthChallenge(respondToAuthChallengeRequest);

 System.out.println("respondToAuthChallengeResult.getAuthenticationResult()"
                + respondToAuthChallengeResult.authenticationResult());
    }
```

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const adminRespondToAuthChallenge = ({
  userPoolId,
  clientId,
  username,
  totp,
  session,
}) => {
  const client = new CognitoIdentityProviderClient({});
  const command = new AdminRespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: totp,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Respond to an authentication challenge.
suspend fun adminRespondToAuthChallenge(userName: String, clientIdVal: String?,
 mfaCode: String, sessionVal: String?) {
    println("SOFTWARE_TOKEN_MFA challenge is generated")
    val challengeResponsesOb = mutableMapOf<String, String>()
    challengeResponsesOb["USERNAME"] = userName
    challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

    val adminRespondToAuthChallengeRequest = AdminRespondToAuthChallengeRequest {
        challengeName = ChallengeNameType.SoftwareTokenMfa
        clientId = clientIdVal
        challengeResponses = challengeResponsesOb
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val respondToAuthChallengeResult =
 identityProviderClient.adminRespondToAuthChallenge(adminRespondToAuthChallengeRequest)
        println("respondToAuthChallengeResult.getAuthenticationResult()
 ${respondToAuthChallengeResult.authenticationResult}")
    }
}
```

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Para responder a un desafío de MFA, proporcione un código generado por una aplicación MFA asociada.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def respond_to_mfa_challenge(self, user_name, session, mfa_code):
        """
        Responds to a challenge for an MFA code. This completes the second step
 of
        a two-factor sign-in. When sign-in is successful, it returns an access
 token
```

```
        that can be used to get AWS credentials from Amazon Cognito.

        :param user_name: The name of the user who is signing in.
        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :param mfa_code: A code generated by the associated MFA application.
        :return: The result of the authentication. When successful, this contains
an
                 access token for the user.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
                "ChallengeName": "SOFTWARE_TOKEN_MFA",
                "Session": session,
                "ChallengeResponses": {
                    "USERNAME": user_name,
                    "SOFTWARE_TOKEN_MFA_CODE": mfa_code,
                },
            }
            if self.client_secret is not None:
                kwargs["ChallengeResponses"]["SECRET_HASH"] = self._secret_hash(
                    user_name
                )
            response =
self.cognito_idp_client.admin_respond_to_auth_challenge(**kwargs)
            auth_result = response["AuthenticationResult"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "ExpiredCodeException":
                logger.warning(
                    "Your MFA code has expired or has been used already. You
might have "
                    "to wait a few seconds until your app shows you a new code."
                )
            else:
                logger.error(
                    "Couldn't respond to mfa challenge for %s. Here's why: %s:
%s",
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
```

```
                raise
        else:
            return auth_result
```

- Para obtener más información sobre la API, consulta AdminRespondToAuthChallengela AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`AdminSetUserPassword`** con un AWS SDK o CLI

En el siguiente ejemplo de código, se muestra cómo usar `AdminSetUserPassword`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Escriba datos de actividad personalizados con una función de Lambda tras la autenticación de usuario de Amazon Cognito

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}
```

```
// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
 string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
  }
 }
 return err
}
```

- Para obtener más información sobre la API, consulta AdminSetUserPasswordla Referencia AWS SDK for Go de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`AssociateSoftwareToken`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `AssociateSoftwareToken`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Get an MFA token to authenticate the user with the authenticator.
    /// </summary>
    /// <param name="session">The session name.</param>
    /// <returns>The session name.</returns>
    public async Task<string> AssociateSoftwareTokenAsync(string session)
    {
        var softwareTokenRequest = new AssociateSoftwareTokenRequest
        {
            Session = session,
        };

        var tokenResponse = await
_cognitoService.AssociateSoftwareTokenAsync(softwareTokenRequest);
        var secretCode = tokenResponse.SecretCode;

        Console.WriteLine($"Use the following secret code to set up the
authenticator: {secretCode}");

        return tokenResponse.Session;
    }
```

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
 client(clientConfig);

        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenRequest
 request;
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenOutcome
 outcome =
                client.AssociateSoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout
                    << "Enter this setup key into an authenticator app, for
 example Google Authenticator."
                    << std::endl;
            std::cout << "Setup key: " << outcome.GetResult().GetSecretCode()
                      << std::endl;
#ifdef USING_QR
            printAsterisksLine();
            std::cout << "\nOr scan the QR code in the file '" << QR_CODE_PATH <<
 "."
                      << std::endl;

            saveQRCode(std::string("otpauth://totp/") + userName + "?secret=" +
                       outcome.GetResult().GetSecretCode());
#endif // USING_QR
            session = outcome.GetResult().GetSession();
```

```
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::AssociateSoftwareToken. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla Referencia AWS SDK for C++ de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static String getSecretForAppMFA(CognitoIdentityProviderClient
identityProviderClient, String session) {
        AssociateSoftwareTokenRequest softwareTokenRequest =
AssociateSoftwareTokenRequest.builder()
                .session(session)
                .build();

        AssociateSoftwareTokenResponse tokenResponse = identityProviderClient
                .associateSoftwareToken(softwareTokenRequest);
        String secretCode = tokenResponse.secretCode();
        System.out.println("Enter this token into Google Authenticator");
        System.out.println(secretCode);
        return tokenResponse.session();
   }
```

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const associateSoftwareToken = (session) => {
  const client = new CognitoIdentityProviderClient({});
  const command = new AssociateSoftwareTokenCommand({
    Session: session,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun getSecretForAppMFA(sessionVal: String?): String? {
    val softwareTokenRequest = AssociateSoftwareTokenRequest {
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
```

```
        val tokenResponse =
 identityProviderClient.associateSoftwareToken(softwareTokenRequest)
        val secretCode = tokenResponse.secretCode
        println("Enter this token into Google Authenticator")
        println(secretCode)
        return tokenResponse.session
    }
}
```

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret
```

```
    def get_mfa_secret(self, session):
        """
        Gets a token that can be used to associate an MFA application with the
user.

        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :return: An MFA token that can be used to set up an MFA application.
        """
        try:
            response =
self.cognito_idp_client.associate_software_token(Session=session)
        except ClientError as err:
            logger.error(
                "Couldn't get MFA secret. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response
```

- Para obtener más información sobre la API, consulta AssociateSoftwareTokenla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ConfirmDevice`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ConfirmDevice`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Initiates and confirms tracking of the device.
    /// </summary>
    /// <param name="accessToken">The user's access token.</param>
    /// <param name="deviceKey">The key of the device from Amazon Cognito.</
param>
    /// <param name="deviceName">The device name.</param>
    /// <returns></returns>
    public async Task<bool> ConfirmDeviceAsync(string accessToken, string
 deviceKey, string deviceName)
    {
        var request = new ConfirmDeviceRequest
        {
            AccessToken = accessToken,
            DeviceKey = deviceKey,
            DeviceName = deviceName
        };

        var response = await _cognitoService.ConfirmDeviceAsync(request);
        return response.UserConfirmationNecessary;
    }
```

- Para obtener más información sobre la API, consulta ConfirmDevicela Referencia AWS SDK for .NET de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const confirmDevice = ({ deviceKey, accessToken, passwordVerifier, salt }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmDeviceCommand({
    DeviceKey: deviceKey,
    AccessToken: accessToken,
    DeviceSecretVerifierConfig: {
      PasswordVerifier: passwordVerifier,
      Salt: salt,
    },
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta ConfirmDevicela Referencia AWS SDK for JavaScript de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
```

```
   """Encapsulates Amazon Cognito actions"""

   def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
       """
       :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
       :param user_pool_id: The ID of an existing Amazon Cognito user pool.
       :param client_id: The ID of a client application registered with the user
pool.
       :param client_secret: The client secret, if the client has a secret.
       """
       self.cognito_idp_client = cognito_idp_client
       self.user_pool_id = user_pool_id
       self.client_id = client_id
       self.client_secret = client_secret


   def confirm_mfa_device(
       self,
       user_name,
       device_key,
       device_group_key,
       device_password,
       access_token,
       aws_srp,
   ):
       """
       Confirms an MFA device to be tracked by Amazon Cognito. When a device is
       tracked, its key and password can be used to sign in without requiring a
new
       MFA code from the MFA application.

       :param user_name: The user that is associated with the device.
       :param device_key: The key of the device, returned by Amazon Cognito.
       :param device_group_key: The group key of the device, returned by Amazon
Cognito.
       :param device_password: The password that is associated with the device.
       :param access_token: The user's access token.
       :param aws_srp: A class that helps with Secure Remote Password (SRP)
                       calculations. The scenario associated with this example
uses
                       the warrant package.
```

```
        :return: True when the user must confirm the device. Otherwise, False.
When
                 False, the device is automatically confirmed and tracked.
        """
        srp_helper = aws_srp.AWSSRP(
            username=user_name,
            password=device_password,
            pool_id="_",
            client_id=self.client_id,
            client_secret=None,
            client=self.cognito_idp_client,
        )
        device_and_pw = f"{device_group_key}{device_key}:{device_password}"
        device_and_pw_hash = aws_srp.hash_sha256(device_and_pw.encode("utf-8"))
        salt = aws_srp.pad_hex(aws_srp.get_random(16))
        x_value = aws_srp.hex_to_long(aws_srp.hex_hash(salt +
device_and_pw_hash))
        verifier = aws_srp.pad_hex(pow(srp_helper.val_g, x_value,
srp_helper.big_n))
        device_secret_verifier_config = {
            "PasswordVerifier": base64.standard_b64encode(
                bytearray.fromhex(verifier)
            ).decode("utf-8"),
            "Salt":
base64.standard_b64encode(bytearray.fromhex(salt)).decode("utf-8"),
        }
        try:
            response = self.cognito_idp_client.confirm_device(
                AccessToken=access_token,
                DeviceKey=device_key,
                DeviceSecretVerifierConfig=device_secret_verifier_config,
            )
            user_confirm = response["UserConfirmationNecessary"]
        except ClientError as err:
            logger.error(
                "Couldn't confirm mfa device %s. Here's why: %s: %s",
                device_key,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return user_confirm
```

- Para obtener más información sobre la API, consulta ConfirmDevicela AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ConfirmForgotPassword`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ConfirmForgotPassword`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Migración en forma automática los usuarios conocidos con una función de Lambda

CLI

AWS CLI

Para confirmar una contraseña olvidada

En este ejemplo se confirma una contraseña olvidada para el nombre de usuario diego@example.com.

Comando:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- Para obtener más información sobre la API, consulte ConfirmForgotPasswordla Referencia de AWS CLI comandos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
 userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}
```

- Para obtener más información sobre la API, consulta ConfirmForgotPasswordla Referencia AWS SDK for Go de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ConfirmSignUp`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ConfirmSignUp`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// Confirm that the user has signed up.
   /// </summary>
   /// <param name="clientId">The Id of this application.</param>
   /// <param name="code">The confirmation code sent to the user.</param>
   /// <param name="userName">The username.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> ConfirmSignupAsync(string clientId, string code,
string userName)
   {
       var signUpRequest = new ConfirmSignUpRequest
       {
           ClientId = clientId,
           ConfirmationCode = code,
```

```
            Username = userName,
        };

        var response = await _cognitoService.ConfirmSignUpAsync(signUpRequest);
        if (response.HttpStatusCode == HttpStatusCode.OK)
        {
            Console.WriteLine($"{userName} was confirmed");
            return true;
        }
        return false;
    }
```

- Para obtener más información sobre la API, consulta ConfirmSignUpla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::ConfirmSignUpRequest request;
        request.SetClientId(clientID);
        request.SetConfirmationCode(confirmationCode);
        request.SetUsername(userName);

        Aws::CognitoIdentityProvider::Model::ConfirmSignUpOutcome outcome =
                client.ConfirmSignUp(request);
```

```
        if (outcome.IsSuccess()) {
            std::cout << "ConfirmSignup was Successful."
                      << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::ConfirmSignUp. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obtener más información sobre la API, consulta ConfirmSignUpla Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para confirmar la inscripción

Este ejemplo confirma el registro del nombre de usuario diego@example.com.

Comando:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --confirmation-code CONF_CODE
```

- Para obtener más información sobre la API, consulta ConfirmSignUpla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    public static void confirmSignUp(CognitoIdentityProviderClient
 identityProviderClient, String clientId, String code,
            String userName) {
        try {
            ConfirmSignUpRequest signUpRequest = ConfirmSignUpRequest.builder()
                    .clientId(clientId)
                    .confirmationCode(code)
                    .username(userName)
                    .build();

            identityProviderClient.confirmSignUp(signUpRequest);
            System.out.println(userName + " was confirmed");

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

- Para obtener más información sobre la API, consulta ConfirmSignUpla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const confirmSignUp = ({ clientId, username, code }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmSignUpCommand({
    ClientId: clientId,
    Username: username,
    ConfirmationCode: code,
  });
```

```
  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta ConfirmSignUpla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun confirmSignUp(clientIdVal: String?, codeVal: String?, userNameVal:
 String?) {
    val signUpRequest = ConfirmSignUpRequest {
        clientId = clientIdVal
        confirmationCode = codeVal
        username = userNameVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.confirmSignUp(signUpRequest)
        println("$userNameVal  was confirmed")
    }
}
```

- Para obtener más información sobre la API, consulta ConfirmSignUpla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def confirm_user_sign_up(self, user_name, confirmation_code):
        """
        Confirms a previously created user. A user must be confirmed before they
        can sign in to Amazon Cognito.

        :param user_name: The name of the user to confirm.
        :param confirmation_code: The confirmation code sent to the user's
 registered
                                  email address.
        :return: True when the confirmation succeeds.
        """
        try:
            kwargs = {
```

```
                "ClientId": self.client_id,
                "Username": user_name,
                "ConfirmationCode": confirmation_code,
            }
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            self.cognito_idp_client.confirm_sign_up(**kwargs)
        except ClientError as err:
            logger.error(
                "Couldn't confirm sign up for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return True
```

- Para obtener más información sobre la API, consulta ConfirmSignUpla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`CreateUserPool`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `CreateUserPool`.

CLI

AWS CLI

Para crear un grupo de usuarios con una configuración mínima

En este ejemplo, se crea un grupo de usuarios denominado MyUserPool con los valores predeterminados. No se requieren atributos ni clientes de aplicación. La MFA y la seguridad avanzada están deshabilitadas.

Comando:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Salida:

```
{
  "UserPool": {
      "SchemaAttributes": [
          {
              "Name": "sub",
              "StringAttributeConstraints": {
                  "MinLength": "1",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": true,
              "AttributeDataType": "String",
              "Mutable": false
          },
          {
              "Name": "name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "given_name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "family_name",
              "StringAttributeConstraints": {
```

```
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "middle_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "nickname",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "preferred_username",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
```

```
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "email",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
```

```
                "Required": false,
                "Name": "email_verified",
                "Mutable": true
            },
            {
                "Name": "gender",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "birthdate",
                "StringAttributeConstraints": {
                    "MinLength": "10",
                    "MaxLength": "10"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "zoneinfo",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "locale",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
```

```
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "updated_at",
            "NumberAttributeConstraints": {
                "MinValue": "0"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "Number",
            "Mutable": true
        }
    ],
```

```
      "MfaConfiguration": "OFF",
      "Name": "MyUserPool",
      "LastModifiedDate": 1547833345.777,
      "AdminCreateUserConfig": {
          "UnusedAccountValidityDays": 7,
          "AllowAdminCreateUserOnly": false
      },
      "EmailConfiguration": {},
      "Policies": {
          "PasswordPolicy": {
              "RequireLowercase": true,
              "RequireSymbols": true,
              "RequireNumbers": true,
              "MinimumLength": 8,
              "RequireUppercase": true
          }
      },
      "CreationDate": 1547833345.777,
      "EstimatedNumberOfUsers": 0,
      "Id": "us-west-2_aaaaaaaaa",
      "LambdaConfig": {}
  }
}
```

Creación de un grupo de usuarios con dos atributos obligatorios

En este ejemplo se crea un grupo de usuarios MyUserPool. El grupo está configurado para aceptar un correo electrónico como atributo de nombre de usuario. También establece la dirección de origen del correo electrónico en una dirección validada mediante Amazon Simple Email Service.

Comando:

```
aws cognito-idp create-user-pool --pool-name MyUserPool --username-
attributes "email" --email-configuration=SourceArn="arn:aws:ses:us-
east-1:111111111111:identity/
jane@example.com",ReplyToEmailAddress="jane@example.com"
```

Salida:

```
{
  "UserPool": {
      "SchemaAttributes": [
```

```
            {
                "Name": "sub",
                "StringAttributeConstraints": {
                    "MinLength": "1",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": true,
                "AttributeDataType": "String",
                "Mutable": false
            },
            {
                "Name": "name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "given_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "family_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
```

```
            {
                "Name": "middle_name",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "nickname",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "preferred_username",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "profile",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
```

```
            {
                "Name": "picture",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "website",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "Name": "email",
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                },
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "AttributeDataType": "String",
                "Mutable": true
            },
            {
                "AttributeDataType": "Boolean",
                "DeveloperOnlyAttribute": false,
                "Required": false,
                "Name": "email_verified",
                "Mutable": true
            },
            {
                "Name": "gender",
                "StringAttributeConstraints": {
                    "MinLength": "0",
```

```
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
```

```
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "updated_at",
            "NumberAttributeConstraints": {
                "MinValue": "0"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "Number",
            "Mutable": true
        }
    ],
    "MfaConfiguration": "OFF",
    "Name": "MyUserPool",
    "LastModifiedDate": 1547837788.189,
    "AdminCreateUserConfig": {
        "UnusedAccountValidityDays": 7,
        "AllowAdminCreateUserOnly": false
    },
    "EmailConfiguration": {
```

```
            "ReplyToEmailAddress": "jane@example.com",
            "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
        },
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "UsernameAttributes": [
            "email"
        ],
        "CreationDate": 1547837788.189,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

- Para obtener más información sobre la API, consulte CreateUserPoolla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
```

```
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolResponse;

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

        final String usage = """

                Usage:
                    <userPoolName>\s

                Where:
                    userPoolName - The name to give your user pool when it's
 created.
                """;

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

        String userPoolName = args[0];
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String id = createPool(cognitoClient, userPoolName);
        System.out.println("User pool ID: " + id);
        cognitoClient.close();
    }
```

```
    public static String createPool(CognitoIdentityProviderClient cognitoClient,
 String userPoolName) {
        try {
            CreateUserPoolRequest request = CreateUserPoolRequest.builder()
                    .poolName(userPoolName)
                    .build();

            CreateUserPoolResponse response =
 cognitoClient.createUserPool(request);
            return response.userPool().id();

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

- Para obtener más información sobre la API, consulta CreateUserPoolla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`CreateUserPoolClient`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `CreateUserPoolClient`.

CLI

AWS CLI

Para crear un cliente de grupo de usuarios

En este ejemplo, se crea un nuevo cliente de grupo de usuarios con dos flujos de autorización explícitos: USER_PASSWORD_AUTH y ADMIN_NO_SRP_AUTH.

Comando:

```
aws cognito-idp create-user-pool-client --user-pool-id us-west-2_aaaaaaaaa
  --client-name MyNewClient --no-generate-secret --explicit-auth-flows
 "USER_PASSWORD_AUTH" "ADMIN_NO_SRP_AUTH"
```

Salida:

```
{
  "UserPoolClient": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientName": "MyNewClient",
      "ClientId": "6p3bs000no6a4ue1idruvd05ad",
      "LastModifiedDate": 1548697449.497,
      "CreationDate": 1548697449.497,
      "RefreshTokenValidity": 30,
      "ExplicitAuthFlows": [
          "USER_PASSWORD_AUTH",
          "ADMIN_NO_SRP_AUTH"
      ],
      "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

- Para obtener CreateUserPoolClientmás AWS CLI información sobre la API, consulte la Referencia de comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
```

```
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientReques
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientRespon

/**
 * A user pool client app is an application that authenticates with Amazon
 * Cognito user pools.
 * When you create a user pool, you can configure app clients that allow mobile
 * or web applications
 * to call API operations to authenticate users, manage user attributes and
 * profiles,
 * and implement sign-up and sign-in flows.
 *
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateUserPoolClient {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <clientName> <userPoolId>\s

                Where:
                    clientName - The name for the user pool client to create.
                    userPoolId - The ID for the user pool.
                """;

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

        String clientName = args[0];
        String userPoolId = args[1];
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();
```

```
        createPoolClient(cognitoClient, clientName, userPoolId);
        cognitoClient.close();
    }

    public static void createPoolClient(CognitoIdentityProviderClient
 cognitoClient, String clientName,
            String userPoolId) {
        try {
            CreateUserPoolClientRequest request =
 CreateUserPoolClientRequest.builder()
                    .clientName(clientName)
                    .userPoolId(userPoolId)
                    .build();

            CreateUserPoolClientResponse response =
 cognitoClient.createUserPoolClient(request);
            System.out.println("User pool " +
 response.userPoolClient().clientName() + " created. ID: "
                    + response.userPoolClient().clientId());

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

- Para obtener más información sobre la API, consulta CreateUserPoolClientla Referencia AWS SDK for Java 2.x de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DeleteUser`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DeleteUser`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Confirmación de manera automática a los usuarios conocidos con una función de Lambda
- Migración en forma automática los usuarios conocidos con una función de Lambda
- Escriba datos de actividad personalizados con una función de Lambda tras la autenticación de usuario de Amazon Cognito

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::DeleteUserRequest request;
        request.SetAccessToken(accessToken);

        Aws::CognitoIdentityProvider::Model::DeleteUserOutcome outcome =
                client.DeleteUser(request);

        if (outcome.IsSuccess()) {
            std::cout << "The user " << userName << " was deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::DeleteUser. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
```

- Para obtener más información sobre la API, consulta DeleteUserla Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para eliminar un usuario

En este ejemplo se elimina un usuario.

Comando:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- Para obtener más información sobre la API, consulta DeleteUserla Referencia de AWS CLI comandos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
 _, err := actor.CognitoClient.DeleteUser(context.TODO(),
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
```

```
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}
```

- Para obtener más información sobre la API, consulta DeleteUserla Referencia AWS SDK for Go de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ForgotPassword`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ForgotPassword`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Migración en forma automática los usuarios conocidos con una función de Lambda

CLI

AWS CLI

Para forzar un cambio de contraseña

En el siguiente `forgot-password` ejemplo, se envía un mensaje a jane@example.com para cambiar su contraseña.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpqsnet7mpld0 --username
 jane@example.com
```

Salida:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
```

```
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

- Para obtener más información sobre la API, consulte ForgotPasswordla Referencia de AWS CLI comandos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
 (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
}
```

- Para obtener más información sobre la API, consulta ForgotPasswordla Referencia AWS SDK for Go de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`InitiateAuth`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `InitiateAuth`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en los siguientes ejemplos de código:

- Confirmación de manera automática a los usuarios conocidos con una función de Lambda
- Migración en forma automática los usuarios conocidos con una función de Lambda
- Registro de un usuario en un grupo de usuarios que requiera MFA
- Escriba datos de actividad personalizados con una función de Lambda tras la autenticación de usuario de Amazon Cognito

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Initiate authorization.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The name of the user who is authenticating.</
param>
```

```
    /// <param name="password">The password for the user who is authenticating.</
param>
    /// <returns>The response from the initiate auth request.</returns>
    public async Task<InitiateAuthResponse> InitiateAuthAsync(string clientId,
 string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var authRequest = new InitiateAuthRequest

        {
            ClientId = clientId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.InitiateAuthAsync(authRequest);
        Console.WriteLine($"Result Challenge is : {response.ChallengeName}");

        return response;
    }
```

- Para obtener más información sobre la API, consulta InitiateAuthla Referencia AWS SDK for .NET de la API.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
```

```
}



// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(clientId string, userName string, password
 string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
 &cognitoidentityprovider.InitiateAuthInput{
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
   log.Println(*resetRequired.Message)
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
}
```

- Para obtener más información sobre la API, consulta InitiateAuthla Referencia AWS SDK for Go de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const initiateAuth = ({ username, password, clientId }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new InitiateAuthCommand({
    AuthFlow: AuthFlowType.USER_PASSWORD_AUTH,
    AuthParameters: {
      USERNAME: username,
      PASSWORD: password,
    },
    ClientId: clientId,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta InitiateAuthla Referencia AWS SDK for JavaScript de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

En este ejemplo, se muestra cómo iniciar la autenticación con un dispositivo del que se hace seguimiento. Para completar el inicio de sesión, el cliente debe responder correctamente a los desafíos relacionados con la contraseña remota segura (SRP).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
```

```
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def sign_in_with_tracked_device(
        self,
        user_name,
        password,
        device_key,
        device_group_key,
        device_password,
        aws_srp,
    ):
        """
        Signs in to Amazon Cognito as a user who has a tracked device. Signing in
        with a tracked device lets a user sign in without entering a new MFA
 code.

        Signing in with a tracked device requires that the client respond to the
 SRP
        protocol. The scenario associated with this example uses the warrant
 package
        to help with SRP calculations.

        For more information on SRP, see https://en.wikipedia.org/wiki/
Secure_Remote_Password_protocol.

        :param user_name: The user that is associated with the device.
        :param password: The user's password.
        :param device_key: The key of a tracked device.
        :param device_group_key: The group key of a tracked device.
        :param device_password: The password that is associated with the device.
        :param aws_srp: A class that helps with SRP calculations. The scenario
                        associated with this example uses the warrant package.
        :return: The result of the authentication. When successful, this contains
 an
                 access token for the user.
```

```
        """
        try:
            srp_helper = aws_srp.AWSSRP(
                username=user_name,
                password=device_password,
                pool_id="_",
                client_id=self.client_id,
                client_secret=None,
                client=self.cognito_idp_client,
            )

            response_init = self.cognito_idp_client.initiate_auth(
                ClientId=self.client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": user_name,
                    "PASSWORD": password,
                    "DEVICE_KEY": device_key,
                },
            )
            if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
                raise RuntimeError(
                    f"Expected DEVICE_SRP_AUTH challenge but got
{response_init['ChallengeName']}."
                )

            auth_params = srp_helper.get_auth_params()
            auth_params["DEVICE_KEY"] = device_key
            response_auth = self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_SRP_AUTH",
                ChallengeResponses=auth_params,
            )
            if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
                raise RuntimeError(
                    f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
                    f"{response_init['ChallengeName']}."
                )

            challenge_params = response_auth["ChallengeParameters"]
            challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
            cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
            cr["USERNAME"] = user_name
```

```
            cr["DEVICE_KEY"] = device_key
            response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_PASSWORD_VERIFIER",
                ChallengeResponses=cr,
            )
            auth_tokens = response_verifier["AuthenticationResult"]
        except ClientError as err:
            logger.error(
                "Couldn't start client sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return auth_tokens
```

- Para obtener más información sobre la API, consulta InitiateAuthla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListUserPools`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ListUserPools`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// List the Amazon Cognito user pools for an account.
    /// </summary>
    /// <returns>A list of UserPoolDescriptionType objects.</returns>
    public async Task<List<UserPoolDescriptionType>> ListUserPoolsAsync()
    {
        var userPools = new List<UserPoolDescriptionType>();

        var userPoolsPaginator = _cognitoService.Paginators.ListUserPools(new
 ListUserPoolsRequest());

        await foreach (var response in userPoolsPaginator.Responses)
        {
            userPools.AddRange(response.UserPools);
        }

        return userPools;
    }
```

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para mostrar los grupos de usuarios

En este ejemplo se muestran hasta 20 grupos de usuarios.

Comando:

```
aws cognito-idp list-user-pools --max-results 20
```

Salida:

```
{
  "UserPools": [
      {
         "CreationDate": 1547763720.822,
```

```
            "LastModifiedDate": 1547763720.822,
            "LambdaConfig": {},
            "Id": "us-west-2_aaaaaaaaa",
            "Name": "MyUserPool"
        }
    ]
}
```

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia de AWS CLI comandos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
package main

import (
 "context"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Notification
 Service
// (Amazon SNS) client and list the topics in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
```

```
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 cognitoClient := cognitoidentityprovider.NewFromConfig(sdkConfig)
 fmt.Println("Let's list the user pools for your account.")
 var pools []types.UserPoolDescriptionType
 paginator := cognitoidentityprovider.NewListUserPoolsPaginator(
  cognitoClient, &cognitoidentityprovider.ListUserPoolsInput{MaxResults:
 aws.Int32(10)})
 for paginator.HasMorePages() {
  output, err := paginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get user pools. Here's why: %v\n", err)
  } else {
   pools = append(pools, output.UserPools...)
  }
 }
 if len(pools) == 0 {
  fmt.Println("You don't have any user pools!")
 } else {
  for _, pool := range pools {
   fmt.Printf("\t%v: %v\n", *pool.Name, *pool.Id)
  }
 }
}
```

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia AWS SDK for Go de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUserPoolsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListUserPools {
    public static void main(String[] args) {
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllUserPools(cognitoClient);
        cognitoClient.close();
    }

    public static void listAllUserPools(CognitoIdentityProviderClient
 cognitoClient) {
```

```
        try {
            ListUserPoolsRequest request = ListUserPoolsRequest.builder()
                    .maxResults(10)
                    .build();

            ListUserPoolsResponse response =
 cognitoClient.listUserPools(request);
            response.userPools().forEach(userpool -> {
                System.out.println("User pool " + userpool.name() + ", User ID "
 + userpool.id());
            });

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

- Para obtener más información sobre la API, consulta ListUserPoolsla Referencia AWS SDK for Java 2.x de la API.

Rust

SDK para Rust

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
async fn show_pools(client: &Client) -> Result<(), Error> {
    let response = client.list_user_pools().max_results(10).send().await?;
    let pools = response.user_pools();
    println!("User pools:");
    for pool in pools {
        println!("  ID:              {}", pool.id().unwrap_or_default());
        println!("  Name:            {}", pool.name().unwrap_or_default());
        println!("  Lambda Config:   {:?}", pool.lambda_config().unwrap());
```

```
        println!(
            "  Last modified:   {}",
            pool.last_modified_date().unwrap().to_chrono_utc()?
        );
        println!(
            "  Creation date:   {:?}",
            pool.creation_date().unwrap().to_chrono_utc()
        );
        println!();
    }
    println!("Next token: {}", response.next_token().unwrap_or_default());

    Ok(())
}
```

- Para obtener más información sobre la API, consulta ListUserPoolsla referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListUsers`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ListUsers`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Get a list of users for the Amazon Cognito user pool.
    /// </summary>
    /// <param name="userPoolId">The user pool ID.</param>
    /// <returns>A list of users.</returns>
    public async Task<List<UserType>> ListUsersAsync(string userPoolId)
    {
        var request = new ListUsersRequest
        {
            UserPoolId = userPoolId
        };

        var users = new List<UserType>();

        var usersPaginator = _cognitoService.Paginators.ListUsers(request);
        await foreach (var response in usersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
    }
```

- Para obtener más información sobre la API, consulta ListUsersla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para mostrar los usuarios

En este ejemplo se muestran hasta 20 usuarios.

Comando:

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

Salida:

```
{
  "Users": [
      {
          "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
          "Enabled": true,
          "UserStatus": "FORCE_CHANGE_PASSWORD",
          "UserCreateDate": 1548089817.683,
          "UserLastModifiedDate": 1548089817.683,
          "Attributes": [
              {
                  "Name": "sub",
                  "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
              },
              {
                  "Name": "email_verified",
                  "Value": "true"
              },
              {
                  "Name": "email",
                  "Value": "mary@example.com"
              }
          ]
      }
  ]
}
```

- Para obtener más información sobre la API, consulta ListUsersla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

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

```
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUsersRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ListUsersResponse;

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

        final String usage = """

                Usage:
                    <userPoolId>\s

                Where:
                    userPoolId - The ID given to your user pool when it's
 created.
                """;

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

        String userPoolId = args[0];
        CognitoIdentityProviderClient cognitoClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllUsers(cognitoClient, userPoolId);
        listUsersFilter(cognitoClient, userPoolId);
```

```
        cognitoClient.close();
    }

    public static void listAllUsers(CognitoIdentityProviderClient cognitoClient,
String userPoolId) {
        try {
            ListUsersRequest usersRequest = ListUsersRequest.builder()
                    .userPoolId(userPoolId)
                    .build();

            ListUsersResponse response = cognitoClient.listUsers(usersRequest);
            response.users().forEach(user -> {
                System.out.println("User " + user.username() + " Status " +
user.userStatus() + " Created "
                        + user.userCreateDate());
            });

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

    // Shows how to list users by using a filter.
    public static void listUsersFilter(CognitoIdentityProviderClient
cognitoClient, String userPoolId) {

        try {
            String filter = "email = \"tblue@noserver.com\"";
            ListUsersRequest usersRequest = ListUsersRequest.builder()
                    .userPoolId(userPoolId)
                    .filter(filter)
                    .build();

            ListUsersResponse response = cognitoClient.listUsers(usersRequest);
            response.users().forEach(user -> {
                System.out.println("User with filter applied " + user.username()
+ " Status " + user.userStatus()
                        + " Created " + user.userCreateDate());
            });

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

```
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListUsersla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const listUsers = ({ userPoolId }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ListUsersCommand({
    UserPoolId: userPoolId,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta ListUsersla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun listAllUsers(userPoolId: String) {

    val request = ListUsersRequest {
        this.userPoolId = userPoolId
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use { cognitoClient ->
        val response = cognitoClient.listUsers(request)
        response.users?.forEach { user ->
            println("The user name is ${user.username}")
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListUsersla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
```

```
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def list_users(self):
        """
        Returns a list of the users in the current user pool.

        :return: The list of users.
        """
        try:
            response =
self.cognito_idp_client.list_users(UserPoolId=self.user_pool_id)
            users = response["Users"]
        except ClientError as err:
            logger.error(
                "Couldn't list users for %s. Here's why: %s: %s",
                self.user_pool_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return users
```

- Para obtener más información sobre la API, consulta [ListUsers](la AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ResendConfirmationCode`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ResendConfirmationCode`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Send a new confirmation code to a user.
    /// </summary>
    /// <param name="clientId">The Id of the client application.</param>
    /// <param name="userName">The username of user who will receive the code.</
param>
    /// <returns>The delivery details.</returns>
    public async Task<CodeDeliveryDetailsType> ResendConfirmationCodeAsync(string
 clientId, string userName)
    {
        var codeRequest = new ResendConfirmationCodeRequest
        {
            ClientId = clientId,
            Username = userName,
        };

        var response = await
 _cognitoService.ResendConfirmationCodeAsync(codeRequest);
```

```
        Console.WriteLine($"Method of delivery is
{response.CodeDeliveryDetails.DeliveryMedium}");

        return response.CodeDeliveryDetails;
    }
```

- Para obtener más información sobre la API, consulta ResendConfirmationCodela Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeRequest
request;
        request.SetUsername(userName);
        request.SetClientId(clientID);

        Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeOutcome
outcome =
                client.ResendConfirmationCode(request);

        if (outcome.IsSuccess()) {
            std::cout
```

```
                    << "CognitoIdentityProvider::ResendConfirmationCode was
 successful."
                    << std::endl;
        }
        else {
            std::cerr << "Error with
 CognitoIdentityProvider::ResendConfirmationCode. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obtener más información sobre la API, consulta ResendConfirmationCodela Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para reenviar un código de confirmación

En el siguiente ejemplo `resend-confirmation-code`, se envía un código de confirmación al usuario `jane`.

```
aws cognito-idp resend-confirmation-code \
    --client-id 12a3b456c7de890f11g123hijk \
    --username jane
```

Salida:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

Para obtener más información, consulte Inscripción y confirmación de cuentas de usuarios en la Guía para desarrolladores de Amazon Cognito.

- Para obtener más información sobre la API, consulta ResendConfirmationCodela Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void resendConfirmationCode(CognitoIdentityProviderClient
identityProviderClient, String clientId,
           String userName) {
       try {
           ResendConfirmationCodeRequest codeRequest =
ResendConfirmationCodeRequest.builder()
                   .clientId(clientId)
                   .username(userName)
                   .build();

           ResendConfirmationCodeResponse response =
identityProviderClient.resendConfirmationCode(codeRequest);
           System.out.println("Method of delivery is " +
response.codeDeliveryDetails().deliveryMediumAsString());

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

- Para obtener más información sobre la API, consulta ResendConfirmationCodela Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const resendConfirmationCode = ({ clientId, username }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ResendConfirmationCodeCommand({
    ClientId: clientId,
    Username: username,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta ResendConfirmationCodela Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun resendConfirmationCode(clientIdVal: String?, userNameVal: String?) {
    val codeRequest = ResendConfirmationCodeRequest {
        clientId = clientIdVal
        username = userNameVal
    }
```

```
    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.resendConfirmationCode(codeRequest)
        println("Method of delivery is " +
 (response.codeDeliveryDetails?.deliveryMedium))
    }
}
```

- Para obtener más información sobre la API, consulta ResendConfirmationCodela referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret
```

```
    def resend_confirmation(self, user_name):
        """
        Prompts Amazon Cognito to resend an email with a new confirmation code.

        :param user_name: The name of the user who will receive the email.
        :return: Delivery information about where the email is sent.
        """
        try:
            kwargs = {"ClientId": self.client_id, "Username": user_name}
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            response = self.cognito_idp_client.resend_confirmation_code(**kwargs)
            delivery = response["CodeDeliveryDetails"]
        except ClientError as err:
            logger.error(
                "Couldn't resend confirmation to %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return delivery
```

- Para obtener más información sobre la API, consulta ResendConfirmationCodela AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`RespondToAuthChallenge`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `RespondToAuthChallenge`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

CLI

AWS CLI

Para responder a un desafío de autorización

Este ejemplo responde a un desafío de autorización iniciado con initiate-auth. Es una respuesta al desafío NEW_PASSWORD_REQUIRED. Establece una contraseña para el usuario jane@example.com.

Comando:

```
aws cognito-idp respond-to-auth-challenge --client-id 3n4b5urk1ft4fl3mg5e62d9ado
 --challenge-name NEW_PASSWORD_REQUIRED --challenge-responses
 USERNAME=jane@example.com,NEW_PASSWORD="password" --session "SESSION_TOKEN"
```

Salida:

```
{
  "ChallengeParameters": {},
  "AuthenticationResult": {
      "AccessToken": "ACCESS_TOKEN",
      "ExpiresIn": 3600,
      "TokenType": "Bearer",
      "RefreshToken": "REFRESH_TOKEN",
      "IdToken": "ID_TOKEN",
      "NewDeviceMetadata": {
          "DeviceKey": "us-west-2_fec070d2-fa88-424a-8ec8-b26d7198eb23",
          "DeviceGroupKey": "-wt2ha1Zd"
      }
  }
}
```

- Para obtener más información sobre la API, consulte RespondToAuthChallengela Referencia de AWS CLI comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const respondToAuthChallenge = ({
  clientId,
  username,
  session,
  userPoolId,
  code,
}) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new RespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: code,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta RespondToAuthChallengela Referencia AWS SDK for JavaScript de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Inicie sesión con un dispositivo con seguimiento. Para completar el inicio de sesión, el cliente debe responder correctamente a los desafíos relacionados con la contraseña remota segura (SRP).

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def sign_in_with_tracked_device(
        self,
        user_name,
        password,
        device_key,
        device_group_key,
        device_password,
        aws_srp,
    ):
```

```
        """
        Signs in to Amazon Cognito as a user who has a tracked device. Signing in
        with a tracked device lets a user sign in without entering a new MFA
 code.

        Signing in with a tracked device requires that the client respond to the
 SRP
        protocol. The scenario associated with this example uses the warrant
 package
        to help with SRP calculations.

        For more information on SRP, see https://en.wikipedia.org/wiki/
Secure_Remote_Password_protocol.

        :param user_name: The user that is associated with the device.
        :param password: The user's password.
        :param device_key: The key of a tracked device.
        :param device_group_key: The group key of a tracked device.
        :param device_password: The password that is associated with the device.
        :param aws_srp: A class that helps with SRP calculations. The scenario
                        associated with this example uses the warrant package.
        :return: The result of the authentication. When successful, this contains
 an
                 access token for the user.
        """
        try:
            srp_helper = aws_srp.AWSSRP(
                username=user_name,
                password=device_password,
                pool_id="_",
                client_id=self.client_id,
                client_secret=None,
                client=self.cognito_idp_client,
            )

            response_init = self.cognito_idp_client.initiate_auth(
                ClientId=self.client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": user_name,
                    "PASSWORD": password,
                    "DEVICE_KEY": device_key,
                },
            )
```

```
            if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
                raise RuntimeError(
                    f"Expected DEVICE_SRP_AUTH challenge but got
{response_init['ChallengeName']}."
                )

            auth_params = srp_helper.get_auth_params()
            auth_params["DEVICE_KEY"] = device_key
            response_auth = self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_SRP_AUTH",
                ChallengeResponses=auth_params,
            )
            if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
                raise RuntimeError(
                    f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
                    f"{response_init['ChallengeName']}."
                )

            challenge_params = response_auth["ChallengeParameters"]
            challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
            cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
            cr["USERNAME"] = user_name
            cr["DEVICE_KEY"] = device_key
            response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_PASSWORD_VERIFIER",
                ChallengeResponses=cr,
            )
            auth_tokens = response_verifier["AuthenticationResult"]
        except ClientError as err:
            logger.error(
                "Couldn't start client sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return auth_tokens
```

- Para obtener más información sobre la API, consulta RespondToAuthChallengela AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`SignUp`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `SignUp`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en los siguientes ejemplos de código:

- Confirmación de manera automática a los usuarios conocidos con una función de Lambda
- Migración en forma automática los usuarios conocidos con una función de Lambda
- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Sign up a new user.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The username to use.</param>
    /// <param name="password">The user's password.</param>
    /// <param name="email">The email address of the user.</param>
    /// <returns>A Boolean value indicating whether the user was confirmed.</
returns>
```

```
    public async Task<bool> SignUpAsync(string clientId, string userName, string
password, string email)
    {
        var userAttrs = new AttributeType
        {
            Name = "email",
            Value = email,
        };

        var userAttrsList = new List<AttributeType>();

        userAttrsList.Add(userAttrs);

        var signUpRequest = new SignUpRequest
        {
            UserAttributes = userAttrsList,
            Username = userName,
            ClientId = clientId,
            Password = password
        };

        var response = await _cognitoService.SignUpAsync(signUpRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obtener más información sobre la API, consulta SignUpla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
```

```
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::SignUpRequest request;
        request.AddUserAttributes(
                Aws::CognitoIdentityProvider::Model::AttributeType().WithName(
                        "email").WithValue(email));
        request.SetUsername(userName);
        request.SetPassword(password);
        request.SetClientId(clientID);
        Aws::CognitoIdentityProvider::Model::SignUpOutcome outcome =
                client.SignUp(request);

        if (outcome.IsSuccess()) {
            std::cout << "The signup request for " << userName << " was
successful."
                      << std::endl;
        }
        else if (outcome.GetError().GetErrorType() ==

Aws::CognitoIdentityProvider::CognitoIdentityProviderErrors::USERNAME_EXISTS) {
            std::cout
                    << "The username already exists. Please enter a different
username."
                    << std::endl;
            userExists = true;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::SignUpRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obtener más información sobre la API, consulta SignUpla Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para inscribir a un usuario

En este ejemplo, se registra jane@example.com.

Comando:

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username jane@example.com --password PASSWORD --user-attributes
 Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

Salida:

```
{
  "UserConfirmed": false,
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- Para obtener más información sobre la API, consulta SignUpla Referencia de AWS CLI comandos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}
```

```
// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(clientId string, userName string, password
 string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(context.TODO(),
 &cognitoidentityprovider.SignUpInput{
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}
```

- Para obtener más información sobre la API, consulta SignUpla Referencia AWS SDK for Go de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   public static void signUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String userName,
           String password, String email) {
       AttributeType userAttrs = AttributeType.builder()
               .name("email")
               .value(email)
               .build();

       List<AttributeType> userAttrsList = new ArrayList<>();
       userAttrsList.add(userAttrs);
       try {
           SignUpRequest signUpRequest = SignUpRequest.builder()
                   .userAttributes(userAttrsList)
                   .username(userName)
                   .clientId(clientId)
                   .password(password)
                   .build();

           identityProviderClient.signUp(signUpRequest);
           System.out.println("User has been signed up ");

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

- Para obtener más información sobre la API, consulta SignUpla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const signUp = ({ clientId, username, password, email }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new SignUpCommand({
    ClientId: clientId,
    Username: username,
    Password: password,
    UserAttributes: [{ Name: "email", Value: email }],
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta SignUpla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun signUp(clientIdVal: String?, userNameVal: String?, passwordVal:
 String?, emailVal: String?) {
    val userAttrs = AttributeType {
        name = "email"
        value = emailVal
    }

    val userAttrsList = mutableListOf<AttributeType>()
    userAttrsList.add(userAttrs)
    val signUpRequest = SignUpRequest {
        userAttributes = userAttrsList
        username = userNameVal
        clientId = clientIdVal
        password = passwordVal
```

```
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.signUp(signUpRequest)
        println("User has been signed up")
    }
}
```

- Para obtener más información sobre la API, consulta SignUpla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
 client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
 pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret
```

```
   def sign_up_user(self, user_name, password, user_email):
       """
       Signs up a new user with Amazon Cognito. This action prompts Amazon
Cognito
       to send an email to the specified email address. The email contains a
code that
       can be used to confirm the user.

       When the user already exists, the user status is checked to determine
whether
       the user has been confirmed.

       :param user_name: The user name that identifies the new user.
       :param password: The password for the new user.
       :param user_email: The email address for the new user.
       :return: True when the user is already confirmed with Amazon Cognito.
                Otherwise, false.
       """
       try:
           kwargs = {
               "ClientId": self.client_id,
               "Username": user_name,
               "Password": password,
               "UserAttributes": [{"Name": "email", "Value": user_email}],
           }
           if self.client_secret is not None:
               kwargs["SecretHash"] = self._secret_hash(user_name)
           response = self.cognito_idp_client.sign_up(**kwargs)
           confirmed = response["UserConfirmed"]
       except ClientError as err:
           if err.response["Error"]["Code"] == "UsernameExistsException":
               response = self.cognito_idp_client.admin_get_user(
                   UserPoolId=self.user_pool_id, Username=user_name
               )
               logger.warning(
                   "User %s exists and is %s.", user_name,
response["UserStatus"]
               )
               confirmed = response["UserStatus"] == "CONFIRMED"
           else:
               logger.error(
                   "Couldn't sign up %s. Here's why: %s: %s",
                   user_name,
```

```
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
    return confirmed
```

- Para obtener más información sobre la API, consulta SignUpla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`UpdateUserPool`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `UpdateUserPool`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en los siguientes ejemplos de código:

- Confirmación de manera automática a los usuarios conocidos con una función de Lambda
- Migración en forma automática los usuarios conocidos con una función de Lambda
- Escriba datos de actividad personalizados con una función de Lambda tras la autenticación de usuario de Amazon Cognito

CLI

AWS CLI

Para actualizar un grupo de usuarios

En este ejemplo, se agregan etiquetas a un grupo de usuarios.

Comando:

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_aaaaaaaaa --user-pool-
tags Team=Blue,Area=West
```

- Para obtener más información sobre la API, consulte UpdateUserPoolla Referencia de AWS CLI comandos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int

const (
 PreSignUp Trigger = iota
 UserMigration
 PostAuthentication
)

type TriggerInfo struct {
 Trigger    Trigger
 HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
 triggers ...TriggerInfo) error {
```

```
 output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
 }
 return err
}
```

- Para obtener más información sobre la API, consulta UpdateUserPoolla Referencia AWS SDK for Go de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`VerifySoftwareToken`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `VerifySoftwareToken`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Registro de un usuario en un grupo de usuarios que requiera MFA

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// Verify the TOTP and register for MFA.
   /// </summary>
   /// <param name="session">The name of the session.</param>
   /// <param name="code">The MFA code.</param>
   /// <returns>The status of the software token.</returns>
   public async Task<VerifySoftwareTokenResponseType>
VerifySoftwareTokenAsync(string session, string code)
   {
       var tokenRequest = new VerifySoftwareTokenRequest
       {
           UserCode = code,
           Session = session,
       };

       var verifyResponse = await
_cognitoService.VerifySoftwareTokenAsync(tokenRequest);

       return verifyResponse.Status;
   }
```

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);

        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenRequest request;
        request.SetUserCode(userCode);
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenOutcome outcome =
                client.VerifySoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout << "Verification of the code was successful."
                      << std::endl;
            session = outcome.GetResult().GetSession();
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::VerifySoftwareToken. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla Referencia AWS SDK for C++ de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   // Verify the TOTP and register for MFA.
   public static void verifyTOTP(CognitoIdentityProviderClient
identityProviderClient, String session, String code) {
       try {
           VerifySoftwareTokenRequest tokenRequest =
VerifySoftwareTokenRequest.builder()
                   .userCode(code)
                   .session(session)
                   .build();

           VerifySoftwareTokenResponse verifyResponse =
identityProviderClient.verifySoftwareToken(tokenRequest);
           System.out.println("The status of the token is " +
verifyResponse.statusAsString());

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

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
const verifySoftwareToken = (totp) => {
  const client = new CognitoIdentityProviderClient({});

  // The 'Session' is provided in the response to 'AssociateSoftwareToken'.
  const session = process.env.SESSION;

  if (!session) {
    throw new Error(
      "Missing a valid Session. Did you run 'admin-initiate-auth'?",
    );
  }

  const command = new VerifySoftwareTokenCommand({
    Session: session,
    UserCode: totp,
  });

  return client.send(command);
};
```

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Verify the TOTP and register for MFA.
suspend fun verifyTOTP(sessionVal: String?, codeVal: String?) {
    val tokenRequest = VerifySoftwareTokenRequest {
        userCode = codeVal
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val verifyResponse =
 identityProviderClient.verifySoftwareToken(tokenRequest)
        println("The status of the token is ${verifyResponse.status}")
    }
}
```

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
class CognitoIdentityProviderWrapper:
```

```
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
client_secret=None):
        """
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def verify_mfa(self, session, user_code):
        """
        Verify a new MFA application that is associated with a user.

        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :param user_code: A code generated by the associated MFA application.
        :return: Status that indicates whether the MFA application is verified.
        """
        try:
            response = self.cognito_idp_client.verify_software_token(
                Session=session, UserCode=user_code
            )
        except ClientError as err:
            logger.error(
                "Couldn't verify MFA. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response
```

- Para obtener más información sobre la API, consulta VerifySoftwareTokenla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escenarios para el uso de SDK por parte del proveedor de identidad de Amazon Cognito AWS

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes en Amazon Cognito Identity Provider con AWS SDK. Estos escenarios muestran cómo llevar a cabo tareas específicas llamando a varias funciones dentro de Amazon Cognito Identity Provider. Cada escenario incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código.

Ejemplos

- Confirme automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un SDK AWS
- Migre automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un SDK AWS
- Registrar un usuario con un grupo de usuarios de Amazon Cognito que requiera MFA mediante un SDK AWS
- Escriba datos de actividad personalizados con una función Lambda tras la autenticación de usuarios de Amazon Cognito mediante un SDK AWS

Confirme automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un SDK AWS

En el siguiente ejemplo de código, se muestra cómo confirmar de manera automática los usuarios conocidos de Amazon Cognito con una función de Lambda.

- Configure un grupo de usuarios para que llame a una función de Lambda para el desencadenador `PreSignUp`.

- Inscripción de un usuario mediante Amazon Cognito
- La función de Lambda escanea una tabla de DynamoDB y confirma de manera automática los usuarios conocidos.
- Inicie sesión con el nuevo usuario y, luego, elimine los recursos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecutar un escenario interactivo en un símbolo del sistema.

```
// AutoConfirm separates the steps of this scenario into individual functions so
 that
// they are simpler to read and understand.
type AutoConfirm struct {
 helper       IScenarioHelper
 questioner   demotools.IQuestioner
 resources    Resources
 cognitoActor *actions.CognitoActions
}

// NewAutoConfirm constructs a new auto confirm runner.
func NewAutoConfirm(sdkConfig aws.Config, questioner demotools.IQuestioner,
 helper IScenarioHelper) AutoConfirm {
 scenario := AutoConfirm{
  helper:       helper,
  questioner:   questioner,
  resources:    Resources{},
  cognitoActor: &actions.CognitoActions{CognitoClient:
 cognitoidentityprovider.NewFromConfig(sdkConfig)},
 }
 scenario.resources.init(scenario.cognitoActor, questioner)
 return scenario
}
```

```
// AddPreSignUpTrigger adds a Lambda handler as an invocation target for the
 PreSignUp trigger.
func (runner *AutoConfirm) AddPreSignUpTrigger(userPoolId string, functionArn
 string) {
 log.Printf("Let's add a Lambda function to handle the PreSignUp trigger from
 Cognito.\n" +
  "This trigger happens when a user signs up, and lets your function take action
 before the main Cognito\n" +
  "sign up processing occurs.\n")
 err := runner.cognitoActor.UpdateTriggers(
  userPoolId,
  actions.TriggerInfo{Trigger: actions.PreSignUp, HandlerArn:
 aws.String(functionArn)})
 if err != nil {
  panic(err)
 }
 log.Printf("Lambda function %v added to user pool %v to handle the PreSignUp
 trigger.\n",
  functionArn, userPoolId)
}

// SignUpUser signs up a user from the known user table with a password you
 specify.
func (runner *AutoConfirm) SignUpUser(clientId string, usersTable string)
 (string, string) {
 log.Println("Let's sign up a user to your Cognito user pool. When the user's
 email matches an email in the\n" +
  "DynamoDB known users table, it is automatically verified and the user is
 confirmed.")

 knownUsers, err := runner.helper.GetKnownUsers(usersTable)
 if err != nil {
  panic(err)
 }
 userChoice := runner.questioner.AskChoice("Which user do you want to use?\n",
 knownUsers.UserNameList())
 user := knownUsers.Users[userChoice]

 var signedUp bool
 var userConfirmed bool
 password := runner.questioner.AskPassword("Enter a password that has at least
 eight characters, uppercase, lowercase, numbers and symbols.\n"+
  "(the password will not display as you type):", 8)
```

```
 for !signedUp {
  log.Printf("Signing up user '%v' with email '%v' to Cognito.\n", user.UserName,
 user.UserEmail)
  userConfirmed, err = runner.cognitoActor.SignUp(clientId, user.UserName,
 password, user.UserEmail)
  if err != nil {
   var invalidPassword *types.InvalidPasswordException
   if errors.As(err, &invalidPassword) {
    password = runner.questioner.AskPassword("Enter another password:", 8)
   } else {
    panic(err)
   }
  } else {
   signedUp = true
  }
 }
 log.Printf("User %v signed up, confirmed = %v.\n", user.UserName, userConfirmed)

 log.Println(strings.Repeat("-", 88))

 return user.UserName, password
}

// SignInUser signs in a user.
func (runner *AutoConfirm) SignInUser(clientId string, userName string, password
 string) string {
 runner.questioner.Ask("Press Enter when you're ready to continue.")
 log.Printf("Let's sign in as %v...\n", userName)
 authResult, err := runner.cognitoActor.SignIn(clientId, userName, password)
 if err != nil {
  panic(err)
 }
 log.Printf("Successfully signed in. Your access token starts with: %v...\n",
 (*authResult.AccessToken)[:10])
 log.Println(strings.Repeat("-", 88))
 return *authResult.AccessToken
}

// Run runs the scenario.
func (runner *AutoConfirm) Run(stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup()
```

```
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]
 runner.helper.PopulateUserTable(stackOutputs["TableName"])

 runner.AddPreSignUpTrigger(stackOutputs["UserPoolId"],
 stackOutputs["AutoConfirmFunctionArn"])
 runner.resources.triggers = append(runner.resources.triggers, actions.PreSignUp)
 userName, password := runner.SignUpUser(stackOutputs["UserPoolClientId"],
 stackOutputs["TableName"])
 runner.helper.ListRecentLogEvents(stackOutputs["AutoConfirmFunction"])
 runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
  runner.SignInUser(stackOutputs["UserPoolClientId"], userName, password))

 runner.resources.Cleanup()

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Controle el desencadenador `PreSignUp` con una función de Lambda.

```
const TABLE_NAME = "TABLE_NAME"

// UserInfo defines structured user data that can be marshalled to a DynamoDB
 format.
type UserInfo struct {
 UserName  string `dynamodbav:"UserName"`
 UserEmail string `dynamodbav:"UserEmail"`
```

```
}

// GetKey marshals the user email value to a DynamoDB key format.
func (user UserInfo) GetKey() map[string]dynamodbtypes.AttributeValue {
 userEmail, err := attributevalue.Marshal(user.UserEmail)
 if err != nil {
  panic(err)
 }
 return map[string]dynamodbtypes.AttributeValue{"UserEmail": userEmail}
}

type handler struct {
 dynamoClient *dynamodb.Client
}

// HandleRequest handles the PreSignUp event by looking up a user in an Amazon
 DynamoDB table and
// specifying whether they should be confirmed and verified.
func (h *handler) HandleRequest(ctx context.Context, event
 events.CognitoEventUserPoolsPreSignup) (events.CognitoEventUserPoolsPreSignup,
 error) {
 log.Printf("Received presignup from %v for user '%v'", event.TriggerSource,
 event.UserName)
 if event.TriggerSource != "PreSignUp_SignUp" {
  // Other trigger sources, such as PreSignUp_AdminInitiateAuth, ignore the
 response from this handler.
  return event, nil
 }
 tableName := os.Getenv(TABLE_NAME)
 user := UserInfo{
  UserEmail: event.Request.UserAttributes["email"],
 }
 log.Printf("Looking up email %v in table %v.\n", user.UserEmail, tableName)
 output, err := h.dynamoClient.GetItem(ctx, &dynamodb.GetItemInput{
  Key:       user.GetKey(),
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Error looking up email %v.\n", user.UserEmail)
  return event, err
 }
 if output.Item == nil {
  log.Printf("Email %v not found. Email verification is required.\n",
 user.UserEmail)
```

```
  return event, err
 }

 err = attributevalue.UnmarshalMap(output.Item, &user)
 if err != nil {
  log.Printf("Couldn't unmarshal DynamoDB item. Here's why: %v\n", err)
  return event, err
 }

 if user.UserName != event.UserName {
  log.Printf("UserEmail %v found, but stored UserName '%v' does not match
 supplied UserName '%v'. Verification is required.\n",
   user.UserEmail, user.UserName, event.UserName)
 } else {
  log.Printf("UserEmail %v found with matching UserName %v. User is confirmed.
\n", user.UserEmail, user.UserName)
  event.Response.AutoConfirmUser = true
  event.Response.AutoVerifyEmail = true
 }

 return event, err
}

func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  log.Panicln(err)
 }
 h := handler{
  dynamoClient: dynamodb.NewFromConfig(sdkConfig),
 }
 lambda.Start(h.HandleRequest)
}
```

Cree una estructura que lleve a cabo las tareas habituales.

```
// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
```

```
 GetStackOutputs(stackName string) (actions.StackOutputs, error)
 PopulateUserTable(tableName string)
 GetKnownUsers(tableName string) (actions.UserList, error)
 AddKnownUser(tableName string, user actions.User)
 ListRecentLogEvents(functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
 example.
type ScenarioHelper struct {
 questioner  demotools.IQuestioner
 dynamoActor *actions.DynamoActions
 cfnActor    *actions.CloudFormationActions
 cwlActor    *actions.CloudWatchLogsActions
 isTestRun   bool
}

// NewScenarioHelper constructs a new scenario helper.
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
 ScenarioHelper {
 scenario := ScenarioHelper{
  questioner:  questioner,
  dynamoActor: &actions.DynamoActions{DynamoClient:
 dynamodb.NewFromConfig(sdkConfig)},
  cfnActor:    &actions.CloudFormationActions{CfnClient:
 cloudformation.NewFromConfig(sdkConfig)},
  cwlActor:    &actions.CloudWatchLogsActions{CwlClient:
 cloudwatchlogs.NewFromConfig(sdkConfig)},
 }
 return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 if !helper.isTestRun {
  time.Sleep(time.Duration(secs) * time.Second)
 }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
 structured format.
func (helper ScenarioHelper) GetStackOutputs(stackName string)
 (actions.StackOutputs, error) {
 return helper.cfnActor.GetOutputs(stackName), nil
```

```
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(tableName string) {
 log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
 this example.\n", tableName)
 err := helper.dynamoActor.PopulateTable(tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(tableName string) (actions.UserList,
 error) {
 knownUsers, err := helper.dynamoActor.Scan(tableName)
 if err != nil {
  log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
 tableName, err)
 }
 return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(tableName string, user actions.User) {
 log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
 table...\n",
  user.UserName, user.UserEmail)
 err := helper.dynamoActor.AddUser(tableName, user)
 if err != nil {
  panic(err)
 }
}

// ListRecentLogEvents gets the most recent log stream and events for the
 specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(functionName string) {
 log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
 helper.Pause(10)
 log.Println("Okay, let's check the logs to find what's happened recently with
 your Lambda function.")
 logStream, err := helper.cwlActor.GetLatestLogStream(functionName)
 if err != nil {
```

```
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(functionName,
 *logStream.LogStreamName, 10)
 if err != nil {
  panic(err)
 }
 for _, event := range events {
  log.Printf("\t%v", *event.Message)
 }
 log.Println(strings.Repeat("-", 88))
}
```

Cree una estructura que ajuste las acciones de Amazon Cognito.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int

const (
 PreSignUp Trigger = iota
 UserMigration
 PostAuthentication
)

type TriggerInfo struct {
 Trigger    Trigger
 HandlerArn *string
}
```

```
// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
 triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
 }
 return err
}



// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(clientId string, userName string, password
 string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(context.TODO(),
 &cognitoidentityprovider.SignUpInput{
```

```
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}



// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(clientId string, userName string, password
 string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
 &cognitoidentityprovider.InitiateAuthInput{
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
   log.Println(*resetRequired.Message)
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
```

```
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
 (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
 userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}
```

```
// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
 _, err := actor.CognitoClient.DeleteUser(context.TODO(),
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}



// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
 userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
 })
 if err != nil {
  var userExists *types.UsernameExistsException
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
 }
 return err
}
```

```
// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
 string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
  }
 }
 return err
}
```

Cree una estructura que ajuste las acciones de DynamoDB.

```
// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
 actions
// used in the examples.
type DynamoActions struct {
 DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
 UserName  string
 UserEmail string
 LastLogin *LoginInfo `dynamodbav:",omitempty"`
}
```

```
// LoginInfo defines structured custom login data.
type LoginInfo struct {
 UserPoolId string
 ClientId   string
 Time       string
}

// UserList defines a list of users.
type UserList struct {
 Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
 strings.
func (users *UserList) UserNameList() []string {
 names := make([]string, len(users.Users))
 for i := 0; i < len(users.Users); i++ {
  names[i] = users.Users[i].UserName
 }
 return names
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(tableName string) error {
 var err error
 var item map[string]types.AttributeValue
 var writeReqs []types.WriteRequest
 for i := 1; i < 4; i++ {
  item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
  if err != nil {
   log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
 err)
   return err
  }
  writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
 &types.PutRequest{Item: item}})
 }
 _, err = actor.DynamoClient.BatchWriteItem(context.TODO(),
 &dynamodb.BatchWriteItemInput{
  RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
 })
 if err != nil {
```

```
  log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
 tableName, err)
 }
 return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(tableName string) (UserList, error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(context.TODO(), &dynamodb.ScanInput{
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
 err)
 } else {
  err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
  if err != nil {
   log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
  }
 }
 return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(tableName string, user User) error {
 userItem, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
 }
 _, err = actor.DynamoClient.PutItem(context.TODO(), &dynamodb.PutItemInput{
  Item:      userItem,
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
 }
 return err
}
```

Crea una estructura que agrupe las acciones de CloudWatch Logs.

```
type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(functionName string)
 (types.LogStream, error) {
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.DescribeLogStreams(context.TODO(),
 &cloudwatchlogs.DescribeLogStreamsInput{
  Descending:   aws.Bool(true),
  Limit:        aws.Int32(1),
  LogGroupName: aws.String(logGroupName),
  OrderBy:      types.OrderByLastEventTime,
 })
 if err != nil {
  log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
 logGroupName, err)
 } else {
  logStream = output.LogStreams[0]
 }
 return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
 stream.
func (actor CloudWatchLogsActions) GetLogEvents(functionName string,
 logStreamName string, eventCount int32) (
 []types.OutputLogEvent, error) {
 var events []types.OutputLogEvent
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.GetLogEvents(context.TODO(),
 &cloudwatchlogs.GetLogEventsInput{
  LogStreamName: aws.String(logStreamName),
  Limit:         aws.Int32(eventCount),
  LogGroupName:  aws.String(logGroupName),
 })
 if err != nil {
  log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
 logStreamName, err)
 } else {
```

```
  events = output.Events
 }
 return events, err
}
```

Crea una estructura que agrupe las acciones. AWS CloudFormation

```
// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
 CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
 structured format.
func (actor CloudFormationActions) GetOutputs(stackName string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(context.TODO(),
 &cloudformation.DescribeStacksInput{
  StackName: aws.String(stackName),
 })
 if err != nil || len(output.Stacks) == 0 {
  log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
 stackName, err)
 }
 stackOutputs := StackOutputs{}
 for _, out := range output.Stacks[0].Outputs {
  stackOutputs[*out.OutputKey] = *out.OutputValue
 }
 return stackOutputs
}
```

Eliminación de recursos.

```
// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {
```

```
 userPoolId       string
 userAccessTokens []string
 triggers         []actions.Trigger

 cognitoActor *actions.CognitoActions
 questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
 demotools.IQuestioner) {
 resources.userAccessTokens = []string{}
 resources.triggers = []actions.Trigger{}
 resources.cognitoActor = cognitoActor
 resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
func (resources *Resources) Cleanup() {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong during cleanup.\n%v\n", r)
   log.Println("Use the AWS Management Console to remove any remaining resources
 \n" +
    "that were created for this scenario.")
  }
 }()

 wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
 resources that were created "+
  "during this demo (y/n)?", "y")
 if wantDelete {
  for _, accessToken := range resources.userAccessTokens {
   err := resources.cognitoActor.DeleteUser(accessToken)
   if err != nil {
    log.Println("Couldn't delete user during cleanup.")
    panic(err)
   }
   log.Println("Deleted user.")
  }
  triggerList := make([]actions.TriggerInfo, len(resources.triggers))
  for i := 0; i < len(resources.triggers); i++ {
   triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
 HandlerArn: nil}
  }
```

```
  err := resources.cognitoActor.UpdateTriggers(resources.userPoolId,
 triggerList...)
  if err != nil {
   log.Println("Couldn't update Cognito triggers during cleanup.")
   panic(err)
  }
  log.Println("Removed Cognito triggers from user pool.")
 } else {
  log.Println("Be sure to remove resources when you're done with them to avoid
 unexpected charges!")
 }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK for Go .
  - DeleteUser
  - InitiateAuth
  - SignUp
  - UpdateUserPool

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Migre automáticamente a los usuarios conocidos de Amazon Cognito con una función Lambda mediante un SDK AWS

En el siguiente ejemplo de código, se muestra cómo migrar de manera automática los usuarios conocidos de Amazon Cognito con una función de Lambda.

- Configure un grupo de usuarios para que llame a una función de Lambda para el desencadenador `MigrateUser`.
- Inicie sesión en Amazon Cognito con un nombre de usuario y un correo electrónico que no estén en el grupo de usuarios.
- La función de Lambda escanea una tabla de DynamoDB y migra de manera automática los usuarios conocidos al grupo de usuarios.

- Realice el flujo en caso de olvido de contraseña para restablecer la contraseña respecto del usuario migrado.
- Inicie sesión como un nuevo usuario y, a continuación, elimine los recursos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecutar un escenario interactivo en un símbolo del sistema.

```
import (
 "errors"
 "fmt"
 "log"
 "strings"
 "user_pools_and_lambda_triggers/actions"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider"
 "github.com/aws/aws-sdk-go-v2/service/cognitoidentityprovider/types"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
)

// MigrateUser separates the steps of this scenario into individual functions so
 that
// they are simpler to read and understand.
type MigrateUser struct {
 helper       IScenarioHelper
 questioner   demotools.IQuestioner
 resources    Resources
 cognitoActor *actions.CognitoActions
}

// NewMigrateUser constructs a new migrate user runner.
```

```
func NewMigrateUser(sdkConfig aws.Config, questioner demotools.IQuestioner,
 helper IScenarioHelper) MigrateUser {
 scenario := MigrateUser{
  helper:       helper,
  questioner:   questioner,
  resources:    Resources{},
  cognitoActor: &actions.CognitoActions{CognitoClient:
 cognitoidentityprovider.NewFromConfig(sdkConfig)},
 }
 scenario.resources.init(scenario.cognitoActor, questioner)
 return scenario
}

// AddMigrateUserTrigger adds a Lambda handler as an invocation target for the
 MigrateUser trigger.
func (runner *MigrateUser) AddMigrateUserTrigger(userPoolId string, functionArn
 string) {
 log.Printf("Let's add a Lambda function to handle the MigrateUser trigger from
 Cognito.\n" +
  "This trigger happens when an unknown user signs in, and lets your function
 take action before Cognito\n" +
  "rejects the user.\n\n")
 err := runner.cognitoActor.UpdateTriggers(
  userPoolId,
  actions.TriggerInfo{Trigger: actions.UserMigration, HandlerArn:
 aws.String(functionArn)})
 if err != nil {
  panic(err)
 }
 log.Printf("Lambda function %v added to user pool %v to handle the MigrateUser
 trigger.\n",
  functionArn, userPoolId)

 log.Println(strings.Repeat("-", 88))
}

// SignInUser adds a new user to the known users table and signs that user in to
 Amazon Cognito.
func (runner *MigrateUser) SignInUser(usersTable string, clientId string) (bool,
 actions.User) {
 log.Println("Let's sign in a user to your Cognito user pool. When the username
 and email matches an entry in the\n" +
  "DynamoDB known users table, the email is automatically verified and the user
 is migrated to the Cognito user pool.")
```

```
 user := actions.User{}
 user.UserName = runner.questioner.Ask("\nEnter a username:")
 user.UserEmail = runner.questioner.Ask("\nEnter an email that you own. This
 email will be used to confirm user migration\n" +
  "during this example:")

 runner.helper.AddKnownUser(usersTable, user)

 var err error
 var resetRequired *types.PasswordResetRequiredException
 var authResult *types.AuthenticationResultType
 signedIn := false
 for !signedIn && resetRequired == nil {
  log.Printf("Signing in to Cognito as user '%v'. The expected result is a
 PasswordResetRequiredException.\n\n", user.UserName)
  authResult, err = runner.cognitoActor.SignIn(clientId, user.UserName, "_")
  if err != nil {
   if errors.As(err, &resetRequired) {
    log.Printf("\nUser '%v' is not in the Cognito user pool but was found in the
 DynamoDB known users table.\n"+
     "User migration is started and a password reset is required.",
 user.UserName)
   } else {
    panic(err)
   }
  } else {
   log.Printf("User '%v' successfully signed in. This is unexpected and probably
 means you have not\n"+
    "cleaned up a previous run of this scenario, so the user exist in the Cognito
 user pool.\n"+
    "You can continue this example and select to clean up resources, or manually
 remove\n"+
    "the user from your user pool and try again.", user.UserName)
   runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
 *authResult.AccessToken)
   signedIn = true
  }
 }

 log.Println(strings.Repeat("-", 88))
 return resetRequired != nil, user
}
```

```
// ResetPassword starts a password recovery flow.
func (runner *MigrateUser) ResetPassword(clientId string, user actions.User) {
 wantCode := runner.questioner.AskBool(fmt.Sprintf("In order to migrate the user
 to Cognito, you must be able to receive a confirmation\n"+
  "code by email at %v. Do you want to send a code (y/n)?", user.UserEmail), "y")
 if !wantCode {
  log.Println("To complete this example and successfully migrate a user to
 Cognito, you must enter an email\n" +
   "you own that can receive a confirmation code.")
  return
 }
 codeDelivery, err := runner.cognitoActor.ForgotPassword(clientId, user.UserName)
 if err != nil {
  panic(err)
 }
 log.Printf("\nA confirmation code has been sent to %v.",
 *codeDelivery.Destination)
 code := runner.questioner.Ask("Check your email and enter it here:")

 confirmed := false
 password := runner.questioner.AskPassword("\nEnter a password that has at least
 eight characters, uppercase, lowercase, numbers and symbols.\n"+
  "(the password will not display as you type):", 8)
 for !confirmed {
  log.Printf("\nConfirming password reset for user '%v'.\n", user.UserName)
  err = runner.cognitoActor.ConfirmForgotPassword(clientId, code, user.UserName,
 password)
  if err != nil {
   var invalidPassword *types.InvalidPasswordException
   if errors.As(err, &invalidPassword) {
    password = runner.questioner.AskPassword("\nEnter another password:", 8)
   } else {
    panic(err)
   }
  } else {
   confirmed = true
  }
 }
 log.Printf("User '%v' successfully confirmed and migrated.\n", user.UserName)
 log.Println("Signing in with your username and password...")
 authResult, err := runner.cognitoActor.SignIn(clientId, user.UserName, password)
 if err != nil {
  panic(err)
 }
```

```
 log.Printf("Successfully signed in. Your access token starts with: %v...\n",
 (*authResult.AccessToken)[:10])
 runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
 *authResult.AccessToken)

 log.Println(strings.Repeat("-", 88))
}

// Run runs the scenario.
func (runner *MigrateUser) Run(stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup()
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]

 runner.AddMigrateUserTrigger(stackOutputs["UserPoolId"],
 stackOutputs["MigrateUserFunctionArn"])
 runner.resources.triggers = append(runner.resources.triggers,
 actions.UserMigration)
 resetNeeded, user := runner.SignInUser(stackOutputs["TableName"],
 stackOutputs["UserPoolClientId"])
 if resetNeeded {
  runner.helper.ListRecentLogEvents(stackOutputs["MigrateUserFunction"])
  runner.ResetPassword(stackOutputs["UserPoolClientId"], user)
 }

 runner.resources.Cleanup()

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
```

```
}
```

Controle el desencadenador `MigrateUser` con una función de Lambda.

```
const TABLE_NAME = "TABLE_NAME"

// UserInfo defines structured user data that can be marshalled to a DynamoDB
 format.
type UserInfo struct {
 UserName  string `dynamodbav:"UserName"`
 UserEmail string `dynamodbav:"UserEmail"`
}

type handler struct {
 dynamoClient *dynamodb.Client
}

// HandleRequest handles the MigrateUser event by looking up a user in an Amazon
 DynamoDB table and
// specifying whether they should be migrated to the user pool.
func (h *handler) HandleRequest(ctx context.Context, event
 events.CognitoEventUserPoolsMigrateUser)
 (events.CognitoEventUserPoolsMigrateUser, error) {
 log.Printf("Received migrate trigger from %v for user '%v'",
 event.TriggerSource, event.UserName)
 if event.TriggerSource != "UserMigration_Authentication" {
  return event, nil
 }
 tableName := os.Getenv(TABLE_NAME)
 user := UserInfo{
  UserName: event.UserName,
 }
 log.Printf("Looking up user '%v' in table %v.\n", user.UserName, tableName)
 filterEx := expression.Name("UserName").Equal(expression.Value(user.UserName))
 expr, err := expression.NewBuilder().WithFilter(filterEx).Build()
 if err != nil {
  log.Printf("Error building expression to query for user '%v'.\n",
 user.UserName)
  return event, err
 }
```

```
 output, err := h.dynamoClient.Scan(ctx, &dynamodb.ScanInput{
  TableName:                 aws.String(tableName),
  FilterExpression:          expr.Filter(),
  ExpressionAttributeNames:  expr.Names(),
  ExpressionAttributeValues: expr.Values(),
 })
 if err != nil {
  log.Printf("Error looking up user '%v'.\n", user.UserName)
  return event, err
 }
 if output.Items == nil || len(output.Items) == 0 {
  log.Printf("User '%v' not found, not migrating user.\n", user.UserName)
  return event, err
 }

 var users []UserInfo
 err = attributevalue.UnmarshalListOfMaps(output.Items, &users)
 if err != nil {
  log.Printf("Couldn't unmarshal DynamoDB items. Here's why: %v\n", err)
  return event, err
 }

 user = users[0]
 log.Printf("UserName '%v' found with email %v. User is migrated and must reset
 password.\n", user.UserName, user.UserEmail)
 event.CognitoEventUserPoolsMigrateUserResponse.UserAttributes =
 map[string]string{
  "email":          user.UserEmail,
  "email_verified": "true", // email_verified is required for the forgot password
 flow.
 }
 event.CognitoEventUserPoolsMigrateUserResponse.FinalUserStatus =
 "RESET_REQUIRED"
 event.CognitoEventUserPoolsMigrateUserResponse.MessageAction = "SUPPRESS"

 return event, err
}

func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  log.Panicln(err)
 }
 h := handler{
```

```
  dynamoClient: dynamodb.NewFromConfig(sdkConfig),
 }
 lambda.Start(h.HandleRequest)
}
```

Cree una estructura que lleve a cabo las tareas habituales.

```
// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
 GetStackOutputs(stackName string) (actions.StackOutputs, error)
 PopulateUserTable(tableName string)
 GetKnownUsers(tableName string) (actions.UserList, error)
 AddKnownUser(tableName string, user actions.User)
 ListRecentLogEvents(functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
 example.
type ScenarioHelper struct {
 questioner  demotools.IQuestioner
 dynamoActor *actions.DynamoActions
 cfnActor    *actions.CloudFormationActions
 cwlActor    *actions.CloudWatchLogsActions
 isTestRun   bool
}

// NewScenarioHelper constructs a new scenario helper.
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
 ScenarioHelper {
 scenario := ScenarioHelper{
  questioner:  questioner,
  dynamoActor: &actions.DynamoActions{DynamoClient:
 dynamodb.NewFromConfig(sdkConfig)},
  cfnActor:    &actions.CloudFormationActions{CfnClient:
 cloudformation.NewFromConfig(sdkConfig)},
  cwlActor:    &actions.CloudWatchLogsActions{CwlClient:
 cloudwatchlogs.NewFromConfig(sdkConfig)},
 }
```

```
 return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 if !helper.isTestRun {
  time.Sleep(time.Duration(secs) * time.Second)
 }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
 structured format.
func (helper ScenarioHelper) GetStackOutputs(stackName string)
 (actions.StackOutputs, error) {
 return helper.cfnActor.GetOutputs(stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(tableName string) {
 log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
 this example.\n", tableName)
 err := helper.dynamoActor.PopulateTable(tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(tableName string) (actions.UserList,
 error) {
 knownUsers, err := helper.dynamoActor.Scan(tableName)
 if err != nil {
  log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
 tableName, err)
 }
 return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(tableName string, user actions.User) {
 log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
 table...\n",
  user.UserName, user.UserEmail)
```

```
 err := helper.dynamoActor.AddUser(tableName, user)
 if err != nil {
  panic(err)
 }
}

// ListRecentLogEvents gets the most recent log stream and events for the
 specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(functionName string) {
 log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
 helper.Pause(10)
 log.Println("Okay, let's check the logs to find what's happened recently with
 your Lambda function.")
 logStream, err := helper.cwlActor.GetLatestLogStream(functionName)
 if err != nil {
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(functionName,
 *logStream.LogStreamName, 10)
 if err != nil {
  panic(err)
 }
 for _, event := range events {
  log.Printf("\t%v", *event.Message)
 }
 log.Println(strings.Repeat("-", 88))
}
```

Cree una estructura que ajuste las acciones de Amazon Cognito.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}
```

```
// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int

const (
 PreSignUp Trigger = iota
 UserMigration
 PostAuthentication
)

type TriggerInfo struct {
 Trigger    Trigger
 HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
 triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
```

```
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
 }
 return err
}



// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(clientId string, userName string, password
 string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(context.TODO(),
 &cognitoidentityprovider.SignUpInput{
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}



// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(clientId string, userName string, password
 string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
 &cognitoidentityprovider.InitiateAuthInput{
```

```
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
   log.Println(*resetRequired.Message)
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
 (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
 userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
```

```
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}



// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
 _, err := actor.CognitoClient.DeleteUser(context.TODO(),
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}



// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
 userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
```

```
 })
 if err != nil {
  var userExists *types.UsernameExistsException
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
 }
 return err
}



// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
 string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
  }
 }
 return err
}
```

Cree una estructura que ajuste las acciones de DynamoDB.

```
// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
 actions
// used in the examples.
type DynamoActions struct {
 DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
 UserName  string
 UserEmail string
 LastLogin *LoginInfo `dynamodbav:",omitempty"`
}

// LoginInfo defines structured custom login data.
type LoginInfo struct {
 UserPoolId string
 ClientId   string
 Time       string
}

// UserList defines a list of users.
type UserList struct {
 Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
 strings.
func (users *UserList) UserNameList() []string {
 names := make([]string, len(users.Users))
 for i := 0; i < len(users.Users); i++ {
  names[i] = users.Users[i].UserName
 }
 return names
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(tableName string) error {
 var err error
 var item map[string]types.AttributeValue
 var writeReqs []types.WriteRequest
 for i := 1; i < 4; i++ {
```

```
  item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
  if err != nil {
   log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
 err)
   return err
  }
  writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
 &types.PutRequest{Item: item}})
 }
 _, err = actor.DynamoClient.BatchWriteItem(context.TODO(),
 &dynamodb.BatchWriteItemInput{
  RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
 })
 if err != nil {
  log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
 tableName, err)
 }
 return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(tableName string) (UserList, error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(context.TODO(), &dynamodb.ScanInput{
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
 err)
 } else {
  err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
  if err != nil {
   log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
  }
 }
 return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(tableName string, user User) error {
 userItem, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
```

```
 }
 _, err = actor.DynamoClient.PutItem(context.TODO(), &dynamodb.PutItemInput{
  Item:      userItem,
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
 }
 return err
}
```

Crea una estructura que agrupe las acciones de CloudWatch Logs.

```
type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(functionName string)
 (types.LogStream, error) {
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.DescribeLogStreams(context.TODO(),
 &cloudwatchlogs.DescribeLogStreamsInput{
  Descending:   aws.Bool(true),
  Limit:        aws.Int32(1),
  LogGroupName: aws.String(logGroupName),
  OrderBy:      types.OrderByLastEventTime,
 })
 if err != nil {
  log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
 logGroupName, err)
 } else {
  logStream = output.LogStreams[0]
 }
 return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
 stream.
```

```
func (actor CloudWatchLogsActions) GetLogEvents(functionName string,
 logStreamName string, eventCount int32) (
 []types.OutputLogEvent, error) {
 var events []types.OutputLogEvent
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.GetLogEvents(context.TODO(),
 &cloudwatchlogs.GetLogEventsInput{
  LogStreamName: aws.String(logStreamName),
  Limit:         aws.Int32(eventCount),
  LogGroupName:  aws.String(logGroupName),
 })
 if err != nil {
  log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
 logStreamName, err)
 } else {
  events = output.Events
 }
 return events, err
}
```

Crea una estructura que agrupe las acciones. AWS CloudFormation

```

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
 CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
 structured format.
func (actor CloudFormationActions) GetOutputs(stackName string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(context.TODO(),
 &cloudformation.DescribeStacksInput{
  StackName: aws.String(stackName),
 })
 if err != nil || len(output.Stacks) == 0 {
  log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
 stackName, err)
 }
```

```
 stackOutputs := StackOutputs{}
 for _, out := range output.Stacks[0].Outputs {
  stackOutputs[*out.OutputKey] = *out.OutputValue
 }
 return stackOutputs
}
```

Eliminación de recursos.

```
// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {
 userPoolId       string
 userAccessTokens []string
 triggers         []actions.Trigger

 cognitoActor *actions.CognitoActions
 questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
 demotools.IQuestioner) {
 resources.userAccessTokens = []string{}
 resources.triggers = []actions.Trigger{}
 resources.cognitoActor = cognitoActor
 resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
func (resources *Resources) Cleanup() {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong during cleanup.\n%v\n", r)
   log.Println("Use the AWS Management Console to remove any remaining resources
 \n" +
    "that were created for this scenario.")
  }
 }()
```

```
 wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
 resources that were created "+
  "during this demo (y/n)?", "y")
 if wantDelete {
  for _, accessToken := range resources.userAccessTokens {
   err := resources.cognitoActor.DeleteUser(accessToken)
   if err != nil {
    log.Println("Couldn't delete user during cleanup.")
    panic(err)
   }
   log.Println("Deleted user.")
  }
  triggerList := make([]actions.TriggerInfo, len(resources.triggers))
  for i := 0; i < len(resources.triggers); i++ {
   triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
 HandlerArn: nil}
  }
  err := resources.cognitoActor.UpdateTriggers(resources.userPoolId,
 triggerList...)
  if err != nil {
   log.Println("Couldn't update Cognito triggers during cleanup.")
   panic(err)
  }
  log.Println("Removed Cognito triggers from user pool.")
 } else {
  log.Println("Be sure to remove resources when you're done with them to avoid
 unexpected charges!")
 }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK for Go .
  - ConfirmForgotPassword
  - DeleteUser
  - ForgotPassword
  - InitiateAuth
  - SignUp
  - UpdateUserPool

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Registrar un usuario con un grupo de usuarios de Amazon Cognito que requiera MFA mediante un SDK AWS

En el siguiente ejemplo de código, se muestra cómo:

- Registre y confirme a un usuario con un nombre de usuario, una contraseña y una dirección de correo electrónico.
- Configure la autenticación multifactor asociando una aplicación MFA al usuario.
- Inicie sesión con una contraseña y un código MFA.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
namespace CognitoBasics;

public class CognitoBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for Amazon Cognito.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
```

```
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonCognitoIdentityProvider>()
            .AddTransient<CognitoWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<CognitoBasics>();

        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        var cognitoWrapper = host.Services.GetRequiredService<CognitoWrapper>();

        Console.WriteLine(new string('-', 80));
        UiMethods.DisplayOverview();
        Console.WriteLine(new string('-', 80));

        // clientId - The app client Id value that you get from the AWS CDK
script.
        var clientId = configuration["ClientId"]; // "*** REPLACE WITH CLIENT ID
VALUE FROM CDK SCRIPT";

        // poolId - The pool Id that you get from the AWS CDK script.
        var poolId = configuration["PoolId"]!; // "*** REPLACE WITH POOL ID VALUE
FROM CDK SCRIPT";
        var userName = configuration["UserName"];
        var password = configuration["Password"];
        var email = configuration["Email"];

        // If the username wasn't set in the configuration file,
        // get it from the user now.
        if (userName is null)
        {
            do
            {
                Console.Write("Username: ");
                userName = Console.ReadLine();
            }
            while (string.IsNullOrEmpty(userName));
```

```
        }
        Console.WriteLine($"\nUsername: {userName}");

        // If the password wasn't set in the configuration file,
        // get it from the user now.
        if (password is null)
        {
            do
            {
                Console.Write("Password: ");
                password = Console.ReadLine();
            }
            while (string.IsNullOrEmpty(password));
        }

        // If the email address wasn't set in the configuration file,
        // get it from the user now.
        if (email is null)
        {
            do
            {
                Console.Write("Email: ");
                email = Console.ReadLine();
            } while (string.IsNullOrEmpty(email));
        }

        // Now sign up the user.
        Console.WriteLine($"\nSigning up {userName} with email address:
{email}");
        await cognitoWrapper.SignUpAsync(clientId, userName, password, email);

        // Add the user to the user pool.
        Console.WriteLine($"Adding {userName} to the user pool");
        await cognitoWrapper.GetAdminUserAsync(userName, poolId);

        UiMethods.DisplayTitle("Get confirmation code");
        Console.WriteLine($"Conformation code sent to {userName}.");
        Console.Write("Would you like to send a new code? (Y/N) ");
        var answer = Console.ReadLine();

        if (answer!.ToLower() == "y")
        {
            await cognitoWrapper.ResendConfirmationCodeAsync(clientId, userName);
            Console.WriteLine("Sending a new confirmation code");
```

```
        }

        Console.Write("Enter confirmation code (from Email): ");
        var code = Console.ReadLine();

        await cognitoWrapper.ConfirmSignupAsync(clientId, code, userName);

        UiMethods.DisplayTitle("Checking status");
        Console.WriteLine($"Rechecking the status of {userName} in the user
pool");
        await cognitoWrapper.GetAdminUserAsync(userName, poolId);

        Console.WriteLine($"Setting up authenticator for {userName} in the user
pool");
        var setupResponse = await cognitoWrapper.InitiateAuthAsync(clientId,
userName, password);

        var setupSession = await
cognitoWrapper.AssociateSoftwareTokenAsync(setupResponse.Session);
        Console.Write("Enter the 6-digit code displayed in Google Authenticator:
");
        var setupCode = Console.ReadLine();

        var setupResult = await
cognitoWrapper.VerifySoftwareTokenAsync(setupSession, setupCode);
        Console.WriteLine($"Setup status: {setupResult}");

        Console.WriteLine($"Now logging in {userName} in the user pool");
        var authSession = await cognitoWrapper.AdminInitiateAuthAsync(clientId,
poolId, userName, password);

        Console.Write("Enter a new 6-digit code displayed in Google
Authenticator: ");
        var authCode = Console.ReadLine();

        var authResult = await
cognitoWrapper.AdminRespondToAuthChallengeAsync(userName, clientId, authCode,
authSession, poolId);
        Console.WriteLine($"Authenticated and received access token:
{authResult.AccessToken}");

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Cognito scenario is complete.");
        Console.WriteLine(new string('-', 80));
```

```
    }
}


using System.Net;

namespace CognitoActions;

/// <summary>
/// Methods to perform Amazon Cognito Identity Provider actions.
/// </summary>
public class CognitoWrapper
{
    private readonly IAmazonCognitoIdentityProvider _cognitoService;

    /// <summary>
    /// Constructor for the wrapper class containing Amazon Cognito actions.
    /// </summary>
    /// <param name="cognitoService">The Amazon Cognito client object.</param>
    public CognitoWrapper(IAmazonCognitoIdentityProvider cognitoService)
    {
        _cognitoService = cognitoService;
    }

    /// <summary>
    /// List the Amazon Cognito user pools for an account.
    /// </summary>
    /// <returns>A list of UserPoolDescriptionType objects.</returns>
    public async Task<List<UserPoolDescriptionType>> ListUserPoolsAsync()
    {
        var userPools = new List<UserPoolDescriptionType>();

        var userPoolsPaginator = _cognitoService.Paginators.ListUserPools(new
 ListUserPoolsRequest());

        await foreach (var response in userPoolsPaginator.Responses)
        {
            userPools.AddRange(response.UserPools);
        }

        return userPools;
    }
```

```
    /// <summary>
    /// Get a list of users for the Amazon Cognito user pool.
    /// </summary>
    /// <param name="userPoolId">The user pool ID.</param>
    /// <returns>A list of users.</returns>
    public async Task<List<UserType>> ListUsersAsync(string userPoolId)
    {
        var request = new ListUsersRequest
        {
            UserPoolId = userPoolId
        };

        var users = new List<UserType>();

        var usersPaginator = _cognitoService.Paginators.ListUsers(request);
        await foreach (var response in usersPaginator.Responses)
        {
            users.AddRange(response.Users);
        }

        return users;
    }


    /// <summary>
    /// Respond to an admin authentication challenge.
    /// </summary>
    /// <param name="userName">The name of the user.</param>
    /// <param name="clientId">The client ID.</param>
    /// <param name="mfaCode">The multi-factor authentication code.</param>
    /// <param name="session">The current application session.</param>
    /// <param name="clientId">The user pool ID.</param>
    /// <returns>The result of the authentication response.</returns>
    public async Task<AuthenticationResultType> AdminRespondToAuthChallengeAsync(
        string userName,
        string clientId,
        string mfaCode,
        string session,
        string userPoolId)
    {
        Console.WriteLine("SOFTWARE_TOKEN_MFA challenge is generated");

        var challengeResponses = new Dictionary<string, string>();
        challengeResponses.Add("USERNAME", userName);
```

```
        challengeResponses.Add("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        var respondToAuthChallengeRequest = new
AdminRespondToAuthChallengeRequest
        {
            ChallengeName = ChallengeNameType.SOFTWARE_TOKEN_MFA,
            ClientId = clientId,
            ChallengeResponses = challengeResponses,
            Session = session,
            UserPoolId = userPoolId,
        };

        var response = await
_cognitoService.AdminRespondToAuthChallengeAsync(respondToAuthChallengeRequest);
        Console.WriteLine($"Response to Authentication
{response.AuthenticationResult.TokenType}");
        return response.AuthenticationResult;
    }


    /// <summary>
    /// Verify the TOTP and register for MFA.
    /// </summary>
    /// <param name="session">The name of the session.</param>
    /// <param name="code">The MFA code.</param>
    /// <returns>The status of the software token.</returns>
    public async Task<VerifySoftwareTokenResponseType>
VerifySoftwareTokenAsync(string session, string code)
    {
        var tokenRequest = new VerifySoftwareTokenRequest
        {
            UserCode = code,
            Session = session,
        };

        var verifyResponse = await
_cognitoService.VerifySoftwareTokenAsync(tokenRequest);

        return verifyResponse.Status;
    }


    /// <summary>
    /// Get an MFA token to authenticate the user with the authenticator.
```

```
    /// </summary>
    /// <param name="session">The session name.</param>
    /// <returns>The session name.</returns>
    public async Task<string> AssociateSoftwareTokenAsync(string session)
    {
        var softwareTokenRequest = new AssociateSoftwareTokenRequest
        {
            Session = session,
        };

        var tokenResponse = await
_cognitoService.AssociateSoftwareTokenAsync(softwareTokenRequest);
        var secretCode = tokenResponse.SecretCode;

        Console.WriteLine($"Use the following secret code to set up the
authenticator: {secretCode}");

        return tokenResponse.Session;
    }


    /// <summary>
    /// Initiate an admin auth request.
    /// </summary>
    /// <param name="clientId">The client ID to use.</param>
    /// <param name="userPoolId">The ID of the user pool.</param>
    /// <param name="userName">The username to authenticate.</param>
    /// <param name="password">The user's password.</param>
    /// <returns>The session to use in challenge-response.</returns>
    public async Task<string> AdminInitiateAuthAsync(string clientId, string
userPoolId, string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var request = new AdminInitiateAuthRequest
        {
            ClientId = clientId,
            UserPoolId = userPoolId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
        };
```

```
        var response = await _cognitoService.AdminInitiateAuthAsync(request);
        return response.Session;
    }

    /// <summary>
    /// Initiate authorization.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The name of the user who is authenticating.</
param>
    /// <param name="password">The password for the user who is authenticating.</
param>
    /// <returns>The response from the initiate auth request.</returns>
    public async Task<InitiateAuthResponse> InitiateAuthAsync(string clientId,
 string userName, string password)
    {
        var authParameters = new Dictionary<string, string>();
        authParameters.Add("USERNAME", userName);
        authParameters.Add("PASSWORD", password);

        var authRequest = new InitiateAuthRequest

        {
            ClientId = clientId,
            AuthParameters = authParameters,
            AuthFlow = AuthFlowType.USER_PASSWORD_AUTH,
        };

        var response = await _cognitoService.InitiateAuthAsync(authRequest);
        Console.WriteLine($"Result Challenge is : {response.ChallengeName}");

        return response;
    }

    /// <summary>
    /// Confirm that the user has signed up.
    /// </summary>
    /// <param name="clientId">The Id of this application.</param>
    /// <param name="code">The confirmation code sent to the user.</param>
    /// <param name="userName">The username.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ConfirmSignupAsync(string clientId, string code,
 string userName)
    {
```

```
        var signUpRequest = new ConfirmSignUpRequest
        {
            ClientId = clientId,
            ConfirmationCode = code,
            Username = userName,
        };

        var response = await _cognitoService.ConfirmSignUpAsync(signUpRequest);
        if (response.HttpStatusCode == HttpStatusCode.OK)
        {
            Console.WriteLine($"{userName} was confirmed");
            return true;
        }
        return false;
    }


    /// <summary>
    /// Initiates and confirms tracking of the device.
    /// </summary>
    /// <param name="accessToken">The user's access token.</param>
    /// <param name="deviceKey">The key of the device from Amazon Cognito.</
param>
    /// <param name="deviceName">The device name.</param>
    /// <returns></returns>
    public async Task<bool> ConfirmDeviceAsync(string accessToken, string
 deviceKey, string deviceName)
    {
        var request = new ConfirmDeviceRequest
        {
            AccessToken = accessToken,
            DeviceKey = deviceKey,
            DeviceName = deviceName
        };

        var response = await _cognitoService.ConfirmDeviceAsync(request);
        return response.UserConfirmationNecessary;
    }


    /// <summary>
    /// Send a new confirmation code to a user.
    /// </summary>
    /// <param name="clientId">The Id of the client application.</param>
```

```
    /// <param name="userName">The username of user who will receive the code.</
param>
    /// <returns>The delivery details.</returns>
    public async Task<CodeDeliveryDetailsType> ResendConfirmationCodeAsync(string
 clientId, string userName)
    {
        var codeRequest = new ResendConfirmationCodeRequest
        {
            ClientId = clientId,
            Username = userName,
        };

        var response = await
 _cognitoService.ResendConfirmationCodeAsync(codeRequest);

        Console.WriteLine($"Method of delivery is
 {response.CodeDeliveryDetails.DeliveryMedium}");

        return response.CodeDeliveryDetails;
    }


    /// <summary>
    /// Get the specified user from an Amazon Cognito user pool with
 administrator access.
    /// </summary>
    /// <param name="userName">The name of the user.</param>
    /// <param name="poolId">The Id of the Amazon Cognito user pool.</param>
    /// <returns>Async task.</returns>
    public async Task<UserStatusType> GetAdminUserAsync(string userName, string
 poolId)
    {
        AdminGetUserRequest userRequest = new AdminGetUserRequest
        {
            Username = userName,
            UserPoolId = poolId,
        };

        var response = await _cognitoService.AdminGetUserAsync(userRequest);

        Console.WriteLine($"User status {response.UserStatus}");
        return response.UserStatus;
    }
```

```
    /// <summary>
    /// Sign up a new user.
    /// </summary>
    /// <param name="clientId">The client Id of the application.</param>
    /// <param name="userName">The username to use.</param>
    /// <param name="password">The user's password.</param>
    /// <param name="email">The email address of the user.</param>
    /// <returns>A Boolean value indicating whether the user was confirmed.</
returns>
    public async Task<bool> SignUpAsync(string clientId, string userName, string
 password, string email)
    {
        var userAttrs = new AttributeType
        {
            Name = "email",
            Value = email,
        };

        var userAttrsList = new List<AttributeType>();

        userAttrsList.Add(userAttrs);

        var signUpRequest = new SignUpRequest
        {
            UserAttributes = userAttrsList,
            Username = userName,
            ClientId = clientId,
            Password = password
        };

        var response = await _cognitoService.SignUpAsync(signUpRequest);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }

}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for .NET .
  - AdminGetUser

- AdminInitiateAuth
- AdminRespondToAuthChallenge
- AssociateSoftwareToken
- ConfirmDevice
- ConfirmSignUp
- InitiateAuth
- ListUsers
- ResendConfirmationCode
- RespondToAuthChallenge
- SignUp
- VerifySoftwareToken

C++

SDK para C++

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

//! Scenario that adds a user to an Amazon Cognito user pool.
/*!
  \sa gettingStartedWithUserPools()
  \param clientID: Client ID associated with an Amazon Cognito user pool.
  \param userPoolID: An Amazon Cognito user pool ID.
  \param clientConfig: Aws client configuration.
  \return bool: Successful completion.
 */
bool AwsDoc::Cognito::gettingStartedWithUserPools(const Aws::String &clientID,
                                                  const Aws::String &userPoolID,
```

```
                                                    const
Aws::Client::ClientConfiguration &clientConfig) {
   printAsterisksLine();
   std::cout
           << "Welcome to the Amazon Cognito example scenario."
           << std::endl;
   printAsterisksLine();

   std::cout
           << "This scenario will add a user to an Amazon Cognito user pool."
           << std::endl;
   const Aws::String userName = askQuestion("Enter a new username: ");
   const Aws::String password = askQuestion("Enter a new password: ");
   const Aws::String email = askQuestion("Enter a valid email for the user: ");

   std::cout << "Signing up " << userName << std::endl;

   Aws::CognitoIdentityProvider::CognitoIdentityProviderClient
client(clientConfig);
   bool userExists = false;
   do {
       // 1. Add a user with a username, password, and email address.
       Aws::CognitoIdentityProvider::Model::SignUpRequest request;
       request.AddUserAttributes(
               Aws::CognitoIdentityProvider::Model::AttributeType().WithName(
                       "email").WithValue(email));
       request.SetUsername(userName);
       request.SetPassword(password);
       request.SetClientId(clientID);
       Aws::CognitoIdentityProvider::Model::SignUpOutcome outcome =
               client.SignUp(request);

       if (outcome.IsSuccess()) {
           std::cout << "The signup request for " << userName << " was
successful."
                     << std::endl;
       }
       else if (outcome.GetError().GetErrorType() ==

Aws::CognitoIdentityProvider::CognitoIdentityProviderErrors::USERNAME_EXISTS) {
           std::cout
                   << "The username already exists. Please enter a different
username."
                   << std::endl;
```

```
            userExists = true;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::SignUpRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (userExists);

    printAsterisksLine();
    std::cout << "Retrieving status of " << userName << " in the user pool."
              << std::endl;
    // 2. Confirm that the user was added to the user pool.
    if (!checkAdminUserStatus(userName, userPoolID, client)) {
        return false;
    }

    std::cout << "A confirmation code was sent to " << email << "." << std::endl;

    bool resend = askYesNoQuestion("Would you like to send a new code? (y/n) ");
    if (resend) {
        // Request a resend of the confirmation code to the email address.
(ResendConfirmationCode)
        Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeRequest
request;
        request.SetUsername(userName);
        request.SetClientId(clientID);

        Aws::CognitoIdentityProvider::Model::ResendConfirmationCodeOutcome
outcome =
                client.ResendConfirmationCode(request);

        if (outcome.IsSuccess()) {
            std::cout
                    << "CognitoIdentityProvider::ResendConfirmationCode was
successful."
                    << std::endl;
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::ResendConfirmationCode. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
```

```
            return false;
        }
    }

    printAsterisksLine();

    {
        // 4. Send the confirmation code that's received in the email.
(ConfirmSignUp)
        const Aws::String confirmationCode = askQuestion(
                "Enter the confirmation code that was emailed: ");
        Aws::CognitoIdentityProvider::Model::ConfirmSignUpRequest request;
        request.SetClientId(clientID);
        request.SetConfirmationCode(confirmationCode);
        request.SetUsername(userName);

        Aws::CognitoIdentityProvider::Model::ConfirmSignUpOutcome outcome =
                client.ConfirmSignUp(request);

        if (outcome.IsSuccess()) {
            std::cout << "ConfirmSignup was Successful."
                      << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::ConfirmSignUp. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    std::cout << "Rechecking the status of " << userName << " in the user pool."
              << std::endl;
    if (!checkAdminUserStatus(userName, userPoolID, client)) {
        return false;
    }

    printAsterisksLine();

    std::cout << "Initiating authorization using the username and password."
              << std::endl;

    Aws::String session;
    // 5. Initiate authorization with username and password. (AdminInitiateAuth)
```

```
    if (!adminInitiateAuthorization(clientID, userPoolID,  userName, password,
 session, client)) {
        return false;
    }

    printAsterisksLine();

    std::cout
            << "Starting setup of time-based one-time password (TOTP) multi-
factor authentication (MFA)."
            << std::endl;

    {
        // 6. Request a setup key for one-time password (TOTP)
        //    multi-factor authentication (MFA). (AssociateSoftwareToken)
        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenRequest
 request;
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AssociateSoftwareTokenOutcome
 outcome =
                client.AssociateSoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout
                    << "Enter this setup key into an authenticator app, for
 example Google Authenticator."
                    << std::endl;
            std::cout << "Setup key: " << outcome.GetResult().GetSecretCode()
                      << std::endl;
#ifdef USING_QR
            printAsterisksLine();
            std::cout << "\nOr scan the QR code in the file '" << QR_CODE_PATH <<
 "."
                      << std::endl;

            saveQRCode(std::string("otpauth://totp/") + userName + "?secret=" +
                       outcome.GetResult().GetSecretCode());
#endif // USING_QR
            session = outcome.GetResult().GetSession();
        }
        else {
            std::cerr << "Error with
 CognitoIdentityProvider::AssociateSoftwareToken. "
```

```
                    << outcome.GetError().GetMessage()
                    << std::endl;
            return false;
        }
    }
    askQuestion("Type enter to continue...", alwaysTrueTest);

    printAsterisksLine();

    {
        Aws::String userCode = askQuestion(
                "Enter the 6 digit code displayed in the authenticator app: ");

        //  7. Send the MFA code copied from an authenticator app.
(VerifySoftwareToken)
        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenRequest request;
        request.SetUserCode(userCode);
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::VerifySoftwareTokenOutcome outcome =
                client.VerifySoftwareToken(request);

        if (outcome.IsSuccess()) {
            std::cout << "Verification of the code was successful."
                    << std::endl;
            session = outcome.GetResult().GetSession();
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::VerifySoftwareToken. "
                    << outcome.GetError().GetMessage()
                    << std::endl;
            return false;
        }
    }

    printAsterisksLine();
    std::cout << "You have completed the MFA authentication setup." << std::endl;
    std::cout << "Now, sign in." << std::endl;

    // 8. Initiate authorization again with username and password.
(AdminInitiateAuth)
    if (!adminInitiateAuthorization(clientID, userPoolID, userName, password,
session, client)) {
```

```
        return false;
    }

    Aws::String accessToken;
    {
        Aws::String mfaCode = askQuestion(
                "Re-enter the 6 digit code displayed in the authenticator app:
");

        // 9. Send a new MFA code copied from an authenticator app.
(AdminRespondToAuthChallenge)
        Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeRequest
request;
        request.AddChallengeResponses("USERNAME", userName);
        request.AddChallengeResponses("SOFTWARE_TOKEN_MFA_CODE", mfaCode);
        request.SetChallengeName(

Aws::CognitoIdentityProvider::Model::ChallengeNameType::SOFTWARE_TOKEN_MFA);
        request.SetClientId(clientID);
        request.SetUserPoolId(userPoolID);
        request.SetSession(session);

        Aws::CognitoIdentityProvider::Model::AdminRespondToAuthChallengeOutcome
outcome =
                client.AdminRespondToAuthChallenge(request);

        if (outcome.IsSuccess()) {
            std::cout << "Here is the response to the challenge.\n" <<

outcome.GetResult().GetAuthenticationResult().Jsonize().View().WriteReadable()
                      << std::endl;

            accessToken =
outcome.GetResult().GetAuthenticationResult().GetAccessToken();
        }
        else {
            std::cerr << "Error with
CognitoIdentityProvider::AdminRespondToAuthChallenge. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

        std::cout << "You have successfully added a user to Amazon Cognito."
```

```
                  << std::endl;
    }

    if (askYesNoQuestion("Would you like to delete the user that you just added?
 (y/n) ")) {
        // 10. Delete the user that you just added. (DeleteUser)
        Aws::CognitoIdentityProvider::Model::DeleteUserRequest request;
        request.SetAccessToken(accessToken);

        Aws::CognitoIdentityProvider::Model::DeleteUserOutcome outcome =
                client.DeleteUser(request);

        if (outcome.IsSuccess()) {
            std::cout << "The user " << userName << " was deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with CognitoIdentityProvider::DeleteUser. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
    }

    return true;
}

//! Routine which checks the user status in an Amazon Cognito user pool.
/*!
 \sa checkAdminUserStatus()
 \param userName: A username.
 \param userPoolID: An Amazon Cognito user pool ID.
 \return bool: Successful completion.
 */
bool AwsDoc::Cognito::checkAdminUserStatus(const Aws::String &userName,
                                           const Aws::String &userPoolID,
                                           const
 Aws::CognitoIdentityProvider::CognitoIdentityProviderClient &client) {
    Aws::CognitoIdentityProvider::Model::AdminGetUserRequest request;
    request.SetUsername(userName);
    request.SetUserPoolId(userPoolID);

    Aws::CognitoIdentityProvider::Model::AdminGetUserOutcome outcome =
            client.AdminGetUser(request);
```

```
    if (outcome.IsSuccess()) {
        std::cout << "The status for " << userName << " is " <<

 Aws::CognitoIdentityProvider::Model::UserStatusTypeMapper::GetNameForUserStatusType(
                          outcome.GetResult().GetUserStatus()) << std::endl;
        std::cout << "Enabled is " << outcome.GetResult().GetEnabled() <<
 std::endl;
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminGetUser. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}

//! Routine which starts authorization of an Amazon Cognito user.
//! This routine requires administrator credentials.
/*!
 \sa adminInitiateAuthorization()
 \param clientID: Client ID of tracked device.
 \param userPoolID: An Amazon Cognito user pool ID.
 \param userName: A username.
 \param password: A password.
 \param sessionResult: String to receive a session token.
 \return bool: Successful completion.
 */
bool AwsDoc::Cognito::adminInitiateAuthorization(const Aws::String &clientID,
                                                 const Aws::String &userPoolID,
                                                 const Aws::String &userName,
                                                 const Aws::String &password,
                                                 Aws::String &sessionResult,
                                                 const
 Aws::CognitoIdentityProvider::CognitoIdentityProviderClient &client) {
    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthRequest request;
    request.SetClientId(clientID);
    request.SetUserPoolId(userPoolID);
    request.AddAuthParameters("USERNAME", userName);
    request.AddAuthParameters("PASSWORD", password);
    request.SetAuthFlow(

 Aws::CognitoIdentityProvider::Model::AuthFlowType::ADMIN_USER_PASSWORD_AUTH);
```

```
    Aws::CognitoIdentityProvider::Model::AdminInitiateAuthOutcome outcome =
            client.AdminInitiateAuth(request);

    if (outcome.IsSuccess()) {
        std::cout << "Call to AdminInitiateAuth was successful." << std::endl;
        sessionResult = outcome.GetResult().GetSession();
    }
    else {
        std::cerr << "Error with CognitoIdentityProvider::AdminInitiateAuth. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for C++ .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

Java

SDK para Java 2.x

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminGetUserRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminGetUserResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminInitiateAuthRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminInitiateAuthResponse;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminRespondToAuthChalleng
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminRespondToAuthChalleng
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenRequ
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenResp
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AttributeType;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AuthFlowType;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ChallengeNameType;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ConfirmSignUpRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ResendConfirmationCodeRequ
```

```
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ResendConfirmationCodeResp
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.SignUpRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.VerifySoftwareTokenRequest
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.VerifySoftwareTokenRespons
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * TIP: To set up the required user pool, run the AWS Cloud Development Kit (AWS
 * CDK) script provided in this GitHub repo at
 * resources/cdk/cognito_scenario_user_pool_with_mfa.
 *
 * This code example performs the following operations:
 *
 * 1. Invokes the signUp method to sign up a user.
 * 2. Invokes the adminGetUser method to get the user's confirmation status.
 * 3. Invokes the ResendConfirmationCode method if the user requested another
 * code.
 * 4. Invokes the confirmSignUp method.
 * 5. Invokes the AdminInitiateAuth to sign in. This results in being prompted
 * to set up TOTP (time-based one-time password). (The response is
 * “ChallengeName”: “MFA_SETUP”).
 * 6. Invokes the AssociateSoftwareToken method to generate a TOTP MFA private
 * key. This can be used with Google Authenticator.
 * 7. Invokes the VerifySoftwareToken method to verify the TOTP and register for
 * MFA.
 * 8. Invokes the AdminInitiateAuth to sign in again. This results in being
```

```
 * prompted to submit a TOTP (Response: “ChallengeName”: “SOFTWARE_TOKEN_MFA”).
 * 9. Invokes the AdminRespondToAuthChallenge to get back a token.
 */

public class CognitoMVP {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws NoSuchAlgorithmException,
 InvalidKeyException {
        final String usage = """

                Usage:
                    <clientId> <poolId>

                Where:
                    clientId - The app client Id value that you can get from the
 AWS CDK script.
                    poolId - The pool Id that you can get from the AWS CDK
 script.\s
                """;

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

        String clientId = args[0];
        String poolId = args[1];
        CognitoIdentityProviderClient identityProviderClient =
 CognitoIdentityProviderClient.builder()
                .region(Region.US_EAST_1)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon Cognito example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Enter your user name");
        Scanner in = new Scanner(System.in);
        String userName = in.nextLine();

        System.out.println("*** Enter your password");
```

```
        String password = in.nextLine();

        System.out.println("*** Enter your email");
        String email = in.nextLine();

        System.out.println("1. Signing up " + userName);
        signUp(identityProviderClient, clientId, userName, password, email);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Getting " + userName + " in the user pool");
        getAdminUser(identityProviderClient, userName, poolId);

        System.out
                .println("*** Conformation code sent to " + userName + ". Would
you like to send a new code? (Yes/No)");
        System.out.println(DASHES);

        System.out.println(DASHES);
        String ans = in.nextLine();

        if (ans.compareTo("Yes") == 0) {
            resendConfirmationCode(identityProviderClient, clientId, userName);
            System.out.println("3. Sending a new confirmation code");
        }
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Enter confirmation code that was emailed");
        String code = in.nextLine();
        confirmSignUp(identityProviderClient, clientId, code, userName);
        System.out.println("Rechecking the status of " + userName + " in the user
pool");
        getAdminUser(identityProviderClient, userName, poolId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Invokes the initiateAuth to sign in");
        AdminInitiateAuthResponse authResponse =
initiateAuth(identityProviderClient, clientId, userName, password,
                poolId);
        String mySession = authResponse.session();
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("6. Invokes the AssociateSoftwareToken method to
generate a TOTP key");
        String newSession = getSecretForAppMFA(identityProviderClient,
mySession);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Enter the 6-digit code displayed in Google
Authenticator");
        String myCode = in.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Verify the TOTP and register for MFA");
        verifyTOTP(identityProviderClient, newSession, myCode);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Re-enter a 6-digit code displayed in Google
Authenticator");
        String mfaCode = in.nextLine();
        AdminInitiateAuthResponse authResponse1 =
initiateAuth(identityProviderClient, clientId, userName, password,
                poolId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9.  Invokes the AdminRespondToAuthChallenge");
        String session2 = authResponse1.session();
        adminRespondToAuthChallenge(identityProviderClient, userName, clientId,
mfaCode, session2);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("All Amazon Cognito operations were successfully
performed");
        System.out.println(DASHES);
    }

    // Respond to an authentication challenge.
    public static void adminRespondToAuthChallenge(CognitoIdentityProviderClient
identityProviderClient,
            String userName, String clientId, String mfaCode, String session) {
```

```
        System.out.println("SOFTWARE_TOKEN_MFA challenge is generated");
        Map<String, String> challengeResponses = new HashMap<>();

        challengeResponses.put("USERNAME", userName);
        challengeResponses.put("SOFTWARE_TOKEN_MFA_CODE", mfaCode);

        AdminRespondToAuthChallengeRequest respondToAuthChallengeRequest =
AdminRespondToAuthChallengeRequest.builder()
                .challengeName(ChallengeNameType.SOFTWARE_TOKEN_MFA)
                .clientId(clientId)
                .challengeResponses(challengeResponses)
                .session(session)
                .build();

        AdminRespondToAuthChallengeResponse respondToAuthChallengeResult =
identityProviderClient
                .adminRespondToAuthChallenge(respondToAuthChallengeRequest);

System.out.println("respondToAuthChallengeResult.getAuthenticationResult()"
                + respondToAuthChallengeResult.authenticationResult());
    }

    // Verify the TOTP and register for MFA.
    public static void verifyTOTP(CognitoIdentityProviderClient
identityProviderClient, String session, String code) {
        try {
            VerifySoftwareTokenRequest tokenRequest =
VerifySoftwareTokenRequest.builder()
                    .userCode(code)
                    .session(session)
                    .build();

            VerifySoftwareTokenResponse verifyResponse =
identityProviderClient.verifySoftwareToken(tokenRequest);
            System.out.println("The status of the token is " +
verifyResponse.statusAsString());

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

```
   public static AdminInitiateAuthResponse
initiateAuth(CognitoIdentityProviderClient identityProviderClient,
           String clientId, String userName, String password, String userPoolId)
{
       try {
           Map<String, String> authParameters = new HashMap<>();
           authParameters.put("USERNAME", userName);
           authParameters.put("PASSWORD", password);

           AdminInitiateAuthRequest authRequest =
AdminInitiateAuthRequest.builder()
                   .clientId(clientId)
                   .userPoolId(userPoolId)
                   .authParameters(authParameters)
                   .authFlow(AuthFlowType.ADMIN_USER_PASSWORD_AUTH)
                   .build();

           AdminInitiateAuthResponse response =
identityProviderClient.adminInitiateAuth(authRequest);
           System.out.println("Result Challenge is : " +
response.challengeName());
           return response;

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

       return null;
   }

   public static String getSecretForAppMFA(CognitoIdentityProviderClient
identityProviderClient, String session) {
       AssociateSoftwareTokenRequest softwareTokenRequest =
AssociateSoftwareTokenRequest.builder()
               .session(session)
               .build();

       AssociateSoftwareTokenResponse tokenResponse = identityProviderClient
               .associateSoftwareToken(softwareTokenRequest);
       String secretCode = tokenResponse.secretCode();
       System.out.println("Enter this token into Google Authenticator");
       System.out.println(secretCode);
       return tokenResponse.session();
```

```
    }

    public static void confirmSignUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String code,
            String userName) {
        try {
            ConfirmSignUpRequest signUpRequest = ConfirmSignUpRequest.builder()
                    .clientId(clientId)
                    .confirmationCode(code)
                    .username(userName)
                    .build();

            identityProviderClient.confirmSignUp(signUpRequest);
            System.out.println(userName + " was confirmed");

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

    public static void resendConfirmationCode(CognitoIdentityProviderClient
identityProviderClient, String clientId,
            String userName) {
        try {
            ResendConfirmationCodeRequest codeRequest =
ResendConfirmationCodeRequest.builder()
                    .clientId(clientId)
                    .username(userName)
                    .build();

            ResendConfirmationCodeResponse response =
identityProviderClient.resendConfirmationCode(codeRequest);
            System.out.println("Method of delivery is " +
response.codeDeliveryDetails().deliveryMediumAsString());

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

    public static void signUp(CognitoIdentityProviderClient
identityProviderClient, String clientId, String userName,
```

```
            String password, String email) {
        AttributeType userAttrs = AttributeType.builder()
                .name("email")
                .value(email)
                .build();

        List<AttributeType> userAttrsList = new ArrayList<>();
        userAttrsList.add(userAttrs);
        try {
            SignUpRequest signUpRequest = SignUpRequest.builder()
                    .userAttributes(userAttrsList)
                    .username(userName)
                    .clientId(clientId)
                    .password(password)
                    .build();

            identityProviderClient.signUp(signUpRequest);
            System.out.println("User has been signed up ");

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

    public static void getAdminUser(CognitoIdentityProviderClient
 identityProviderClient, String userName,
            String poolId) {
        try {
            AdminGetUserRequest userRequest = AdminGetUserRequest.builder()
                    .username(userName)
                    .userPoolId(poolId)
                    .build();

            AdminGetUserResponse response =
 identityProviderClient.adminGetUser(userRequest);
            System.out.println("User status " + response.userStatusAsString());

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

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Para obtener la mejor experiencia, clone el GitHub repositorio y ejecute este ejemplo. El código siguiente es una muestra de la aplicación de ejemplo completa.

```
import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { signUp } from "../../../actions/sign-up.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getSecondValuesFromEntries } from "@aws-doc-sdk-examples/lib/utils/util-
csv.js";
```

```
const validateClient = (clientId) => {
  if (!clientId) {
    throw new Error(
      `App client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateUser = (username, password, email) => {
  if (!(username && password && email)) {
    throw new Error(
      `Username, password, and email must be provided as arguments to the 'sign-
up' command.`,
    );
  }
};

const signUpHandler = async (commands) => {
  const [_, username, password, email] = commands;

  try {
    validateUser(username, password, email);
    /**
     * @type {string[]}
     */
    const values = getSecondValuesFromEntries(FILE_USER_POOLS);
    const clientId = values[0];
    validateClient(clientId);
    log(`Signing up.`);
    await signUp({ clientId, username, password, email });
    log(`Signed up. A confirmation email has been sent to: ${email}.`);
    log(`Run 'confirm-sign-up ${username} <code>' to confirm your account.`);
  } catch (err) {
    log(err);
  }
};

export { signUpHandler };

const signUp = ({ clientId, username, password, email }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new SignUpCommand({
    ClientId: clientId,
```

```
    Username: username,
    Password: password,
    UserAttributes: [{ Name: "email", Value: email }],
  });

  return client.send(command);
};

import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { confirmSignUp } from "../../../actions/confirm-sign-up.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getSecondValuesFromEntries } from "@aws-doc-sdk-examples/lib/utils/util-
csv.js";

const validateClient = (clientId) => {
  if (!clientId) {
    throw new Error(
      `App client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateUser = (username) => {
  if (!username) {
    throw new Error(
      `Username name is missing. It must be provided as an argument to the
 'confirm-sign-up' command.`,
    );
  }
};

const validateCode = (code) => {
  if (!code) {
    throw new Error(
      `Verification code is missing. It must be provided as an argument to the
 'confirm-sign-up' command.`,
    );
  }
};

const confirmSignUpHandler = async (commands) => {
  const [_, username, code] = commands;

  try {
```

```
    validateUser(username);
    validateCode(code);
    /**
     * @type {string[]}
     */
    const values = getSecondValuesFromEntries(FILE_USER_POOLS);
    const clientId = values[0];
    validateClient(clientId);
    log(`Confirming user.`);
    await confirmSignUp({ clientId, username, code });
    log(
      `User confirmed. Run 'admin-initiate-auth ${username} <password>' to sign
 in.`,
    );
  } catch (err) {
    log(err);
  }
};

export { confirmSignUpHandler };

const confirmSignUp = ({ clientId, username, code }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmSignUpCommand({
    ClientId: clientId,
    Username: username,
    ConfirmationCode: code,
  });

  return client.send(command);
};

import qrcode from "qrcode-terminal";
import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { adminInitiateAuth } from "../../../actions/admin-initiate-auth.js";
import { associateSoftwareToken } from "../../../actions/associate-software-
token.js";
import { FILE_USER_POOLS } from "./constants.js";
import { getFirstEntry } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";

const handleMfaSetup = async (session, username) => {
  const { SecretCode, Session } = await associateSoftwareToken(session);
```

```
  // Store the Session for use with 'VerifySoftwareToken'.
  process.env.SESSION = Session;

  console.log(
    "Scan this code in your preferred authenticator app, then run 'verify-
software-token' to finish the setup.",
  );
  qrcode.generate(
    `otpauth://totp/${username}?secret=${SecretCode}`,
    { small: true },
    console.log,
  );
};

const handleSoftwareTokenMfa = (session) => {
  // Store the Session for use with 'AdminRespondToAuthChallenge'.
  process.env.SESSION = session;
};

const validateClient = (id) => {
  if (!id) {
    throw new Error(
      `User pool client id is missing. Did you run 'create-user-pool'?`,
    );
  }
};

const validateId = (id) => {
  if (!id) {
    throw new Error(`User pool id is missing. Did you run 'create-user-pool'?`);
  }
};

const validateUser = (username, password) => {
  if (!(username && password)) {
    throw new Error(
      `Username and password must be provided as arguments to the 'admin-
initiate-auth' command.`,
    );
  }
};

const adminInitiateAuthHandler = async (commands) => {
  const [_, username, password] = commands;
```

```
  try {
    validateUser(username, password);

    const [userPoolId, clientId] = getFirstEntry(FILE_USER_POOLS);
    validateId(userPoolId);
    validateClient(clientId);

    log("Signing in.");
    const { ChallengeName, Session } = await adminInitiateAuth({
      clientId,
      userPoolId,
      username,
      password,
    });

    if (ChallengeName === "MFA_SETUP") {
      log("MFA setup is required.");
      return handleMfaSetup(Session, username);
    }

    if (ChallengeName === "SOFTWARE_TOKEN_MFA") {
      handleSoftwareTokenMfa(Session);
      log(`Run 'admin-respond-to-auth-challenge ${username} <totp>'`);
    }
  } catch (err) {
    log(err);
  }
};

export { adminInitiateAuthHandler };

const adminInitiateAuth = ({ clientId, userPoolId, username, password }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new AdminInitiateAuthCommand({
    ClientId: clientId,
    UserPoolId: userPoolId,
    AuthFlow: AuthFlowType.ADMIN_USER_PASSWORD_AUTH,
    AuthParameters: { USERNAME: username, PASSWORD: password },
  });

  return client.send(command);
};
```

```
import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { adminRespondToAuthChallenge } from "../../../actions/admin-respond-to-
auth-challenge.js";
import { getFirstEntry } from "@aws-doc-sdk-examples/lib/utils/util-csv.js";
import { FILE_USER_POOLS } from "./constants.js";

const verifyUsername = (username) => {
  if (!username) {
    throw new Error(
      `Username is missing. It must be provided as an argument to the 'admin-
respond-to-auth-challenge' command.`,
    );
  }
};

const verifyTotp = (totp) => {
  if (!totp) {
    throw new Error(
      `Time-based one-time password (TOTP) is missing. It must be provided as an
 argument to the 'admin-respond-to-auth-challenge' command.`,
    );
  }
};

const storeAccessToken = (token) => {
  process.env.AccessToken = token;
};

const adminRespondToAuthChallengeHandler = async (commands) => {
  const [_, username, totp] = commands;

  try {
    verifyUsername(username);
    verifyTotp(totp);

    const [userPoolId, clientId] = getFirstEntry(FILE_USER_POOLS);
    const session = process.env.SESSION;

    const { AuthenticationResult } = await adminRespondToAuthChallenge({
      clientId,
      userPoolId,
      username,
      totp,
```

```
      session,
    });

    storeAccessToken(AuthenticationResult.AccessToken);

    log("Successfully authenticated.");
  } catch (err) {
    log(err);
  }
};

export { adminRespondToAuthChallengeHandler };

const respondToAuthChallenge = ({
  clientId,
  username,
  session,
  userPoolId,
  code,
}) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new RespondToAuthChallengeCommand({
    ChallengeName: ChallengeNameType.SOFTWARE_TOKEN_MFA,
    ChallengeResponses: {
      SOFTWARE_TOKEN_MFA_CODE: code,
      USERNAME: username,
    },
    ClientId: clientId,
    UserPoolId: userPoolId,
    Session: session,
  });

  return client.send(command);
};

import { log } from "@aws-doc-sdk-examples/lib/utils/util-log.js";
import { verifySoftwareToken } from "../../../actions/verify-software-token.js";

const validateTotp = (totp) => {
  if (!totp) {
    throw new Error(
      `Time-based one-time password (TOTP) must be provided to the 'validate-
software-token' command.`,
```

```
    );
  }
};
const verifySoftwareTokenHandler = async (commands) => {
  const [_, totp] = commands;

  try {
    validateTotp(totp);

    log("Verifying TOTP.");
    await verifySoftwareToken(totp);
    log("TOTP Verified. Run 'admin-initiate-auth' again to sign-in.");
  } catch (err) {
    console.log(err);
  }
};

export { verifySoftwareTokenHandler };

const verifySoftwareToken = (totp) => {
  const client = new CognitoIdentityProviderClient({});

  // The 'Session' is provided in the response to 'AssociateSoftwareToken'.
  const session = process.env.SESSION;

  if (!session) {
    throw new Error(
      "Missing a valid Session. Did you run 'admin-initiate-auth'?",
    );
  }

  const command = new VerifySoftwareTokenCommand({
    Session: session,
    UserCode: totp,
  });

  return client.send(command);
};
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for JavaScript .
  - AdminGetUser

- AdminInitiateAuth
- AdminRespondToAuthChallenge
- AssociateSoftwareToken
- ConfirmDevice
- ConfirmSignUp
- InitiateAuth
- ListUsers
- ResendConfirmationCode
- RespondToAuthChallenge
- SignUp
- VerifySoftwareToken

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más en marcha GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/**
 Before running this Kotlin code example, set up your development environment,
 including your credentials.

 For more information, see the following documentation:
 https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

 TIP: To set up the required user pool, run the AWS Cloud Development
 Kit (AWS CDK) script provided in this GitHub repo at resources/cdk/
cognito_scenario_user_pool_with_mfa.

 This code example performs the following operations:

 1. Invokes the signUp method to sign up a user.
 2. Invokes the adminGetUser method to get the user's confirmation status.
```

```
 3. Invokes the ResendConfirmationCode method if the user requested another code.
 4. Invokes the confirmSignUp method.
 5. Invokes the initiateAuth to sign in. This results in being prompted to
 set up TOTP (time-based one-time password). (The response is “ChallengeName”:
 “MFA_SETUP”).
 6. Invokes the AssociateSoftwareToken method to generate a TOTP MFA private key.
 This can be used with Google Authenticator.
 7. Invokes the VerifySoftwareToken method to verify the TOTP and register for
 MFA.
 8. Invokes the AdminInitiateAuth to sign in again. This results in being
 prompted to submit a TOTP (Response: “ChallengeName”: “SOFTWARE_TOKEN_MFA”).
 9. Invokes the AdminRespondToAuthChallenge to get back a token.
 */

suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <clientId> <poolId>
        Where:
            clientId - The app client Id value that you can get from the AWS CDK
 script.
            poolId - The pool Id that you can get from the AWS CDK script.
    """

    if (args.size != 2) {
        println(usage)
        exitProcess(1)
    }

    val clientId = args[0]
    val poolId = args[1]

    // Use the console to get data from the user.
    println("*** Enter your use name")
    val inOb = Scanner(System.`in`)
    val userName = inOb.nextLine()
    println(userName)

    println("*** Enter your password")
    val password: String = inOb.nextLine()

    println("*** Enter your email")
    val email = inOb.nextLine()
```

```
    println("*** Signing up $userName")
    signUp(clientId, userName, password, email)

    println("*** Getting $userName in the user pool")
    getAdminUser(userName, poolId)

    println("*** Conformation code sent to $userName. Would you like to send a
 new code? (Yes/No)")
    val ans = inOb.nextLine()

    if (ans.compareTo("Yes") == 0) {
        println("*** Sending a new confirmation code")
        resendConfirmationCode(clientId, userName)
    }
    println("*** Enter the confirmation code that was emailed")
    val code = inOb.nextLine()
    confirmSignUp(clientId, code, userName)

    println("*** Rechecking the status of $userName in the user pool")
    getAdminUser(userName, poolId)

    val authResponse = checkAuthMethod(clientId, userName, password, poolId)
    val mySession = authResponse.session
    val newSession = getSecretForAppMFA(mySession)
    println("*** Enter the 6-digit code displayed in Google Authenticator")
    val myCode = inOb.nextLine()

    // Verify the TOTP and register for MFA.
    verifyTOTP(newSession, myCode)
    println("*** Re-enter a 6-digit code displayed in Google Authenticator")
    val mfaCode: String = inOb.nextLine()
    val authResponse1 = checkAuthMethod(clientId, userName, password, poolId)
    val session2 = authResponse1.session
    adminRespondToAuthChallenge(userName, clientId, mfaCode, session2)
}

suspend fun checkAuthMethod(clientIdVal: String, userNameVal: String,
 passwordVal: String, userPoolIdVal: String): AdminInitiateAuthResponse {
    val authParas = mutableMapOf<String, String>()
    authParas["USERNAME"] = userNameVal
    authParas["PASSWORD"] = passwordVal

    val authRequest = AdminInitiateAuthRequest {
        clientId = clientIdVal
```

```
        userPoolId = userPoolIdVal
        authParameters = authParas
        authFlow = AuthFlowType.AdminUserPasswordAuth
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminInitiateAuth(authRequest)
        println("Result Challenge is ${response.challengeName}")
        return response
    }
}

suspend fun resendConfirmationCode(clientIdVal: String?, userNameVal: String?) {
    val codeRequest = ResendConfirmationCodeRequest {
        clientId = clientIdVal
        username = userNameVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.resendConfirmationCode(codeRequest)
        println("Method of delivery is " +
 (response.codeDeliveryDetails?.deliveryMedium))
    }
}

// Respond to an authentication challenge.
suspend fun adminRespondToAuthChallenge(userName: String, clientIdVal: String?,
 mfaCode: String, sessionVal: String?) {
    println("SOFTWARE_TOKEN_MFA challenge is generated")
    val challengeResponsesOb = mutableMapOf<String, String>()
    challengeResponsesOb["USERNAME"] = userName
    challengeResponsesOb["SOFTWARE_TOKEN_MFA_CODE"] = mfaCode

    val adminRespondToAuthChallengeRequest = AdminRespondToAuthChallengeRequest {
        challengeName = ChallengeNameType.SoftwareTokenMfa
        clientId = clientIdVal
        challengeResponses = challengeResponsesOb
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
```

```
        val respondToAuthChallengeResult =
 identityProviderClient.adminRespondToAuthChallenge(adminRespondToAuthChallengeRequest)
        println("respondToAuthChallengeResult.getAuthenticationResult()
 ${respondToAuthChallengeResult.authenticationResult}")
    }
}

// Verify the TOTP and register for MFA.
suspend fun verifyTOTP(sessionVal: String?, codeVal: String?) {
    val tokenRequest = VerifySoftwareTokenRequest {
        userCode = codeVal
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val verifyResponse =
 identityProviderClient.verifySoftwareToken(tokenRequest)
        println("The status of the token is ${verifyResponse.status}")
    }
}

suspend fun getSecretForAppMFA(sessionVal: String?): String? {
    val softwareTokenRequest = AssociateSoftwareTokenRequest {
        session = sessionVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val tokenResponse =
 identityProviderClient.associateSoftwareToken(softwareTokenRequest)
        val secretCode = tokenResponse.secretCode
        println("Enter this token into Google Authenticator")
        println(secretCode)
        return tokenResponse.session
    }
}

suspend fun confirmSignUp(clientIdVal: String?, codeVal: String?, userNameVal:
 String?) {
    val signUpRequest = ConfirmSignUpRequest {
        clientId = clientIdVal
        confirmationCode = codeVal
        username = userNameVal
```

```
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.confirmSignUp(signUpRequest)
        println("$userNameVal  was confirmed")
    }
}

suspend fun getAdminUser(userNameVal: String?, poolIdVal: String?) {
    val userRequest = AdminGetUserRequest {
        username = userNameVal
        userPoolId = poolIdVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        val response = identityProviderClient.adminGetUser(userRequest)
        println("User status ${response.userStatus}")
    }
}

suspend fun signUp(clientIdVal: String?, userNameVal: String?, passwordVal:
 String?, emailVal: String?) {
    val userAttrs = AttributeType {
        name = "email"
        value = emailVal
    }

    val userAttrsList = mutableListOf<AttributeType>()
    userAttrsList.add(userAttrs)
    val signUpRequest = SignUpRequest {
        userAttributes = userAttrsList
        username = userNameVal
        clientId = clientIdVal
        password = passwordVal
    }

    CognitoIdentityProviderClient { region = "us-east-1" }.use
 { identityProviderClient ->
        identityProviderClient.signUp(signUpRequest)
        println("User has been signed up")
    }
}
```

- Para obtener información acerca de la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Kotlin.
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree una clase que incluya las funciones de Amazon Cognito que se utilizan en el escenario.

```
class CognitoIdentityProviderWrapper:
    """Encapsulates Amazon Cognito actions"""

    def __init__(self, cognito_idp_client, user_pool_id, client_id,
 client_secret=None):
        """
```

```
        :param cognito_idp_client: A Boto3 Amazon Cognito Identity Provider
client.
        :param user_pool_id: The ID of an existing Amazon Cognito user pool.
        :param client_id: The ID of a client application registered with the user
pool.
        :param client_secret: The client secret, if the client has a secret.
        """
        self.cognito_idp_client = cognito_idp_client
        self.user_pool_id = user_pool_id
        self.client_id = client_id
        self.client_secret = client_secret


    def _secret_hash(self, user_name):
        """
        Calculates a secret hash from a user name and a client secret.

        :param user_name: The user name to use when calculating the hash.
        :return: The secret hash.
        """
        key = self.client_secret.encode()
        msg = bytes(user_name + self.client_id, "utf-8")
        secret_hash = base64.b64encode(
            hmac.new(key, msg, digestmod=hashlib.sha256).digest()
        ).decode()
        logger.info("Made secret hash for %s: %s.", user_name, secret_hash)
        return secret_hash

    def sign_up_user(self, user_name, password, user_email):
        """
        Signs up a new user with Amazon Cognito. This action prompts Amazon
Cognito
        to send an email to the specified email address. The email contains a
code that
        can be used to confirm the user.

        When the user already exists, the user status is checked to determine
whether
        the user has been confirmed.

        :param user_name: The user name that identifies the new user.
        :param password: The password for the new user.
        :param user_email: The email address for the new user.
        :return: True when the user is already confirmed with Amazon Cognito.
```

```
                 Otherwise, false.
        """
        try:
            kwargs = {
                "ClientId": self.client_id,
                "Username": user_name,
                "Password": password,
                "UserAttributes": [{"Name": "email", "Value": user_email}],
            }
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            response = self.cognito_idp_client.sign_up(**kwargs)
            confirmed = response["UserConfirmed"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "UsernameExistsException":
                response = self.cognito_idp_client.admin_get_user(
                    UserPoolId=self.user_pool_id, Username=user_name
                )
                logger.warning(
                    "User %s exists and is %s.", user_name,
response["UserStatus"]
                )
                confirmed = response["UserStatus"] == "CONFIRMED"
            else:
                logger.error(
                    "Couldn't sign up %s. Here's why: %s: %s",
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        return confirmed


    def resend_confirmation(self, user_name):
        """
        Prompts Amazon Cognito to resend an email with a new confirmation code.

        :param user_name: The name of the user who will receive the email.
        :return: Delivery information about where the email is sent.
        """
        try:
            kwargs = {"ClientId": self.client_id, "Username": user_name}
            if self.client_secret is not None:
```

```
                kwargs["SecretHash"] = self._secret_hash(user_name)
            response = self.cognito_idp_client.resend_confirmation_code(**kwargs)
            delivery = response["CodeDeliveryDetails"]
        except ClientError as err:
            logger.error(
                "Couldn't resend confirmation to %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return delivery


    def confirm_user_sign_up(self, user_name, confirmation_code):
        """
        Confirms a previously created user. A user must be confirmed before they
        can sign in to Amazon Cognito.

        :param user_name: The name of the user to confirm.
        :param confirmation_code: The confirmation code sent to the user's
registered
                                  email address.
        :return: True when the confirmation succeeds.
        """
        try:
            kwargs = {
                "ClientId": self.client_id,
                "Username": user_name,
                "ConfirmationCode": confirmation_code,
            }
            if self.client_secret is not None:
                kwargs["SecretHash"] = self._secret_hash(user_name)
            self.cognito_idp_client.confirm_sign_up(**kwargs)
        except ClientError as err:
            logger.error(
                "Couldn't confirm sign up for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
```

```
            return True


    def list_users(self):
        """
        Returns a list of the users in the current user pool.

        :return: The list of users.
        """
        try:
            response =
 self.cognito_idp_client.list_users(UserPoolId=self.user_pool_id)
            users = response["Users"]
        except ClientError as err:
            logger.error(
                "Couldn't list users for %s. Here's why: %s: %s",
                self.user_pool_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return users


    def start_sign_in(self, user_name, password):
        """
        Starts the sign-in process for a user by using administrator credentials.
        This method of signing in is appropriate for code running on a secure
 server.

        If the user pool is configured to require MFA and this is the first sign-
in
        for the user, Amazon Cognito returns a challenge response to set up an
        MFA application. When this occurs, this function gets an MFA secret from
        Amazon Cognito and returns it to the caller.

        :param user_name: The name of the user to sign in.
        :param password: The user's password.
        :return: The result of the sign-in attempt. When sign-in is successful,
 this
                 returns an access token that can be used to get AWS credentials.
 Otherwise,
                 Amazon Cognito returns a challenge to set up an MFA application,
```

```
                or a challenge to enter an MFA code from a registered MFA
application.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
                "AuthFlow": "ADMIN_USER_PASSWORD_AUTH",
                "AuthParameters": {"USERNAME": user_name, "PASSWORD": password},
            }
            if self.client_secret is not None:
                kwargs["AuthParameters"]["SECRET_HASH"] =
self._secret_hash(user_name)
            response = self.cognito_idp_client.admin_initiate_auth(**kwargs)
            challenge_name = response.get("ChallengeName", None)
            if challenge_name == "MFA_SETUP":
                if (
                    "SOFTWARE_TOKEN_MFA"
                    in response["ChallengeParameters"]["MFAS_CAN_SETUP"]
                ):
                    response.update(self.get_mfa_secret(response["Session"]))
                else:
                    raise RuntimeError(
                        "The user pool requires MFA setup, but the user pool is
not "
                        "configured for TOTP MFA. This example requires TOTP
MFA."
                    )
        except ClientError as err:
            logger.error(
                "Couldn't start sign in for %s. Here's why: %s: %s",
                user_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response


    def get_mfa_secret(self, session):
        """
```

```
        Gets a token that can be used to associate an MFA application with the
user.

        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :return: An MFA token that can be used to set up an MFA application.
        """
        try:
            response =
self.cognito_idp_client.associate_software_token(Session=session)
        except ClientError as err:
            logger.error(
                "Couldn't get MFA secret. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            response.pop("ResponseMetadata", None)
            return response


    def verify_mfa(self, session, user_code):
        """
        Verify a new MFA application that is associated with a user.

        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :param user_code: A code generated by the associated MFA application.
        :return: Status that indicates whether the MFA application is verified.
        """
        try:
            response = self.cognito_idp_client.verify_software_token(
                Session=session, UserCode=user_code
            )
        except ClientError as err:
            logger.error(
                "Couldn't verify MFA. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

```
        else:
            response.pop("ResponseMetadata", None)
            return response


    def respond_to_mfa_challenge(self, user_name, session, mfa_code):
        """
        Responds to a challenge for an MFA code. This completes the second step
of
        a two-factor sign-in. When sign-in is successful, it returns an access
token
        that can be used to get AWS credentials from Amazon Cognito.

        :param user_name: The name of the user who is signing in.
        :param session: Session information returned from a previous call to
initiate
                        authentication.
        :param mfa_code: A code generated by the associated MFA application.
        :return: The result of the authentication. When successful, this contains
an
                 access token for the user.
        """
        try:
            kwargs = {
                "UserPoolId": self.user_pool_id,
                "ClientId": self.client_id,
                "ChallengeName": "SOFTWARE_TOKEN_MFA",
                "Session": session,
                "ChallengeResponses": {
                    "USERNAME": user_name,
                    "SOFTWARE_TOKEN_MFA_CODE": mfa_code,
                },
            }
            if self.client_secret is not None:
                kwargs["ChallengeResponses"]["SECRET_HASH"] = self._secret_hash(
                    user_name
                )
            response =
self.cognito_idp_client.admin_respond_to_auth_challenge(**kwargs)
            auth_result = response["AuthenticationResult"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "ExpiredCodeException":
                logger.warning(
```

```
                    "Your MFA code has expired or has been used already. You
might have "
                    "to wait a few seconds until your app shows you a new code."
                )
            else:
                logger.error(
                    "Couldn't respond to mfa challenge for %s. Here's why: %s:
%s",
                    user_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return auth_result


    def confirm_mfa_device(
        self,
        user_name,
        device_key,
        device_group_key,
        device_password,
        access_token,
        aws_srp,
    ):
        """
        Confirms an MFA device to be tracked by Amazon Cognito. When a device is
        tracked, its key and password can be used to sign in without requiring a
new
        MFA code from the MFA application.

        :param user_name: The user that is associated with the device.
        :param device_key: The key of the device, returned by Amazon Cognito.
        :param device_group_key: The group key of the device, returned by Amazon
Cognito.
        :param device_password: The password that is associated with the device.
        :param access_token: The user's access token.
        :param aws_srp: A class that helps with Secure Remote Password (SRP)
                        calculations. The scenario associated with this example
uses
                        the warrant package.
        :return: True when the user must confirm the device. Otherwise, False.
When
```

```
                False, the device is automatically confirmed and tracked.
        """
        srp_helper = aws_srp.AWSSRP(
            username=user_name,
            password=device_password,
            pool_id="_",
            client_id=self.client_id,
            client_secret=None,
            client=self.cognito_idp_client,
        )
        device_and_pw = f"{device_group_key}{device_key}:{device_password}"
        device_and_pw_hash = aws_srp.hash_sha256(device_and_pw.encode("utf-8"))
        salt = aws_srp.pad_hex(aws_srp.get_random(16))
        x_value = aws_srp.hex_to_long(aws_srp.hex_hash(salt +
device_and_pw_hash))
        verifier = aws_srp.pad_hex(pow(srp_helper.val_g, x_value,
srp_helper.big_n))
        device_secret_verifier_config = {
            "PasswordVerifier": base64.standard_b64encode(
                bytearray.fromhex(verifier)
            ).decode("utf-8"),
            "Salt":
base64.standard_b64encode(bytearray.fromhex(salt)).decode("utf-8"),
        }
        try:
            response = self.cognito_idp_client.confirm_device(
                AccessToken=access_token,
                DeviceKey=device_key,
                DeviceSecretVerifierConfig=device_secret_verifier_config,
            )
            user_confirm = response["UserConfirmationNecessary"]
        except ClientError as err:
            logger.error(
                "Couldn't confirm mfa device %s. Here's why: %s: %s",
                device_key,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return user_confirm


    def sign_in_with_tracked_device(
```

```
        self,
        user_name,
        password,
        device_key,
        device_group_key,
        device_password,
        aws_srp,
    ):
        """
        Signs in to Amazon Cognito as a user who has a tracked device. Signing in
        with a tracked device lets a user sign in without entering a new MFA
 code.

        Signing in with a tracked device requires that the client respond to the
 SRP
        protocol. The scenario associated with this example uses the warrant
 package
        to help with SRP calculations.

        For more information on SRP, see https://en.wikipedia.org/wiki/
Secure_Remote_Password_protocol.

        :param user_name: The user that is associated with the device.
        :param password: The user's password.
        :param device_key: The key of a tracked device.
        :param device_group_key: The group key of a tracked device.
        :param device_password: The password that is associated with the device.
        :param aws_srp: A class that helps with SRP calculations. The scenario
                        associated with this example uses the warrant package.
        :return: The result of the authentication. When successful, this contains
 an
                 access token for the user.
        """
        try:
            srp_helper = aws_srp.AWSSRP(
                username=user_name,
                password=device_password,
                pool_id="_",
                client_id=self.client_id,
                client_secret=None,
                client=self.cognito_idp_client,
            )

            response_init = self.cognito_idp_client.initiate_auth(
```

```
                ClientId=self.client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": user_name,
                    "PASSWORD": password,
                    "DEVICE_KEY": device_key,
                },
            )
            if response_init["ChallengeName"] != "DEVICE_SRP_AUTH":
                raise RuntimeError(
                    f"Expected DEVICE_SRP_AUTH challenge but got
{response_init['ChallengeName']}."
                )

            auth_params = srp_helper.get_auth_params()
            auth_params["DEVICE_KEY"] = device_key
            response_auth = self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_SRP_AUTH",
                ChallengeResponses=auth_params,
            )
            if response_auth["ChallengeName"] != "DEVICE_PASSWORD_VERIFIER":
                raise RuntimeError(
                    f"Expected DEVICE_PASSWORD_VERIFIER challenge but got "
                    f"{response_init['ChallengeName']}."
                )

            challenge_params = response_auth["ChallengeParameters"]
            challenge_params["USER_ID_FOR_SRP"] = device_group_key + device_key
            cr = srp_helper.process_challenge(challenge_params, {"USERNAME":
user_name})
            cr["USERNAME"] = user_name
            cr["DEVICE_KEY"] = device_key
            response_verifier =
self.cognito_idp_client.respond_to_auth_challenge(
                ClientId=self.client_id,
                ChallengeName="DEVICE_PASSWORD_VERIFIER",
                ChallengeResponses=cr,
            )
            auth_tokens = response_verifier["AuthenticationResult"]
        except ClientError as err:
            logger.error(
                "Couldn't start client sign in for %s. Here's why: %s: %s",
                user_name,
```

```
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return auth_tokens
```

Crear una clase que ejecute el escenario. En este ejemplo, también se registra un dispositivo MFA del que Amazon Cognito realiza un seguimiento y se muestra cómo iniciar sesión con una contraseña y la información del dispositivo del que se realiza el seguimiento. Esto evita la necesidad de introducir un nuevo código de MFA.

```
def run_scenario(cognito_idp_client, user_pool_id, client_id):
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Welcome to the Amazon Cognito user signup with MFA demo.")
    print("-" * 88)

    cog_wrapper = CognitoIdentityProviderWrapper(
        cognito_idp_client, user_pool_id, client_id
    )

    user_name = q.ask("Let's sign up a new user. Enter a user name: ",
 q.non_empty)
    password = q.ask("Enter a password for the user: ", q.non_empty)
    email = q.ask("Enter a valid email address that you own: ", q.non_empty)
    confirmed = cog_wrapper.sign_up_user(user_name, password, email)
    while not confirmed:
        print(
            f"User {user_name} requires confirmation. Check {email} for "
            f"a verification code."
        )
        confirmation_code = q.ask("Enter the confirmation code from the email: ")
        if not confirmation_code:
            if q.ask("Do you need another confirmation code (y/n)? ",
 q.is_yesno):
                delivery = cog_wrapper.resend_confirmation(user_name)
                print(
```

```
                    f"Confirmation code sent by {delivery['DeliveryMedium']} "
                    f"to {delivery['Destination']}."
                )
        else:
            confirmed = cog_wrapper.confirm_user_sign_up(user_name,
confirmation_code)
    print(f"User {user_name} is confirmed and ready to use.")
    print("-" * 88)

    print("Let's get a list of users in the user pool.")
    q.ask("Press Enter when you're ready.")
    users = cog_wrapper.list_users()
    if users:
        print(f"Found {len(users)} users:")
        pp(users)
    else:
        print("No users found.")
    print("-" * 88)

    print("Let's sign in and get an access token.")
    auth_tokens = None
    challenge = "ADMIN_USER_PASSWORD_AUTH"
    response = {}
    while challenge is not None:
        if challenge == "ADMIN_USER_PASSWORD_AUTH":
            response = cog_wrapper.start_sign_in(user_name, password)
            challenge = response["ChallengeName"]
        elif response["ChallengeName"] == "MFA_SETUP":
            print("First, we need to set up an MFA application.")
            qr_img = qrcode.make(
                f"otpauth://totp/{user_name}?secret={response['SecretCode']}"
            )
            qr_img.save("qr.png")
            q.ask(
                "Press Enter to see a QR code on your screen. Scan it into an MFA
"
                "application, such as Google Authenticator."
            )
            webbrowser.open("qr.png")
            mfa_code = q.ask(
                "Enter the verification code from your MFA application: ",
q.non_empty
            )
            response = cog_wrapper.verify_mfa(response["Session"], mfa_code)
```

```
            print(f"MFA device setup {response['Status']}")
            print("Now that an MFA application is set up, let's sign in again.")
            print(
                "You might have to wait a few seconds for a new MFA code to
appear in "
                "your MFA application."
            )
            challenge = "ADMIN_USER_PASSWORD_AUTH"
        elif response["ChallengeName"] == "SOFTWARE_TOKEN_MFA":
            auth_tokens = None
            while auth_tokens is None:
                mfa_code = q.ask(
                    "Enter a verification code from your MFA application: ",
q.non_empty
                )
                auth_tokens = cog_wrapper.respond_to_mfa_challenge(
                    user_name, response["Session"], mfa_code
                )
            print(f"You're signed in as {user_name}.")
            print("Here's your access token:")
            pp(auth_tokens["AccessToken"])
            print("And your device information:")
            pp(auth_tokens["NewDeviceMetadata"])
            challenge = None
        else:
            raise Exception(f"Got unexpected challenge
{response['ChallengeName']}")
    print("-" * 88)

    device_group_key = auth_tokens["NewDeviceMetadata"]["DeviceGroupKey"]
    device_key = auth_tokens["NewDeviceMetadata"]["DeviceKey"]
    device_password = base64.standard_b64encode(os.urandom(40)).decode("utf-8")

    print("Let's confirm your MFA device so you don't have re-enter MFA tokens
for it.")
    q.ask("Press Enter when you're ready.")
    cog_wrapper.confirm_mfa_device(
        user_name,
        device_key,
        device_group_key,
        device_password,
        auth_tokens["AccessToken"],
        aws_srp,
    )
```

```
    print(f"Your device {device_key} is confirmed.")
    print("-" * 88)

    print(
        f"Now let's sign in as {user_name} from your confirmed device
 {device_key}.\n"
        f"Because this device is tracked by Amazon Cognito, you won't have to re-
enter an MFA code."
    )
    q.ask("Press Enter when ready.")
    auth_tokens = cog_wrapper.sign_in_with_tracked_device(
        user_name, password, device_key, device_group_key, device_password,
 aws_srp
    )
    print("You're signed in. Your access token is:")
    pp(auth_tokens["AccessToken"])
    print("-" * 88)

    print("Don't forget to delete your user pool when you're done with this
 example.")
    print("\nThanks for watching!")
    print("-" * 88)


def main():
    parser = argparse.ArgumentParser(
        description="Shows how to sign up a new user with Amazon Cognito and
 associate "
        "the user with an MFA application for multi-factor authentication."
    )
    parser.add_argument(
        "user_pool_id", help="The ID of the user pool to use for the example."
    )
    parser.add_argument(
        "client_id", help="The ID of the client application to use for the
 example."
    )
    args = parser.parse_args()
    try:
        run_scenario(boto3.client("cognito-idp"), args.user_pool_id,
 args.client_id)
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

```
if __name__ == "__main__":
    main()
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Python (Boto3).
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escriba datos de actividad personalizados con una función Lambda tras la autenticación de usuarios de Amazon Cognito mediante un SDK AWS

En el siguiente ejemplo de código, se muestra cómo escribir datos de actividad personalizados con una función de Lambda tras la autenticación de usuarios de Amazon Cognito.

- Utilice las funciones de administrador para añadir un usuario a un grupo de usuarios.
- Configure un grupo de usuarios para que llame a una función de Lambda para el desencadenador `PostAuthentication`.
- Inicie sesión con el nuevo usuario en Amazon Cognito.

- La función Lambda escribe información personalizada en los CloudWatch registros y en una tabla de DynamoDB.
- Obtenga y exhiba los datos personalizados de la tabla de DynamoDB y, a continuación, elimine los recursos.

Go

SDK para Go V2

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecutar un escenario interactivo en un símbolo del sistema.

```
// ActivityLog separates the steps of this scenario into individual functions so
 that
// they are simpler to read and understand.
type ActivityLog struct {
 helper       IScenarioHelper
 questioner   demotools.IQuestioner
 resources    Resources
 cognitoActor *actions.CognitoActions
}

// NewActivityLog constructs a new activity log runner.
func NewActivityLog(sdkConfig aws.Config, questioner demotools.IQuestioner,
 helper IScenarioHelper) ActivityLog {
 scenario := ActivityLog{
  helper:       helper,
  questioner:   questioner,
  resources:    Resources{},
  cognitoActor: &actions.CognitoActions{CognitoClient:
 cognitoidentityprovider.NewFromConfig(sdkConfig)},
 }
 scenario.resources.init(scenario.cognitoActor, questioner)
 return scenario
}
```

```
// AddUserToPool selects a user from the known users table and uses administrator
 credentials to add the user to the user pool.
func (runner *ActivityLog) AddUserToPool(userPoolId string, tableName string)
 (string, string) {
 log.Println("To facilitate this example, let's add a user to the user pool using
 administrator privileges.")
 users, err := runner.helper.GetKnownUsers(tableName)
 if err != nil {
  panic(err)
 }
 user := users.Users[0]
 log.Printf("Adding known user %v to the user pool.\n", user.UserName)
 err = runner.cognitoActor.AdminCreateUser(userPoolId, user.UserName,
 user.UserEmail)
 if err != nil {
  panic(err)
 }
 pwSet := false
 password := runner.questioner.AskPassword("\nEnter a password that has at least
 eight characters, uppercase, lowercase, numbers and symbols.\n"+
  "(the password will not display as you type):", 8)
 for !pwSet {
  log.Printf("\nSetting password for user '%v'.\n", user.UserName)
  err = runner.cognitoActor.AdminSetUserPassword(userPoolId, user.UserName,
 password)
  if err != nil {
   var invalidPassword *types.InvalidPasswordException
   if errors.As(err, &invalidPassword) {
    password = runner.questioner.AskPassword("\nEnter another password:", 8)
   } else {
    panic(err)
   }
  } else {
   pwSet = true
  }
 }

 log.Println(strings.Repeat("-", 88))

 return user.UserName, password
}
```

```
// AddActivityLogTrigger adds a Lambda handler as an invocation target for the
 PostAuthentication trigger.
func (runner *ActivityLog) AddActivityLogTrigger(userPoolId string,
 activityLogArn string) {
 log.Println("Let's add a Lambda function to handle the PostAuthentication
 trigger from Cognito.\n" +
  "This trigger happens after a user is authenticated, and lets your function
 take action, such as logging\n" +
  "the outcome.")
 err := runner.cognitoActor.UpdateTriggers(
  userPoolId,
  actions.TriggerInfo{Trigger: actions.PostAuthentication, HandlerArn:
 aws.String(activityLogArn)})
 if err != nil {
  panic(err)
 }
 runner.resources.triggers = append(runner.resources.triggers,
 actions.PostAuthentication)
 log.Printf("Lambda function %v added to user pool %v to handle
 PostAuthentication Cognito trigger.\n",
  activityLogArn, userPoolId)

 log.Println(strings.Repeat("-", 88))
}

// SignInUser signs in as the specified user.
func (runner *ActivityLog) SignInUser(clientId string, userName string, password
 string) {
 log.Printf("Now we'll sign in user %v and check the results in the logs and the
 DynamoDB table.", userName)
 runner.questioner.Ask("Press Enter when you're ready.")
 authResult, err := runner.cognitoActor.SignIn(clientId, userName, password)
 if err != nil {
  panic(err)
 }
 log.Println("Sign in successful.",
  "The PostAuthentication Lambda handler writes custom information to CloudWatch
 Logs.")

 runner.resources.userAccessTokens = append(runner.resources.userAccessTokens,
 *authResult.AccessToken)
}
```

```
// GetKnownUserLastLogin gets the login info for a user from the Amazon DynamoDB
 table and displays it.
func (runner *ActivityLog) GetKnownUserLastLogin(tableName string, userName
 string) {
 log.Println("The PostAuthentication handler also writes login data to the
 DynamoDB table.")
 runner.questioner.Ask("Press Enter when you're ready to continue.")
 users, err := runner.helper.GetKnownUsers(tableName)
 if err != nil {
  panic(err)
 }
 for _, user := range users.Users {
  if user.UserName == userName {
   log.Println("The last login info for the user in the known users table is:")
   log.Printf("\t%+v", *user.LastLogin)
  }
 }
 log.Println(strings.Repeat("-", 88))
}

// Run runs the scenario.
func (runner *ActivityLog) Run(stackName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   runner.resources.Cleanup()
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Printf("Welcome\n")

 log.Println(strings.Repeat("-", 88))

 stackOutputs, err := runner.helper.GetStackOutputs(stackName)
 if err != nil {
  panic(err)
 }
 runner.resources.userPoolId = stackOutputs["UserPoolId"]
 runner.helper.PopulateUserTable(stackOutputs["TableName"])
 userName, password := runner.AddUserToPool(stackOutputs["UserPoolId"],
 stackOutputs["TableName"])
```

```
 runner.AddActivityLogTrigger(stackOutputs["UserPoolId"],
 stackOutputs["ActivityLogFunctionArn"])
 runner.SignInUser(stackOutputs["UserPoolClientId"], userName, password)
 runner.helper.ListRecentLogEvents(stackOutputs["ActivityLogFunction"])
 runner.GetKnownUserLastLogin(stackOutputs["TableName"], userName)

 runner.resources.Cleanup()

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Controle el desencadenador `PostAuthentication` con una función de Lambda.

```
const TABLE_NAME = "TABLE_NAME"

// LoginInfo defines structured login data that can be marshalled to a DynamoDB
 format.
type LoginInfo struct {
 UserPoolId string `dynamodbav:"UserPoolId"`
 ClientId   string `dynamodbav:"ClientId"`
 Time       string `dynamodbav:"Time"`
}

// UserInfo defines structured user data that can be marshalled to a DynamoDB
 format.
type UserInfo struct {
 UserName  string    `dynamodbav:"UserName"`
 UserEmail string    `dynamodbav:"UserEmail"`
 LastLogin LoginInfo `dynamodbav:"LastLogin"`
}

// GetKey marshals the user email value to a DynamoDB key format.
func (user UserInfo) GetKey() map[string]dynamodbtypes.AttributeValue {
 userEmail, err := attributevalue.Marshal(user.UserEmail)
 if err != nil {
  panic(err)
 }
 return map[string]dynamodbtypes.AttributeValue{"UserEmail": userEmail}
```

```
}

type handler struct {
 dynamoClient *dynamodb.Client
}

// HandleRequest handles the PostAuthentication event by writing custom data to
 the logs and
// to an Amazon DynamoDB table.
func (h *handler) HandleRequest(ctx context.Context,
 event events.CognitoEventUserPoolsPostAuthentication)
 (events.CognitoEventUserPoolsPostAuthentication, error) {
 log.Printf("Received post authentication trigger from %v for user '%v'",
 event.TriggerSource, event.UserName)
 tableName := os.Getenv(TABLE_NAME)
 user := UserInfo{
  UserName:  event.UserName,
  UserEmail: event.Request.UserAttributes["email"],
  LastLogin: LoginInfo{
   UserPoolId: event.UserPoolID,
   ClientId:   event.CallerContext.ClientID,
   Time:       time.Now().Format(time.UnixDate),
  },
 }
 // Write to CloudWatch Logs.
 fmt.Printf("%#v", user)

 // Also write to an external system. This examples uses DynamoDB to demonstrate.
 userMap, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshal to DynamoDB map. Here's why: %v\n", err)
 } else if len(userMap) == 0 {
  log.Printf("User info marshaled to an empty map.")
 } else {
  _, err := h.dynamoClient.PutItem(ctx, &dynamodb.PutItemInput{
   Item:      userMap,
   TableName: aws.String(tableName),
  })
  if err != nil {
   log.Printf("Couldn't write to DynamoDB. Here's why: %v\n", err)
  } else {
   log.Printf("Wrote user info to DynamoDB table %v.\n", tableName)
  }
 }
```

```
 return event, nil
}

func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  log.Panicln(err)
 }
 h := handler{
  dynamoClient: dynamodb.NewFromConfig(sdkConfig),
 }
 lambda.Start(h.HandleRequest)
}
```

Cree una estructura que lleve a cabo las tareas habituales.

```
// IScenarioHelper defines common functions used by the workflows in this
 example.
type IScenarioHelper interface {
 Pause(secs int)
 GetStackOutputs(stackName string) (actions.StackOutputs, error)
 PopulateUserTable(tableName string)
 GetKnownUsers(tableName string) (actions.UserList, error)
 AddKnownUser(tableName string, user actions.User)
 ListRecentLogEvents(functionName string)
}

// ScenarioHelper contains AWS wrapper structs used by the workflows in this
 example.
type ScenarioHelper struct {
 questioner  demotools.IQuestioner
 dynamoActor *actions.DynamoActions
 cfnActor    *actions.CloudFormationActions
 cwlActor    *actions.CloudWatchLogsActions
 isTestRun   bool
}

// NewScenarioHelper constructs a new scenario helper.
```

```
func NewScenarioHelper(sdkConfig aws.Config, questioner demotools.IQuestioner)
 ScenarioHelper {
 scenario := ScenarioHelper{
  questioner:  questioner,
  dynamoActor: &actions.DynamoActions{DynamoClient:
 dynamodb.NewFromConfig(sdkConfig)},
  cfnActor:    &actions.CloudFormationActions{CfnClient:
 cloudformation.NewFromConfig(sdkConfig)},
  cwlActor:    &actions.CloudWatchLogsActions{CwlClient:
 cloudwatchlogs.NewFromConfig(sdkConfig)},
 }
 return scenario
}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 if !helper.isTestRun {
  time.Sleep(time.Duration(secs) * time.Second)
 }
}

// GetStackOutputs gets the outputs from the specified CloudFormation stack in a
 structured format.
func (helper ScenarioHelper) GetStackOutputs(stackName string)
 (actions.StackOutputs, error) {
 return helper.cfnActor.GetOutputs(stackName), nil
}

// PopulateUserTable fills the known user table with example data.
func (helper ScenarioHelper) PopulateUserTable(tableName string) {
 log.Printf("First, let's add some users to the DynamoDB %v table we'll use for
 this example.\n", tableName)
 err := helper.dynamoActor.PopulateTable(tableName)
 if err != nil {
  panic(err)
 }
}

// GetKnownUsers gets the users from the known users table in a structured
 format.
func (helper ScenarioHelper) GetKnownUsers(tableName string) (actions.UserList,
 error) {
 knownUsers, err := helper.dynamoActor.Scan(tableName)
 if err != nil {
```

```
  log.Printf("Couldn't get known users from table %v. Here's why: %v\n",
 tableName, err)
 }
 return knownUsers, err
}

// AddKnownUser adds a user to the known users table.
func (helper ScenarioHelper) AddKnownUser(tableName string, user actions.User) {
 log.Printf("Adding user '%v' with email '%v' to the DynamoDB known users
 table...\n",
  user.UserName, user.UserEmail)
 err := helper.dynamoActor.AddUser(tableName, user)
 if err != nil {
  panic(err)
 }
}

// ListRecentLogEvents gets the most recent log stream and events for the
 specified Lambda function and displays them.
func (helper ScenarioHelper) ListRecentLogEvents(functionName string) {
 log.Println("Waiting a few seconds to let Lambda write to CloudWatch Logs...")
 helper.Pause(10)
 log.Println("Okay, let's check the logs to find what's happened recently with
 your Lambda function.")
 logStream, err := helper.cwlActor.GetLatestLogStream(functionName)
 if err != nil {
  panic(err)
 }
 log.Printf("Getting some recent events from log stream %v\n",
 *logStream.LogStreamName)
 events, err := helper.cwlActor.GetLogEvents(functionName,
 *logStream.LogStreamName, 10)
 if err != nil {
  panic(err)
 }
 for _, event := range events {
  log.Printf("\t%v", *event.Message)
 }
 log.Println(strings.Repeat("-", 88))
}
```

Cree una estructura que ajuste las acciones de Amazon Cognito.

```
type CognitoActions struct {
 CognitoClient *cognitoidentityprovider.Client
}



// Trigger and TriggerInfo define typed data for updating an Amazon Cognito
 trigger.
type Trigger int

const (
 PreSignUp Trigger = iota
 UserMigration
 PostAuthentication
)

type TriggerInfo struct {
 Trigger    Trigger
 HandlerArn *string
}

// UpdateTriggers adds or removes Lambda triggers for a user pool. When a trigger
 is specified with a `nil` value,
// it is removed from the user pool.
func (actor CognitoActions) UpdateTriggers(userPoolId string,
 triggers ...TriggerInfo) error {
 output, err := actor.CognitoClient.DescribeUserPool(context.TODO(),
 &cognitoidentityprovider.DescribeUserPoolInput{
  UserPoolId: aws.String(userPoolId),
 })
 if err != nil {
  log.Printf("Couldn't get info about user pool %v. Here's why: %v\n",
 userPoolId, err)
  return err
 }
 lambdaConfig := output.UserPool.LambdaConfig
 for _, trigger := range triggers {
  switch trigger.Trigger {
  case PreSignUp:
   lambdaConfig.PreSignUp = trigger.HandlerArn
```

```
  case UserMigration:
   lambdaConfig.UserMigration = trigger.HandlerArn
  case PostAuthentication:
   lambdaConfig.PostAuthentication = trigger.HandlerArn
  }
 }
 _, err = actor.CognitoClient.UpdateUserPool(context.TODO(),
 &cognitoidentityprovider.UpdateUserPoolInput{
  UserPoolId:   aws.String(userPoolId),
  LambdaConfig: lambdaConfig,
 })
 if err != nil {
  log.Printf("Couldn't update user pool %v. Here's why: %v\n", userPoolId, err)
 }
 return err
}



// SignUp signs up a user with Amazon Cognito.
func (actor CognitoActions) SignUp(clientId string, userName string, password
 string, userEmail string) (bool, error) {
 confirmed := false
 output, err := actor.CognitoClient.SignUp(context.TODO(),
 &cognitoidentityprovider.SignUpInput{
  ClientId: aws.String(clientId),
  Password: aws.String(password),
  Username: aws.String(userName),
  UserAttributes: []types.AttributeType{
   {Name: aws.String("email"), Value: aws.String(userEmail)},
  },
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't sign up user %v. Here's why: %v\n", userName, err)
  }
 } else {
  confirmed = output.UserConfirmed
 }
 return confirmed, err
}
```

```
// SignIn signs in a user to Amazon Cognito using a username and password
 authentication flow.
func (actor CognitoActions) SignIn(clientId string, userName string, password
 string) (*types.AuthenticationResultType, error) {
 var authResult *types.AuthenticationResultType
 output, err := actor.CognitoClient.InitiateAuth(context.TODO(),
 &cognitoidentityprovider.InitiateAuthInput{
  AuthFlow:       "USER_PASSWORD_AUTH",
  ClientId:       aws.String(clientId),
  AuthParameters: map[string]string{"USERNAME": userName, "PASSWORD": password},
 })
 if err != nil {
  var resetRequired *types.PasswordResetRequiredException
  if errors.As(err, &resetRequired) {
   log.Println(*resetRequired.Message)
  } else {
   log.Printf("Couldn't sign in user %v. Here's why: %v\n", userName, err)
  }
 } else {
  authResult = output.AuthenticationResult
 }
 return authResult, err
}



// ForgotPassword starts a password recovery flow for a user. This flow typically
 sends a confirmation code
// to the user's configured notification destination, such as email.
func (actor CognitoActions) ForgotPassword(clientId string, userName string)
 (*types.CodeDeliveryDetailsType, error) {
 output, err := actor.CognitoClient.ForgotPassword(context.TODO(),
 &cognitoidentityprovider.ForgotPasswordInput{
  ClientId: aws.String(clientId),
  Username: aws.String(userName),
 })
 if err != nil {
  log.Printf("Couldn't start password reset for user '%v'. Here;s why: %v\n",
 userName, err)
 }
 return output.CodeDeliveryDetails, err
```

```
}



// ConfirmForgotPassword confirms a user with a confirmation code and a new
 password.
func (actor CognitoActions) ConfirmForgotPassword(clientId string, code string,
 userName string, password string) error {
 _, err := actor.CognitoClient.ConfirmForgotPassword(context.TODO(),
 &cognitoidentityprovider.ConfirmForgotPasswordInput{
  ClientId:         aws.String(clientId),
  ConfirmationCode: aws.String(code),
  Password:         aws.String(password),
  Username:         aws.String(userName),
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't confirm user %v. Here's why: %v", userName, err)
  }
 }
 return err
}



// DeleteUser removes a user from the user pool.
func (actor CognitoActions) DeleteUser(userAccessToken string) error {
 _, err := actor.CognitoClient.DeleteUser(context.TODO(),
 &cognitoidentityprovider.DeleteUserInput{
  AccessToken: aws.String(userAccessToken),
 })
 if err != nil {
  log.Printf("Couldn't delete user. Here's why: %v\n", err)
 }
 return err
}



// AdminCreateUser uses administrator credentials to add a user to a user pool.
 This method leaves the user
```

```
// in a state that requires they enter a new password next time they sign in.
func (actor CognitoActions) AdminCreateUser(userPoolId string, userName string,
 userEmail string) error {
 _, err := actor.CognitoClient.AdminCreateUser(context.TODO(),
 &cognitoidentityprovider.AdminCreateUserInput{
  UserPoolId:     aws.String(userPoolId),
  Username:       aws.String(userName),
  MessageAction:  types.MessageActionTypeSuppress,
  UserAttributes: []types.AttributeType{{Name: aws.String("email"), Value:
 aws.String(userEmail)}},
 })
 if err != nil {
  var userExists *types.UsernameExistsException
  if errors.As(err, &userExists) {
   log.Printf("User %v already exists in the user pool.", userName)
   err = nil
  } else {
   log.Printf("Couldn't create user %v. Here's why: %v\n", userName, err)
  }
 }
 return err
}



// AdminSetUserPassword uses administrator credentials to set a password for a
 user without requiring a
// temporary password.
func (actor CognitoActions) AdminSetUserPassword(userPoolId string, userName
 string, password string) error {
 _, err := actor.CognitoClient.AdminSetUserPassword(context.TODO(),
 &cognitoidentityprovider.AdminSetUserPasswordInput{
  Password:   aws.String(password),
  UserPoolId: aws.String(userPoolId),
  Username:   aws.String(userName),
  Permanent:  true,
 })
 if err != nil {
  var invalidPassword *types.InvalidPasswordException
  if errors.As(err, &invalidPassword) {
   log.Println(*invalidPassword.Message)
  } else {
   log.Printf("Couldn't set password for user %v. Here's why: %v\n", userName,
 err)
```

```
  }
 }
 return err
}
```

Cree una estructura que ajuste las acciones de DynamoDB.

```
// DynamoActions encapsulates the Amazon Simple Notification Service (Amazon SNS)
 actions
// used in the examples.
type DynamoActions struct {
 DynamoClient *dynamodb.Client
}

// User defines structured user data.
type User struct {
 UserName  string
 UserEmail string
 LastLogin *LoginInfo `dynamodbav:",omitempty"`
}

// LoginInfo defines structured custom login data.
type LoginInfo struct {
 UserPoolId string
 ClientId   string
 Time       string
}

// UserList defines a list of users.
type UserList struct {
 Users []User
}

// UserNameList returns the usernames contained in a UserList as a list of
 strings.
func (users *UserList) UserNameList() []string {
 names := make([]string, len(users.Users))
 for i := 0; i < len(users.Users); i++ {
  names[i] = users.Users[i].UserName
 }
```

```
 return names
}

// PopulateTable adds a set of test users to the table.
func (actor DynamoActions) PopulateTable(tableName string) error {
 var err error
 var item map[string]types.AttributeValue
 var writeReqs []types.WriteRequest
 for i := 1; i < 4; i++ {
  item, err = attributevalue.MarshalMap(User{UserName: fmt.Sprintf("test_user_
%v", i), UserEmail: fmt.Sprintf("test_email_%v@example.com", i)})
  if err != nil {
   log.Printf("Couldn't marshall user into DynamoDB format. Here's why: %v\n",
 err)
   return err
  }
  writeReqs = append(writeReqs, types.WriteRequest{PutRequest:
 &types.PutRequest{Item: item}})
 }
 _, err = actor.DynamoClient.BatchWriteItem(context.TODO(),
 &dynamodb.BatchWriteItemInput{
  RequestItems: map[string][]types.WriteRequest{tableName: writeReqs},
 })
 if err != nil {
  log.Printf("Couldn't populate table %v with users. Here's why: %v\n",
 tableName, err)
 }
 return err
}

// Scan scans the table for all items.
func (actor DynamoActions) Scan(tableName string) (UserList, error) {
 var userList UserList
 output, err := actor.DynamoClient.Scan(context.TODO(), &dynamodb.ScanInput{
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't scan table %v for items. Here's why: %v\n", tableName,
 err)
 } else {
  err = attributevalue.UnmarshalListOfMaps(output.Items, &userList.Users)
  if err != nil {
   log.Printf("Couldn't unmarshal items into users. Here's why: %v\n", err)
  }
```

```
 }
 return userList, err
}

// AddUser adds a user item to a table.
func (actor DynamoActions) AddUser(tableName string, user User) error {
 userItem, err := attributevalue.MarshalMap(user)
 if err != nil {
  log.Printf("Couldn't marshall user to item. Here's why: %v\n", err)
 }
 _, err = actor.DynamoClient.PutItem(context.TODO(), &dynamodb.PutItemInput{
  Item:      userItem,
  TableName: aws.String(tableName),
 })
 if err != nil {
  log.Printf("Couldn't put item in table %v. Here's why: %v", tableName, err)
 }
 return err
}
```

Crea una estructura que agrupe las acciones de CloudWatch Logs.

```
type CloudWatchLogsActions struct {
 CwlClient *cloudwatchlogs.Client
}

// GetLatestLogStream gets the most recent log stream for a Lambda function.
func (actor CloudWatchLogsActions) GetLatestLogStream(functionName string)
 (types.LogStream, error) {
 var logStream types.LogStream
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.DescribeLogStreams(context.TODO(),
 &cloudwatchlogs.DescribeLogStreamsInput{
  Descending:   aws.Bool(true),
  Limit:        aws.Int32(1),
  LogGroupName: aws.String(logGroupName),
  OrderBy:      types.OrderByLastEventTime,
 })
 if err != nil {
```

```
  log.Printf("Couldn't get log streams for log group %v. Here's why: %v\n",
 logGroupName, err)
 } else {
  logStream = output.LogStreams[0]
 }
 return logStream, err
}

// GetLogEvents gets the most recent eventCount events from the specified log
 stream.
func (actor CloudWatchLogsActions) GetLogEvents(functionName string,
 logStreamName string, eventCount int32) (
 []types.OutputLogEvent, error) {
 var events []types.OutputLogEvent
 logGroupName := fmt.Sprintf("/aws/lambda/%s", functionName)
 output, err := actor.CwlClient.GetLogEvents(context.TODO(),
 &cloudwatchlogs.GetLogEventsInput{
  LogStreamName: aws.String(logStreamName),
  Limit:         aws.Int32(eventCount),
  LogGroupName:  aws.String(logGroupName),
 })
 if err != nil {
  log.Printf("Couldn't get log event for log stream %v. Here's why: %v\n",
 logStreamName, err)
 } else {
  events = output.Events
 }
 return events, err
}
```

Crea una estructura que agrupe las acciones. AWS CloudFormation

```
// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
 CfnClient *cloudformation.Client
}
```

```
// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
 structured format.
func (actor CloudFormationActions) GetOutputs(stackName string) StackOutputs {
 output, err := actor.CfnClient.DescribeStacks(context.TODO(),
 &cloudformation.DescribeStacksInput{
  StackName: aws.String(stackName),
 })
 if err != nil || len(output.Stacks) == 0 {
  log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
 stackName, err)
 }
 stackOutputs := StackOutputs{}
 for _, out := range output.Stacks[0].Outputs {
  stackOutputs[*out.OutputKey] = *out.OutputValue
 }
 return stackOutputs
}
```

Eliminación de recursos.

```

// Resources keeps track of AWS resources created during an example and handles
// cleanup when the example finishes.
type Resources struct {
 userPoolId       string
 userAccessTokens []string
 triggers         []actions.Trigger

 cognitoActor *actions.CognitoActions
 questioner   demotools.IQuestioner
}

func (resources *Resources) init(cognitoActor *actions.CognitoActions, questioner
 demotools.IQuestioner) {
 resources.userAccessTokens = []string{}
 resources.triggers = []actions.Trigger{}
 resources.cognitoActor = cognitoActor
 resources.questioner = questioner
}

// Cleanup deletes all AWS resources created during an example.
```

```
func (resources *Resources) Cleanup() {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong during cleanup.\n%v\n", r)
   log.Println("Use the AWS Management Console to remove any remaining resources
 \n" +
    "that were created for this scenario.")
  }
 }()

 wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
 resources that were created "+
  "during this demo (y/n)?", "y")
 if wantDelete {
  for _, accessToken := range resources.userAccessTokens {
   err := resources.cognitoActor.DeleteUser(accessToken)
   if err != nil {
    log.Println("Couldn't delete user during cleanup.")
    panic(err)
   }
   log.Println("Deleted user.")
  }
  triggerList := make([]actions.TriggerInfo, len(resources.triggers))
  for i := 0; i < len(resources.triggers); i++ {
   triggerList[i] = actions.TriggerInfo{Trigger: resources.triggers[i],
 HandlerArn: nil}
  }
  err := resources.cognitoActor.UpdateTriggers(resources.userPoolId,
 triggerList...)
  if err != nil {
   log.Println("Couldn't update Cognito triggers during cleanup.")
   panic(err)
  }
  log.Println("Removed Cognito triggers from user pool.")
 } else {
  log.Println("Be sure to remove resources when you're done with them to avoid
 unexpected charges!")
 }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK for Go .
  - AdminCreateUser
  - AdminSetUserPassword
  - DeleteUser
  - InitiateAuth
  - UpdateUserPool

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de este servicio con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código para Amazon Cognito Sync mediante SDK AWS

Los siguientes ejemplos de código muestran cómo utilizar Amazon Cognito Sync con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Acciones para Amazon Cognito Sync mediante SDK AWS
  - Úselo ListIdentityPoolUsage con un AWS SDK o CLI

Acciones para Amazon Cognito Sync mediante SDK AWS

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Cognito Sync con AWS los SDK. Estos fragmentos llaman a la API de sincronización de Amazon Cognito y son fragmentos de código de programas más grandes que se deben ejecutar en contexto. Cada

ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para consultar la lista completa, vea la Amazon Cognito Sync API Reference (Referencia de la API de Amazon Cognito Sync).

Ejemplos

- Úselo ListIdentityPoolUsage con un AWS SDK o CLI

Úselo `ListIdentityPoolUsage` con un AWS SDK o CLI

En el siguiente ejemplo de código, se muestra cómo usar `ListIdentityPoolUsage`.

Rust

SDK para Rust

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
async fn show_pools(client: &Client) -> Result<(), Error> {
    let response = client
        .list_identity_pool_usage()
        .max_results(10)
        .send()
        .await?;

    let pools = response.identity_pool_usages();
    println!("Identity pools:");

    for pool in pools {
        println!(
            "  Identity pool ID:    {}",
            pool.identity_pool_id().unwrap_or_default()
        );
        println!(
```

```
            "  Data storage:        {}",
            pool.data_storage().unwrap_or_default()
        );
        println!(
            "  Sync sessions count: {}",
            pool.sync_sessions_count().unwrap_or_default()
        );
        println!(
            "  Last modified:       {}",
            pool.last_modified_date().unwrap().to_chrono_utc()?
        );
        println!();
    }

    println!("Next token: {}", response.next_token().unwrap_or_default());

    Ok(())
}
```

- Para obtener más información sobre la API, consulta ListIdentityPoolUsagela referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Prácticas recomendadas de aplicaciones de varios inquilinos

Los grupos de usuarios de Amazon Cognito funcionan con aplicaciones de varios inquilinos que generan un volumen de solicitudes que deben permanecer dentro de las cuotas de Amazon Cognito. Para ampliar esta capacidad cuando su base de clientes crezca, puede adquirir una capacidad de cuota adicional.

> **Note**
>
> Las cuotas de Amazon Cognito se aplican por Cuenta de AWS y. Región de AWS Estas cuotas se comparten entre todos los inquilinos de la aplicación. Revise las cuotas de servicio de Amazon Cognito y asegúrese de que la cuota cumpla con el volumen y el número de inquilinos esperados en su solicitud.

En esta sección se describen los métodos que puede implementar para separar los inquilinos entre los recursos de Amazon Cognito de la misma región y. Cuenta de AWS También puede dividir a sus inquilinos en más de una Cuenta de AWS o más regiones y asignar a cada uno de ellos su propia cuota. Otras ventajas de la tenencia multirregional incluyen el mayor nivel de aislamiento posible, el menor tiempo de tránsito de la red para los usuarios distribuidos por todo el mundo y la adhesión a los modelos de distribución existentes en su organización.

La tenencia múltiple de una sola región también puede suponer ventajas para sus clientes y administradores.

En la siguiente lista se describen algunas de las ventajas de la tenencia múltiple con recursos compartidos.

Ventajas de la tenencia múltiple

Directorio de usuarios común

La multitenencia admite modelos en los que los clientes tienen cuentas en más de una aplicación. Puede vincular identidades de proveedores externos en un único perfil de grupo de usuarios coherente. En los casos en que los perfiles de usuario son exclusivos de su arrendatario, cualquier estrategia de arrendamiento múltiple con un único grupo de usuarios tiene un punto de entrada para la administración de usuarios.

Seguridad común

En un grupo de usuarios compartido, puede crear un único estándar de seguridad y aplicar la misma seguridad avanzada, autenticación multifactor (MFA) AWS WAFy estándares a todos los inquilinos. Como una ACL AWS WAF web debe estar en el mismo lugar Región de AWS que el recurso al que se asocia, la opción de arrendamiento múltiple ofrece acceso compartido a un recurso complejo. Si desea mantener una configuración de seguridad uniforme en las aplicaciones de Amazon Cognito de varias regiones, debe aplicar estándares operativos que repliquen la configuración entre los recursos.

Personalización común

Puede personalizar los grupos de usuarios y los grupos de identidades con AWS Lambda. La configuración de los activadores de Lambda en los grupos de usuarios y los eventos de Amazon Cognito en los grupos de identidades puede resultar compleja. Las funciones de Lambda deben estar en el Región de AWS mismo grupo de usuarios o grupo de identidades. Las funciones Lambda compartidas pueden aplicar estándares para los flujos de autenticación personalizados, la migración de usuarios, la generación de tokens y otras funciones dentro de una región.

Mensajería común

Amazon Simple Notification Service (Amazon SNS) requiere una configuración adicional en una región antes de poder enviar mensajes SMS a sus usuarios. Puede enviar mensajes de correo electrónico con identidades y dominios verificados por Amazon Simple Email Service (Amazon SES) que se encuentren dentro de una región.

Con la multitenencia, puede compartir esta configuración y los gastos de mantenimiento entre todos sus inquilinos. Dado que Amazon SNS y Amazon SES no están disponibles en todas partes Regiones de AWS, es necesario tener en cuenta la posibilidad de dividir los recursos entre regiones.

Cuando utiliza proveedores de mensajería personalizados, obtiene la personalización habitual de una sola función de Lambda para gestionar la entrega de mensajes.

La interfaz de usuario alojada establece una cookie de sesión en el navegador para que reconozca a un usuario que ya se ha autenticado. Al autenticar a los usuarios locales en un grupo de usuarios, su cookie de sesión los autentica para todos los clientes de aplicaciones del mismo grupo de usuarios. Un usuario local existe exclusivamente en el directorio del grupo de usuarios sin federación a través de un IdP externo. La cookie de sesión es válida durante una hora. No puede cambiar la duración de la sesión de la cookie.

Hay dos formas de impedir el inicio de sesión en todos los clientes de aplicaciones con una cookie de sesión de interfaz de usuario alojada.

- Separe a los usuarios en grupos de usuarios por inquilino.
- Sustituya el inicio de sesión de la interfaz de usuario alojada por el inicio de sesión en la API de los grupos de usuarios de Amazon Cognito.

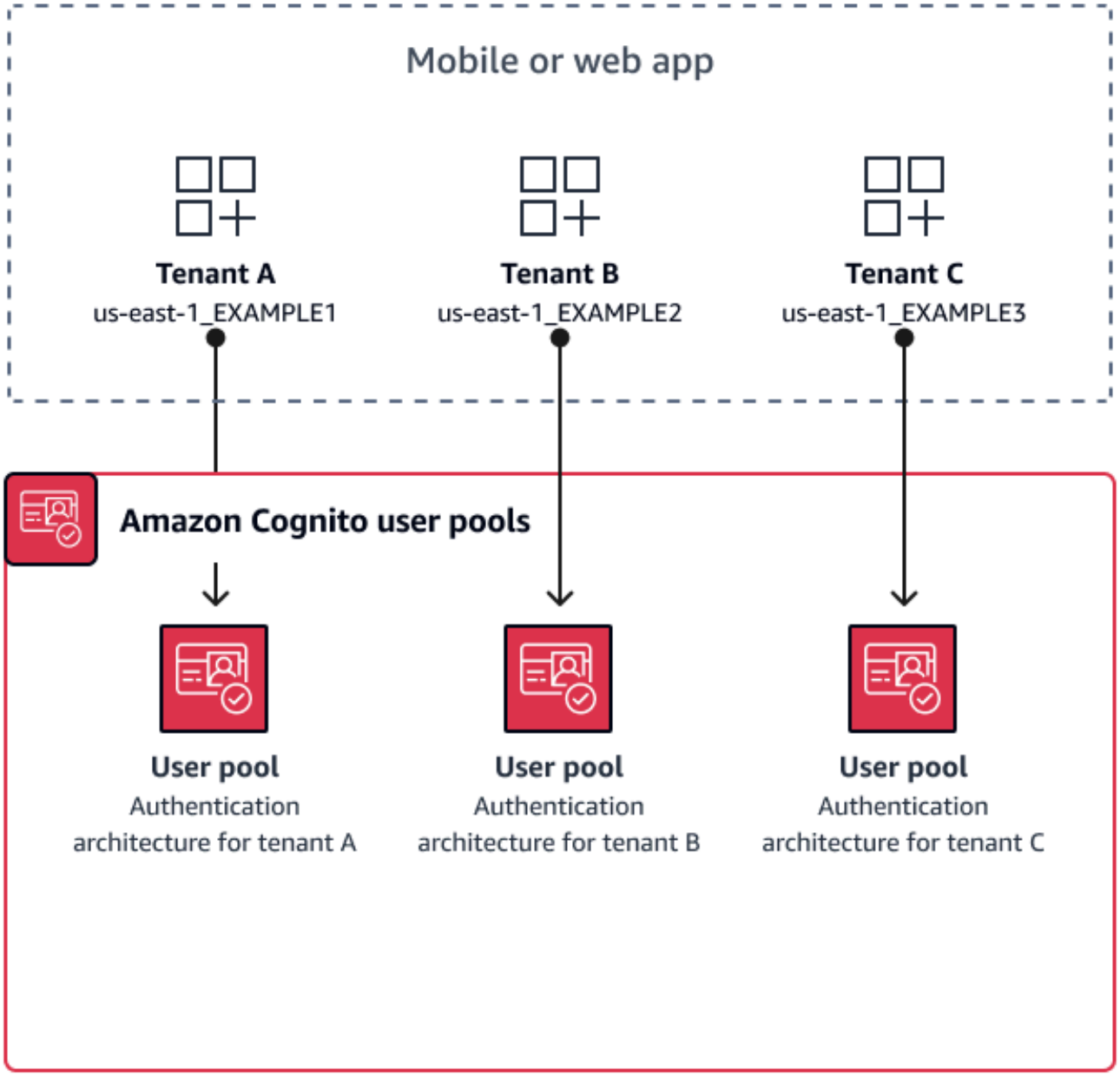
Temas

Prácticas recomendadas para un grupo de usuarios con varios arrendatarios

Crea un grupo de usuarios para cada inquilino de tu aplicación. Este enfoque aporta el máximo aislamiento a cada inquilino. Puede implementar diferentes configuraciones para cada inquilino. El aislamiento de inquilinos por grupo de usuarios le brinda flexibilidad a la hora de user-to-tenant mapear. Puede crear varios perfiles para el mismo usuario. Sin embargo, cada usuario tiene que registrarse de manera individual para cada inquilino al que tenga acceso.

Con este enfoque, puede configurar una interfaz de usuario alojada para cada inquilino de forma independiente y redirigir a los usuarios a la instancia de su aplicación específica para cada inquilino. También puedes usar este enfoque para integrarte con servicios de backend como Amazon API Gateway.

El siguiente diagrama muestra a cada inquilino con un grupo de usuarios dedicado.



¿Cuándo implementar la multitenencia con grupos de usuarios

Cuando el aislamiento y la personalización son sus principales preocupaciones. La relación entre los usuarios y los inquilinos puede ser compleja en una arquitectura con varios grupos de usuarios. Considere un ejemplo en el que tiene dos inquilinos educativos. El mismo usuario puede ser un estudiante con acceso limitado en una aplicación y un profesor con un alto nivel de permisos en otra. Es posible que necesites MFA en una aplicación pero no en otra, o que tengas una política

de contraseñas diferente. Como los usuarios locales pueden iniciar sesión en varios clientes de aplicaciones en grupos de usuarios con la interfaz de usuario alojada, la opción multiusuario también es ideal cuando se quiere que más de uno de sus inquilinos inicie sesión con la interfaz de usuario alojada.

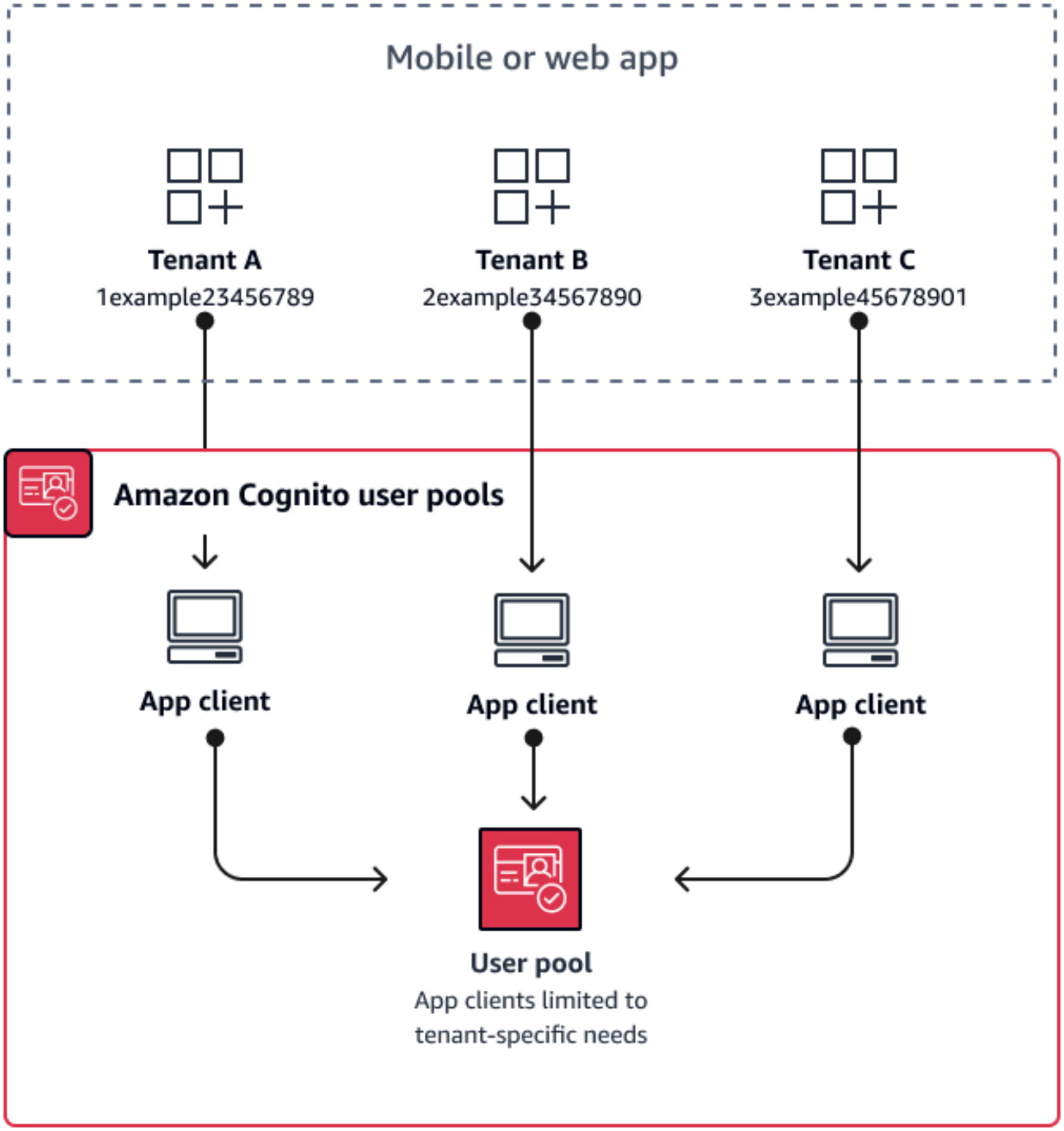
Nivel de esfuerzo

El nivel de esfuerzo de desarrollo y operación para utilizar este enfoque es alto. Para garantizar resultados consistentes y predecibles para su familia de aplicaciones, debe integrar los recursos de Amazon Cognito con sus herramientas de automatización y mantener sus bases de referencia a medida que la arquitectura de autenticación se hace más compleja. Si desea crear un punto de partida único para sus aplicaciones, debe crear los elementos de la interfaz de usuario (UI) para capturar la decisión inicial que dirige a los usuarios al recurso correcto.

Prácticas recomendadas para la multitenencia de aplicaciones y clientes

Crea un cliente de aplicación para cada inquilino de tu aplicación. Con la multitenencia entre aplicaciones y clientes, puedes asignar cualquier usuario a clientes de aplicaciones vinculadas a inquilinos y conservar un único perfil de usuario. Como puede asignar uno o todos los proveedores de identidad (IdPs) de su grupo de usuarios a un cliente de aplicaciones, un cliente de aplicación arrendatario puede permitir el inicio de sesión con un IdP específico del inquilino. Si hay usuarios en varios arrendatarios, puedes vincular sus perfiles con varios IdPs para ofrecer una experiencia de usuario coherente.

En el siguiente diagrama, se muestra a cada inquilino con un cliente de aplicación dedicado en un grupo de usuarios compartido.



¿Cuándo implementar la multitenencia entre aplicaciones y clientes

Cuando puede elegir una configuración universal para los ajustes a nivel de grupo de usuarios, como los activadores de Lambda, la política de contraseñas y el contenido y los métodos de entrega de

los mensajes de correo electrónico y SMS. Como los usuarios de un grupo de usuarios compartido pueden iniciar sesión en cualquier cliente de aplicaciones, la opción de tenencia múltiple app-cliente es ideal para iniciar sesión con la API de grupos de usuarios de Amazon Cognito o con la API de grupos de usuarios de app-client-specific IdPs Amazon Cognito. La multitenencia entre aplicaciones y clientes también es adecuada para one-to-many entornos en los que se quiere permitir a los usuarios realizar la transición entre varias aplicaciones.

Nivel de esfuerzo

La multitenencia de aplicaciones y clientes requiere un esfuerzo moderado. Uno de los principales desafíos de la multitenencia entre aplicaciones y clientes es la posibilidad de que los inquilinos presenten una cookie de interfaz de usuario alojada y cambien de una aplicación a otra. En una arquitectura multiusuario entre una aplicación y un cliente, evite el inicio de sesión en la interfaz de usuario alojada cuando sea necesario aislarla. Puedes distribuir tu aplicación móvil o los enlaces a tu aplicación web con la lógica del cliente de la aplicación integrada, o puedes crear elementos de interfaz de usuario iniciales que determinen la tenencia de los usuarios. El nivel de esfuerzo es menor porque no es necesario estandarizar ni mantener la configuración en varios grupos de usuarios y grupos de identidades.

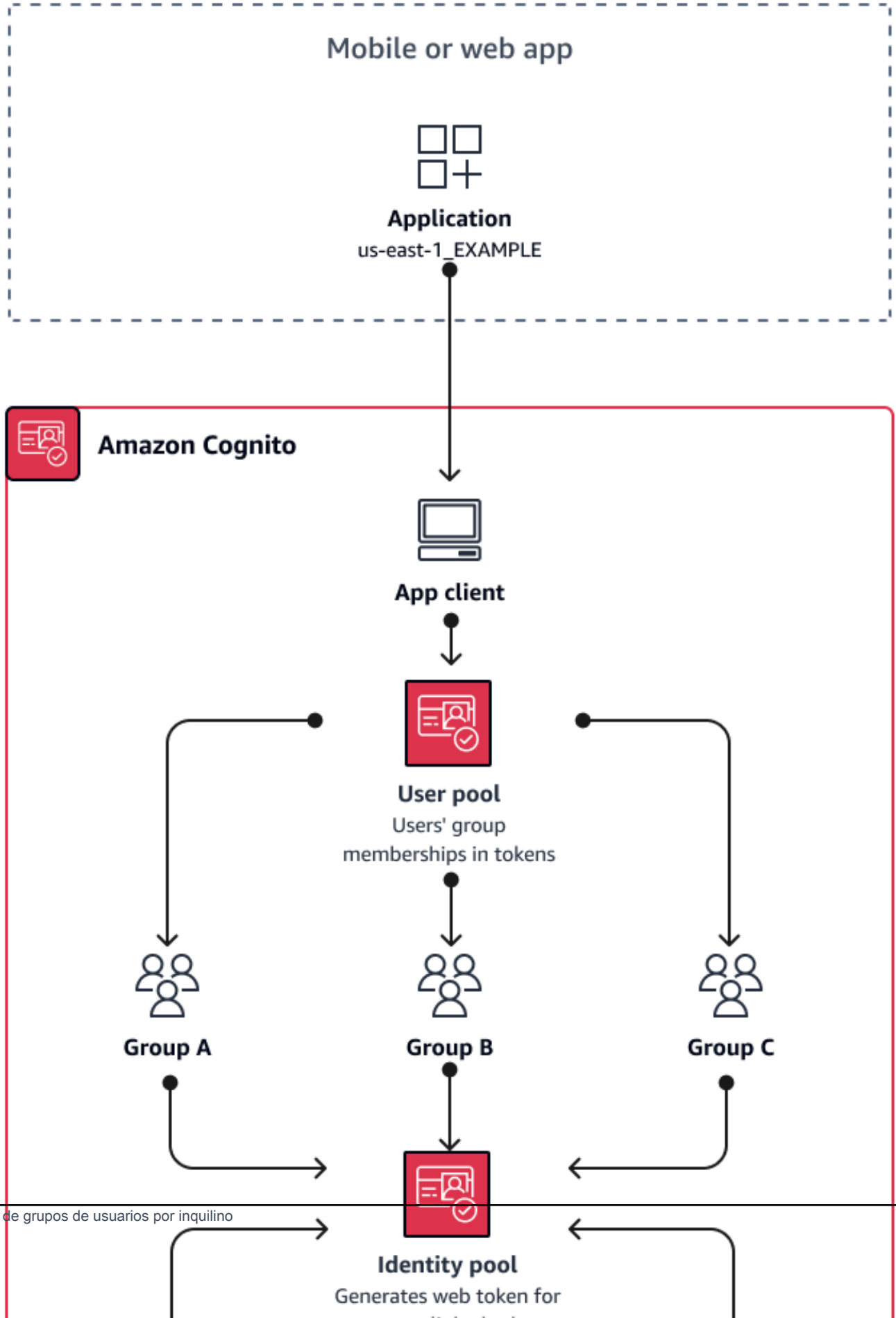
Mejores prácticas de arrendamiento múltiple para grupos de usuarios

La multitenencia basada en grupos funciona mejor cuando la arquitectura requiere grupos de usuarios de Amazon Cognito con grupos de identidades.

El identificador del grupo de usuarios y los tokens de acceso contienen una declaración. `cognito:groups` Además, los tokens de identificación contienen `cognito:preferred_role` reclamos `cognito:roles` y reclamos. Cuando el resultado principal de la autenticación en tu aplicación son AWS las credenciales temporales de un grupo de identidades, la pertenencia a un grupo de usuarios puede determinar la función de IAM y los permisos que reciben.

Como ejemplo, pensemos en tres arrendatarios, cada uno de los cuales almacena activos de aplicaciones en su propio bucket de Amazon S3. Asigne los usuarios de cada inquilino a un grupo asociado, configure un rol preferido para el grupo y otorgue a ese rol acceso de lectura a su bucket.

En el siguiente diagrama, se muestra a los inquilinos que comparten un cliente de aplicaciones y un grupo de usuarios, con grupos específicos en el grupo de usuarios que determinan si cumplen los requisitos para un rol de IAM.


Mobile or web app
Application
us-east-1_EXAMPLE
Amazon Cognito
App client
User pool
Users' group
memberships in tokens
Group A
Group B
Group C
Identity pool
Generates web token for

¿Cuándo implementar la tenencia múltiple grupal

Cuando el acceso a AWS los recursos es su principal preocupación. Grupos en los grupos de usuarios de Amazon Cognito Los grupos de usuarios son un mecanismo de control de acceso basado en roles (RBAC). Puede configurar muchos grupos en un grupo de usuarios y tomar decisiones complejas sobre el RBAC con prioridad de grupo. Los grupos de identidades pueden asignar credenciales al rol con mayor prioridad, a cualquier rol del grupo o a partir de otros atributos de los tokens de un usuario.

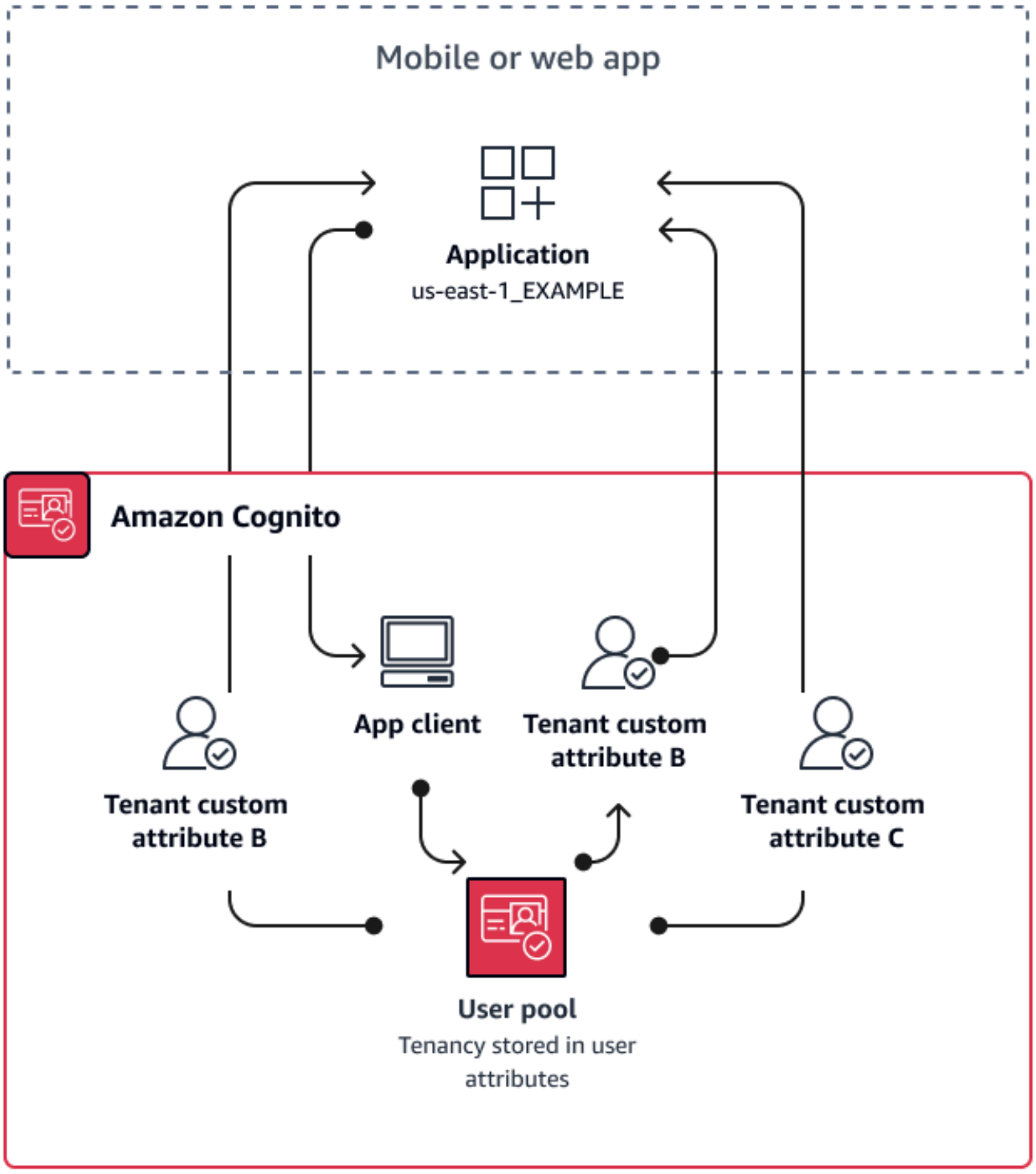
Nivel de esfuerzo

El nivel de esfuerzo para mantener la multitenencia solo con la pertenencia a un grupo es bajo. Sin embargo, para ampliar la función de los grupos de usuarios más allá de la capacidad integrada de selección de roles de IAM, debe crear una lógica de aplicación que procese la pertenencia a los grupos en los tokens de los usuarios y determine qué hacer en el cliente. Puedes integrar Amazon Verified Permissions con tus aplicaciones para tomar decisiones de autorización por parte del cliente. Los identificadores de grupo no se procesan actualmente en las operaciones de la IsAuthorizedWithTokenAPI de permisos verificados, pero puedes desarrollar un código personalizado que analice el contenido de los tokens, incluidas las solicitudes de pertenencia a grupos.

Mejores prácticas de tenencia múltiple con atributos personalizados

Amazon Cognito admite atributos personalizados con los nombres que usted elija. Un escenario en el que los atributos personalizados son útiles es cuando distinguen la tenencia de los usuarios en un grupo de usuarios compartido. Cuando asignas a los usuarios un valor para un atributo como`custom:tenantID`, por ejemplo, tu aplicación puede asignar el acceso a los recursos específicos del inquilino en consecuencia. Un atributo personalizado que defina un ID de inquilino debe ser inmutable o de solo lectura para el cliente de la aplicación.

En el siguiente diagrama, se muestra a los inquilinos que comparten un cliente de aplicaciones y un grupo de usuarios, con atributos personalizados en el grupo de usuarios que indican el inquilino al que pertenecen.



Cuando los atributos personalizados determinan la tenencia, puedes distribuir una única aplicación o URL de inicio de sesión. Una vez que el usuario inicie sesión, la aplicación podrá procesar la

`custom:tenantID` reclamación y determinar qué activos cargar, qué imagen de marca aplicar y qué funciones mostrar. Para tomar decisiones avanzadas de control de acceso a partir de los atributos del usuario, configure su grupo de usuarios como proveedor de identidades en Amazon Verified Permissions y genere decisiones de acceso a partir del contenido de los identificadores o los tokens de acceso.

¿Cuándo implementar la multitenencia con atributos personalizados

Cuando el arrendamiento es a nivel de superficie. Un atributo de inquilino puede contribuir a los resultados de la marca y el diseño. Si quieres lograr un aislamiento significativo entre los inquilinos, los atributos personalizados no son la mejor opción. Cualquier diferencia entre los inquilinos que deben configurarse a nivel de grupo de usuarios o de aplicación y cliente, como la MFA o la marca de la interfaz de usuario alojada, requiere que cree distinciones entre los inquilinos de una manera que los atributos personalizados no ofrezcan. Con los grupos de identidades, puedes incluso elegir el rol de IAM entre tus usuarios a partir del atributo personalizado que aparece en su token de identificación.

Nivel de esfuerzo

Dado que la multitenencia con atributos personalizados transfiere la responsabilidad de las decisiones de autorización basadas en el inquilino a tu aplicación, el nivel de esfuerzo suele ser elevado. Si ya conoce bien una configuración de cliente que analiza las reclamaciones de OIDC o los permisos verificados de Amazon, es posible que este enfoque requiera el nivel de esfuerzo más bajo.

Recomendaciones de seguridad para la arquitectura de varios inquilinos

Para garantizar que su aplicación sea más segura le recomendamos lo siguiente:

- Valida el arrendamiento en tu aplicación con Amazon Verified Permissions. Cree políticas que examinen los derechos del grupo de usuarios, los clientes de la aplicación, los grupos o los atributos personalizados antes de permitir la solicitud de un usuario en su aplicación. AWS creó fuentes de identidad de permisos verificados teniendo en cuenta los grupos de usuarios de Amazon Cognito. Verified Permissions incluye una guía adicional para la administración de múltiples inquilinos.
- Use únicamente una dirección de correo electrónico verificada para autorizar el acceso de usuario a un inquilino en función de la coincidencia de dominio. No confíe en las direcciones de correo electrónico y los números de teléfono a menos que su aplicación las verifique o que el IdP externo

proporcione una prueba de verificación. Para obtener más detalles sobre la configuración de estos permisos, consulte Permisos y ámbitos de los atributos.

- Utilice atributos personalizados inmutables o de solo lectura para los atributos del perfil de usuario que identifican a los inquilinos. Solo puede establecer el valor de los atributos inmutables cuando crea un usuario o cuando un usuario se registra en su grupo de usuarios. Además, proporcione a los clientes de aplicaciones acceso de solo lectura a los atributos.
- Utilice un mapeo 1:1 entre el IDP externo del inquilino y el cliente de la aplicación para evitar el acceso no autorizado entre inquilinos. Un usuario que ha sido autenticado por un IdP externo y que tiene una cookie de sesión de Amazon Cognito válida, puede acceder a otras aplicaciones de inquilino que confían en el mismo IdP.
- Al implementar la lógica de autorización y coincidencia de inquilinos en la aplicación, asegúrese de que los propios usuarios no puedan modificar los criterios utilizados para autorizar el acceso de los usuarios a los inquilinos. Además, si se está utilizando un IdP externo para la federación, restrinja a los administradores de proveedores de identidad de los inquilinos para que no puedan modificar el acceso de usuarios.

Situaciones comunes de Amazon Cognito

En este tema, se describen seis situaciones comunes del uso de Amazon Cognito.

Los dos componentes principales de Amazon Cognito son los grupos de usuarios y los grupos de identidades. Los grupos de usuarios son directorios de usuarios que proporcionan opciones de registro y de inicio de sesión para los usuarios de la aplicación web y la móvil. Los grupos de identidades proporcionan AWS credenciales temporales para conceder a los usuarios acceso a otros Servicios de AWS.

Un grupo de usuarios es un directorio de usuarios en Amazon Cognito. Los usuarios de tu aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidad (IdP) externo. El grupo de usuarios gestiona la sobrecarga de gestión de los tokens que se devuelven al iniciar sesión en redes sociales a través de Facebook, Google, Amazon y Apple, y desde OpenID Connect (OIDC) y SAML. IdPs Tanto si los usuarios inician sesión directamente o a través de un tercero, todos los miembros del grupo de usuarios tienen un perfil de directorio al que puede obtener acceso a través de un SDK.

Con un grupo de identidades, los usuarios pueden obtener AWS credenciales temporales para acceder a AWS servicios, como Amazon S3 y DynamoDB. Los grupos de identidades admiten usuarios invitados anónimos, así como la federación a través de terceros. IdPs

Temas

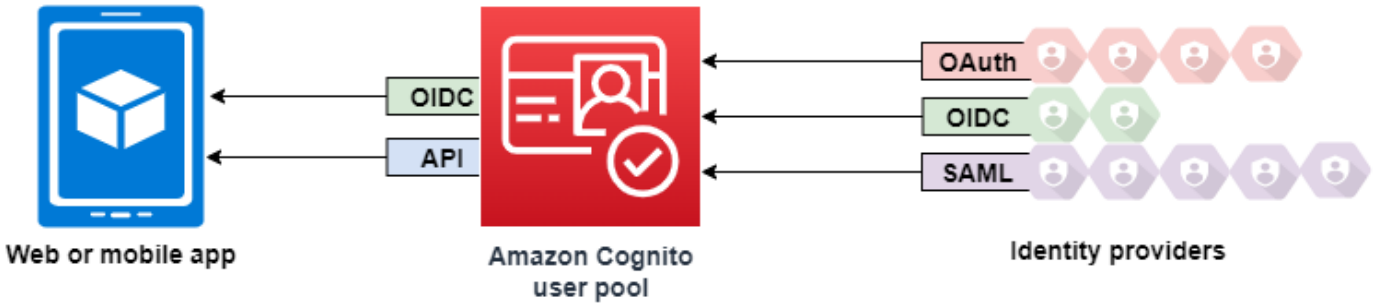
Autenticar con un grupo de usuarios

Puede permitir que los usuarios se autentiquen con un grupo de usuarios. Los usuarios de tu aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidad (IdP) externo. El grupo de usuarios gestiona la sobrecarga

de gestión de los tokens que se devuelven al iniciar sesión en redes sociales a través de Facebook, Google, Amazon y Apple, y desde OpenID Connect (OIDC) y SAML. IdPs

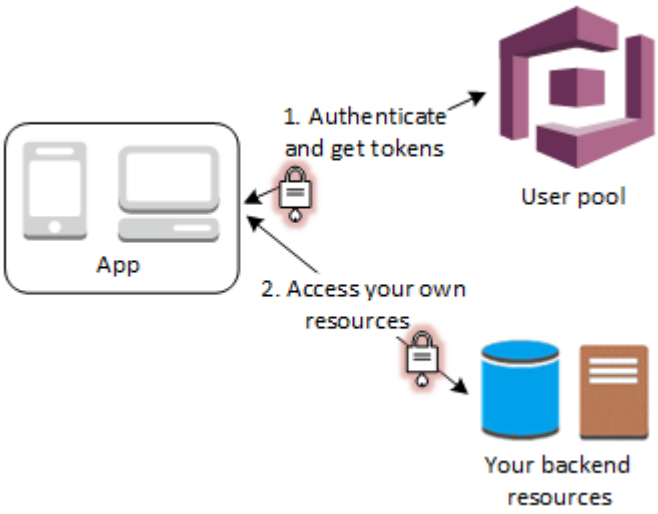
Tras una autenticación correcta, la aplicación web o móvil recibirá tokens de grupos de usuarios desde Amazon Cognito. Puede usar esos tokens para recuperar AWS las credenciales que permiten a su aplicación acceder a otros AWS servicios, o puede optar por usarlos para controlar el acceso a los recursos del lado del servidor o a Amazon API Gateway.

Para obtener más información, consulte Flujo de autenticación de los grupos de usuarios y Uso de tokens con grupos de usuarios.



Acceso a los recursos del lado del servidor con un grupo de usuarios

Tras un inicio de sesión de grupo de usuarios correcto, la aplicación web o móvil recibirá tokens de grupos de usuarios desde Amazon Cognito. Puede utilizar los tokens para controlar el acceso a los recursos del lado del servidor. También puede crear conjuntos de grupos de usuarios para administrar permisos y representar diferentes tipos de usuarios. Para obtener más información sobre el uso de grupos para controlar el acceso a los recursos, consulte Agregar grupos a un grupo de usuarios.



Después de configurar un dominio para el grupo de usuarios, Amazon Cognito aprovisiona una IU web alojada que le permite agregar páginas de registro e inicio de sesión a la aplicación. Con esta base de OAuth 2.0, puede crear su propio servidor de recursos y permitir que los usuarios obtengan acceso a los recursos protegidos. Para obtener más información, consulte Autorización de alcances, M2M y API con servidores de recursos.

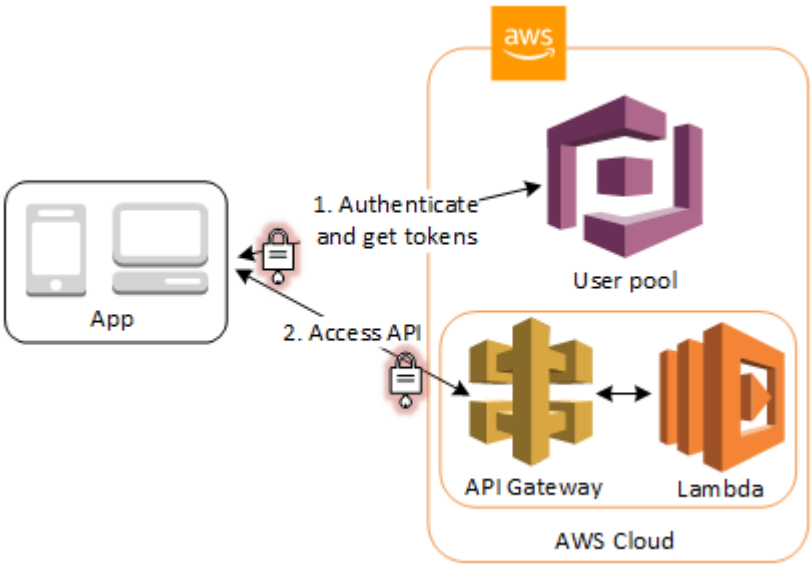
Para obtener más información sobre la autenticación de grupos de usuarios, consulte Flujo de autenticación de los grupos de usuarios y Uso de tokens con grupos de usuarios.

Acceso a los recursos con API Gateway y Lambda mediante un grupo de usuarios

Puede habilitar a los usuarios para que accedan a la API a través de API Gateway. API Gateway valida los tokens a partir de una autenticación correcta de grupos de usuarios y los utiliza para conceder acceso a sus usuarios a los recursos, incluidas las funciones de Lambda o su propia API.

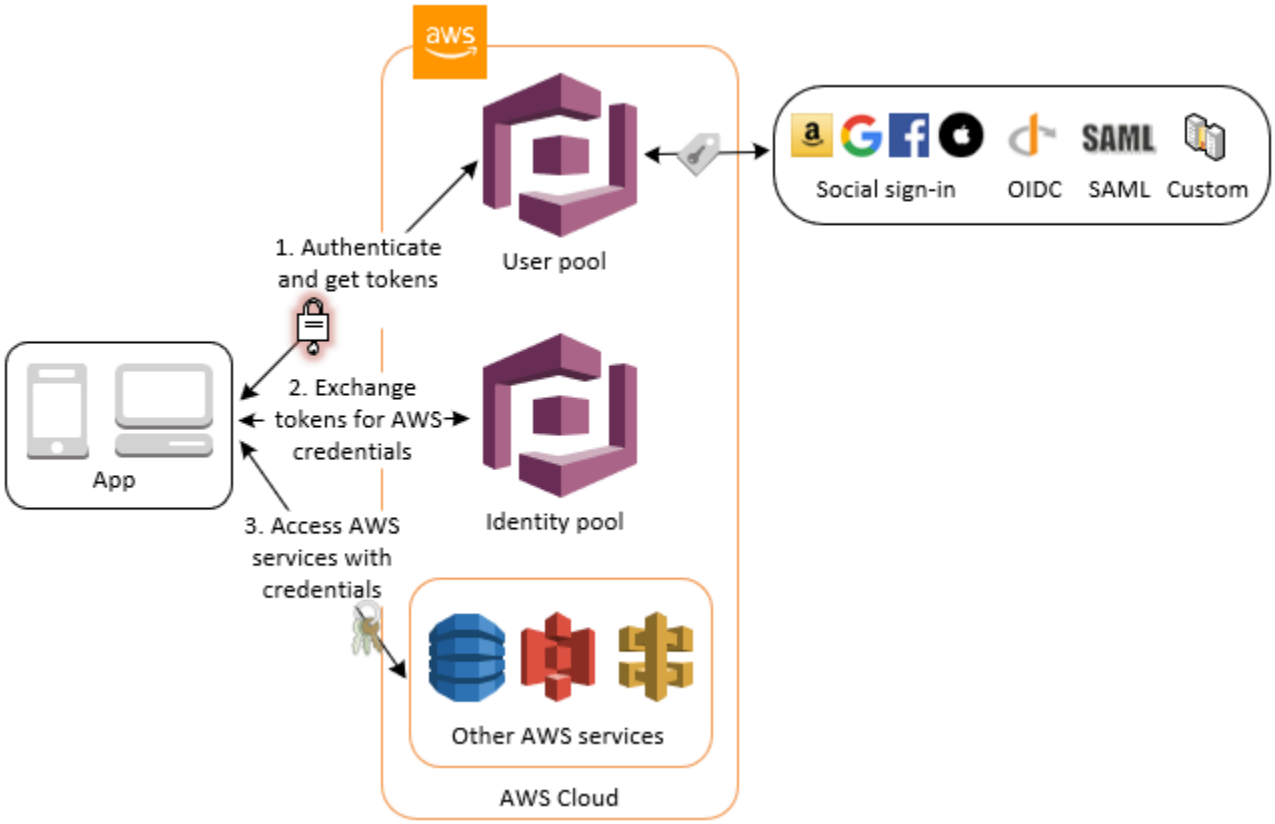
Puede utilizar grupos en un grupo de usuarios a fin de controlar permisos con API Gateway al mapear membresías de grupo a roles de IAM. Los grupos a los que pertenece un usuario están incluidos en el token de ID proporcionado por un grupo de usuarios cuando el usuario de la aplicación inicia sesión. Para obtener más información acerca de los conjuntos de grupos de usuarios, consulte Agregar grupos a un grupo de usuarios.

Puede enviar sus tokens de grupo de usuarios con una solicitud a API Gateway para que los verifique una función de Lambda del autorizador de Amazon Cognito. Para obtener más información acerca de API Gateway, consulte Uso de API Gateway con grupos de usuarios de Amazon Cognito.



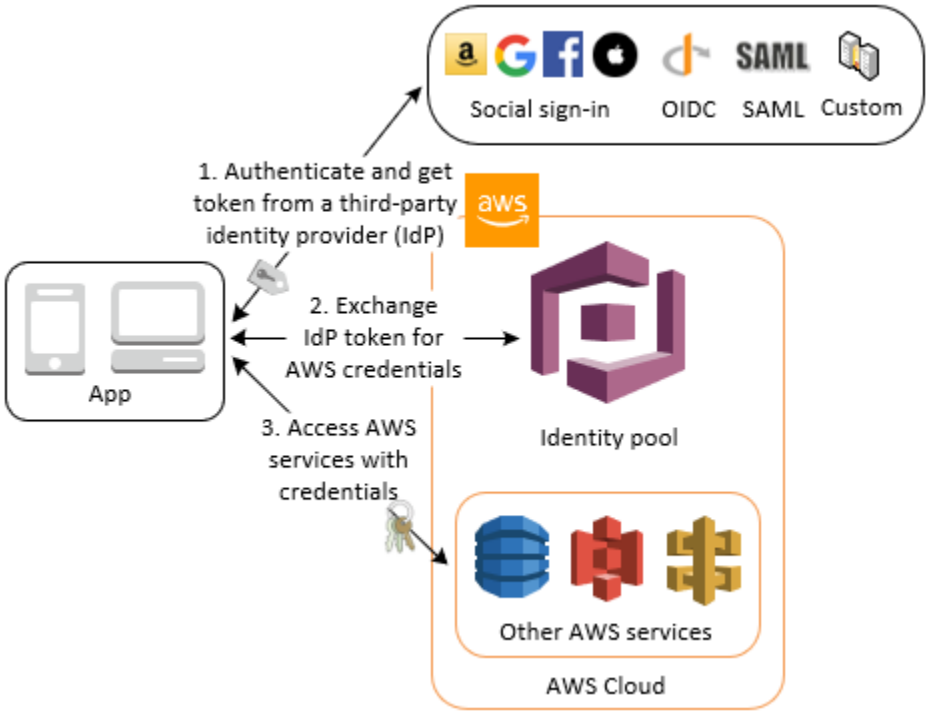
Acceda a AWS los servicios con un grupo de usuarios y un grupo de identidades

Tras una autenticación correcta mediante el grupo de usuarios, la aplicación web o móvil recibirá tokens de grupos de usuarios desde Amazon Cognito. Puede cambiarlos por un acceso temporal a otros AWS servicios con un grupo de identidades. Para obtener más información, consulte Acceder Servicios de AWS mediante un grupo de identidades después de iniciar sesión y Introducción a los grupos de identidades de Amazon Cognito.



Autenticación con un tercero y acceso a los servicios de AWS con un grupo de identidades

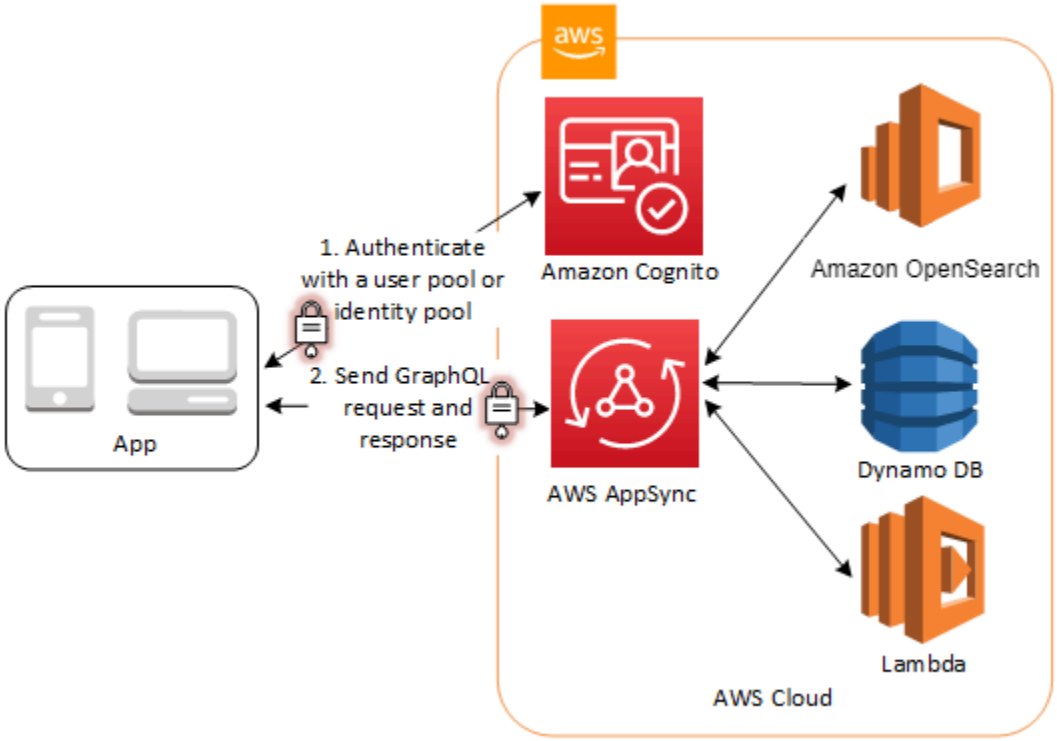
Puede permitir que sus usuarios accedan a los AWS servicios a través de un grupo de identidades. Un grupo de identidades requiere un token de proveedor de identidad de un usuario que se haya autenticado mediante un proveedor de identidad de terceros (o nada si se trata de un invitado anónimo). A cambio, el grupo de identidades otorga AWS credenciales temporales que puede usar para acceder a otros AWS servicios. Para obtener más información, consulte Introducción a los grupos de identidades de Amazon Cognito.



Acceda a AWS AppSync los recursos con Amazon Cognito

Puede conceder a sus usuarios acceso a los AWS AppSync recursos con los tokens de una autenticación correcta del grupo de usuarios de Amazon Cognito. Para obtener más información, consulte la autorización de AMAZON_COGNITO_USER_POOLS en la Guía para desarrolladores de AWS AppSync .

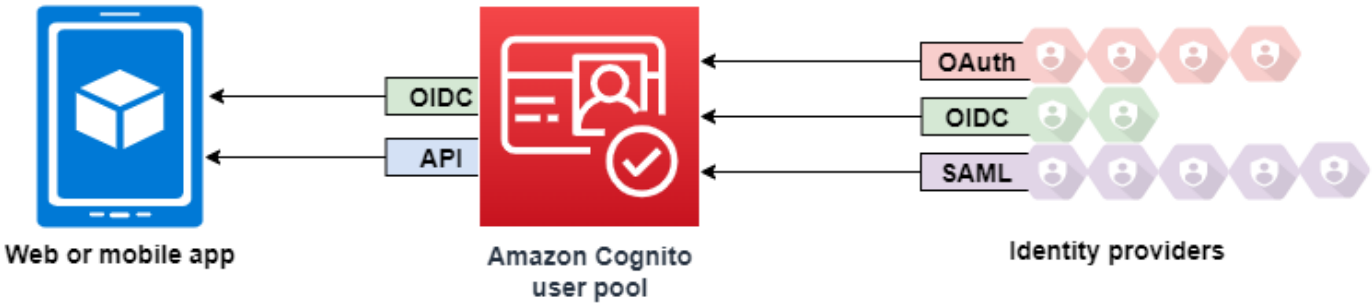
También puedes firmar las solicitudes a la API de AWS AppSync GraphQL con las credenciales de IAM que recibas de un grupo de identidades. Consulte Autorización AWS_IAM.


aws
Amazon Cognito
1. Authenticate with a user pool or identity pool
2. Send GraphQL request and response
App
AWS AppSync
Amazon OpenSearch
Dynamo DB
Lambda
AWS Cloud

Grupos de usuarios de Amazon Cognito

Un grupo de usuarios de Amazon Cognito es un directorio de usuarios para la autenticación y autorización de aplicaciones web y móviles. Desde la perspectiva de la aplicación, un grupo de usuarios de Amazon Cognito es un proveedor de identidades (IdP) OpenID Connect (OIDC). Un grupo de usuarios agrega capas de características adicionales para la seguridad, la federación de identidades, la integración de aplicaciones y la personalización de la experiencia del usuario.

Puede, por ejemplo, comprobar que las sesiones de los usuarios provengan de orígenes fiables. Puede combinar el directorio de Amazon Cognito con un proveedor de identidad externo. Con el AWS SDK que prefieras, puedes elegir el modelo de autorización de API que mejor se adapte a tu aplicación. Además, puede agregar funciones de AWS Lambda que modifiquen o revisen el comportamiento predeterminado de Amazon Cognito.



Temas

- Características
- Autenticación con un grupo de usuarios
- Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios
- Actualización de la configuración del grupo de usuarios
- Configuración y uso de la interfaz de usuario alojada y los puntos de conexión de federación de Amazon Cognito
- Autorización de alcances, M2M y API con servidores de recursos
- Agregar inicio de sesión de grupo de usuarios a través de un tercero
- Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda
- Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito

- Administración de usuarios en el grupo de usuarios
- Configuración de correo electrónico para grupos de usuarios de Amazon Cognito
- Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito
- Uso de tokens con grupos de usuarios
- Acceso a los recursos después de una autenticación correcta con el grupo de usuarios
- Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito

Características

Los grupos de usuarios de Amazon Cognito cuentan con las características siguientes.

Sign-up (Registro)

Los grupos de usuarios de Amazon Cognito cuentan con métodos programáticos, impulsados por el usuario y por el administrador para agregar perfiles de usuario al grupo de usuarios. Los grupos de usuarios de Amazon Cognito admiten los siguientes modelos de registro. Puede usar cualquier combinación de estos modelos en la aplicación.

> **Important**
>
> Si activa el registro de usuarios en el grupo de usuarios, cualquier usuario de Internet podrá crear una cuenta e iniciar sesión en las aplicaciones. No habilite el registro automático en el grupo de usuarios a menos que quiera abrir la aplicación para que el público se registre. Para cambiar esta configuración, actualiza el registro de autoservicio en la pestaña Experiencia de registro de la consola del grupo de usuarios o actualiza el valor de una solicitud AllowAdminCreateUserOnlyde API CreateUserPool. UpdateUserPool Para obtener información sobre las características de seguridad que puede configurar en los grupos de usuarios, consulte Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito.

1. Los usuarios pueden ingresar la información en la aplicación y crear un perfil de usuario nativo para el grupo de usuarios. Puede realizar operaciones de registro de la API para registrar a los usuarios en el grupo de usuarios. Puedes abrir estas operaciones de registro a cualquier persona o puedes autorizarlas con un secreto de cliente o credenciales. AWS

2. Puede redirigir a los usuarios a un IdP de terceros al que puedan autorizar a transmitir la información a Amazon Cognito. Amazon Cognito procesa los tokens de ID de OIDC, los datos `userInfo` de OAuth 2.0 y las afirmaciones de SAML 2.0 en los perfiles de usuario del grupo de usuarios. Controla los atributos que desea que reciba Amazon Cognito en función de las reglas de mapeo de atributos.
3. Puede omitir el registro público o federado y crear usuarios en función del propio origen de datos y esquema. Agregue usuarios directamente en la consola o la API de Amazon Cognito. Importe usuarios desde un archivo CSV. Ejecute una just-in-time AWS Lambda función que busque al nuevo usuario en un directorio existente y complete su perfil de usuario a partir de los datos existentes.

Después de que los usuarios se registren, puede agregarlos a los grupos que Amazon Cognito muestra en los tokens de acceso e ID. También puede enlazar grupos de grupos de usuarios a roles de IAM al pasar el token de ID a un grupo de identidades.

Temas relacionados de

- Administración de usuarios en el grupo de usuarios
- Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios
- Ejemplos de código para Amazon Cognito Identity Provider mediante SDK AWS

Sign-in (Inicio de sesión)

Amazon Cognito puede ser un directorio de usuarios independiente y proveedor de identidades (IdP) para la aplicación. Los usuarios pueden iniciar sesión con una interfaz de usuario alojada por Amazon Cognito o con su propia interfaz de usuario a través de la API de grupos de usuarios de Amazon Cognito. El nivel de la aplicación que está detrás de la interfaz de usuario personalizada de frontend puede autorizar las solicitudes en el backend con cualquiera de varios métodos para confirmar las solicitudes legítimas.

Para iniciar sesión en los usuarios con un directorio externo, combinado opcionalmente con el directorio de usuarios integrado en Amazon Cognito, puede agregar las siguientes integraciones.

1. Inicie sesión e importe datos de usuarios de consumidores con el inicio de sesión social de OAuth 2.0. Amazon Cognito admite el inicio de sesión con Google, Facebook, Amazon y Apple a través de OAuth 2.0.

2. Inicie sesión e importe datos de usuarios empresariales con el inicio de sesión de SAML y OIDC. También puede configurar Amazon Cognito para aceptar reclamaciones de cualquier proveedor de identidades (IdP) de SAML u OpenID Connect (OIDC).
3. Enlace los perfiles de usuario externos a los perfiles de usuario nativos. Un usuario enlazado puede iniciar sesión con una identidad de usuario de terceros y recibir el acceso que asigne a un usuario en el directorio integrado.

Temas relacionados de

- Agregar inicio de sesión de grupo de usuarios a través de un tercero
- Vinculación de usuarios federados a un perfil de usuario existente

Mi autorización achine-to-machine

Algunas sesiones no son una human-to-machine interacción. Es posible que necesite una cuenta de servicio que pueda autorizar una solicitud a una API mediante un proceso automatizado. Para generar tokens de acceso para la machine-to-machine autorización con los ámbitos de OAuth 2.0, puedes añadir un cliente de aplicación que genere concesiones de credenciales de cliente.

Temas relacionados de

- Autorización de alcances, M2M y API con servidores de recursos

IU alojada

Si no desea crear una interfaz de usuario, puede presentar a los usuarios una interfaz de usuario alojada en Amazon Cognito personalizada. La interfaz de usuario alojada es un conjunto de páginas web para registrarse, iniciar sesión, autenticación multifactor (MFA) y restablecer contraseña. Puedes añadir la interfaz de usuario alojada a tu dominio existente o usar un identificador de prefijo en un subdominio. AWS

Temas relacionados de

- Configuración y uso de la interfaz de usuario alojada y los puntos de conexión de federación de Amazon Cognito
- Configuración de un dominio del grupo de usuarios

Seguridad

Los usuarios locales pueden proporcionar un factor de autenticación adicional con un código de un mensaje SMS o una aplicación que genere códigos de autenticación multifactor (MFA). Puede crear mecanismos para configurar y procesar la MFA en la aplicación o puede dejar que la interfaz de usuario alojada la administre. Los grupos de usuarios de Amazon Cognito pueden omitir la MFA cuando los usuarios inician sesión desde dispositivos de confianza.

Si no desea solicitar inicialmente la MFA a los usuarios, puede exigirla de forma condicional. Gracias a las características de seguridad avanzadas, Amazon Cognito puede detectar posibles actividades malintencionadas y solicitar al usuario que configure la MFA o bloquee el inicio de sesión.

Si el tráfico de red hacia tu grupo de usuarios puede ser malintencionado, puedes supervisarlo y tomar medidas con las ACL AWS WAF web.

Temas relacionados de

- Adición de MFA a un grupo de usuarios.
- Adición de seguridad avanzada a un grupo de usuarios
- Asociar una ACL AWS WAF web a un grupo de usuarios

Personalizar la experiencia del usuario

En la mayoría de las etapas del registro, el inicio de sesión o la actualización del perfil de un usuario, puede personalizar la forma en que Amazon Cognito gestiona la solicitud. Con los desencadenadores de Lambda, puede modificar un token de ID o rechazar una solicitud de registro en función de las condiciones personalizadas. Puede crear su propio flujo de autenticación personalizado.

Puede cargar CSS y logotipos personalizados para dar a la interfaz de usuario alojada un aspecto familiar para los usuarios.

Temas relacionados de

- Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda
- Desencadenadores de Lambda de desafío de autenticación personalizado
- Personalizar las páginas web integradas de registro e inicio de sesión

Monitoreo y análisis

Los grupos de usuarios de Amazon Cognito registran las solicitudes de la API, incluidas las solicitudes a la interfaz de usuario alojada, en AWS CloudTrail. Puede revisar las métricas de rendimiento de Amazon CloudWatch Logs, insertar registros personalizados CloudWatch con activadores de Lambda y supervisar el volumen de solicitudes de API en la consola de Service Quotas.

También puede registrar los datos del dispositivo y de la sesión de las solicitudes de la API en una campaña de Amazon Pinpoint. Con Amazon Pinpoint, puede enviar notificaciones push desde la aplicación en función del análisis de la actividad de los usuarios.

Temas relacionados de

- Registrar llamadas a la API de Amazon Cognito con AWS CloudTrail
- Seguimiento de las cuotas CloudWatch y el uso en Service Quotas
- Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito

Integración de los grupos de identidades de Amazon Cognito

La otra mitad de Amazon Cognito son grupos de identidades. Los grupos de identidades proporcionan credenciales que autorizan y supervisan las solicitudes de API de sus usuarios a Servicios de AWS, por ejemplo, Amazon DynamoDB o Amazon S3. Puede crear políticas de acceso basadas en la identidad que protejan los datos en función de la forma en que clasifique a los usuarios del grupo de usuarios. Los grupos de identidades también pueden aceptar tokens y aserciones SAML 2.0 de diversos proveedores de identidades, independientemente de la autenticación del grupo de usuarios.

Temas relacionados de

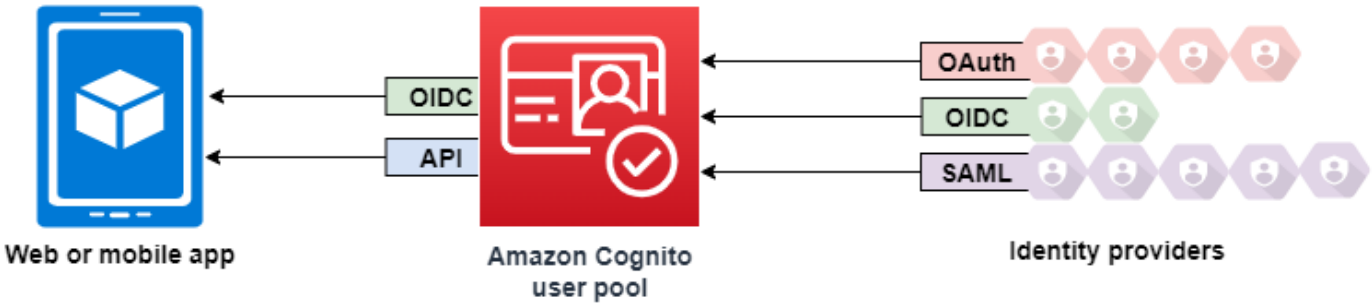
- Acceder Servicios de AWS mediante un grupo de identidades después de iniciar sesión
- Grupos de identidades de Amazon Cognito

Autenticación con un grupo de usuarios

Los usuarios de tu aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidad (IdP) externo. El grupo de usuarios gestiona

la sobrecarga de gestión de los tokens que se devuelven al iniciar sesión en redes sociales a través de Facebook, Google, Amazon y Apple, y desde OpenID Connect (OIDC) y SAML. IdPs

Después de una autenticación correcta, Amazon Cognito devuelve tokens de grupos de usuarios a la aplicación. Puede utilizar los tokens para conceder a sus usuarios acceso a sus propios recursos del lado del servidor o a Amazon API Gateway. O bien, puedes cambiarlos por AWS credenciales para acceder a otros servicios. AWS



La manipulación y administración de los tokens de grupos de usuarios para su aplicación web o móvil se realiza en el lado del cliente por medio de los SDK de Amazon Cognito. Del mismo modo, el SDK para móviles para iOS y el SDK para móviles para Android actualizan de forma automática los tokens de ID y de acceso si existe un token de actualización válido (no caducado) y los tokens de ID y de acceso tienen una validez mínima restante de 5 minutos. Para obtener información sobre los SDK y códigos de muestra para Android e iOS JavaScript, consulte los SDK del grupo de usuarios de Amazon Cognito.

Una vez que el usuario de la aplicación haya iniciado sesión de forma correcta, Amazon Cognito crea una sesión y devuelve un token de ID, de acceso y de actualización para el usuario autenticado.

JavaScript

```
// Amazon Cognito creates a session which includes the id, access, and refresh
 tokens of an authenticated user.

var authenticationData = {
        Username : 'username',
        Password : 'password',
    };
    var authenticationDetails = new
 AmazonCognitoIdentity.AuthenticationDetails(authenticationData);
    var poolData = { UserPoolId : 'us-east-1_ExaMPle',
        ClientId : '1example23456789'
```

```
    };
    var userPool = new AmazonCognitoIdentity.CognitoUserPool(poolData);
    var userData = {
        Username : 'username',
        Pool : userPool
    };
    var cognitoUser = new AmazonCognitoIdentity.CognitoUser(userData);
    cognitoUser.authenticateUser(authenticationDetails, {
        onSuccess: function (result) {
            var accessToken = result.getAccessToken().getJwtToken();

            /* Use the idToken for Logins Map when Federating User Pools with
 identity pools or when passing through an Authorization Header to an API Gateway
 Authorizer */
            var idToken = result.idToken.jwtToken;
        },

        onFailure: function(err) {
            alert(err);
        },

});
```

Android

```
// Session is an object of the type CognitoUserSession, and includes the id, access,
 and refresh tokens for a user.

String idToken = session.getIdToken().getJWTToken();
String accessToken = session.getAccessToken().getJWT();
```

iOS - swift

```
// AWSCognitoIdentityUserSession includes id, access, and refresh tokens for a user.

- (AWSTask<AWSCognitoIdentityUserSession *> *)getSession;
```

iOS - objective-C

```
// AWSCognitoIdentityUserSession includes the id, access, and refresh tokens for a
 user.
```

```
[[user getSession:@"username" password:@"password" validationData:nil scopes:nil]
 continueWithSuccessBlock:^id _Nullable(AWSTask<AWSCognitoIdentityUserSession *> *
 _Nonnull task) {
    // success, task.result has user session
    return nil;
}];
```

Temas

- Flujo de autenticación de los grupos de usuarios
- Clientes de aplicación de grupo de usuarios
- Uso de dispositivos de usuario en el grupos de usuarios

Flujo de autenticación de los grupos de usuarios

Amazon Cognito incluye varios métodos para autenticar a los usuarios. Todos los grupos de usuarios, tengan o no un dominio, pueden autenticar usuarios en la API de grupos de usuarios. Si agrega un dominio al grupo de usuarios, puede utilizar los puntos de conexión del grupo de usuarios. La API de grupos de usuarios admite una variedad de modelos de autorización y flujos de solicitud para las solicitudes de API.

Para verificar la identidad de los usuarios, Amazon Cognito admite flujos de autenticación que incorporan nuevos tipos de desafíos, además de las contraseñas. La autenticación de Amazon Cognito suele requerir que implemente dos operaciones de la API en el siguiente orden:

Public authentication

1. InitiateAuth
2. RespondToAuthChallenge

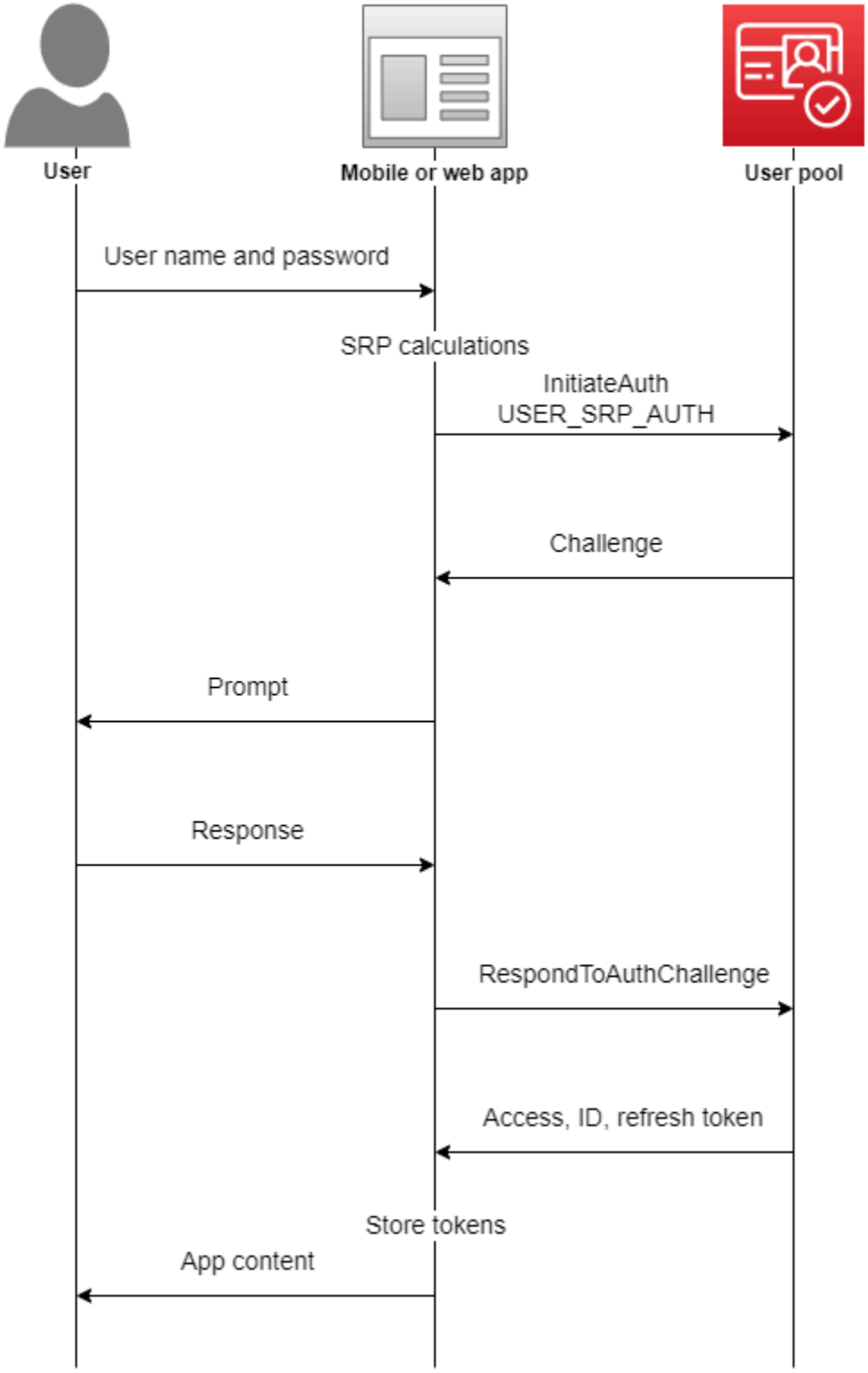
`InitiateAuth` y `RespondToAuthChallenge` son API no autenticadas que se usan con clientes de aplicaciones públicas del lado del cliente.

Server-side authentication

1. AdminInitiateAuth
2. AdminRespondToAuthChallenge

`AdminInitiateAuth` y `AdminRespondToAuthChallenge` requieren credenciales de IAM y son adecuadas para clientes de aplicaciones confidenciales del lado del servidor.

Un usuario se autentica respondiendo a desafíos sucesivos hasta que se produce un error de autenticación o Amazon Cognito emite tokens para el usuario. Puede repetir estos pasos con Amazon Cognito, en un proceso que incluye diferentes desafíos, para admitir cualquier flujo de autenticación personalizado.


User
Mobile or web app
User pool
User name and password
SRP calculations
InitiateAuth
USER_SRP_AUTH
Challenge
Prompt
Response
RespondToAuthChallenge
Access, ID, refresh token
Store tokens
App content

Por lo general, la aplicación genera un mensaje para recopilar información del usuario y envía esa información en una solicitud de API a Amazon Cognito. Considere un flujo de `InitiateAuth` en un grupo de usuarios en el que ha configurado el usuario con autenticación multifactor (MFA).

1. La aplicación pide a los usuarios el nombre de usuario y la contraseña.
2. El nombre de usuario y la contraseña se incluyen como parámetros en `InitiateAuth`.
3. Amazon Cognito devuelve un desafío de `SMS_MFA` y un identificador de sesión.
4. La aplicación solicita al usuario el código de MFA desde el teléfono.
5. Debe incluir ese código y el identificador de sesión en la solicitud `RespondToAuthChallenge`.

Según las características del grupo de usuarios, puede terminar respondiendo a varios desafíos para `InitiateAuth` antes de que la aplicación recupere los tokens de Amazon Cognito. Amazon Cognito incluye una cadena de sesión en la respuesta a cada solicitud. Para combinar las solicitudes de la API en un flujo de autenticación, incluya la cadena de sesión de la respuesta a la solicitud anterior en cada solicitud posterior. De forma predeterminada, los usuarios tienen tres minutos para completar cada desafío antes de que caduque la cadena de sesión. Para ajustar este periodo, cambie el cliente de la aplicación Duración de la sesión de flujo de autenticación. En el siguiente procedimiento, se describe cómo cambiar esta configuración en la configuración del cliente de la aplicación.

> **Note**
>
> La configuración de la duración de la sesión del flujo de autenticación se aplica a la autenticación con la API de los grupos de usuarios de Amazon Cognito. La interfaz de usuario alojada en Amazon Cognito establece la duración de la sesión en 3 minutos para la autenticación multifactorial y 8 minutos para los códigos de restablecimiento de contraseña.

Amazon Cognito console

Para configurar la duración de la sesión del flujo de autenticación del cliente (AWS Management Console)

1. En la pestaña App integration (Integración de aplicaciones) de su grupo de usuarios, seleccione el nombre de su cliente de aplicaciones en el contenedor App clients and analytics (Clientes de aplicaciones y análisis).
2. Elija Editar en el contenedor de Información de cliente de aplicaciones.

3. Cambie el valor de Authentication flow session duration (Duración de la sesión de flujo de autenticación) a la duración de validez que desee, en minutos, para los códigos MFA de SMS. Esto también cambia la cantidad de tiempo que tiene cualquier usuario para completar cualquier desafío de autenticación en el cliente de la aplicación.
4. Elija Guardar cambios.

Amazon Cognito API

Para configurar la duración de la sesión del flujo de autenticación del cliente (API Amazon Cognito)

1. Prepare una solicitud `UpdateUserPoolClient` con la configuración de su grupo de usuarios existente desde una solicitud `DescribeUserPoolClient`. Su solicitud `UpdateUserPoolClient` debe incluir todas las propiedades del cliente de la aplicación existentes.
2. Cambie el valor de `AuthSessionValidity` a la duración de validez que desee, en minutos, para los códigos MFA de SMS. Esto también cambia la cantidad de tiempo que tiene cualquier usuario para completar cualquier desafío de autenticación en el cliente de la aplicación.

Para obtener más información acerca de los clientes de aplicación, consulte Clientes de aplicación de grupo de usuarios.

Puede usar AWS Lambda activadores para personalizar la forma en que los usuarios se autentican. Estos disparadores emiten y verifican sus propios desafíos durante el flujo de autenticación.

También puede utilizar el flujo de autenticación de administrador para servidores backend seguros. También puede utilizar el flujo de autenticación de migración de usuarios para permitir la migración de usuarios sin necesidad de que estos restablezcan sus contraseñas.

Comportamiento de bloqueo de Amazon Cognito por intentos de inicio de sesión con error

Tras cinco intentos infructuosos de inicio de sesión no autenticado o de inicio de sesión autenticado por IAM con una contraseña, Amazon Cognito bloquea al usuario durante un segundo. La duración del bloqueo se duplica después de cada intento fallido adicional, hasta un máximo de aproximadamente 15 minutos. Los intentos realizados durante un periodo de bloqueo generan una excepción `Password attempts exceeded` y no afectan a la duración de los periodos de bloqueo posteriores. Para un número acumulado de intentos de inicio de sesión con error n, sin

incluir las excepciones `Password attempts exceeded`, Amazon Cognito bloquea a su usuario durante 2^(n-5) segundos. Para restablecer el bloqueo a su estado inicial n=0, su usuario debe iniciar sesión correctamente después de que venza un periodo de bloqueo, o no iniciar ningún intento de inicio de sesión durante 15 minutos consecutivos en cualquier momento después de un bloqueo. Este comportamiento está sujeto a cambios. Este comportamiento no se aplica a los desafíos personalizados, a menos que también realicen una autenticación basada en contraseña.

Temas

Flujo de autenticación en el lado del cliente

El siguiente proceso funciona para las aplicaciones de usuario en el cliente que cree con AWS Amplify o los SDK de AWS.

1. El usuario introduce su nombre de usuario y contraseña en la aplicación.
2. La aplicación llama a la operación `InitiateAuth` con el nombre de usuario y los detalles de contraseña remota segura (SRP) del usuario.

   Esta operación de la API devuelve los parámetros de autenticación.

   > **Note**
   >
   > La aplicación genera detalles de SRP con las funciones SRP de Amazon Cognito que están integradas en los SDK de AWS .

3. La aplicación llama a la operación `RespondToAuthChallenge`. Si la llamada se realiza correctamente, Amazon Cognito devuelve los tokens del usuario y el flujo de autenticación finaliza.

Si Amazon Cognito necesita otro desafío, la llamada a `RespondToAuthChallenge` no devuelve ningún token. En su lugar, la llamada devuelve una sesión.

4. Si `RespondToAuthChallenge` devuelve una sesión, la aplicación llama de nuevo a `RespondToAuthChallenge`, esta vez con la sesión y la respuesta al desafío (por ejemplo, código de MFA).

Flujo de autenticación en el lado del servidor

Si no dispone de una aplicación de usuario, sino que usa una aplicación segura del backend o del lado del servidor en Java, Ruby o Node.js, puede utilizar la API autenticada del lado del servidor para los grupos de usuarios de Amazon Cognito.

En el caso de las aplicaciones del lado del servidor, la autenticación de grupos de usuarios es similar a la de las aplicaciones del lado del cliente, excepto en el siguiente caso:

- La aplicación del lado del servidor llama a la operación de API `AdminInitiateAuth` (en lugar de `InitiateAuth`). Esta operación requiere AWS credenciales con permisos que incluyan `cognito-idp:AdminInitiateAuth` y`cognito-idp:AdminRespondToAuthChallenge`. Esta operación devuelve los parámetros de autenticación requeridos.
- Una vez que la aplicación del lado del servidor tiene los parámetros de autenticación, llama a la operación de la API `AdminRespondToAuthChallenge` (en lugar de `RespondToAuthChallenge`). La operación `AdminRespondToAuthChallenge` de la API solo se realiza correctamente si se proporcionan AWS las credenciales.

Para obtener más información sobre cómo firmar las solicitudes de la API de Amazon Cognito con AWS credenciales, consulte el proceso de firma de la versión 4 de Signature en la Referencia AWS general.

Las operaciones `AdminInitiateAuth` y la `AdminRespondToAuthChallenge` API no pueden aceptar credenciales de username-and-password usuario para el inicio de sesión de administrador, a menos que las habilite explícitamente de una de las siguientes maneras:

- Incluya `ALLOW_ADMIN_USER_PASSWORD_AUTH` (anteriormente llamado `ADMIN_NO_SRP_AUTH`) en el parámetro `ExplicitAuthFlow` cuando llame a `CreateUserPoolClient` o a `UpdateUserPoolClient`.

- Agregue `ALLOW_ADMIN_USER_PASSWORD_AUTH` a la lista de Flujos de autenticación para el cliente de la aplicación. Configure clientes de aplicaciones en la pestaña App integration (Integración de aplicaciones) en el grupo de usuarios, bajo App clients and analytics (Clientes de aplicaciones y análisis). Para obtener más información, consulte Clientes de aplicación de grupo de usuarios.

Flujo de autenticación personalizado

Los grupos de usuarios de Amazon Cognito también permiten utilizar flujos de autenticación personalizados, lo que puede ayudarle a crear un modelo de autenticación basado en desafíos/respuestas mediante activadores. AWS Lambda

> **Note**
>
> No puede utilizar las funciones de seguridad avanzadas para las credenciales comprometidas ni la autenticación adaptativa con flujos de autenticación personalizados. Para obtener más información, consulte Adición de seguridad avanzada a un grupo de usuarios.

El flujo de autenticación personalizado hace posible los ciclos de desafíos y respuestas personalizados para satisfacer diferentes requisitos. El flujo comienza con una llamada a la operación de la API `InitiateAuth`, que indica el tipo de autenticación que debe utilizarse y proporciona todos los parámetros de autenticación iniciales. Amazon Cognito responde a la llamada `InitiateAuth` con uno de los siguientes tipos de información:

- Un desafío para el usuario junto con una sesión y parámetros.
- Un error si el usuario no se autentica correctamente.
- Tokens de ID, acceso y actualización, si los parámetros proporcionados en la llamada `InitiateAuth` son suficientes para que el usuario inicie sesión. (Por lo general, el usuario o la aplicación deben responder primero a un desafío, pero el código personalizado debe determinarlo).

Si Amazon Cognito responde a la llamada `InitiateAuth` con un desafío, la aplicación reunirá más información y llamará a la operación `RespondToAuthChallenge`, lo que proporciona las respuestas al desafío y vuelve a pasar la sesión. Amazon Cognito responde a la llamada `RespondToAuthChallenge` de forma similar a la llamada `InitiateAuth`. Si el usuario ha iniciado

sesión, Amazon Cognito proporciona tokens o si el usuario no ha iniciado sesión, Amazon Cognito proporciona otro desafío o un error. Si devuelve otro desafío, la secuencia se repite y la aplicación llama a `RespondToAuthChallenge` hasta que el usuario inicie sesión correctamente o se devuelva un error. Para obtener más información sobre las operaciones de la API `InitiateAuth` and `RespondToAuthChallenge`, consulte la documentación de la API.

Flujo de autenticación integrado y desafíos

Amazon Cognito contiene algunos valores de `AuthFlow` y `ChallengeName` integrados para que un flujo de autenticación estándar pueda validar el nombre de usuario y la contraseña mediante el protocolo de contraseña remota segura (SRP). Los AWS SDK cuentan con soporte integrado para estos flujos con Amazon Cognito.

El flujo comienza enviando `USER_SRP_AUTH` como el `AuthFlow` a `InitiateAuth`. También envía los valores `USERNAME` y `SRP_A` en `AuthParameters`. Si la llamada `InitiateAuth` tiene éxito, la respuesta incluye `PASSWORD_VERIFIER` como `ChallengeName` y `SRP_B` en los parámetros del desafío. La aplicación llamará a continuación a `RespondToAuthChallenge` con el `ChallengeName` `PASSWORD_VERIFIER` y los parámetros necesarios en `ChallengeResponses`. Si la llamada a `RespondToAuthChallenge` se efectúa de manera correcta y el usuario inicia sesión, Amazon Cognito emite tokens. Si ha activado la autenticación multifactor (MFA) para el usuario, Amazon Cognito devuelve el `ChallengeName` de `SMS_MFA`. La aplicación puede proporcionar el código necesario a través de otra llamada a `RespondToAuthChallenge`.

Flujo de autenticación personalizado y desafíos

Una aplicación puede iniciar un flujo de autenticación personalizado llamando a `InitiateAuth` con `CUSTOM_AUTH` como `Authflow`. En el caso de un flujo de autenticación personalizado, tres desencadenadores de Lambda controlan los desafíos y la verificación de las respuestas.

- El desencadenador de Lambda `DefineAuthChallenge` toma como entrada una matriz de sesiones de desafíos y respuestas anteriores. Luego genera los siguientes nombres de desafíos y valores booleanos que indican si el usuario está autenticado y se le deben otorgar tokens. Este desencadenador de Lambda es una máquina de estado que controla la ruta que sigue el usuario a través de los desafíos.
- El desencadenador de Lambda `CreateAuthChallenge` toma el nombre de un desafío como entrada y genera el desafío y los parámetros para evaluar la respuesta. Cuando `DefineAuthChallenge` devuelve `CUSTOM_CHALLENGE` como el siguiente desafío, el flujo de autenticación llama a `CreateAuthChallenge`. El desencadenador de Lambda

`CreateAuthChallenge` supera el siguiente tipo de desafío del parámetro de metadatos del desafío.

- La función de Lambda `VerifyAuthChallengeResponse` evalúa la respuesta y devuelve un valor booleano para indicar si la respuesta ha sido válida.

Un flujo de autenticación personalizado también puede utilizar una combinación de desafíos integrados, como la verificación de contraseñas SRP y MFA mediante SMS. Puede usar desafíos personalizados como CAPTCHA o preguntas secretas.

Usar la verificación de contraseña de SRP en el flujo de autenticación personalizado

Si desea incluir SRP en un flujo de autenticación personalizado, debe comenzar con SRP.

- Para iniciar la verificación por contraseña de SRP en un flujo personalizado, la aplicación llama a `InitiateAuth` con `CUSTOM_AUTH` como `Authflow`. En la asignación de `AuthParameters`, la solicitud de la aplicación incluye `SRP_A:` (el valor de SRP A) y `CHALLENGE_NAME: SRP_A`.
- El flujo `CUSTOM_AUTH` invoca el desencadenador de Lambda `DefineAuthChallenge` con una sesión inicial de `challengeName: SRP_A` y `challengeResult: true`. La función de Lambda responder con `challengeName: PASSWORD_VERIFIER`, `issueTokens: false` y `failAuthentication: false`.
- A continuación, la aplicación debe llamar a `RespondToAuthChallenge` con `challengeName: PASSWORD_VERIFIER` y los demás parámetros necesarios para SRP en el mapa `challengeResponses`.
- Si Amazon Cognito verifica la contraseña, `RespondToAuthChallenge` llama al desencadenador de Lambda `DefineAuthChallenge` con una segunda sesión de `challengeName: PASSWORD_VERIFIER` y `challengeResult: true`. En ese momento, el desencadenador de Lambda `DefineAuthChallenge` responde con `challengeName: CUSTOM_CHALLENGE` para iniciar el desafío personalizado.
- Si MFA está habilitado para un usuario, una vez que Amazon Cognito verifique la contraseña, se le pide al usuario que configure o inicie sesión con MFA.

> **Note**
>
> La página web de inicio de sesión alojada de Amazon Cognito no puede activar Desencadenadores de Lambda de desafío de autenticación personalizado.

Para obtener más información sobre los desencadenadores de Lambda, incluido el código de muestra, consulte Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda.

Flujo de autenticación de administrador

La práctica recomendada para la autenticación consiste en utilizar las operaciones de la API descritas en Flujo de autenticación personalizado con SRP para la verificación de contraseñas. Los AWS SDK utilizan ese enfoque, y este enfoque les ayuda a utilizar SRP. Sin embargo, si desea evitar los cálculos de SRP, hay disponible un conjunto alternativo de operaciones de la API de administrador que se usa en servidores backend seguros. Para estas implementaciones de administrador de backend, utilice `AdminInitiateAuth` en lugar de `InitiateAuth`. También utilice `AdminRespondToAuthChallenge` en lugar de `RespondToAuthChallenge`. Dado que puede enviar la contraseña como texto sin formato, no tiene que realizar cálculos de SRP al utilizar estas operaciones. A continuación se muestra un ejemplo:

```
AdminInitiateAuth Request {
    "AuthFlow":"ADMIN_USER_PASSWORD_AUTH",
    "AuthParameters":{
        "USERNAME":"<username>",
        "PASSWORD":"<password>"
        },
    "ClientId":"<clientId>",
    "UserPoolId":"<userPoolId>"
}
```

Estas operaciones de autenticación de administrador requieren credenciales de desarrollador y el uso del proceso de firma de AWS Signature Version 4 (SigV4). Estas operaciones están disponibles en los SDK estándar de AWS , que incluyen Node.js, lo que es de gran utilidad para las funciones de Lambda. Para utilizar estas operaciones y hacer que acepten contraseñas como texto sin formato, debe activarlas para la aplicación en la consola. Alternativamente, puede pasar `ADMIN_USER_PASSWORD_AUTH` en el parámetro `ExplicitAuthFlow` en llamadas a `CreateUserPoolClient` o `UpdateUserPoolClient`. Las operaciones `InitiateAuth` y `RespondToAuthChallenge` no aceptan `ADMIN_USER_PASSWORD_AUTH` y `AuthFlow`.

En la respuesta `AdminInitiateAuth` de `ChallengeParameters`, el atributo `USER_ID_FOR_SRP`, si existe, contiene el verdadero nombre de usuario y no el alias del usuario (como la dirección de correo electrónico o un número de teléfono). En la llamada a `AdminRespondToAuthChallenge`, en `ChallengeResponses`, debe pasar este nombre de usuario en el parámetro `USERNAME`.

> **Note**
>
> Dado que las implementaciones de administrador de backend usan el flujo de autenticación de administrador, el flujo no admite el seguimiento de dispositivos. Si el seguimiento de dispositivos está activado, la autenticación de administrador se realiza correctamente, pero las llamadas de actualización del token de acceso fallan.

Flujo de autenticación de migración de usuarios

Un desencadenador de Lambda para la migración de usuarios ayuda a migrar usuarios desde un sistema de administración de usuarios heredado a un grupo de usuarios. Si elige el flujo de autenticación `USER_PASSWORD_AUTH`, no es necesario que los usuarios restablezcan sus contraseñas durante la migración de usuarios. Durante la autenticación, este flujo envía las contraseñas de los usuarios al servicio a través de una conexión SSL cifrada.

Cuando haya migrado todos los usuarios, cambie los flujos al flujo SRP más seguro. El flujo SRP no envía ninguna contraseña a través de la red.

Para obtener más información sobre los desencadenadores de Lambda, consulte Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda.

Para obtener más información acerca de la migración de usuarios con un desencadenador de Lambda, consulte Importación de usuarios a grupos de usuarios con un desencadenador de Lambda para la migración de usuarios.

Clientes de aplicación de grupo de usuarios

Un cliente de la aplicación del grupo de usuarios es una configuración dentro de un grupo de usuarios que interactúa con una aplicación móvil o web que se autentica con Amazon Cognito. Los clientes de aplicaciones pueden llamar a las operaciones de la API autenticadas y no autenticadas y leer o modificar algunos o todos los atributos de los usuarios. La aplicación se debe identificar ante el cliente de la aplicación en las operaciones para registrar, iniciar sesión y gestionar las contraseñas olvidadas. Estas solicitudes de la API deben incluir la autoidentificación con un ID de cliente de la aplicación y la autorización con un secreto de cliente opcional. Debe asegurar los ID o secretos de cliente de aplicación para que solo las aplicaciones de cliente autorizadas puedan llamar a estas operaciones no autenticadas. Además, si configuras tu aplicación para firmar solicitudes de API autenticadas con AWS credenciales, debes proteger tus credenciales para que no sean inspeccionadas por los usuarios.

Puede crear varias aplicaciones para un grupo de usuarios. Es posible que el cliente de una aplicación esté vinculado a la plataforma de código de una aplicación o a un inquilino independiente del grupo de usuarios. Por ejemplo, puede crear una aplicación para una aplicación del lado del servidor y una aplicación de Android diferente. Cada aplicación tiene su propio ID de cliente de aplicación.

Tipos de cliente de aplicación

Al crear un cliente de aplicación en Amazon Cognito, puede rellenar previamente las opciones según los tipos de cliente estándar de OAuth cliente público y cliente confidencial. Configure un cliente confidencial con un secreto del cliente. Para obtener más información sobre los tipos de cliente, consulte IETF RFC 6749 #2.1.

Cliente público

Un cliente público se ejecuta en un navegador o en un dispositivo móvil. Debido a que no tiene recursos de confianza del lado del servidor, no incluye ningún secreto del cliente.

Cliente confidencial

Un cliente confidencial tiene recursos del lado del servidor a los que se puede confiar un secreto del cliente para operaciones de la API no autenticadas. Es posible que la aplicación se ejecute como un daemon o script de shell en el servidor backend.

Secreto del cliente

Un secreto de cliente, o una contraseña de cliente, es una cadena fija que la aplicación debe usar en todas las solicitudes API al cliente de la aplicación. El cliente de la aplicación debe tener un secreto del cliente para ejecutar concesiones de `client_credentials`. Para obtener más información, consulte IETF RFC 6749 #2.3.1.

No puede cambiar secretos del cliente después de crear una aplicación. Puede crear una nueva aplicación con un nuevo secreto si quiere rotar el secreto. También puede eliminar una aplicación para bloquear el acceso de aplicaciones que utilizan el ID de cliente de dicha aplicación.

Puede utilizar un cliente confidencial y un secreto del cliente con una aplicación pública. Usa un CloudFront proxy de Amazon para añadir un objeto `SECRET_HASH` en tránsito. Para obtener más información, consulte Proteger los clientes públicos de Amazon Cognito mediante un CloudFront proxy de Amazon en el AWS blog.

Tokens web JSON

Los clientes de aplicación de Amazon Cognito pueden emitir tokens web JSON (JWT) de los siguientes tipos.

Token de identidad (ID)

Una instrucción verificable de que su usuario está autenticado a partir de su grupo de usuarios. OpenID Connect (OIDC) ha agregado la especificación de token de ID a los estándares de token de acceso y actualización definidos por OAuth 2.0. El token de ID contiene información de identidad, como atributos de usuario, que su aplicación puede utilizar para crear un perfil de usuario y aprovisionar recursos. Para obtener más información, consulte Uso del token de ID.

Token de acceso

Una instrucción verificable de los derechos de acceso de su usuario. El token de acceso contiene ámbitos, una característica de OIDC y OAuth 2.0. Su aplicación puede presentar ámbitos para recursos backend y demostrar que su grupo de usuarios autorizó a un usuario o máquina para acceder a datos de una API, o a sus propios datos de usuario. Un token de acceso con ámbitos personalizados, a menudo procedente de una concesión de credenciales de cliente M2M, autoriza el acceso a un servidor de recursos. Para obtener más información, consulte Uso del token de acceso.

Token de actualización

Una instrucción cifrada de autenticación inicial que su aplicación puede presentar a su grupo de usuarios cuando caduquen sus tokens de usuario. Una solicitud de actualización de token devuelve tokens de acceso e ID nuevos y no caducados. Para obtener más información, consulte Uso del token de actualización.

Puede establecer la caducidad de estos tokens para cada cliente de aplicación desde la pestaña Integración de aplicaciones de su grupo de usuarios en la consola de Amazon Cognito.

Condiciones de uso de la aplicación

Los siguientes términos son propiedades disponibles de los clientes de aplicación en la consola de Amazon Cognito.

URL de devolución de llamada permitidas

Una URL de devolución de llamada indica adónde se redirigirá al usuario tras iniciar sesión correctamente. Elija al menos una URL de devolución de llamada. La URL de devolución de llamada debe:

- Ser una URI absoluta.
- Estar registrada previamente con un cliente.
- No incluir un componente fragmento.

Consulte OAuth 2.0 - redirection endpoint (punto de enlace de redirección).

Amazon Cognito requiere `HTTPS` sobre `HTTP`, excepto para `http://localhost` solo con fines de prueba.

También se admiten las URL de devolución de llamada de aplicación como `myapp://example`.

URL de cierre de sesión permitidas

Una URL de cierre de sesión indica adónde se redirigirá al usuario después de cerrar la sesión.

Permisos de lectura y escritura de atributos

Es posible que su grupo de usuarios tenga muchos clientes, cada uno con su propio cliente de aplicación y IdPs. Puede configurar su cliente de aplicación para que tenga acceso de lectura y escritura solo a los atributos de usuario que sean relevantes para la aplicación. En casos como la autorización machine-to-machine (M2M), no puedes conceder acceso a ninguno de tus atributos de usuario.

Consideraciones para la configuración de los permisos de lectura y escritura de atributos

- Cuando crea un cliente de aplicación y no personaliza los permisos de lectura y escritura de atributos, Amazon Cognito concede permisos de lectura y escritura a todos los atributos del grupo de usuarios.
- Puede conceder acceso de escritura a atributos personalizados inmutables. Su cliente de aplicaciones puede escribir valores en atributos inmutables al crear o registrar un usuario. Después de esto, no puede escribir valores en ningún atributo personalizado inmutable para el usuario.
- Los clientes de aplicaciones deben tener acceso de escritura a los atributos requeridos de su grupo de usuarios. La consola de Amazon Cognito establece automáticamente los atributos requeridos para que se puedan escribir.

- No puede permitir que un cliente de aplicaciones tenga acceso de escritura a `email_verified` o `phone_number_verified`. Un administrador de grupo de usuarios puede modificar estos valores. Un usuario solo puede cambiar el valor de estos atributos mediante la verificación de atributos.

Flujos de autenticación

Los métodos que el cliente de su aplicación permite para el inicio de sesión. Su aplicación puede admitir autenticación con nombre de usuario y contraseña, contraseña remota segura (SRP), autenticación personalizada con desencadenadores de Lambda y actualización de token. Como práctica recomendada en materia de seguridad, utilice la autenticación SRP como método principal de inicio de sesión. La IU alojada inicia automáticamente la sesión de los usuarios con SRP.

Ámbitos personalizados

Un ámbito personalizado es el que se define para un servidor de recursos propio en Resource Servers (Recursos de servidores). El formato es *`resource-server-identifier`*/*`scope`*. Consulte Autorización de alcances, M2M y API con servidores de recursos.

URI de redireccionamiento predeterminado

Sustituye el `redirect_uri` parámetro en las solicitudes de autenticación de usuarios por otro de terceros IdPs. Configure esta configuración del cliente de la aplicación con el `DefaultRedirectURI` parámetro de una solicitud de UpdateUserPoolClientAPI CreateUserPoolClienteo una solicitud. Esta URL también debe ser miembro de la del cliente `CallbackURLs` de la aplicación. Amazon Cognito redirige las sesiones autenticadas a esta URL cuando:

1. El cliente de su aplicación tiene un proveedor de identidad asignado y varias URL de devolución de llamadas definidas. Tu grupo de usuarios redirige las solicitudes de autenticación al servidor de autorización al URI de redireccionamiento predeterminado cuando no incluyen ningún parámetro. `redirect_uri`
2. El cliente de la aplicación tiene un proveedor de identidad asignado y una URL de devolución de llamada definida. En este escenario, no es necesario definir una URL de devolución de llamada predeterminada. Las solicitudes que no incluyen un `redirect_uri` parámetro se redirigen a la única URL de devolución de llamada disponible.

Proveedores de identidades

Puedes elegir algunos o todos los proveedores de identidad externos (IdPs) de tu grupo de usuarios para autenticar a tus usuarios. Su cliente de aplicación también puede autenticar solo a

los usuarios locales de su grupo de usuarios. Cuando agregue un IdP a su cliente de aplicación, podrá generar enlaces de autorización al IdP y mostrarlos en su página de inicio de sesión de la interfaz de usuario alojada. Puede asignar varios IdPs, pero debe asignar al menos uno. Para obtener más información sobre el uso de fuentes externas IdPs, consulteAgregar inicio de sesión de grupo de usuarios a través de un tercero.

Ámbitos de OpenID Connect

Elija uno o varios de los siguientes ámbitos `OAuth` para especificar los privilegios de acceso que se pueden solicitar para los tokens de acceso.

- El ámbito de `openid` declara que desea recuperar un token de ID y un ID único de usuario. También solicita todos o algunos atributos de usuario, en función de los ámbitos adicionales de la solicitud. Amazon Cognito no devuelve un token de ID a menos que se solicite el ámbito `openid`. El ámbito de `openid` autoriza las reclamaciones de los token de ID estructurales, como la fecha de caducidad y el ID de clave y determina los atributos de usuario que se reciben en una respuesta de Punto de conexión de UserInfo.
  - Cuando `openid` es el único ámbito que solicita, Amazon Cognito rellena el token de ID con todos los atributos de usuario que el cliente de la aplicación actual pueda leer. La respuesta de `userInfo` a un token de acceso con este ámbito por sí solo devuelve todos los atributos del usuario.
  - Cuando solicita `openid` con otros ámbitos como `phone`, `email` o `profile`, el token de ID y `userInfo` devuelven el ID único del usuario y los atributos definidos por los ámbitos adicionales.
- El ámbito `phone` concede acceso a las notificaciones `phone_number` y `phone_number_verified`. Este ámbito solo se puede solicitar con el ámbito `openid`.
- El ámbito `email` concede acceso a las notificaciones `email` y `email_verified`. Este ámbito solo se puede solicitar con el ámbito `openid`.
- El `aws.cognito.signin.user.admin` ámbito otorga acceso a las operaciones de API de los grupos de usuarios de Amazon Cognito que requieren tokens de acceso, como UpdateUserAttributesy. VerifyUserAttribute
- El ámbito `profile` concede acceso a todos los atributos de usuario que el cliente puede leer. Este ámbito solo se puede solicitar con el ámbito `openid`.

Para obtener más información sobre los ámbitos, consulte la lista de ámbitos de OIDC estándar.

Tipos de concesión de OAuth

Una concesión de OAuth es un método de autenticación que recupera tokens de grupo de usuarios. Amazon Cognito admite los siguientes tipos de concesiones. Para integrar estas concesiones de OAuth en su aplicación, debe agregar un dominio a su grupo de usuarios.

Concesión de código de autorización

La concesión de código de autorización genera un código que su aplicación puede intercambiar por tokens de grupo de usuarios con el Punto de conexión de token. Cuando intercambia un código de autorización, su aplicación recibe tokens de identificación, acceso y actualización. Este flujo de OAuth, al igual que la concesión implícita, se produce en los navegadores de sus usuarios. Una concesión de código de autorización es la concesión más segura que ofrece Amazon Cognito, porque los tokens no son visibles en las sesiones de sus usuarios. En su lugar, su aplicación genera la solicitud que devuelve los tokens y puede almacenarlos en caché en un almacenamiento protegido. Para obtener más información, consulte Código de autorización en IETF RFC 6749 #1.3.1

> **Note**
>
> Como práctica recomendada de seguridad en las aplicaciones de cliente público, active solo el flujo OAuth de concesión de código de autorización e implemente la clave de prueba para el intercambio de códigos (PKCE) a fin de restringir el intercambio de tokens. Con PKCE, un cliente solo puede intercambiar un código de autorización cuando ha proporcionado al punto de conexión del token el mismo secreto que se presentó en la solicitud de autenticación original. Para obtener más información sobre PKCE, consulte IETF RFC 7636.

Implicit grant (Concesión implícita)

La concesión implícita entrega un token de acceso y de ID, pero no de actualización, a la sesión del navegador de su usuario directamente desde el Autorizar punto de conexión. Una concesión implícita elimina el requisito de una solicitud independiente al punto de conexión de tokens, pero no es compatible con PKCE y no devuelve tokens de actualización. Esta concesión se adapta a los escenarios de prueba y a la arquitectura de las aplicaciones que no pueden completar las concesiones de códigos de autorización. Para obtener más información, consulte Concesión implícita en IETF RFC 6749 #1.3.2. Puede activar tanto la concesión de código de autorización

como la concesión implícita en un cliente de aplicación y, a continuación, utilizar cada concesión según sea necesario.

Concesión de credenciales de cliente

La concesión de credenciales de cliente es para comunicaciones machine-to-machine (M2M). Las concesiones de código de autorización e implícitas emiten tokens a los usuarios humanos autenticados. Las credenciales de cliente conceden una autorización basada en el alcance desde un sistema no interactivo a una API. Su aplicación puede solicitar las credenciales del cliente directamente desde el punto de conexión del token y recibir un token de acceso. Para obtener más información, consulte Credenciales de cliente en IETF RFC 6749 #1.3.4. Solo puede activar concesiones de credenciales de cliente en clientes de aplicación que tengan un secreto de cliente y que no admitan concesiones de código de autorización o implícitas.

> **Note**
>
> Debido a que no invoca el flujo de credenciales de cliente como usuario, esta concesión solo puede agregar ámbitos personalizados a los tokens de acceso. Un ámbito personalizado es el que se puede definir para un servidor de recursos propio. Los ámbitos predeterminados como `openid` y `profile` no se aplican a los usuarios no humanos. Dado que los tokens de ID son una validación de los atributos de usuario, no son relevantes para la comunicación M2M, y un cliente de concesión de credenciales no los emite. Consulte Autorización de alcances, M2M y API con servidores de recursos.

La concesión de credenciales de cliente añade costes a su AWS factura. Para obtener más información, consulte Precios de Amazon Cognito.

Creación de un cliente de aplicación

AWS Management Console

Para crear un cliente de aplicación (consola)

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Seleccione la pestaña App integration (Integración de aplicaciones).

5. En App clients (Clientes de aplicaciones), seleccione Create an app client (Crear un cliente de aplicación).
6. Seleccione unaApp type:Cliente público,Cliente confidencial, o bienOtro.
7. Ingrese un nombre de cliente de aplicación.
8. Elija Generar un secreto de cliente para que Amazon Cognito genere un secreto de cliente para usted. Los secretos de cliente suelen asociarse a clientes confidenciales.
9. Seleccione los Authentication flows (Flujos de autenticación) que quiera permitir en su cliente de aplicaciones.
10. Configure la Authentication flow session duration (Duración de la sesión de flujo de autenticación). Esta es la cantidad de tiempo que tienen los usuarios para completar cada desafío de autenticación antes de que caduque el token de sesión.
11. (Opcional) Si desea configurar la caducidad del token, siga los pasos que se describen a continuación:

    a. Especifique el vencimiento del token de actualización para la aplicación. El valor predeterminado es 30 días. Puede cambiarlo por cualquier valor comprendido entre 1 hora y 10 años.

    b. Especifique el Access token expiration (Vencimiento del token de acceso) para el cliente de la aplicación. El valor predeterminado es de 1 hora. Puede cambiarlo por cualquier valor comprendido entre 5 minutos y 24 horas.

    c. Especifique el ID token expiration (Vencimiento del token de ID) para el cliente de la aplicación. El valor predeterminado es de 1 hora. Puede cambiarlo por cualquier valor comprendido entre 5 minutos y 24 horas.

    > **Important**
    >
    > Si utiliza la IU alojada y configura la duración del token con menos de una hora, el usuario final podrá obtener nuevos tokens basados en la duración de su cookie de sesión, que, en este momento, está fijada en una hora.

12. Elija si desea Enable token revocation (Habilitar revocación de tokens) para este cliente de aplicación. Esto aumentará el tamaño de los tokens que emite Amazon Cognito.
13. Elija si desea Evitar errores de existencia de usuarios para este cliente de aplicaciones. Amazon Cognito responderá a las solicitudes de inicio de sesión de usuarios inexistentes con un mensaje genérico que indica que el nombre de usuario o la contraseña son incorrectos.

14. Si desea utilizar la interfaz de usuario alojada con este cliente de aplicaciones, configure los Ajustes de la interfaz de usuario alojada.

    a. Ingrese una o más URL de devolución de llamada permitidas. Estas son las URL de la web o de la aplicación a las que desea que Amazon Cognito redirija a los usuarios una vez finalizada la autenticación.

    b. Ingrese una o más URL de cierre de sesión permitidas. Estas son las URL que quiere que la aplicación acepte en las solicitudes a Punto de conexión Logout.

    c. Elija uno o más Proveedores de identidad con los que quiera que puedan iniciar sesión los usuarios en la aplicación. Puede elegir cualquier combinación de las existentes IdPs. Puede autenticar a los usuarios solo con su grupo de usuarios o con uno o más terceros IdPs que haya configurado en su grupo de usuarios.

    d. Elija los Tipos de concesión OAuth 2.0 que quiere que acepte el cliente de aplicaciones.

    - Seleccione Concesión de códigos de autorización para transferir códigos a la aplicación que pueda canjearlos por tokens con Punto de conexión de token.
    - Seleccione Concesión implícita para transferir el ID y los tokens de acceso directamente a la aplicación. El flujo de concesiones implícitas expone los tokens directamente a los usuarios.
    - Seleccione Credenciales de cliente para transferir los tokens de acceso a la aplicación en función de su conocimiento no de las credenciales de usuario, sino del secreto del cliente. El flujo de concesión de credenciales del cliente se excluye mutuamente, con el código de autorización y los flujos de concesión implícitos.

    e. Elija los Ámbitos OpenID Connect que desea autorizar para usar con el cliente de aplicaciones. Puede generar tokens de acceso solo con el ámbito `aws.cognito.signin.user.admin` a través de la API de grupos de usuarios. Para obtener ámbitos adicionales, debe solicitar los tokens de acceso de Punto de conexión de token.

    f. Elija los Ámbitos personalizados que desea autorizar con el cliente de aplicaciones. Los ámbitos personalizados se utilizan con mayor frecuencia para autorizar el acceso a las API de terceros.

15. Configure Permisos de lectura y escritura de atributos para este cliente de aplicaciones. El cliente de aplicaciones puede tener permiso para leer y escribir todo, un subconjunto limitado del esquema de atributos del grupo de usuarios.

16. Elija Create app client (Crear cliente de aplicación).

17. Anote el Id de cliente. Esto identificará al cliente de aplicación en las solicitudes de registro e inicio de sesión.

AWS CLI

```
aws cognito-idp create-user-pool-client --user-pool-id MyUserPoolID --client-
name myApp
```

> **Note**
>
> Utilice el formato JSON para las direcciones URL de devolución de llamada y de cierre de sesión con el fin de evitar que la CLI las considere archivos de parámetros remotos:
>
> ```
> --callback-urls "["https://example.com"]"
> --logout-urls "["https://example.com"]"
> ```

Consulte la referencia de AWS CLI comandos para obtener más información: create-user-pool-client

Amazon Cognito user pools API

Genera una solicitud CreateUserPoolClientde API. Debe especificar un valor para todos los parámetros que no desee establecer en un valor predeterminado.

Actualización de un grupo de usuarios, una aplicación, un cliente (AWS CLI y una AWS API)

En el AWS CLI, introduzca el siguiente comando:

```
aws cognito-idp update-user-pool-client --user-pool-id  "MyUserPoolID" --client-id
 "MyAppClientID" --allowed-o-auth-flows-user-pool-client --allowed-o-auth-flows "code"
 "implicit" --allowed-o-auth-scopes "openid" --callback-urls "["https://example.com"]"
 --supported-identity-providers "["MySAMLIdP", "LoginWithAmazon"]"
```

Si el comando se ejecuta correctamente, AWS CLI devuelve una confirmación:

```
{
```

```
    "UserPoolClient": {
        "ClientId": "MyClientID",
        "SupportedIdentityProviders": [
            "LoginWithAmazon",
            "MySAMLIdP"
        ],
        "CallbackURLs": [
            "https://example.com"
        ],
        "AllowedOAuthScopes": [
            "openid"
        ],
        "ClientName": "Example",
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "RefreshTokenValidity": 30,
        "AuthSessionValidity": 3,
        "CreationDate": 1524628110.29,
        "AllowedOAuthFlowsUserPoolClient": true,
        "UserPoolId": "MyUserPoolID",
        "LastModifiedDate": 1530055177.553
    }
}
```

Consulte la referencia de AWS CLI comandos para obtener más información: update-user-pool-client.

AWS API: UpdateUserPoolClient

Obtener información sobre un grupo de usuarios, un cliente de aplicaciones (AWS CLI y una AWS API)

```
aws cognito-idp describe-user-pool-client --user-pool-id MyUserPoolID --client-
id MyClientID
```

Consulte la referencia de AWS CLI comandos para obtener más información: describe-user-pool-client.

AWS API: DescribeUserPoolClient

Listar toda la información del cliente de la aplicación en un grupo de usuarios (AWS CLI y AWS API)

```
aws cognito-idp list-user-pool-clients --user-pool-id "MyUserPoolID" --max-results 3
```

Consulte la referencia de AWS CLI comandos para obtener más información: list-user-pool-clients.

AWS API: ListUserPoolClients

Eliminar un grupo de usuarios, una aplicación, un cliente (AWS CLI y una AWS API)

```
aws cognito-idp delete-user-pool-client --user-pool-id "MyUserPoolID" --client-id
 "MyAppClientID"
```

Consulte la referencia de AWS CLI comandos para obtener más información: delete-user-pool-client

AWS API: DeleteUserPoolClient

Uso de dispositivos de usuario en el grupos de usuarios

Al iniciar sesión en los usuarios de un grupo de usuarios local con la API de grupos de usuarios de Amazon Cognito, puede asociar los registros de actividad de los usuarios procedentes de las características de seguridad avanzadas a cada uno de sus dispositivos y, de forma opcional, permitir que los usuarios se salten la autenticación multifactor (MFA) si utilizan un dispositivo de confianza. Amazon Cognito incluye una clave de dispositivo en la respuesta a cualquier inicio de sesión que no incluya información del dispositivo. La clave del dispositivo está en el formato *Region*_*UUID*. Con una clave de dispositivo, una biblioteca de contraseñas remotas seguras (SRP) y un grupo de usuarios que permita la autenticación de dispositivos, puede pedir a los usuarios de su aplicación que confíen en el dispositivo actual y dejar de solicitar un código de MFA al iniciar sesión.

Temas

- Configuración de dispositivos recordados
- Obtención de la clave del dispositivo
- Inicio de sesión con un dispositivo
- Visualización, actualización y olvido de dispositivos

Configuración de dispositivos recordados

Con los grupos de usuarios de Amazon Cognito, puede asociar los dispositivos de cada uno de sus usuarios a un identificador de dispositivo único: una clave de dispositivo. Al presentar la clave del dispositivo y realizar la autenticación del dispositivo al iniciar sesión, puede utilizar dos características.

1. Gracias a las características de seguridad avanzadas, puede monitorizar la actividad de los usuarios en dispositivos concretos con fines de seguridad y análisis. Cuando los usuarios inician sesión, su aplicación tiene la opción de autenticar a cada usuario y su dispositivo, agregando la información del dispositivo a sus registros de actividad.
2. La función de recordar dispositivos también admite un flujo de autenticación de dispositivos de confianza, en el que los usuarios pueden optar por iniciar sesión sin MFA durante el periodo de tiempo adecuado a los requisitos de seguridad de la aplicación. Cuando quiera volver a solicitar al usuario que envíe un código de MFA, puede cambiar el estado recordado de su dispositivo.

   Los dispositivos recordados solo pueden anular la MFA en grupos de usuarios con MFA activa.

Cuando el usuario inicia sesión con un dispositivo recordado, debe realizar una autenticación de dispositivo adicional durante el flujo de autenticación. Para obtener más información, consulte Inicio de sesión con un dispositivo.

Configure su grupo de usuarios para que recuerde dispositivos en la pestaña Experiencia de inicio de sesión del grupo de usuarios, en Seguimiento de dispositivos. Cuando configura la funcionalidad de recordar los dispositivos a través de la consola de Amazon Cognito, dispone de tres opciones: Always (Siempre), User Opt-In (Activación por usuario) y No.

No recordar

Su grupo de usuarios no sugiere a los usuarios que se recuerden los dispositivos cuando inician sesión.

Recordar siempre

Cuando la aplicación confirma el dispositivo de un usuario, su grupo de usuarios siempre recuerda el dispositivo y no devuelve errores de MFA cuando se inicia sesión correctamente en el dispositivo en el futuro.

Opción de usuario

> Cuando la aplicación confirma el dispositivo de un usuario, su grupo de usuarios no suprime automáticamente los desafíos de la MFA. Debe presentar un mensaje para que el usuario elija si quiere que se recuerde su dispositivo.

Al elegir Recordar siempre o Opción de usuario, Amazon Cognito genera una clave de identificación del dispositivo y un secreto cada vez que un usuario inicia sesión desde un dispositivo no identificado. La clave del dispositivo es el identificador inicial que la aplicación envía al grupo de usuarios cuando el usuario autentica el dispositivo.

Con cada dispositivo de usuario confirmado, ya sea que se recuerde automáticamente o por opción de usuario, puede usar la clave y el secreto del identificador del dispositivo para autenticar un dispositivo cada vez que un usuario inicie sesión.

También puede configurar los ajustes de dispositivos recordados para su grupo de usuarios en una solicitud de API CreateUserPool o UpdateUserPool. Para obtener más información, consulte la propiedad DeviceConfiguration.

La API de grupos de usuarios de Amazon Cognito tiene operaciones adicionales para recordar dispositivos.

1. ListDevices y AdminListDevices devuelven una lista de claves del dispositivo y sus metadatos para un usuario.
2. GetDevice y AdminGetDevice devuelven la clave del dispositivo y los metadatos de un solo dispositivo.
3. UpdateDeviceStatus y AdminUpdateDeviceStatus establecen el dispositivo del usuario como recordado o no recordado.
4. ForgetDevice y AdminForgetDevice eliminan el dispositivo confirmado de un usuario de su perfil.

Las operaciones de API con nombres que comiencen por `Admin` se utilizan en aplicaciones del lado del servidor y deben autorizarse con credenciales de IAM. Para obtener más información, consulte Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios.

Obtención de la clave del dispositivo

Cada vez que el usuario inicia sesión con la API de grupos de usuarios y no incluye una clave de dispositivo en los parámetros de autenticación como `DEVICE_KEY`, Amazon Cognito devuelve una nueva clave de dispositivo en la respuesta. En su aplicación pública del lado del cliente, coloque la clave del dispositivo en el almacenamiento de la aplicación para poder incluirla en futuras solicitudes. En su aplicación confidencial del lado del servidor, configure una cookie del navegador u otro token del lado del cliente con la clave del dispositivo de su usuario.

Para que el usuario pueda iniciar sesión con su dispositivo de confianza, la aplicación debe confirmar la clave del dispositivo y proporcionar información adicional. Genere una solicitud ConfirmDevice para Amazon Cognito que confirme el dispositivo de su usuario con la clave del dispositivo, un nombre descriptivo, un verificador de contraseñas y una sal. Si ha configurado su grupo de usuarios para la autenticación de dispositivos según la opción del usuario, Amazon Cognito responde a su solicitud `ConfirmDevice` con una pregunta para que el usuario elija si desea que se recuerde el dispositivo actual. Responda con la selección de su usuario en una solicitud UpdateDeviceStatus.

Cuando confirma el dispositivo de su usuario pero no lo configura como recordado, Amazon Cognito guarda la asociación, pero continúa con el inicio de sesión sin dispositivo cuando proporciona la clave del dispositivo. Los dispositivos pueden generar registros que son útiles para la seguridad del usuario y la solución de problemas. Un dispositivo confirmado pero no recordado no aprovecha la característica de inicio de sesión, pero sí la característica de registros de monitorización de la seguridad. Al activar las características de seguridad avanzadas para el cliente de la aplicación y codificar la huella de un dispositivo en la solicitud, Amazon Cognito asocia los eventos de usuario con el dispositivo confirmado.

Para obtener una nueva clave de dispositivo

1. Empiece la sesión de inicio de sesión de su usuario con una solicitud de API InitiateAuth.
2. Responda a todos los desafíos de autenticación con RespondToAuthChallenge hasta recibir los tokens web JSON (JWT) que marcan que la sesión de inicio de sesión de su usuario ha finalizado.
3. En su aplicación, registre los valores que Amazon Cognito devuelve en `NewDeviceMetadata` en su respuesta `RespondToAuthChallenge` o `InitiateAuth`: `DeviceGroupKey` y `DeviceKey`.
4. Genere un nuevo secreto de SRP para su usuario: una sal y un verificador de contraseñas. Esta función está disponible en los SDK que proporcionan bibliotecas SRP.
5. Solicite al usuario un nombre de dispositivo o genere uno a partir de las características del dispositivo del usuario.

6. Proporcione el token de acceso, la clave del dispositivo, el nombre del dispositivo y el secreto SRP del usuario en una solicitud de API ConfirmDevice. Si su grupo de usuarios está configurado para Recordar siempre los dispositivos, el registro del usuario se habrá completado.
7. Si Amazon Cognito responde a `ConfirmDevice` con `"UserConfirmationNecessary": true`, pida al usuario que elija si quiere que se recuerde el dispositivo. Si afirma que quiere que se recuerde el dispositivo, genere una solicitud a la API UpdateDeviceStatus con el token de acceso del usuario, la clave del dispositivo y `"DeviceRememberedStatus": "remembered"`.
8. Si ha indicado a Amazon Cognito que recuerde el dispositivo, la próxima vez que inicie sesión, en lugar de un desafío de MFA, se le presentará un desafío `DEVICE_SRP_AUTH`.

Inicio de sesión con un dispositivo

Tras configurar el dispositivo de un usuario para que se recuerde, Amazon Cognito ya no le exige que envíe un código de MFA cuando inicie sesión con la misma clave de dispositivo. La autenticación del dispositivo solo reemplaza el desafío de autenticación de MFA por un desafío de autenticación del dispositivo. Los usuarios no pueden iniciar sesión únicamente con la autenticación del dispositivo. El usuario debe completar primero la autenticación con su contraseña o con un desafío personalizado. A continuación se muestra el proceso de autenticación de un usuario en un dispositivo recordado.

Para realizar la autenticación del dispositivo en un flujo que utilice Desencadenadores de Lambda de desafío de autenticación personalizado, transfiera un parámetro `DEVICE_KEY` en su solicitud de API InitiateAuth. Cuando el usuario supere todos los desafíos y el desafío `CUSTOM_CHALLENGE` devuelva un valor `issueTokens` de `true`, Amazon Cognito devolverá un último desafío `DEVICE_SRP_AUTH`.

Para iniciar sesión con un dispositivo

1. Recupere la clave del dispositivo de su usuario del almacenamiento del cliente.
2. Empiece la sesión de inicio de sesión de su usuario con una solicitud de API InitiateAuth. Elija un `AuthFlow` de `USER_SRP_AUTH`, `REFRESH_TOKEN_AUTH`, `USER_PASSWORD_AUTH` o `CUSTOM_AUTH`. En `AuthParameters`, agregue la clave del dispositivo de su usuario al parámetro `DEVICE_KEY` e incluya los demás parámetros necesarios para el flujo de inicio de sesión seleccionado.
   a. También puede transferir `DEVICE_KEY` en los parámetros de una respuesta `PASSWORD_VERIFIER` a un desafío de autenticación.
3. Complete las respuestas al desafío hasta que reciba un desafío `DEVICE_SRP_AUTH` en la respuesta.

4. En una solicitud de API RespondToAuthChallenge, envíe un `ChallengeName` de `DEVICE_SRP_AUTH` y parámetros para `USERNAME`, `DEVICE_KEY` y `SRP_A`.
5. Amazon Cognito responde con un desafío `DEVICE_PASSWORD_VERIFIER`. Esta respuesta al desafío incluye valores para `SECRET_BLOCK` y `SRP_B`.
6. Con su biblioteca SRP, genere y envíe los parámetros `PASSWORD_CLAIM_SIGNATURE`, `PASSWORD_CLAIM_SECRET_BLOCK`, `TIMESTAMP`, `USERNAME` y `DEVICE_KEY`. Envíelos en una solicitud `RespondToAuthChallenge` adicional.
7. Complete los desafíos adicionales hasta recibir los JWT del usuario.

El siguiente pseudocódigo muestra cómo calcular los valores para la respuesta al desafío `DEVICE_PASSWORD_VERIFIER`.

```
PASSWORD_CLAIM_SECRET_BLOCK = SECRET_BLOCK
TIMESTAMP = Tue Sep 25 00:09:40 UTC 2018
PASSWORD_CLAIM_SIGNATURE = Base64(SHA256_HMAC(K_USER, DeviceGroupKey + DeviceKey +
 PASSWORD_CLAIM_SECRET_BLOCK + TIMESTAMP))
K_USER = SHA256_HASH(S_USER)
S_USER = (SRP_B - k * gx)(a + ux)
x = SHA256_HASH(salt + FULL_PASSWORD)
u = SHA256_HASH(SRP_A + SRP_B)
k = SHA256_HASH(N + g)
```

Visualización, actualización y olvido de dispositivos

Puede implementar las siguientes características en su aplicación con la API de Amazon Cognito.

1. Mostrar información sobre el dispositivo actual de un usuario.
2. Mostrar una lista de todos los dispositivos del usuario.
3. Olvidar un dispositivo.
4. Actualizar el estado recordado de un dispositivo.

Los tokens de acceso que autorizan las solicitudes de API en las siguientes descripciones deben incluir el ámbito `aws.cognito.signin.user.admin`. Amazon Cognito agrega una notificación para este ámbito a todos los tokens de acceso que genere con la API de grupos de usuarios de Amazon Cognito. Los IdP de terceros deben gestionar por separado los dispositivos y la MFA de los usuarios que se autentiquen en Amazon Cognito. En la interfaz de usuario alojada, puede

solicitar el ámbito `aws.cognito.signin.user.admin`, pero la interfaz de usuario alojada agrega automáticamente la información del dispositivo a los registros de usuarios de seguridad avanzada y no permite recordar los dispositivos.

Visualización de información sobre un dispositivo

Puede consultar información sobre el dispositivo de un usuario para determinar si todavía está en uso. Por ejemplo, es posible que desees desactivar los dispositivos recordados después de que no hayan iniciado sesión durante 90 días.

- Para mostrar la información del dispositivo del usuario en una aplicación de cliente público, envíe la clave de acceso y la clave del dispositivo del usuario en una solicitud de API GetDevice.
- Para mostrar la información del dispositivo de su usuario en una aplicación de cliente confidencial, firme una solicitud de API AdminGetDevice con las credenciales de AWS y envíe el nombre del usuario, la clave del dispositivo y el grupo de usuarios del usuario.

Visualización de una lista de todos los dispositivos del usuario

Puede mostrar una lista de todos los dispositivos de sus usuarios y sus propiedades. Por ejemplo, es posible que desee comprobar que el dispositivo actual coincide con un dispositivo recordado.

- En una aplicación de cliente público, envíe el token de acceso del usuario en una solicitud de API ListDevices.
- En una aplicación de cliente confidencial, firme una solicitud de API AdminListDevices con las credenciales de AWS y envíe el nombre del usuario y el grupo de usuarios del usuario.

Olvido de un dispositivo

Puede eliminar la clave del dispositivo de un usuario. Puede que desee hacer esto cuando determine que el usuario ya no usa un dispositivo o cuando detecte una actividad inusual y desee solicitar al usuario que vuelva a completar la MFA. Para volver a registrar el dispositivo más adelante, debe generar y almacenar una nueva clave de dispositivo.

- En una aplicación de cliente público, envíe la clave del dispositivo y el token de acceso del usuario en una solicitud de API ForgetDevice.
- En una aplicación de cliente confidencial, envíe la clave del dispositivo y el token de acceso del usuario en una solicitud de API AdminForgetDevice.

Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios

Para el registro, el inicio de sesión y la administración de usuarios de su grupo de usuarios, tiene dos opciones.

1. Los puntos de conexión del grupo de usuarios incluyen la interfaz de usuario alojada y los puntos de conexión de federación. Constituyen un paquete de páginas web públicas que Amazon Cognito activa cuando elija un dominio para el grupo de usuarios. Para comenzar rápidamente a utilizar las características de autenticación y autorización de los grupos de usuarios de Amazon Cognito, incluidas las páginas de registro, inicio de sesión, administración de contraseñas y autenticación multifactor (MFA), utilice la interfaz de usuario integrada de la interfaz de usuario alojada. Los otros puntos de conexión del grupo de usuarios facilitan la autenticación con proveedores de identidades (IdP) de terceros. Los servicios que se prestan incluyen los siguientes.
   a. Puntos de conexión de devolución de llamadas del proveedor de servicios para reclamaciones autenticadas de los IdP, como `saml2/idpresponse` y `oauth2/idpresponse`. Cuando Amazon Cognito es un proveedor de servicios (SP) intermedio entre la aplicación y el IdP, los puntos de conexión de devolución de llamada representan el servicio.
   b. Puntos de conexión que proporcionan información sobre el entorno, como `oauth2/userInfo` y `jwks.json`. La aplicación usa estos puntos de conexión cuando verifica los tokens o recupera los datos del perfil de usuario con los AWS SDK y las bibliotecas de OAuth 2.0.
2. La API de grupos de usuarios de Amazon Cognito es un conjunto de herramientas para la aplicación web o móvil, después recopila información de inicio de sesión en el frontend propio personalizado, para autenticar a los usuarios. La autenticación de la API de grupos de usuarios produce los siguientes JSON Web Tokens.
   a. Un token de identidad con afirmaciones de atributos verificables por parte del usuario.
   b. Un token de acceso que autoriza al usuario a crear solicitudes de API autorizadas por tokens para un punto de conexión de servicio de AWS.

   > **Note**
   >
   > De forma predeterminada, los tokens de acceso de la autenticación de la API de los grupos de usuarios solo contienen el ámbito de `aws.cognito.signin.user.admin`. Para generar un token de acceso con ámbitos adicionales para, por ejemplo, autorizar una solicitud a una API de terceros, solicite ámbitos durante la autenticación a través de los puntos de conexión del grupo de usuarios o agregue ámbitos personalizados en una

Desencadenador de Lambda anterior a la generación del token. La personalización del token de acceso agrega costes a su factura de AWS.

Puede vincular un usuario federado, que normalmente iniciaría sesión a través de los puntos de conexión de los grupos de usuarios, con un usuario cuyo perfil sea local del grupo de usuarios. Un usuario local existe exclusivamente en el directorio del grupo de usuarios sin federación a través de un IdP externo. Si vincula la identidad federada a un usuario local en una solicitud de la API AdminLinkProviderForUser, este podrá iniciar sesión con la API de los grupos de usuarios. Para obtener más información, consulte Vinculación de usuarios federados a un perfil de usuario existente.

La API de grupos de usuarios de Amazon Cognito tiene una doble finalidad. Crea y configura sus recursos de grupos de usuarios de Amazon Cognito. Por ejemplo, puede crear grupos de usuarios, agregar desencadenadores de AWS Lambda y configurar el dominio de su IU alojada. La API de los grupos de usuarios también realiza el registro, el inicio de sesión y otras operaciones de usuario para los usuarios locales y enlazados.

Escenario de ejemplo con la API de grupos de usuarios de Amazon Cognito

1. El usuario selecciona el botón "Create an account" (Crear una cuenta) que ha creado en su aplicación. Ingresa una dirección de correo electrónico y una contraseña.
2. Su aplicación envía una solicitud de API SignUp y crea un nuevo usuario en su grupo de usuarios.
3. Su aplicación solicita a su usuario un código de confirmación por correo electrónico. Su usuario ingresa el código que ha recibido en un mensaje de correo electrónico.
4. Su aplicación envía una solicitud de API ConfirmSignUp con el código de confirmación del usuario.
5. La aplicación solicita al usuario el nombre de usuario y la contraseña y este ingresa las información.
6. Su aplicación envía una solicitud de API InitiateAuth y almacena un token de ID, un token de acceso y un token de actualización. Su aplicación llama a las bibliotecas OIDC para administrar los tokens de usuarios y mantener una sesión persistente para ese usuario.

En la API de grupos de usuarios de Amazon Cognito, no se puede registrar a los usuarios que se federan a través de un IdP. Debe autenticar a estos usuarios mediante los puntos de conexión del grupo de usuarios. Para obtener más información sobre los puntos de conexión del grupo de

usuarios que incluyen la interfaz de usuario alojada, consulte Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada. Sus usuarios federados pueden comenzar en la interfaz de usuario alojada y seleccionar su IdP, o puede omitir la interfaz de usuario alojada y enviar a sus usuarios directamente a su IdP para iniciar sesión. Cuando su solicitud de API al Autorizar punto de conexión incluye un parámetro de IdP, Amazon Cognito redirige silenciosamente a su usuario a la página de inicio de sesión del IdP.

Escenario de ejemplo con puntos de conexión de grupo de usuarios

1. El usuario selecciona el botón "Create an account" (Crear una cuenta) que ha creado en su aplicación.
2. Se presenta a su usuario una lista de los proveedores de identidades sociales en los que ha registrado credenciales de desarrollador. El usuario elige Apple.
3. Su aplicación inicia una solicitud al Autorizar punto de conexión con nombre de proveedor `SignInWithApple`.
4. El navegador de su usuario abre la página de autorización de OAuth de Apple. Su usuario elige permitir que Amazon Cognito lea su información de perfil.
5. Amazon Cognito confirma el token de acceso de Apple y consulta el perfil de Apple de su usuario.
6. El usuario presenta un código de autorización de Amazon Cognito en la aplicación.
7. Su aplicación intercambia el código de autorización con el Punto de conexión de token y almacena un token de ID, un token de acceso y un token de actualización. Su aplicación llama a las bibliotecas OIDC para administrar los tokens de usuarios y mantener una sesión persistente para ese usuario.

La API y los puntos de conexión de los grupos de usuarios admiten diversos escenarios, que se describen a lo largo de esta guía. En las secciones siguientes se examina cómo la API de grupos de usuarios se divide a su vez en clases que respaldan los requisitos de registro, inicio de sesión y administración de recursos.

Operaciones de API autenticadas y no autenticadas de los grupos de usuarios de Amazon Cognito

La API de grupos de usuarios de Amazon Cognito, que es a la vez una interfaz de administración de recursos y una interfaz de autenticación y autorización orientada al usuario, combina en sus operaciones los modelos de autorización que se indican a continuación. Según la operación de

la API, es posible que tenga que proporcionar autorización con credenciales de IAM, un token de acceso, un token de sesión, un secreto de cliente o una combinación de estos. Para muchas operaciones de autenticación y autorización de usuarios, puede elegir entre versiones autenticadas y no autenticadas de la solicitud. Las operaciones no autenticadas son la práctica recomendada en materia de seguridad para las aplicaciones que distribuye a sus usuarios, como las aplicaciones móviles; no necesita incluir ningún secreto en el código.

Solo puede asignar permisos en las políticas de IAM para Operaciones de administración autenticadas por IAM y Operaciones de usuario autenticadas por IAM.

Operaciones de administración autenticadas por IAM

Las operaciones de administración autenticadas por IAM le permiten modificar y visualizar la configuración de sus grupos de usuarios y clientes de aplicación, como lo haría en la AWS Management Console.

Por ejemplo, para modificar su grupo de usuarios en una solicitud de API UpdateUserPool, debe presentar credenciales de AWS y permisos de IAM para actualizar el recurso.

Para autorizar estas solicitudes en la AWS Command Line Interface (AWS CLI) o en un SDK de AWS, configure su entorno con variables de entorno o una configuración de cliente que agregue credenciales de IAM a su solicitud. Para obtener más información, consulte Acceso a AWS con las credenciales de AWS en la Referencia general de AWS. También puede enviar solicitudes directamente a los puntos de conexión del servicio para la API de los grupos de usuarios de Amazon Cognito. Debe autorizar, o firmar, estas solicitudes con las credenciales de AWS que inserte en el encabezado de la solicitud. Para obtener más información, consulte Firma de solicitudes de API de AWS.

Operaciones de administración autenticadas por IAM

`AddCustomAttributes`

`CreateGroup`

`CreateIdentityProvider`

`CreateResourceServer`

`CreateUserImportJob`

Operaciones de administración autenticadas por IAM

`CreateUserPool`

`CreateUserPoolClient`

`CreateUserPoolDomain`

`DeleteGroup`

`DeleteIdentityProvider`

`DeleteResourceServer`

`DeleteUserPool`

`DeleteUserPoolClient`

`DeleteUserPoolDomain`

`DescribeIdentityProvider`

`DescribeResourceServer`

`DescribeRiskConfiguration`

`DescribeUserImportJob`

`DescribeUserPool`

`DescribeUserPoolClient`

`DescribeUserPoolDomain`

`GetCSVHeader`

`GetGroup`

`GetIdentityProviderByIdentifier`

`GetSigningCertificate`

`GetUICustomization`

Operaciones de administración autenticadas por IAM

`GetUserPoolMfaConfig`

`ListGroups`

`ListIdentityProviders`

`ListResourceServers`

`ListTagsForResource`

`ListUserImportJobs`

`ListUserPoolClients`

`ListUserPools`

`ListUsers`

`ListUsersInGroup`

`SetRiskConfiguration`

`SetUICustomization`

`SetUserPoolMfaConfig`

`StartUserImportJob`

`StopUserImportJob`

`TagResource`

`UntagResource`

`UpdateGroup`

`UpdateIdentityProvider`

`UpdateResourceServer`

`UpdateUserPool`

Operaciones de administración autenticadas por IAM

`UpdateUserPoolClient`

`UpdateUserPoolDomain`

Operaciones de usuario autenticadas por IAM

Las operaciones de usuario autenticadas por IAM permiten el registro, inicio de sesión, administración de credenciales, modificación y visualización de sus usuarios.

Por ejemplo, puede tener un nivel de aplicación en el servidor que respalde un front-end web. Su aplicación en el servidor es un cliente confidencial de OAuth al que usted confía el acceso con privilegios a sus recursos de Amazon Cognito. Para registrar un usuario en la aplicación, su servidor puede incluir credenciales de AWS en una solicitud de API AdminCreateUser. Para obtener más información sobre los tipos de cliente de OAuth, consulte Client Types (Tipos de cliente) en The OAuth 2.0 Authorization Framework (Marco de autorización de OAuth 2.0).

Para autorizar estas solicitudes en la AWS CLI o en un SDK de AWS, configure su entorno de aplicaciones en el servidor con variables de entorno o una configuración de cliente que agregue credenciales de IAM a su solicitud. Para obtener más información, consulte Acceso a AWS con las credenciales de AWS en la Referencia general de AWS. También puede enviar solicitudes directamente a los puntos de conexión del servicio para la API de los grupos de usuarios de Amazon Cognito. Debe autorizar, o firmar, estas solicitudes con las credenciales de AWS que inserte en el encabezado de la solicitud. Para obtener más información, consulte Firma de solicitudes de API de AWS.

Si su cliente de aplicación tiene un secreto de cliente, deberá proporcionar tanto sus credenciales de IAM como, en función de la operación, el parámetro `SecretHash` o el valor `SECRET_HASH` en `AuthParameters`. Para obtener más información, consulte Cálculo de los valores de hash secretos.

Operaciones de usuario autenticadas por IAM

`AdminAddUserToGroup`

`AdminConfirmSignUp`

`AdminCreateUser`

Operaciones de usuario autenticadas por IAM

`AdminDeleteUser`

`AdminDeleteUserAttributes`

`AdminDisableProviderForUser`

`AdminDisableUser`

`AdminEnableUser`

`AdminForgetDevice`

`AdminGetDevice`

`AdminGetUser`

`AdminInitiateAuth`

`AdminLinkProviderForUser`

`AdminListDevices`

`AdminListGroupsForUser`

`AdminListUserAuthEvents`

`AdminRemoveUserFromGroup`

`AdminResetUserPassword`

`AdminRespondToAuthChallenge`

`AdminSetUserMFAPreference`

`AdminSetUserPassword`

`AdminSetUserSettings`

`AdminUpdateAuthEventFeedback`

`AdminUpdateDeviceStatus`

Operaciones de usuario autenticadas por IAM

`AdminUpdateUserAttributes`

`AdminUserGlobalSignOut`

Operaciones de usuario no autenticadas

Operaciones de usuario no autenticadas: registran, inician sesión e inician el restablecimiento de contraseñas para sus usuarios. Utilice operaciones de API no autenticadas, o públicas, cuando desee que cualquier usuario de Internet se registre e inicie sesión en su aplicación.

Por ejemplo, para registrar a un usuario en su aplicación, puede distribuir un cliente público de OAuth que no proporcione ningún acceso con privilegios a los secretos. Puede registrar a este usuario con la operación de API no autenticada SignUp.

Para enviar estas solicitudes en un cliente público que haya desarrollado con un SDK de AWS, no necesita configurar ninguna credencial. También puede enviar solicitudes directamente a los puntos de conexión del servicio para la API de grupos de usuarios de Amazon Cognito sin autorización adicional.

Si su cliente de aplicación tiene un secreto de cliente, deberá proporcionar, según la operación, el parámetro `SecretHash` o el valor `SECRET_HASH` en `AuthParameters`. Para obtener más información, consulte Cálculo de los valores de hash secretos.

Operaciones de usuario no autenticadas

`SignUp`

`ConfirmSignUp`

`ResendConfirmationCode`

`ForgotPassword`

`ConfirmForgotPassword`

`InitiateAuth`

Operaciones de usuario autorizadas por tokens

Las operaciones de usuario autorizadas por token permiten cerrar la sesión de los usuarios, administrar las credenciales de los usuarios, modificarlos y visualizarlos después de que hayan iniciado sesión o hayan comenzado dicho proceso. Utilice las operaciones de la API autorizadas por token cuando no desee distribuir secretos en la aplicación y desee autorizar las solicitudes con las propias credenciales del usuario. Si el usuario ha completado el inicio de sesión, debe autorizar la solicitud de la API autorizada por token con un token de acceso. Si el usuario se encuentra en medio de un proceso de inicio de sesión, deberá autorizar la solicitud de la API autorizada por token con un token de sesión que Amazon Cognito le haya devuelto en la respuesta a la solicitud anterior.

Por ejemplo, en un cliente público, es posible que desee actualizar el perfil de un usuario de forma que se restrinja el acceso de escritura solo al propio perfil del usuario. Para realizar esta actualización, su cliente puede incluir el token de acceso del usuario en una solicitud de API UpdateUserAttributes.

Para enviar estas solicitudes en un cliente público que haya desarrollado con un SDK de AWS, no necesita configurar ninguna credencial. Incluya un parámetro `AccessToken` o `Session` en su solicitud. También puede enviar solicitudes directamente a los puntos de conexión del servicio para la API de los grupos de usuarios de Amazon Cognito. Para autorizar una solicitud a un punto de conexión de servicio, incluya el token de acceso o de sesión en el cuerpo POST de la solicitud.

Para firmar una solicitud de la API para una operación autorizada por un token, incluya el token de acceso como un encabezado de `Authorization` en la solicitud, en el formato `Bearer` *`<Base64-encoded access token>`*.

Operaciones de usuario autorizadas por tokens	AccessToken	Session (Sesión)
`RespondToAuthChallenge`		✓
`ChangePassword`	✓	
`GetUser`	✓	
`UpdateUserAttributes`	✓	

Operaciones de usuario autorizadas por tokens	AccessToken	Session (Sesión)
DeleteUserAttributes	✓	
DeleteUser	✓	
ConfirmDevice	✓	
ForgetDevice	✓	
GetDevice	✓	
ListDevices	✓	
UpdateDeviceStatus	✓	
GetUserAttributeVerificationCode	✓	
VerifyUserAttribute	✓	
SetUserSettings	✓	
SetUserMFAPreference	✓	
GlobalSignOut	✓	
AssociateSoftwareToken	✓	✓
UpdateAuthEventFeedback		✓

Operaciones de usuario autorizadas por tokens	AccessToken	Session (Sesión)
`VerifySoftwareToken`	✓	✓
`RevokeToken` [1]		

[1] RevokeToken toma un token de actualización como parámetro. El token de actualización sirve de token de autorización y de recurso de destino.

Actualización de la configuración del grupo de usuarios

Para cambiar la configuración de los grupos de usuarios de Amazon Cognito en AWS Management Console, navegue por las pestañas basadas en funciones de la configuración del grupo de usuarios y actualice los campos tal y como se describe en otras áreas de esta guía. No se pueden cambiar algunos ajustes después de crear un grupo de usuarios. Si desea cambiar la siguiente configuración, debe crear un nuevo grupo de usuarios o un cliente de aplicaciones.

Nombre de grupo de usuarios

Nombre del parámetro de la API: PoolName

Nombre descriptivo que ha asignado a su grupo de usuarios. Para cambiar el nombre de un grupo de usuarios, cree un nuevo grupo de usuarios.

Opciones de inicio de sesión del grupo de usuarios de Amazon Cognito

Nombres de los parámetros de la API: AliasAttributesy UsernameAttributes

Los atributos que los usuarios pueden pasar como nombre de usuario cuando inician sesión. Cuando se crea un grupo de usuarios, se puede optar por permitir el inicio de sesión con el nombre de usuario, la dirección de correo electrónico, el número de teléfono o un nombre de usuario preferido. Para cambiar las opciones de inicio de sesión del grupo de usuarios, cree un nuevo grupo de usuarios.

Make user name case sensitive (En el nombre de usuario se distinguirán mayúsculas de minúsculas)

Nombre del parámetro de la API: UsernameConfiguration

Cuando cree un nombre de usuario que coincida con otro nombre de usuario, excepto por la distinción de mayúsculas y minúsculas, Amazon Cognito puede tratarlo como el mismo usuario o como usuarios únicos. Para obtener más información, consulte Sensibilidad de mayúsculas y minúsculas en el grupo de usuarios. Para cambiar la distinción de mayúsculas y minúsculas, cree un nuevo grupo de usuarios.

Secreto del cliente

Nombre del parámetro de la API: GenerateSecret

Cuando crea un cliente de aplicación, puede generar un secreto de cliente para que solo las fuentes de confianza puedan realizar solicitudes a su grupo de usuarios. Para obtener más información, consulte Clientes de aplicación de grupo de usuarios. Para cambiar un secreto de cliente, cree un nuevo cliente de aplicaciones en el mismo grupo de usuarios.

Atributos obligatorios

Nombre del parámetro de API: Schema

Los atributos para los que los usuarios deben proporcionar valores cuando se registran o cuando los crea. Para obtener más información, consulte Custom pool attributes (. Para cambiar los atributos obligatorios, cree un nuevo grupo de usuarios.

Custom attributes (Atributos personalizados)

Nombre del parámetro de API: Schema

Atributos con nombres personalizados. Puede cambiar el valor del atributo personalizado de un usuario, pero no puede eliminar un atributo personalizado de su grupo de usuarios. Para obtener más información, consulte Custom pool attributes (. Si alcanza el número máximo de atributos personalizados y desea modificar la lista, cree un nuevo grupo de usuarios.

Configuración de SMS

Después de activar los mensajes SMS en su grupo de usuarios, no podrá desactivarlos.

- Si elige configurar los mensajes SMS al crear un grupo de usuarios, no podrá desactivar los SMS una vez finalizada la configuración.
- Puede activar los mensajes SMS en un grupo de usuarios que haya creado, pero después no podrá desactivar los SMS.

- Amazon Cognito puede utilizar los mensajes SMS para la invitación y recuperación de cuentas de usuario, la verificación de atributos y la autenticación multifactorial (MFA). Tras activar los mensajes SMS, puede activar o desactivar los mensajes SMS para estas funciones en cualquier momento.
- La configuración de los mensajes SMS incluye una función de IAM que puede delegar en Amazon Cognito para enviar mensajes con Amazon SNS. Puede cambiar el rol asignado en cualquier momento.

Actualizar un grupo de usuarios con un AWS SDK o una API REST AWS CDK

En la consola de Amazon Cognito, puede cambiar la configuración del grupo de usuarios parámetro por parámetro. Por ejemplo, para agregar un disparador Lambda, elija Agregar disparador Lambda y elija la función y el tipo de disparador. La API de grupos de usuarios de Amazon Cognito está estructurada de manera que las operaciones de actualización de los grupos de usuarios y los clientes de aplicaciones requieren el conjunto completo de parámetros del grupo de usuarios. Sin embargo, la consola automatiza de forma transparente esta operación de actualización con las demás configuraciones del grupo de usuarios.

Es posible que, en ocasiones, un cambio en alguna parte de su Cuenta de AWS equipo provoque que las actualizaciones generen un error si no están relacionadas con la configuración que desea cambiar. Una identidad de Amazon SES eliminada o un cambio en un permiso de IAM AWS WAF, por ejemplo. Si uno de los parámetros actuales ya no es válido, no podrá actualizar la configuración hasta que lo corrija. Cuando te encuentres con un error de este tipo, examina la respuesta al error y valida la configuración que menciona.

Los AWS Cloud Development Kit (AWS CDK)AWSSDK y la API REST de los grupos de usuarios de Amazon Cognito son herramientas para la automatización y la configuración programática de los recursos de Amazon Cognito. Las solicitudes con estas herramientas también deben, como la consola de Amazon Cognito, actualizar una configuración con una configuración de recursos completa en el cuerpo de la solicitud. En un nivel superior, debe realizar el siguiente proceso.

1. Capture el resultado de una operación que describa la configuración del recurso existente.
2. Modifique la salida con los cambios de configuración.
3. Envíe la configuración modificada en una operación que actualice su recurso.

El siguiente procedimiento actualiza la configuración con la operación de la UpdateUserPoolAPI. El mismo enfoque, con diferentes campos de entrada, se aplica a UpdateUserPoolClient.

> **Important**
>
> Si no proporciona valores para parámetros existentes, Amazon Cognito los establece en valores predeterminados. Por ejemplo, cuando tienes `LambdaConfig` y envías un `UpdateUserPool` con un `LambdaConfig` vacío, elimina la asignación de todas las funciones de Lambda de los desencadenadores del grupo de usuarios. Planifique en consecuencia cuando desee automatizar los cambios en la configuración del grupo de usuarios.

1. Capture el estado actual de su grupo de usuarios con DescribeUserPool.
2. Asigne el formato a la salida de `DescribeUserPool` para coincidir con los parámetros de solicitud de `UpdateUserPool`. Elimine los siguientes campos de nivel superior y sus objetos secundarios del JSON de salida.
   - `Arn`
   - `CreationDate`
   - `CustomDomain`
     - Actualice este campo con la operación UpdateUserPoolDomainde la API.
   - `Domain`
     - Actualiza este campo con la operación UpdateUserPoolDomainde la API.
   - `EmailConfigurationFailure`
   - `EstimatedNumberOfUsers`
   - `Id`
   - `LastModifiedDate`
   - `Name`
   - `SchemaAttributes`
   - `SmsConfigurationFailure`
   - `Status`
3. Confirme que el JSON resultante coincida con los parámetros de solicitud de `UpdateUserPool`.

4. Modifique los parámetros que desee cambiar en el JSON resultante.
5. Envíe una solicitud de API `UpdateUserPool` con el JSON modificado como entrada de solicitud.

También puede utilizar esta salida de `DescribeUserPool` modificada en el parámetro `--cli-input-json` de `update-user-pool` en la AWS CLI.

Como alternativa, ejecute el siguiente AWS CLI comando para generar JSON con valores en blanco para los campos de entrada aceptados. `update-user-pool` A continuación, puede rellenar estos campos con los valores existentes de su grupo de usuarios.

```
aws cognito-idp update-user-pool --generate-cli-skeleton --output json
```

Utilice el siguiente comando para generar el mismo objeto JSON para un cliente de aplicación.

```
aws cognito-idp update-user-pool-client --generate-cli-skeleton --output json
```

Configuración y uso de la interfaz de usuario alojada y los puntos de conexión de federación de Amazon Cognito

Un grupo de usuarios de Amazon Cognito con un dominio es un servidor de autorización compatible con OAuth-2.0 y una interfaz de usuario (UI) ready-to-use alojada para la autenticación. El servidor de autorización enruta las solicitudes de autenticación, emite y administra los tokens web JSON (JWT) y proporciona información sobre los atributos del usuario. La interfaz de usuario alojada es un conjunto de interfaces web para las actividades básicas de registro, inicio de sesión, autenticación multifactorial y restablecimiento de contraseñas en el grupo de usuarios. También es un centro central para la autenticación con los proveedores de identidad externos (IdPs) que asocie a su aplicación. La aplicación puede invocar la interfaz de usuario alojada y los puntos de conexión de autorización cuando desee autenticar y autorizar a los usuarios. Puede hacer que la experiencia de usuario de la interfaz de usuario alojada se adapte a la marca con el logotipo propio y la personalización de CSS. Para obtener más información sobre los componentes de la interfaz de usuario alojada y el servidor de autorización, consulte Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada.

> **Note**
>
> La interfaz de usuario alojada de Amazon Cognito no admite autenticación personalizada con desencadenadores de Lambda de desafío de autenticación personalizados.

Temas

Configurar la interfaz de usuario alojada con AWS Amplify

Si utilizas la autenticación AWS Amplify para tu aplicación web o móvil, puedes configurar la interfaz de usuario alojada mediante la interfaz de línea de comandos (CLI) y las bibliotecas del AWS Amplify marco. Para añadir la autenticación a la aplicación, utilice la CLI de AWS Amplify para añadir la categoría `Auth` al proyecto. A continuación, en el código de cliente, utiliza las AWS Amplify bibliotecas para autenticar a los usuarios con su grupo de usuarios de Amazon Cognito.

Puede mostrar una interfaz de usuario alojada prediseñada o federar a usuarios a través de un punto de enlace de OAuth 2.0 que redirige a un proveedor de inicio de sesión de redes sociales, como Facebook, Google, Amazon o Apple. Después de que un usuario se autentique correctamente con el proveedor social, AWS Amplify crea un nuevo usuario en el grupo de usuarios si es necesario y proporciona el token de OIDC del usuario a la aplicación.

Los siguientes ejemplos muestran cómo configurar la interfaz AWS Amplify de usuario alojada con los proveedores sociales de su aplicación.

- AWS Amplify autenticación para JavaScript.
- AWS Amplify autenticación para Swift.
- AWS Amplify autenticación para Flutter.

- AWS Amplify autenticación para Android.

Configuración de la IU alojada con la consola de Amazon Cognito

Creación de un cliente de aplicación

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Seleccione la pestaña App integration (Integración de aplicaciones).
5. En App clients (Clientes de aplicaciones), seleccione Create an app client (Crear un cliente de aplicación).
6. Seleccione un App type (Tipo de aplicación): Public client (Cliente público), Confidential client (Cliente confidencial), o bien Other (Otro). Un Public client (Cliente público) normalmente funciona desde los dispositivos de los usuarios y utiliza API no autenticadas y autenticadas por tokens. Un cliente confidencial normalmente funciona desde una aplicación en un servidor central en el que se confían los secretos del cliente y las credenciales de la API, y utiliza encabezados y AWS Identity and Access Management credenciales de autorización para firmar las solicitudes. Si su caso de uso es diferente de la configuración del cliente de aplicación preconfigurada para un Public client (Cliente público) o un Confidential client (Cliente confidencial), seleccione Other (Otro).
7. Introduzca un nombre de cliente de aplicación.
8. Seleccione los Authentication flows (Flujos de autenticación) que quiera permitir en su cliente de aplicaciones.
9. Configure la Authentication flow session duration (Duración de la sesión de flujo de autenticación). Esta es la cantidad de tiempo que tienen los usuarios para completar cada desafío de autenticación antes de que caduque el token de sesión.
10. (Opcional) Configure la caducidad del token.

    a. Especifique el Refresh token expiration (Vencimiento del token de actualización) para el cliente de la aplicación. El valor predeterminado es 30 días. Puede cambiarlo por cualquier valor comprendido entre 1 hora y 10 años.

    b. Especifique el Access token expiration (Vencimiento del token de acceso) para el cliente de la aplicación. El valor predeterminado es de 1 hora. Puede cambiarlo por cualquier valor comprendido entre 5 minutos y 24 horas.

c. Especifique el ID token expiration (Vencimiento del token de ID) para el cliente de la aplicación. El valor predeterminado es de 1 hora. Puede cambiarlo por cualquier valor comprendido entre 5 minutos y 24 horas.

> **Important**
>
> Si utiliza la IU alojada y configura la duración del token con menos de una hora, el usuario final podrá obtener nuevos tokens basados en la duración de su cookie de sesión, que, en este momento, está fijada en una hora.

11. Elija Generate client secret (Generar secreto de cliente) para que Amazon Cognito genere un secreto de cliente para usted. Los secretos de cliente suelen asociarse a clientes confidenciales.
12. Elija si desea Enable token revocation (Habilitar revocación de tokens) para este cliente de aplicación. Esto aumentará el tamaño de los tokens. Para obtener más información, consulte Revoking Tokens (Revocación de tokens).
13. Elija si desea Prevent error messages that reveal user existence (Evitar los mensajes de error que revelan la existencia del usuario) para este cliente de aplicación. Amazon Cognito responderá a las solicitudes de inicio de sesión de usuarios inexistentes con un mensaje genérico que indica que el nombre de usuario o la contraseña son incorrectos.
14. (Opcional) Configure Permisos de lectura y escritura de atributos para este cliente de aplicación. El cliente de aplicaciones puede tener permiso para leer y escribir un subconjunto limitado del esquema de atributos de su grupo de usuarios.
15. Seleccione Crear.
16. Anote el Id de cliente. Esto identificará al cliente de aplicación en las solicitudes de registro e inicio de sesión.

Configuración de la aplicación

1. En el pestaña App integration (Integración de aplicaciones), seleccione su cliente de aplicación en App clients (Clientes de aplicaciones). Revisión de la información actual sobre la Hosted UI (IU alojada).
2. Add a callback URL (Agregar una URL de devolución de llamada) en Allowed callback URL(s) (Direcciones URL de devolución de llamada permitidas). Una URL de devolución de llamada indica adónde se redirigirá al usuario tras iniciar sesión correctamente.

3. Add a sign-out URL (Agregar una URL de cierre de sesión) en Allowed sign-out URL(s) (Direcciones URL de cierre de sesión permitidas). Una URL de cierre de sesión indica adónde se redirigirá al usuario después de cerrar la sesión.
4. Agregue al menos una de las opciones que se muestran de la lista de Identity providers (Proveedores de identidad).
5. En OAuth 2.0 grant types (Tipos de concesiones OAuth 2.0), seleccione Authorization code grant (Concesión de código de autorización) para devolver un código de autorización que se intercambie por tokens de grupos de usuarios. Debido a que los tokens nunca se exponen directamente a un usuario final, es menos probable que se vean comprometidos. Sin embargo, se requiere una aplicación personalizada en el backend para intercambiar el código de autorización para tokens de grupos de usuarios. Por motivos de seguridad, le recomendamos utilizar el flujo de concesión de código de autorización junto con PKCE (Proof Key for Code Exchange, clave de prueba para intercambio de código) para las aplicaciones móviles.
6. En Allowed OAuth Flows (Flujos de OAuth 2.0 permitidos), seleccione Implicit grant (Concesión implícita) para que se devuelvan tokens web de JSON (JWT) del grupo de usuarios desde Amazon Cognito. Puede utilizar este flujo cuando no hay un backend disponible para intercambiar un código de autorización para tokens. También es útil para depurar tokens.
7. Puede habilitar tanto Authorization code grant (Concesión de código de autorización) como Implicit code grant (Concesión de código implícita) y, a continuación, utilizar cada concesión según sea necesario. Si no se seleccionan las concesiones Authorization code (Código de autorización) o Implicit code (Código implícito) y su cliente de aplicación tiene un secreto de cliente, puede habilitar las concesiones de Client credentials (Credenciales del cliente). Seleccione Client credentials (Credenciales del cliente) solo si la aplicación debe solicitar tokens de acceso en su propio nombre y no en nombre de un usuario.
8. Seleccione los OpenID Connect scopes (Ámbitos OpenID Connect) que desea autorizar para este cliente de aplicación.
9. Elija Guardar cambios.

Configuración de un dominio

1. Vaya a la pestaña App integration (Integración de aplicaciones) para su grupo de usuarios.
2. Junto a Domain (Dominio), elija Actions (Acciones) y seleccione Create custom domain (Crear dominio personalizado) o Create Cognito domain (Crear dominio Cognito). Si ya ha configurado un dominio de grupo de usuarios, elija Delete Cognito domain (Eliminar dominio de Cognito)

o Delete custom domain (Eliminar dominio personalizado) antes de crear el nuevo dominio personalizado.

3. Introduzca un prefijo de dominio disponible para utilizarlo con un Cognito domain (Dominio Cognito). Para obtener información sobre cómo configurar un Custom domain (Dominio personalizado), consulte Uso de un propio dominio con la interfaz de usuario alojada
4. Seleccione Crear.

Consulta de la página de inicio de sesión

En la consola de Amazon Cognito, seleccione el botón View Hosted UI (Ver IU alojada) en la configuración del cliente de la aplicación, en App clients and analytics (Clientes de aplicaciones y análisis) en la pestaña App integration (Integración de aplicaciones). Este botón le llevará a una página de inicio de sesión en su IU alojada con los siguientes parámetros básicos.

- El ID de cliente de aplicación.
- Una solicitud de concesión de código de autorización
- Una solicitud para todos los ámbitos que ha activado para el cliente de la aplicación actual
- La primera URL de devolución de llamada de la lista para el cliente de aplicación actual

El botón View hosted UI (Ver IU alojada) es útil cuando se quiere probar las funciones básicas de la interfaz de usuario alojada. Puede personalizar la URL de inicio de sesión con parámetros adicionales y modificados. En la mayoría de casos, los parámetros generados automáticamente del enlace de View hosted UI (Ver IU alojada) no se ajustan completamente a las necesidades de su aplicación. En estos casos, debe personalizar la URL que invoca su aplicación cuando inicia sesión en sus usuarios. Para obtener más información acerca de los parámetros y valores de los parámetros, consulte Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada.

La página web de inicio de sesión de la interfaz de usuario alojada utiliza el siguiente formato. En este ejemplo se solicita la concesión de un código de autorización con el parámetro `response_type=code`.

```
https://<your domain>/oauth2/authorize?response_type=code&client_id=<your app client
 id>&redirect_uri=<your callback url>
```

Puede recuperar la cadena de dominio del grupo de usuarios desde la pestaña Integración de aplicaciones. En la misma pestaña, puede identificar los ID de los clientes de la aplicación, las URL de devolución de llamadas, los ámbitos permitidos y otras configuraciones en Clientes y análisis de aplicaciones.

Cuando navegue hasta el punto de conexión de `/oauth2/authorize` con sus parámetros personalizados, Amazon Cognito lo redirige al punto de conexión de `/oauth2/login` o, si tiene un parámetro `identity_provider` o `idp_identifier`, lo redirige silenciosamente a la página de inicio de sesión de su IdP. Para ver un ejemplo de URL que omite la interfaz de usuario alojada, consulte Inicio de sesión SAML en grupos de usuarios de Amazon Cognito.

Solicitud de interfaz de usuario alojada de ejemplo para una adjudicación implícita

Puede ver la página web de inicio de sesión de la interfaz de usuario alojada con la siguiente dirección URL para la adjudicación de código implícita, donde `response_type=token`. Tras un inicio de sesión correcto, Amazon Cognito devuelve tokens de grupo de usuarios a su barra de direcciones de navegador web.

```
            https://mydomain.us-east-1.amazoncognito.com/authorize?
response_type=token&client_id=1example23456789&redirect_uri=https://
mydomain.example.com
```

Los tokens de identidad y acceso aparecen como parámetros adjuntos a la URL de redireccionamiento.

A continuación, se muestra una respuesta de ejemplo de una solicitud de concesión implícita.

```
            https://mydomain.example.com/
#id_token=eyJraaBcDeF1234567890&access_token=eyJraGhIjKlM1112131415&expires_in=3600&token_type=
```

Información que debe saber sobre la interfaz de usuario alojada en los grupos de usuarios de Amazon Cognito

La interfaz de usuario alojada y la confirmación de los usuarios como administradores

Para los usuarios locales del grupo de usuarios, la interfaz de usuario alojada funciona mejor cuando se configura el grupo de usuarios para Permitir que Cognito envíe mensajes automáticamente para verificar y confirmar. Al habilitar esta configuración, Amazon Cognito envía un mensaje con un código de confirmación a los usuarios que se registren. Si, en cambio, confirma a los usuarios como administradores del grupo de usuarios, la interfaz de usuario alojada muestra un mensaje de error tras el registro. En este estado, Amazon Cognito ha creado el nuevo usuario, pero no ha podido enviar un mensaje de verificación. Aún puede confirmar a los usuarios como administradores, pero es posible que se pongan en contacto con el servicio de asistencia cuando detecten un error. Para obtener más información sobre la confirmación administrativa, consulte Permitir que los usuarios se registren en la aplicación, pero con confirmación del administrador del grupo de usuarios.

Consulta de los cambios en la configuración de la interfaz de usuario alojada

Si los cambios en las páginas de la IU alojada no aparecen inmediatamente, espere unos minutos y, a continuación, actualice la página.

Descodificación de los tokens del grupo de usuarios

Los tokens del grupo de usuarios de Amazon Cognito se firman con un algoritmo RS256. Para decodificar y verificar los tokens del grupo de usuarios AWS Lambda, consulte Decodificar y verificar los tokens JWT de Amazon Cognito en. GitHub

La interfaz de usuario alojada y la versión TLS

La interfaz de usuario alojada requiere el cifrado en tránsito. Los dominios de grupos de usuarios proporcionados por Amazon Cognito requieren una versión TLS 1.2 como mínimo. Los dominios personalizados admiten la versión 1.2 de TLS, pero no la requieren. Dado que Amazon Cognito administra la configuración de la interfaz de usuario alojada y los puntos de enlace del servidor de autorización, no puede modificar los requisitos de TLS del dominio de su grupo de usuarios.

La interfaz de usuario alojada y las políticas de CORS

La IU alojada de Amazon Cognito no admite políticas de origen de uso compartido de recursos entre orígenes (CORS). Una política CORS en la interfaz de usuario alojada impediría que los usuarios pasaran parámetros de autenticación en sus solicitudes. En su lugar, implemente una política de CORS en la interfaz web de su aplicación. Amazon Cognito devuelve un encabezado de respuesta `Access-Control-Allow-Origin: *` a las solicitudes a los siguientes puntos de conexión de OAuth.

1. Punto de conexión de token

2. Revocación de puntos de conexión
3. Punto de conexión de UserInfo

Cookies de interfaz de usuario alojada y servidor de autorización

Los puntos finales del grupo de usuarios de Amazon Cognito configuran cookies en los navegadores de los usuarios. Las cookies cumplen con los requisitos de algunos navegadores de que los sitios no instalen cookies de terceros. Están dirigidas únicamente a los puntos finales de su grupo de usuarios e incluyen lo siguiente:

- Una `XSRF-TOKEN` cookie para cada solicitud.
- Una `csrf-state` cookie para garantizar la coherencia de la sesión cuando se redirige a un usuario.
- Cookie de `cognito` sesión que conserva los intentos de inicio de sesión correctos durante una hora.

Configuración de un dominio del grupo de usuarios

Después de configurar un cliente de aplicación, puede configurar la dirección de las páginas web de inscripción e inicio de sesión. Puede utilizar un dominio alojado de Amazon Cognito y elegir un prefijo de dominio disponible, o puede utilizar su propia dirección web como dominio personalizado.

Para agregar un cliente de aplicación y un dominio de Amazon Cognito alojado con la AWS Management Console, consulte Adición de una aplicación para habilitar la IU web alojada.

> **Note**
>
> No puede usar el texto `aws`, `amazon` ni `cognito` en el prefijo de dominio.

Temas

- Uso del dominio de Amazon Cognito con la IU alojada
- Uso de un dominio propio con la IU alojada

Uso del dominio de Amazon Cognito con la IU alojada

Después de configurar un cliente de aplicación, puede configurar la dirección de las páginas web de registro e inicio de sesión. Puede utilizar el dominio de Amazon Cognito alojado con su propio prefijo de dominio.

> **Note**
>
> Para aumentar la seguridad de sus aplicaciones de Amazon Cognito, los dominios principales de los puntos de conexión del grupo de usuarios se registran en la lista pública de sufijos (PSL). La PSL ayuda a los navegadores web de sus usuarios a establecer una comprensión coherente de los puntos de conexión de su grupo de usuarios y de las cookies que establecen.
> Los dominios principales de punto de conexión de grupo de usuarios adoptan los siguientes formatos.
>
> ```
> auth.Region.amazoncognito.com
> auth-fips.Region.amazoncognito.com
> ```

Para añadir un cliente de aplicación y un dominio alojado en Amazon Cognito con AWS Management Console, consulte. Creación de un cliente de aplicación

Temas

- Requisitos previos
- Paso 1: Configurar un dominio alojado de grupo de usuarios
- Paso 2: Verificar la página de inicio de sesión

Requisitos previos

Antes de comenzar, necesitará:

- Un grupo de usuarios con un cliente de aplicación. Para obtener más información, consulte Introducción a los grupos de usuarios.

Paso 1: Configurar un dominio alojado de grupo de usuarios

Para configurar un dominio alojado de grupo de usuarios

Puede usar la API o la AWS Management Console AWS CLI o para configurar un dominio de grupo de usuarios.

Amazon Cognito console

Configuración de un dominio

1. Vaya a la pestaña App integration (Integración de aplicaciones) para su grupo de usuarios.
2. Junto a Dominio, elija Acciones y seleccione Crear dominio personalizado o Crear dominio de Amazon Cognito. Si ya ha configurado un dominio de grupo de usuarios, elija Eliminar dominio de Amazon Cognito o Eliminar dominio personalizado antes de crear el nuevo dominio personalizado.
3. Ingrese un prefijo de dominio disponible para utilizarlo con un dominio de Amazon Cognito. Para obtener información sobre cómo configurar un Custom domain (Dominio personalizado), consulte Uso de un propio dominio con la interfaz de usuario alojada
4. Seleccione Crear.

CLI/API

Utilice los siguientes comandos para crear un prefijo de dominio y asignarlo al grupo de usuarios.

Para configurar un dominio de grupo de usuarios

- AWS CLI: `aws cognito-idp create-user-pool-domain`

  Ejemplo: `aws cognito-idp create-user-pool-domain --user-pool-id <user_pool_id> --domain <domain_name>`
- AWS API: CreateUserPoolDomain

Para obtener información acerca de un dominio

- AWS CLI: `aws cognito-idp describe-user-pool-domain`

  Ejemplo: `aws cognito-idp describe-user-pool-domain --domain <domain_name>`

- AWS API: DescribeUserPoolDomain

Para eliminar un dominio

- AWS CLI: `aws cognito-idp delete-user-pool-domain`

  Ejemplo: `aws cognito-idp delete-user-pool-domain --domain <domain_name>`
- AWS API: DeleteUserPoolDomain

Paso 2: Verificar la página de inicio de sesión

- Compruebe si la página de inicio de sesión está disponible desde el dominio alojado de Amazon Cognito.

```
https://<your_domain>/login?
response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback_url>
```

El dominio aparece en la página Domain name (Nombre del dominio) de la consola de Amazon Cognito. El ID del cliente de aplicación y la URL de devolución de llamada se muestran en la página App client settings (Configuración del cliente de aplicación).

Uso de un dominio propio con la IU alojada

Después de configurar un cliente de aplicación, puede configurar el grupo de usuarios con un dominio personalizado para la IU alojada de Amazon Cognito y puntos de conexión de la API Auth. Con los dominios personalizados, los usuarios pueden iniciar sesión en la aplicación utilizando su propia dirección web.

Temas

- Adición de un dominio personalizado a un grupo de usuarios
- Cambio del certificado SSL en el dominio personalizado

Adición de un dominio personalizado a un grupo de usuarios

Para agregar un dominio personalizado al grupo de usuarios, debe especificar el nombre de dominio en la consola de Amazon Cognito y proporcionar un certificado que administre con AWS Certificate

Manager (ACM). Una vez agregado el dominio, Amazon Cognito ofrece un destino de alias, que debe agregarse a la configuración de DNS.

Requisitos previos

Antes de comenzar, necesitará:

- Un grupo de usuarios con un cliente de aplicación. Para obtener más información, consulte Introducción a los grupos de usuarios.
- Un dominio web de su propiedad. Su dominio principal debe tener un registro DNS A válido. Puede asignar cualquier valor a este registro. El elemento principal puede ser la raíz del dominio o un dominio secundario que esté un paso más arriba en la jerarquía de dominios. Por ejemplo, si el dominio personalizado es auth.xyz.example.com, Amazon Cognito debe poder resolver xyz.example.com a una dirección IP. Para evitar un impacto accidental en la infraestructura del cliente, Amazon Cognito no admite el uso de dominios de nivel superior (TLD) para dominios personalizados. Para obtener más información, consulte Nombres de dominio.
- Tener la capacidad para crear un subdominio en el dominio personalizado. Recomendamos utilizar auth como subdominio. Por ejemplo: *auth.example.com*.

  > **Note**
  >
  > Si no dispone de un certificado comodín, es posible que tenga que obtener un nuevo certificado para el subdominio del dominio personalizado.

- Un certificado de Capa de conexión segura (SSL) administrado por ACM.

  > **Note**
  >
  > Debe cambiar la AWS región a EE.UU. Este (Virginia del Norte) en la consola de ACM antes de solicitar o importar un certificado.

- Aplicación que permite al servidor de autorización de su grupo de usuarios añadir cookies a las sesiones de los usuarios. Amazon Cognito establece varias cookies obligatorias para la interfaz de usuario alojada. Entre ellos se encuentran `cognito`, `cognito-fl` y `XSRF-TOKEN`. Si bien cada cookie individual se ajusta a los límites de tamaño del navegador, los cambios en la configuración del grupo de usuarios pueden provocar que las cookies de la interfaz de usuario alojada aumenten de tamaño. Un servicio intermedio, como un Application Load Balancer (ALB), delante de su dominio personalizado puede imponer un tamaño máximo de encabezado o un tamaño total de

cookie. Si tu aplicación también establece sus propias cookies, es posible que las sesiones de tus usuarios superen estos límites. Te recomendamos que, para evitar conflictos con los límites de tamaño, tu aplicación no establezca cookies en el subdominio de la interfaz de usuario alojada.

- Permiso para actualizar las CloudFront distribuciones de Amazon. Puede hacerlo adjuntando la siguiente declaración de política de IAM a un usuario en su Cuenta de AWS:

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Sid": "AllowCloudFrontUpdateDistribution",
            "Effect": "Allow",
            "Action": [
                "cloudfront:updateDistribution"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Para obtener más información sobre cómo autorizar acciones en CloudFront, consulte Uso de políticas basadas en la identidad (políticas de IAM) para. CloudFront

Amazon Cognito utiliza inicialmente sus permisos de IAM para configurar la CloudFront distribución, pero la gestión de la distribución corre a cargo de. AWS No puede cambiar la configuración de la CloudFront distribución que Amazon Cognito asoció a su grupo de usuarios. Por ejemplo, puede actualizar las versiones de TLS compatibles en la política de seguridad.

Paso 1: Introducir el nombre de dominio personalizado

Puede agregar el dominio al grupo de usuarios con una API o la consola de Amazon Cognito.

Amazon Cognito console

Para agregar un dominio al grupo de usuarios mediante la consola de Amazon Cognito:

1. Inicie sesión en la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS .

2. Elija User pools (Grupos de usuarios).
3. Elija el grupo de usuarios en el que desee agregar el dominio.
4. Elija la pestaña App integration (Integración de aplicaciones).
5. Junto a Domain (Dominio) , elijaActions (Acciones)y, después, elija Create custom domain (Crear dominio personalizado).

> **Note**
>
> Si ya ha configurado un dominio de grupo de usuarios, elija Delete Cognito domain (Eliminar dominio de Cognito) o Delete custom domain (Eliminar dominio personalizado) para eliminar el dominio existente antes de crear el nuevo dominio personalizado.

6. Para Custom domain (Dominio personalizado), introduzca la URL del dominio que desea utilizar con Amazon Cognito. El nombre de dominio solo puede incluir letras minúsculas, números y guiones. No utilice un guion en el primer carácter ni en el último. Utilice puntos para separar los nombres de los subdominios.
7. En ACM certificate (Certificado de ACM), elija el certificado SSL que desee utilizar con el dominio. Solo los certificados ACM de EE. UU. Este (Virginia del Norte) son aptos para su uso con un dominio personalizado de Amazon Cognito, independientemente Región de AWS del grupo de usuarios.

   Si no dispone de un certificado disponible, puede utilizar ACM para aprovisionar uno en EE. UU. Este (Norte de Virginia). Para obtener más información, consulte la introducción de la Guía del usuario de AWS Certificate Manager .
8. Seleccione Create (Crear).
9. Amazon Cognito le devuelve a la pestaña App integration (Integración de aplicaciones). Se muestra un mensaje titulado Create an alias record in your domain's DNS (Cree un registro de alias en el DNS de su dominio). Anote el Domain (Dominio) y el Alias Target (Destino de alias) que se muestra en la consola. Se utilizarán en el paso siguiente para dirigir el tráfico a su dominio personalizado.

API

Para agregar un dominio al grupo de usuarios mediante la API de Amazon Cognito:

- Utilice la acción CreateUserPoolDomain.

Paso 2: Agregar un destino de alias y un subdominio

En este paso, configurará un alias mediante el proveedor de servicios de servidor de nombres de dominio (DNS) que apunta al destino de alias del paso anterior. Si utiliza Amazon Route 53 para la resolución de direcciones DNS, elija la sección To add an alias target and subdomain using Route 53 (Para agregar un destino de alias y un subdominio con Route 53).

Para añadir un destino de alias y un subdominio a la configuración de DNS actual

- Si no utiliza Route 53 para la resolución de direcciones de DNS, entonces debe usar las herramientas de configuración del proveedor de servicios de DNS para agregar el destino de alias del paso anterior al registro del DNS del dominio. El proveedor de DNS también deberá configurar el subdominio para el dominio personalizado.

Para agregar un destino de alias y un subdominio con Route 53, siga estos pasos:

1. Inicie sesión en la consola de Route 53. Si se le solicita, escriba sus credenciales de AWS .
2. Si no tiene una zona alojada en Route 53, cree una con una raíz que sea la principal de su dominio personalizado. Para obtener más información, consulte

   a. Elija Create Hosted Zone (Crear zona alojada).

   b. Introduzca el dominio principal, por ejemplo, *auth.ejemplo.com*, de su dominio personalizado, por ejemplo, *myapp.auth.example.com*, desde la lista Domain Name (Nombre de dominio).

   c. Introduzca una Descripción para su zona alojada.

   d. Elija una zona alojada Type (Tipo) de Public hosted zone (Zona alojada pública) para permitir que los clientes públicos resuelvan su dominio personalizado. Elegir una Private hosted zone (Zona alojada privada) no es compatible.

   e. Aplique Tags (Etiquetas) como desee.

   f. Elija Crear zona alojada.

> **Note**
>
> También puede crear una nueva zona alojada para su dominio personalizado y crear un conjunto de delegación en la zona alojada principal que dirija las consultas a la zona alojada del subdominio. De lo contrario, cree un registro A. Este método ofrece más flexibilidad y seguridad con las zonas alojadas. Para obtener más información, consulte Creating a subdomain for a domain hosted through Amazon Route 53 (Creación de un subdominio para un dominio alojado mediante Amazon Route 53).

3. En la página Hosted Zones (Zonas alojadas), elija el nombre de la zona alojada.
4. Agrega un registro DNS para el dominio principal de tu dominio personalizado, si aún no tienes uno. Agrega un `A` registro DNS para el dominio principal y selecciona Crear registros. A continuación se muestra un registro como ejemplo de dominio.*`auth.ejemplo.com`*.

   ```
   auth.example.com. 60 IN A 198.51.100.1
   ```

   > **Note**
   >
   > Amazon Cognito verifica que haya un registro DNS para el dominio principal de su dominio personalizado para protegerlo contra la apropiación accidental de dominios de producción. Si no tiene un registro DNS para el dominio principal, Amazon Cognito devolverá un error cuando intente establecer el dominio personalizado. Un registro de inicio de autoridad (SOA) no es un registro de DNS suficiente para verificar el dominio principal.

5. Agrega un registro DNS para tu dominio personalizado. El registro debe apuntar al destino Alias del dominio personalizado, por ejemplo`123example.cloudfront.net`. Elija de nuevo Create Record (Crear registro).
6. Introduzca un Record name (Nombre del registro) que coincida con su dominio personalizado, por ejemplo, *`myapp`* para crear un registro para *`myapp.auth.example.com`*.
7. Habilite la opción Alias.
8. Elija Route traffic to (Dirigir tráfico a), un Alias to Cloudfront distribution (Alias a distribución de CloudFront). Introduzca el Alias Target (Destino de alias) proporcionado por Amazon Cognito al crear su dominio personalizado.
9. Elija Create records (Crear registros).

> **Note**
>
> Los nuevos registros pueden tardar unos 60 segundos en propagarse a todos los servidores DNS de Route 53. Puede usar el método de la GetChangeAPI de Route 53 para comprobar que los cambios se han propagado.

Paso 3: Verificar la página de inicio de sesión

- Compruebe que la página de inicio de sesión está disponible desde el dominio personalizado.

  Inicie sesión con el dominio personalizado y el subdominio; para ello, introduzca esta dirección en el navegador. Esta es una URL de ejemplo de un dominio personalizado *example.com* con el subdominio *auth*:

```
https://myapp.auth.example.com/login?
response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback_url>
```

Cambio del certificado SSL en el dominio personalizado

Si es necesario, puede utilizar Amazon Cognito para cambiar el certificado que se ha aplicado al dominio personalizado.

Esta operación no suele ser necesaria si se mantiene una renovación rutinaria de certificados con ACM. Cuando se renueva el certificado actual en ACM, el ARN del certificado sigue siendo el mismo, y el nombre de dominio personalizado utiliza el nuevo certificado de manera automática.

Sin embargo, si el certificado actual se sustituye por otro nuevo, ACM proporciona otro ARN al nuevo certificado. Para aplicar el nuevo certificado al dominio personalizado, debe proporcionar este ARN a Amazon Cognito.

Una vez proporcionado el certificado, Amazon Cognito puede necesitar hasta una hora para distribuirlo en el dominio personalizado.

> **Antes de empezar**
>
> Para poder cambiar el certificado en Amazon Cognito, debe agregarlo a ACM. Para obtener más información, consulte la introducción de la Guía del usuario de AWS Certificate Manager .
>
> Cuando añada el certificado a ACM, debe seleccionar US East (N. Virginia) [Este de EE. UU. (Norte de Virginia)] como región de AWS .

Puede cambiar el certificado con una API o la consola de Amazon Cognito.

AWS Management Console

Para renovar un certificado mediante la consola de Amazon Cognito:

1. Inicie sesión en la consola de Amazon Cognito AWS Management Console y ábrala en. https://console.aws.amazon.com/cognito/home
2. Elija User Pools (Grupos de usuarios).
3. Elija el grupo de usuarios para el que desea actualizar el certificado.
4. Elija la pestaña App integration (Integración de aplicaciones).
5. Elija Actions (Acciones), Edit ACM certificate (Editar certificado de ACM).
6. Seleccione el nuevo certificado que desea asociar a su dominio personalizado.
7. Elija Guardar cambios.

API

Para renovar un certificado (API de Amazon Cognito)

- Utilice la acción UpdateUserPoolDomain.

Personalizar las páginas web integradas de registro e inicio de sesión

Puede utilizar la AWS Management Console, la AWS CLI o la API para especificar la configuración de personalización de la interfaz de usuario de la aplicación integrada. Puede cargar una imagen de logotipo personalizada para que se muestre en la aplicación. También puede usar cascading style sheets (CSS, hojas de estilos en cascada) para personalizar el aspecto de la IU.

Puede especificar los ajustes de personalización de interfaz de usuario de la aplicación para un solo cliente (con un `clientId` específico) o para todos los clientes (cambiando la configuración de `clientId` a `ALL`). Si especifica `ALL`, se utilizará la configuración predeterminada para cada cliente que no tiene ninguna personalización de interfaz de usuario determinada. Si especifica ajustes de personalización de interfaz de usuario para un cliente específico, no volverá a usar la configuración `ALL`.

La solicitud que establece la personalización de su IU no debe superar los 135 KB de tamaño. En casos excepcionales, la suma de los encabezados de solicitud, su archivo CSS y su logotipo podría superar los 135 KB. Amazon Cognito codifica el archivo de imagen en Base64. Esto aumenta el tamaño de una imagen de 100 KB a 130 KB, lo que mantiene cinco KB para los encabezados de solicitud y su CSS. Si la solicitud es demasiado grande, la AWS Management Console o su solicitud de API `SetUICustomization` devuelve un error `request parameters too large`. Ajuste la imagen de su logotipo para que no supere los 100 KB y su archivo CSS para que no supere los 3 KB. No puede establecer la personalización de CSS y logotipo por separado.

> **Note**
>
> Para personalizar su IU, debe establecer un dominio para su grupo de usuarios.

Especificar un logotipo personalizado para la aplicación

Amazon Cognito centra su logotipo personalizado encima de los campos de entrada en el Punto de conexión Login.

Elija un archivo PNG, JPG o JPEG que pueda escalarse a 350 por 178 píxeles para su logotipo de IU alojado personalizado. El archivo del logotipo no puede tener un tamaño superior a 100 KB o 130 KB después de que Amazon Cognito lo codifique en Base64. Para establecer un `ImageFile` en SetUICustomization en la API, convierta su archivo en una cadena de texto codificada en Base64 o, en la AWS CLI, proporcione una ruta de archivo y deje que Amazon Cognito la codifique automáticamente.

Especificar ajustes personalizados de CSS para la aplicación

Puede personalizar el CSS de las páginas de la aplicación alojada, con las siguientes restricciones:

- Puede utilizar cualquiera de los siguientes nombres de clase de CSS:

- background-customizable
- banner-customizable
- errorMessage-customizable
- idpButton-customizable
- idpButton-customizable:hover
- idpDescription-customizable
- inputField-customizable
- inputField-customizable:focus
- label-customizable
- legalText-customizable
- logo-customizable
- passwordCheck-valid-customizable
- passwordCheck-notValid-customizable
- redirect-customizable
- socialButton-customizable
- submitButton-customizable
- submitButton-customizable:hover
- textDescription-customizable

- Los valores de propiedad pueden contener HTML, excepto los siguientes valores: `@import`, `@supports`, `@page`, o bien instrucciones de `@media` o Javascript.

Puede personalizar las siguientes propiedades CSS.

Etiquetas

- font-weight (peso de fuente) es un múltiplo de 100 entre 100 y 900.

Campos de entrada

- width (anchura) es la anchura del bloque contenedor como un porcentaje.
- height (altura) es la altura del campo de entrada en píxeles (px).
- color es el color del texto. Puede ser cualquier valor de color CSS estándar.
- background-color (color de fondo) es el color de fondo del campo de entrada. Puede ser cualquier valor de color CSS estándar.

- border (borde) es un valor de borde CSS estándar que especifica la anchura, la transparencia y el color del borde de la ventana de la aplicación. La anchura puede ser cualquier valor entre 1 px y 100 px. La transparencia puede ser sólida o ninguna. El color puede ser cualquier valor de color estándar.

Descripciones de texto

- padding-top (relleno superior) es la cantidad de relleno por encima de la descripción de texto.
- padding-bottom (relleno inferior) es la cantidad de relleno por debajo de la descripción de texto.
- display (visualización) puede ser `block` o `inline`.
- font-size (tamaño de fuente) es el tamaño de fuente de las descripciones de texto.

Botón de envío

- font-size (tamaño de fuente) es el tamaño de fuente del botón de envío.
- font-weight (peso de fuente) es el peso de fuente del texto del botón: `bold`, `italic` o `normal`.
- margen es una cadena de cuatro valores que indica el tamaño de margen de las partes superior, inferior, derecha e izquierda del botón.
- font-size (tamaño de fuente) es el tamaño de fuente de las descripciones de texto.
- width (anchura) es la anchura del texto del botón como porcentaje del bloque contenedor.
- height (altura) es la altura del botón en píxeles (px).
- color es el color del texto del botón. Puede ser cualquier valor de color CSS estándar.
- background-color (color de fondo) es el color de fondo del botón. Puede ser cualquier valor de color estándar.

Banner

- relleno es una cadena de cuatro valores que indica el tamaño del relleno de las partes superior, inferior, derecha e izquierda del banner.
- background-color (color de fondo) es el color de fondo del banner. Puede ser cualquier valor de color CSS estándar.

Ajustes al mantener el puntero sobre el botón de envío

- color es el color de primer plano del botón al pasar el puntero sobre él. Puede ser cualquier valor de color CSS estándar.
- background-color (color de fondo) es el color de fondo del botón al pasar el puntero sobre él. Puede ser cualquier valor de color CSS estándar.

Ajustes al mantener el puntero sobre el botón de proveedor de identidad

- color es el color de primer plano del botón al pasar el puntero sobre él. Puede ser cualquier valor de color CSS estándar.
- background-color (color de fondo) es el color de fondo del botón al pasar el puntero sobre él. Puede ser cualquier valor de color CSS estándar.

Comprobación de contraseña no válida

- color es el color del texto del mensaje `"Password check not valid"`. Puede ser cualquier valor de color CSS estándar.

Introducción

- background-color (color de fondo) es el color de fondo de la ventana de la aplicación. Puede ser cualquier valor de color CSS estándar.

Mensajes de error

- margen es una cadena de cuatro valores que indica el tamaño de margen de las partes superior, inferior, derecha e izquierda.
- padding (relleno) es el tamaño del relleno.
- font-size (tamaño de fuente) es el tamaño de la fuente.
- width (anchura) es la anchura del mensaje de error como porcentaje del bloque contenedor.
- background-color (color de fondo) es el color de fondo del mensaje de error. Puede ser cualquier valor de color CSS estándar.
- borde es una cadena de tres valores que especifica el ancho, la transparencia y el color del borde.
- color es el color del texto del mensaje de error. Puede ser cualquier valor de color CSS estándar.
- box-sizing (tamaño de cuadro) se utiliza para indicar al navegador qué deben incluir las propiedades de tamaño (anchura y altura).

Botones de proveedor de identidad

- height (altura) es la altura del botón en píxeles (px).
- width (anchura) es la anchura del texto del botón como porcentaje del bloque contenedor.
- text-align (alineación de texto) es el ajuste de alineación del texto. Puede ser `left`, `right`, o `center`.
- margin-bottom (margen inferior) es el ajuste del margen inferior.
- color es el color del texto del botón. Puede ser cualquier valor de color CSS estándar.

- background-color (color de fondo) es el color de fondo del botón. Puede ser cualquier valor de color CSS estándar.
- border-color (color de borde) es el color de borde del botón. Puede ser cualquier valor de color CSS estándar.

Descripciones de proveedor de identidad

- padding-top (relleno superior) es la cantidad de relleno por encima de la descripción.
- padding-bottom (relleno inferior) es la cantidad de relleno por debajo de la descripción.
- display (visualización) puede ser `block` o `inline`.
- font-size (tamaño de fuente) es el tamaño de fuente de las descripciones.

Texto legal

- color es el color del texto. Puede ser cualquier valor de color CSS estándar.
- font-size (tamaño de fuente) es el tamaño de la fuente.

> **Note**
>
> Cuando personaliza texto legal, está personalizando el mensaje. No publicaremos nada en ninguna de sus cuentas sin pedir antes que se muestre en los proveedores de identidad social en la página de inicio de sesión.

Logo

- max-width (anchura máx.) es la anchura máxima como porcentaje del bloque contenedor.
- max-height (altura máx.) es la altura máxima como porcentaje del bloque contenedor.

Foco del campo de entrada

- border-color (color de borde) es el color del campo de entrada. Puede ser cualquier valor de color CSS estándar.
- outline (contorno) es la anchura del borde del campo de entrada en píxeles (px).

Botones sociales

- height (altura) es la altura del botón en píxeles (px).
- text-align (alineación de texto) es el ajuste de alineación del texto. Puede ser `left`, `right`, o `center`.
- width (anchura) es la anchura del texto del botón como porcentaje del bloque contenedor.
- margin-bottom (margen inferior) es el ajuste del margen inferior.

Comprobación de contraseña válida

- color es el color del texto del mensaje `"Password check valid"`. Puede ser cualquier valor de color CSS estándar.

Especificación de la configuración de personalización de la IU de la aplicación para un grupo de usuarios (AWS Management Console)

Puede utilizar la AWS Management Console para especificar la configuración de personalización de la interfaz de usuario de la aplicación.

> **Note**
>
> Para ver la interfaz de usuario alojada y sus personalizaciones, escriba en un navegador la siguiente URL con los datos específicos de su grupo de usuarios: `https://<your_domain>/login?response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callbac`
> Posiblemente deba esperar hasta un minuto para actualizar la ventana del navegador y que aparezcan los cambios realizados en la consola.
> Su dominio aparece en la pestaña App integration (Integración de aplicaciones) en Domain (Dominio). Su ID de cliente de aplicación y URL de devolución de llamada aparecen en App clients (Clientes de la aplicación).

Para especificar la configuración de personalización de la interfaz de usuario

1. Inicie sesión en la consola de Amazon Cognito.
2. En el panel de navegación, elija User Pools (Grupos de usuarios), y elija el grupo de usuarios que desea editar.
3. Elija la pestaña App integration (Integración de aplicaciones).
4. Para personalizar la configuración de la IU de todos los clientes de aplicaciones, localice Hosted UI customization (Personalización de IU alojada)y seleccione Edit (Editar).
5. Para personalizar la configuración de la IU de un cliente de aplicación, localice Clientes de aplicaciones y seleccione el cliente de la aplicación que quiere modificar y, a continuación, localice Personalización de IU alojada y seleccione Editar. Para cambiar un cliente de aplicación de la personalización predeterminada del grupo de usuarios a la personalización específica del cliente, seleccione Use client-level settings (Utilizar la configuración de nivel de cliente).

6. Para cargar su propio archivo de imagen de logotipo, elija Choose file (Elegir archivo) o Replace current file (Reemplazar archivo actual).
7. Para personalizar CSS de la interfaz de usuario alojada, descargue CSS template.css y modifique la plantilla con los valores que quiera personalizar. Solo las claves incluidas en la plantilla se pueden utilizar con la IU alojada. Las claves CSS añadidas no se reflejarán en la IU. Después de personalizar el archivo CSS, elija Choose file (Elegir archivo) o Replace current file (Reemplazar archivo actual) para cargar su archivo CSS personalizado.

Especificación de la configuración de personalización de la IU de la aplicación para un grupo de usuarios (AWS CLI y API de AWS)

Utilice los siguientes comandos para especificar la configuración de la personalización de interfaz de usuario de la aplicación para su grupo de usuarios.

Para obtener la configuración de personalización de la IU de aplicación integrada de un grupo de usuarios, utilice las siguientes operaciones de API.

- AWS CLI: `aws cognito-idp get-ui-customization`
- API de AWS: GetUICustomization

Para establecer la configuración de personalización de la IU de aplicación integrada de un grupo de usuarios, utilice las siguientes operaciones de API.

- AWS CLI del archivo de imagen: `aws cognito-idp set-ui-customization --user-pool-id <your-user-pool-id> --client-id <your-app-client-id> --image-file fileb://"<path-to-logo-image-file>" --css ".label-customizable{ color: <color>;}"`
- AWS CLI con imagen codificada como texto binario Base64: `aws cognito-idp set-ui-customization --user-pool-id <your-user-pool-id> --client-id <your-app-client-id> --image-file <base64-encoded-image-file> --css ".label-customizable{ color: <color>;}"`
- API de AWS: SetUICustomization

Registro e inicio de sesión con la interfaz de usuario alojada

Después de configurar y personalizar la interfaz de usuario alojada en Amazon Cognito para su grupo de usuarios y clientes de aplicaciones, la aplicación podrá presentarla a sus usuarios. La interfaz de usuario alojada admite varias operaciones de autenticación de Amazon Cognito, incluidos los siguientes ejemplos.

- Registrarse como nuevo usuario en la aplicación
- Verificar una dirección de correo electrónico o un número de teléfono
- Configurar autenticación multifactor (MFA)
- Inicio de sesión con un nombre de usuario y una contraseña locales
- Inicio de sesión con un proveedor de identidades (IdP) externo
- Restablezca una contraseña

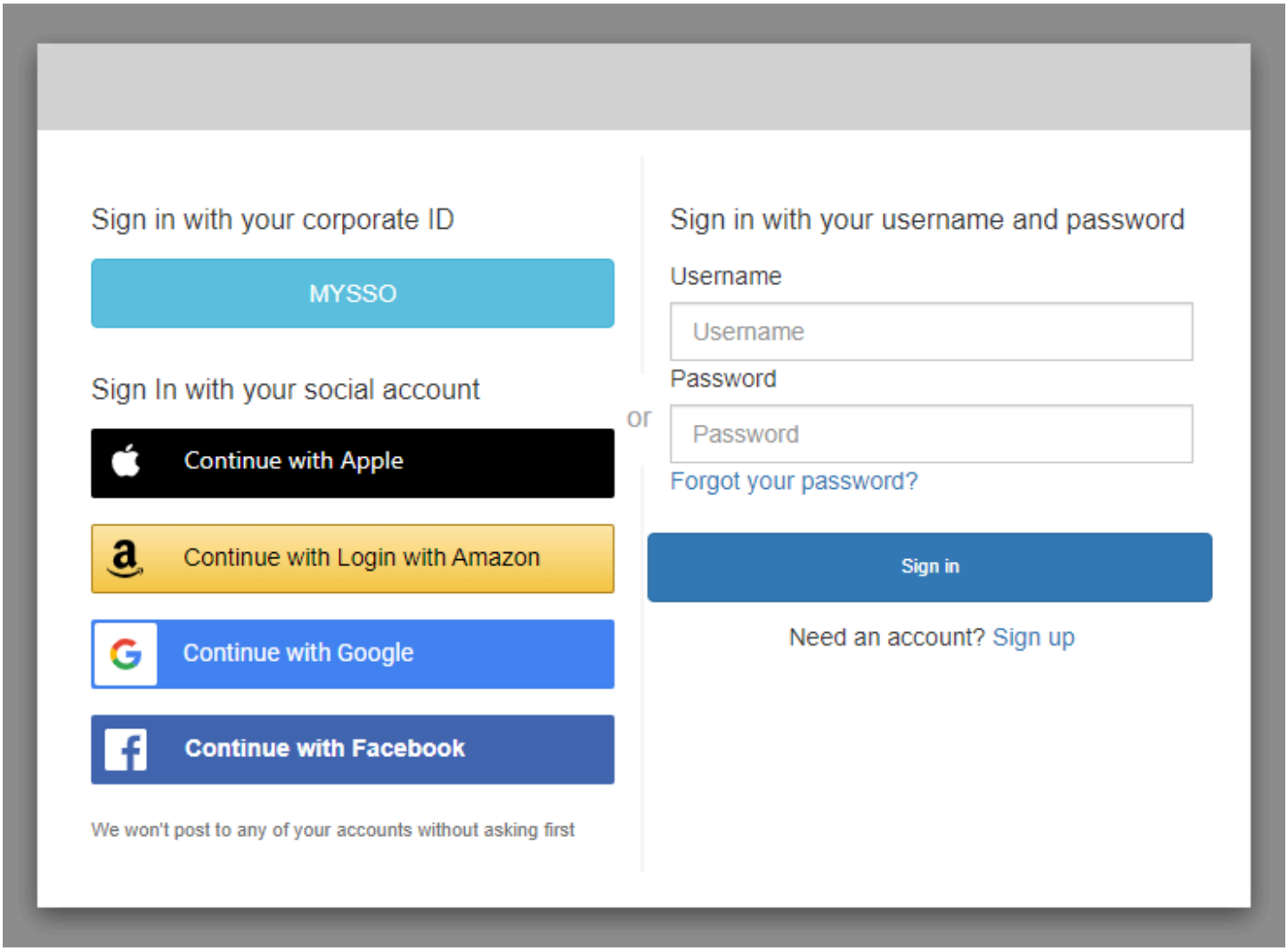
La interfaz de usuario alojada en Amazon Cognito comienza en el Punto de conexión Login. La URL de su página de inicio de sesión es una combinación del dominio que ha elegido para su grupo de usuarios y los parámetros que reflejan las concesiones de OAuth 2.0 que desea conceder, el cliente de la aplicación, la ruta a la aplicación y los ámbitos de OpenID Connect (OIDC) que desea solicitar.

```
https://<your user pool domain>/authorize?client_id=<your app client
 ID>&response_type=<code/token>&scope=<scopes to request>&redirect_uri=<your callback
 URL>
```

La siguiente URL reemplaza los campos de marcadores de posición anteriores por valores de ejemplo.

```
https://auth.example.com/authorize? /
client_id=1example23456789 /
&response_type=code /
&scope=aws.cognito.signin.user.admin+email+openid+profile /
&redirect_uri=https%3A%2F%2Faws.amazon.com
```

La página de inicio de sesión de la interfaz de usuario alojada en Amazon Cognito tiene opciones para iniciar sesión a través del grupo de usuarios o de cualquier proveedor de identidades (IdP) que haya asignado al cliente de la aplicación que solicita su usuario. También incluye enlaces para crear una nueva cuenta de usuario en el grupo de usuarios o para restablecer una contraseña olvidada.



Temas

- Cómo registrarse en una nueva cuenta en la IU alojada de Amazon Cognito
- Cómo registrarse con la IU alojada de Amazon Cognito
- Cómo restablecer una contraseña con la IU alojada de Amazon Cognito

Cómo registrarse en una nueva cuenta en la IU alojada de Amazon Cognito

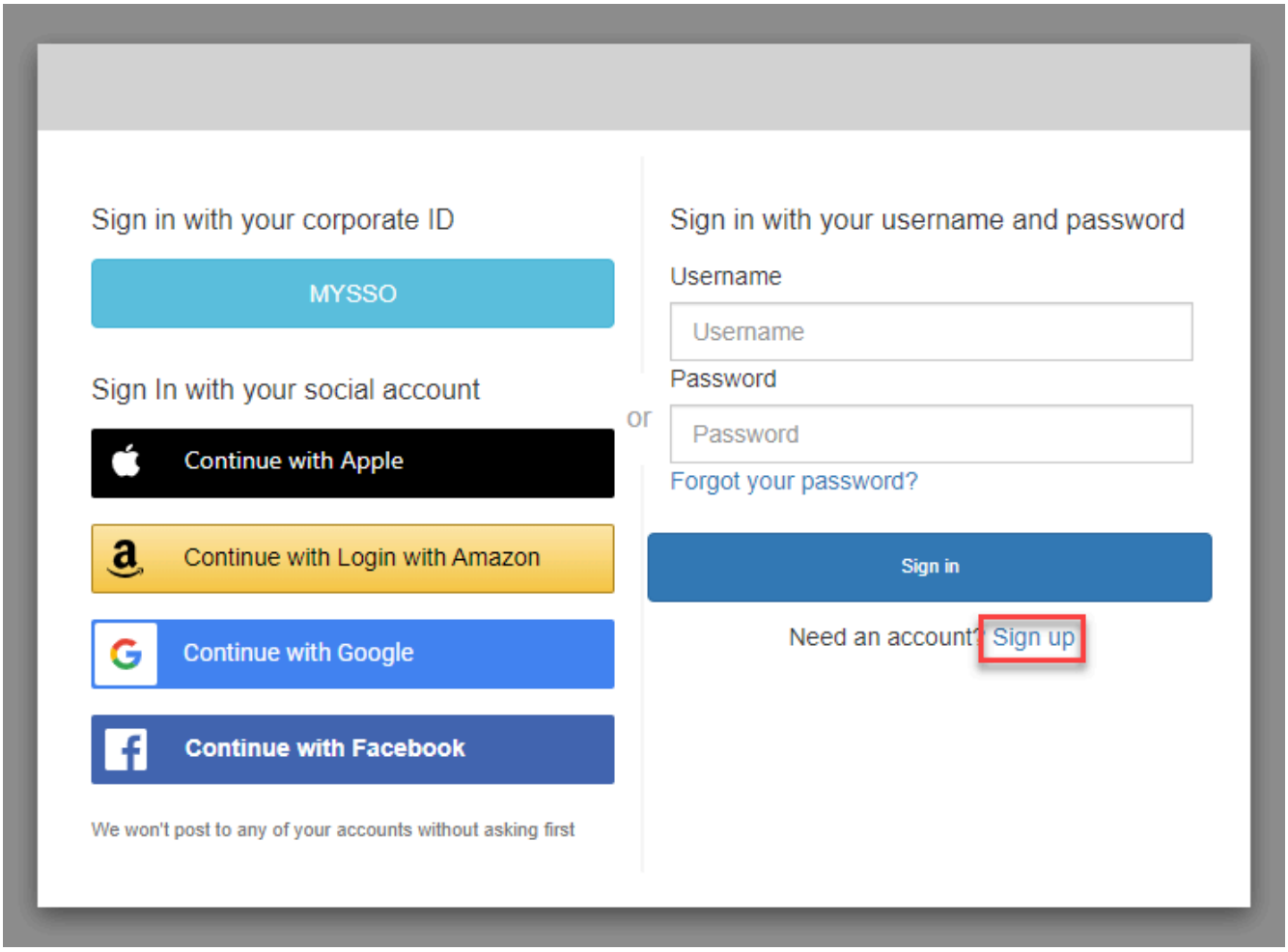
En esta guía se muestra cómo registrarse para obtener una cuenta de usuario en aplicaciones que usan Amazon Cognito.

> **Note**
>
> Al iniciar sesión en una aplicación que utiliza la interfaz de usuario (IU) alojada de Amazon Cognito, es posible que vea una página que el propietario de la aplicación haya personalizado más allá de la configuración básica que se muestra en esta guía.

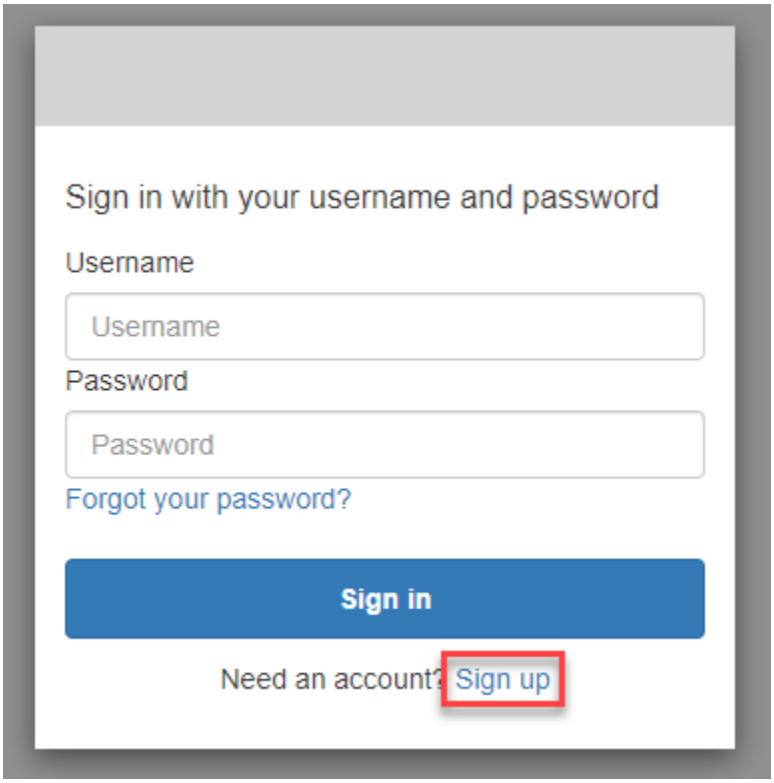
1. Elija Sign up (Registrarse) en la página de inicio de sesión si tiene intención de iniciar sesión a través de Amazon Cognito con un nombre de usuario y una contraseña, en lugar de uno de los proveedores de inicio de sesión de terceros que ha publicado el propietario de la aplicación.

   Si su proveedor de inicio de sesión es otro proveedor que no sea Amazon Cognito, el registro se habrá completado después de seleccionar el botón correspondiente a su proveedor externo. Según las opciones que haya elegido el propietario de la aplicación, es posible que pueda elegir los proveedores con los que iniciar sesión o que solo vea un mensaje con un nombre de usuario y una contraseña.

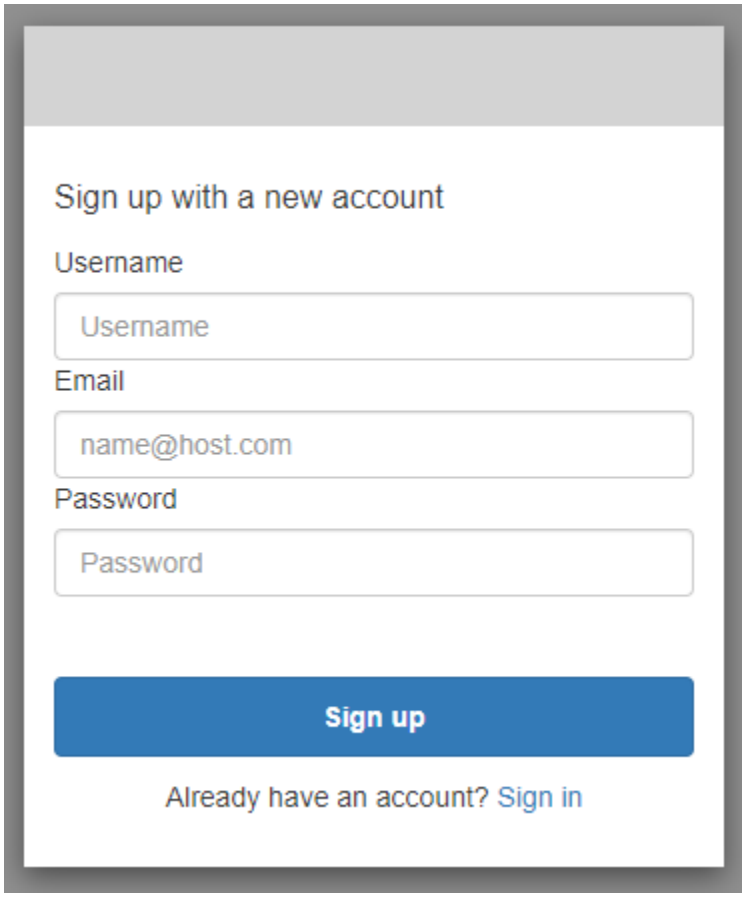
With multiple sign-in providers



With only Amazon Cognito as a sign-in provider

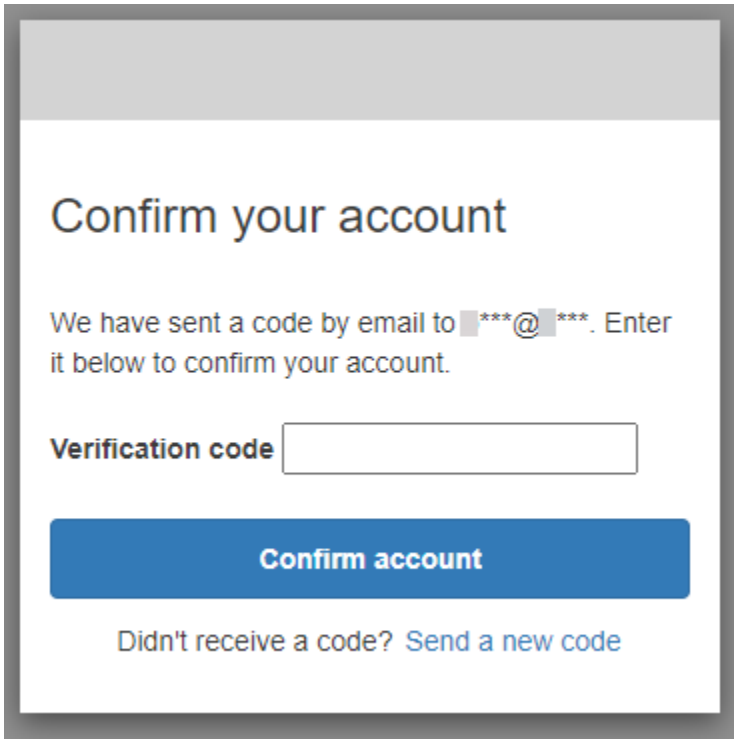


2. En la página Sign up with a new account (Registrarse con una cuenta nueva), el propietario de la aplicación solicita la información que necesita para registrarse. Es posible que le pidan un nombre de usuario, una dirección de correo electrónico o un número de teléfono. Introduzca la información requerida y elija una contraseña.



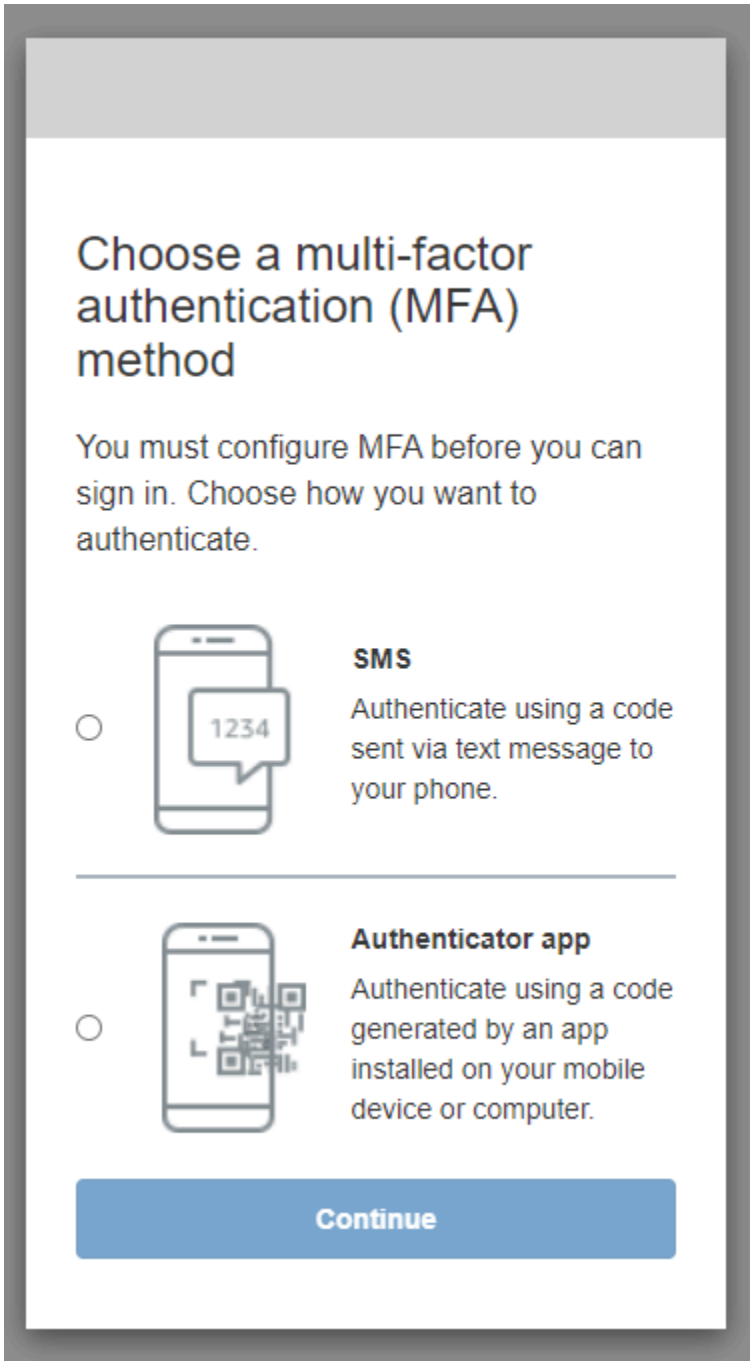
3. En la página Confirm your account (Confirmar la cuenta), es posible que el propietario de la aplicación le pida que confirme su cuenta para verificar que puede recibir mensajes en la dirección de correo electrónico o el número de teléfono facilitado.

   Recibirá un código en el correo electrónico o un mensaje SMS. Introduzca el código en el formulario para confirmar que ha introducido la información de contacto correcta.



4. El propietario de la aplicación puede solicitar que configure la autenticación multifactor (MFA). Es posible que aparezca un mensaje para elegir el método de MFA o que la aplicación vaya directamente al paso siguiente.

   En la página Choose a multi-factor authentication (MFA) method (Elija un método de autenticación multifactor (MFA)), elija un método de MFA. Si elige SMS, recibirá códigos de acceso MFA en los mensajes de texto SMS. Si elige Aplicación Authenticator (Autenticador de aplicaciones), debe instalar una aplicación en su dispositivo para generar códigos de acceso de MFA basados en el tiempo. Debe elegir en un plazo de 3 minutos.



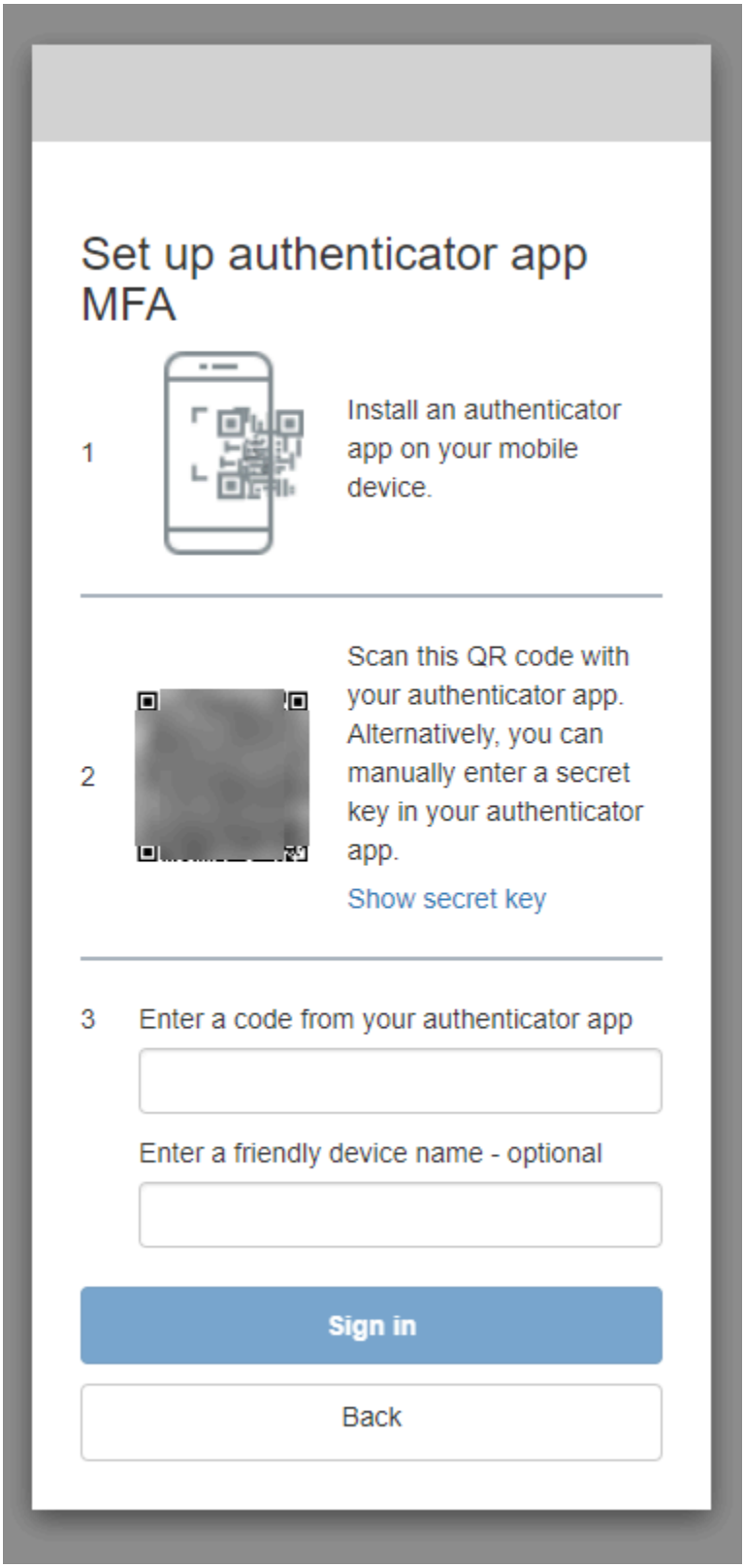
5. Amazon Cognito le pide un código de su aplicación de autenticación o mensaje de texto SMS. Ingrese el código que recibió en 3 minutos.

   Authenticator app

   1. Abra la aplicación de autenticación descargada.
   2. Escanea con la cámara el código QR de la página. Es posible que tenga que autorizar a la aplicación para usar su cámara.

Si no puede escanear el código QR, elija Show secret key (Mostrar la clave secreta) para mostrar un código que puede introducir manualmente en su aplicación de autenticación.

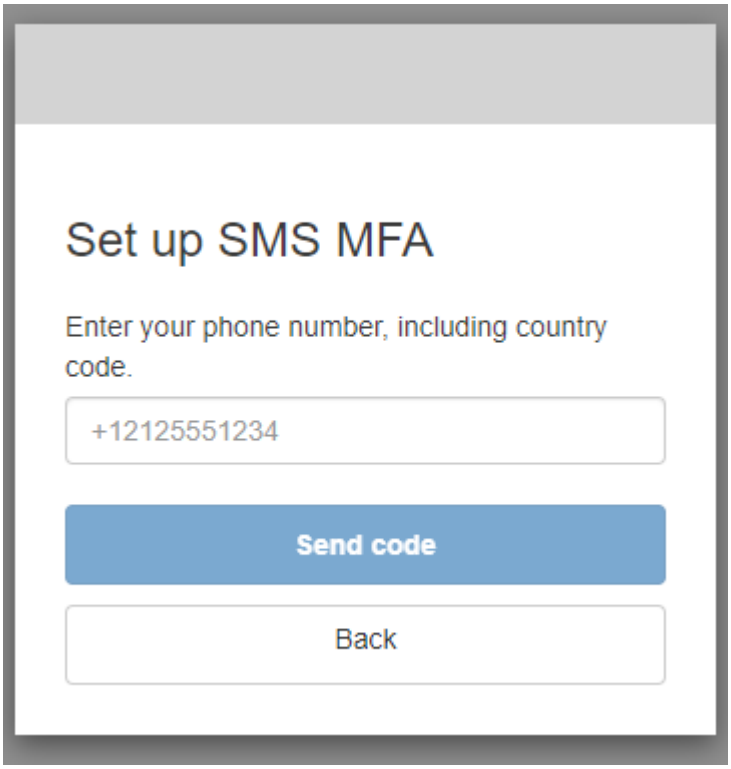
3. La aplicación de autenticación comienza a mostrar códigos que cambian cada varios segundos. Introduzca un código actual de la aplicación.
4. (Opcional) En la página Set up authenticator app MFA (Configurar la aplicación de autenticación MFA), elija un nombre para su dispositivo. Cuando inicie sesión, Amazon Cognito le pedirá un código del dispositivo con el nombre aquí indicado.


Set up authenticator app MFA
1
Install an authenticator app on your mobile device.
2
Scan this QR code with your authenticator app. Alternatively, you can manually enter a secret key in your authenticator app.
Show secret key
3
Enter a code from your authenticator app
Enter a friendly device name - optional
Sign in
Back

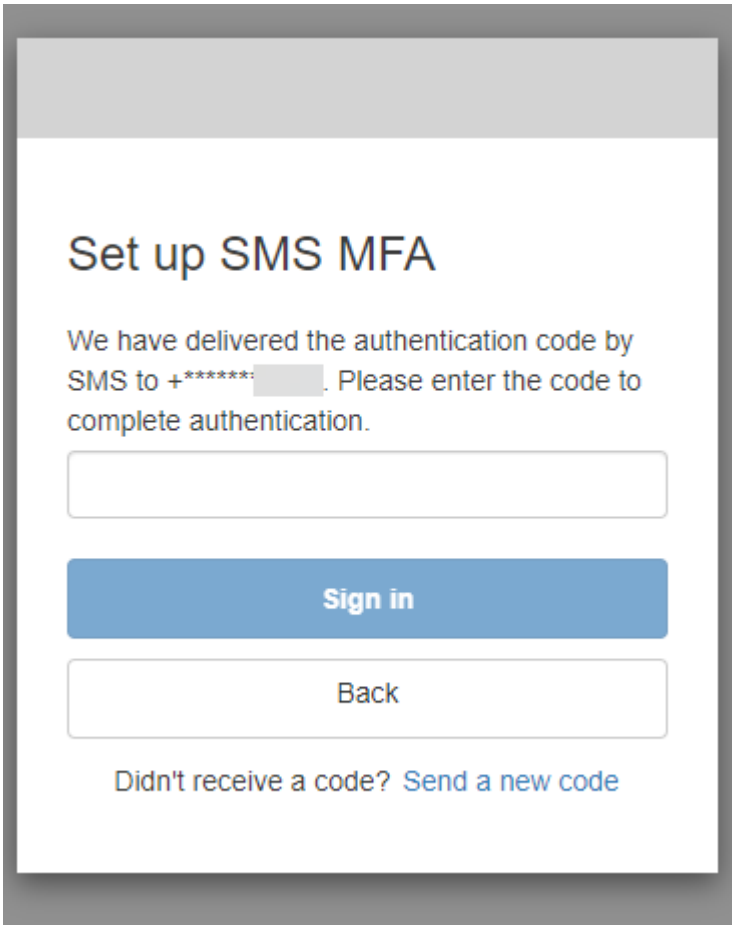
SMS text message

1. Si el propietario de la aplicación aún no ha recopilado su número de teléfono, Amazon Cognito lo solicitará.

   En la página Set up SMS MFA (Configurar SMS MFA), introduzca un número de teléfono que incluya un signo + y un código de país, por ejemplo +12125551234.



2. Recibirás un mensaje SMS con un código. En la página Set up SMS MFA (Configurar SMS MFA), introduzca el código. Si no ha recibido ningún código y quiere volver a intentarlo, seleccione Send a new code (Enviar un código nuevo). Seleccione Back (Volver) para introducir un número de teléfono nuevo.



6. La primera vez que se registre y confirme sus datos, Amazon Cognito le concederá acceso a su aplicación una vez finalizado este proceso.

Cómo registrarse con la IU alojada de Amazon Cognito

En esta guía se muestra cómo registrarse en aplicaciones que usan Amazon Cognito.

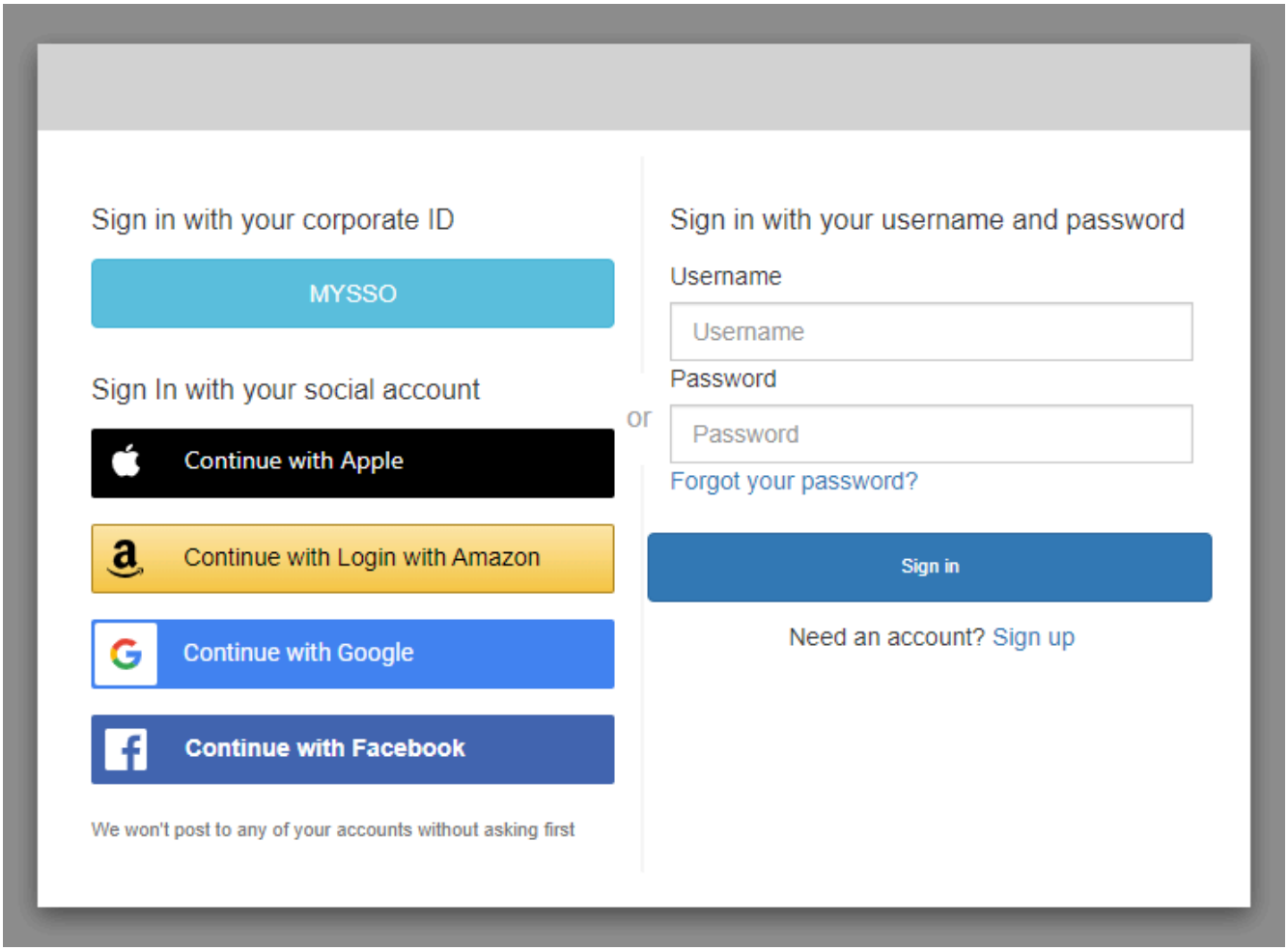
> **Note**
>
> Al iniciar sesión en una aplicación que utiliza la interfaz de usuario (IU) alojada de Amazon Cognito, es posible que vea una página que el propietario de la aplicación haya personalizado más allá de la configuración básica que se muestra en esta guía.

1. Según las opciones que haya elegido el propietario de la aplicación, es posible que pueda elegir los proveedores con los que iniciar sesión o que solo vea un mensaje con un nombre de usuario y una contraseña. Al iniciar sesión con un nombre de usuario y una contraseña de esta página,

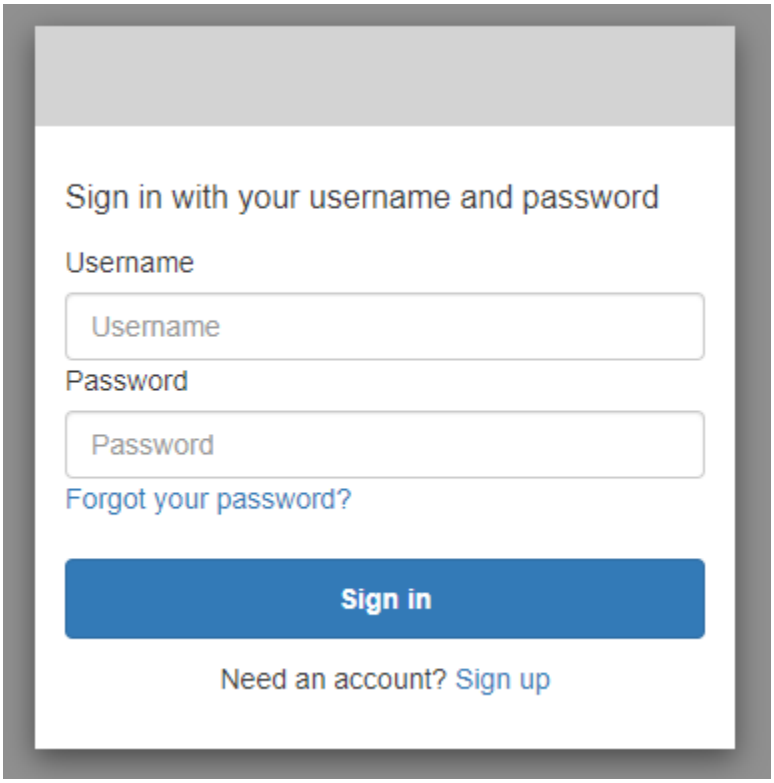
Amazon Cognito es su proveedor de inicio de sesión. De lo contrario, el proveedor de inicio de sesión se representa mediante el botón que elija.

Puede elegir un proveedor aquí o introducir un nombre de usuario y una contraseña y acceder a su aplicación de inmediato. Si Amazon Cognito es su proveedor de inicio de sesión, es posible que el propietario de la aplicación también necesite una autenticación multifactor.

With multiple sign-in providers

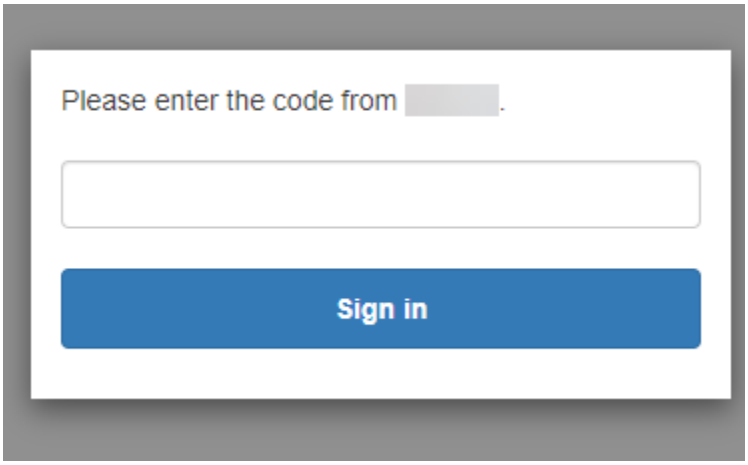


With only Amazon Cognito as a sign-in provider

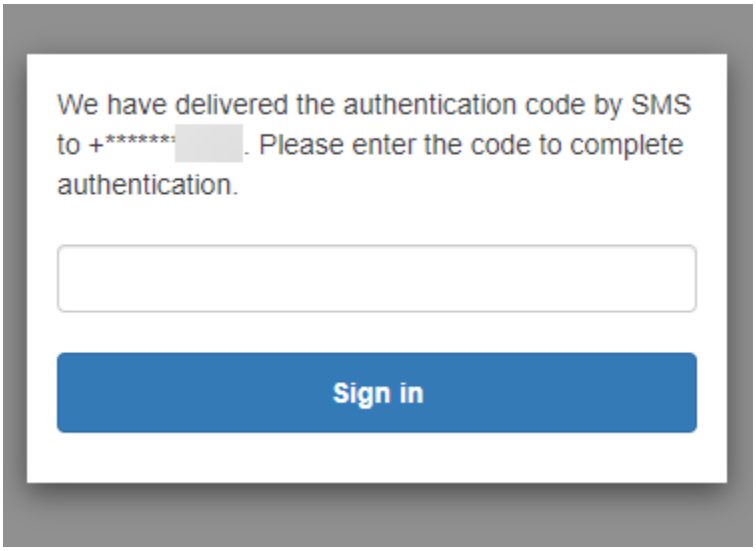


2. Es posible que haya configurado MFA al registrarse en la aplicación. Introduzca el código MFA que recibió en un mensaje SMS o que aparece en la aplicación de autenticación. Debe ingresar este código en un plazo de 3 minutos.

   With an authenticator app



With an SMS code



3. Tras iniciar sesión y completar la MFA, Amazon Cognito concede acceso a su aplicación.

Cómo restablecer una contraseña con la IU alojada de Amazon Cognito

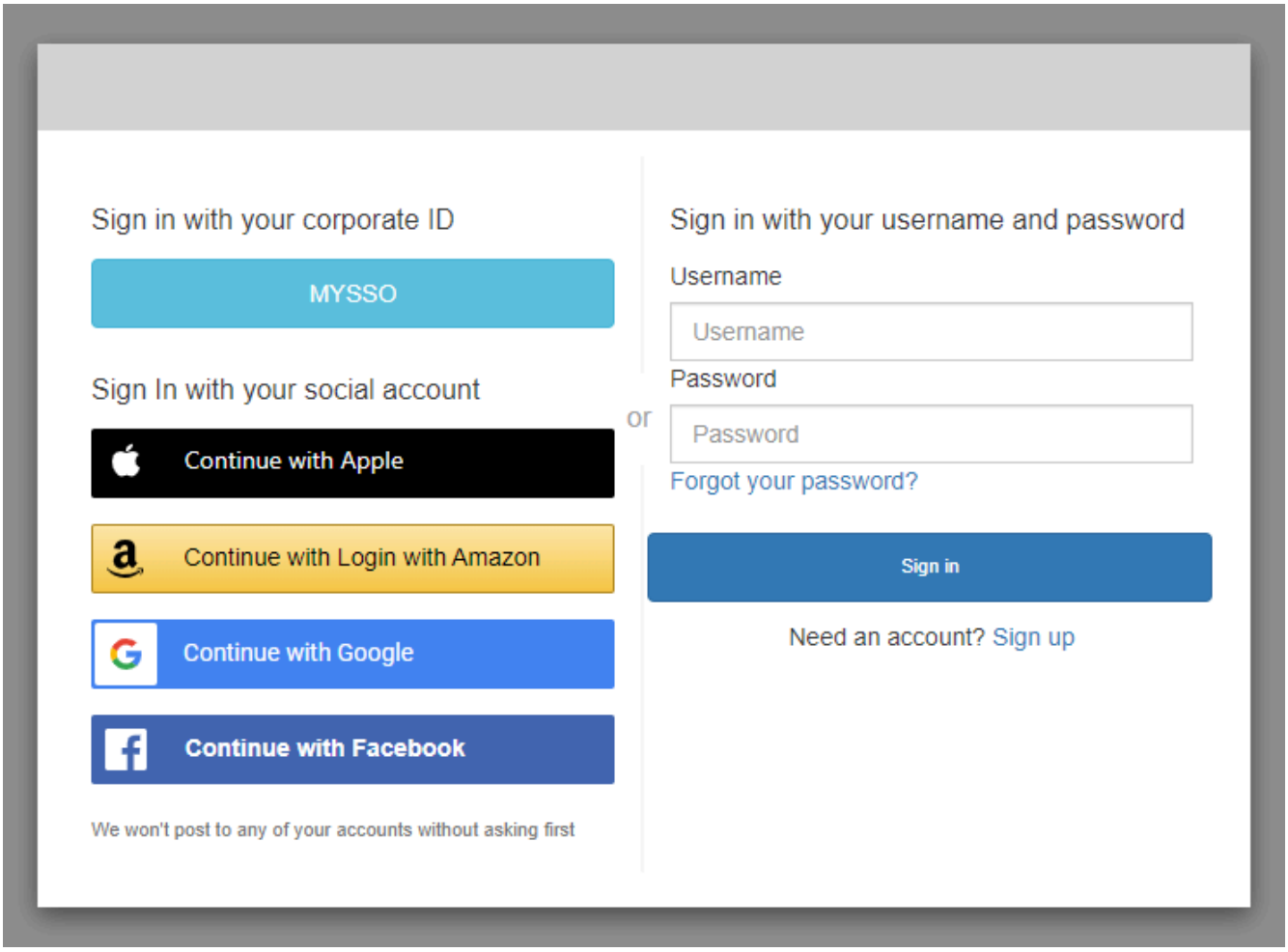
En esta guía se muestra cómo restablecer una contraseña en aplicaciones que usan Amazon Cognito.

> **Note**
>
> Al iniciar sesión en una aplicación que utiliza la interfaz de usuario (IU) alojada de Amazon Cognito, es posible que vea una página que el propietario de la aplicación haya personalizado más allá de la configuración básica que se muestra en esta guía.

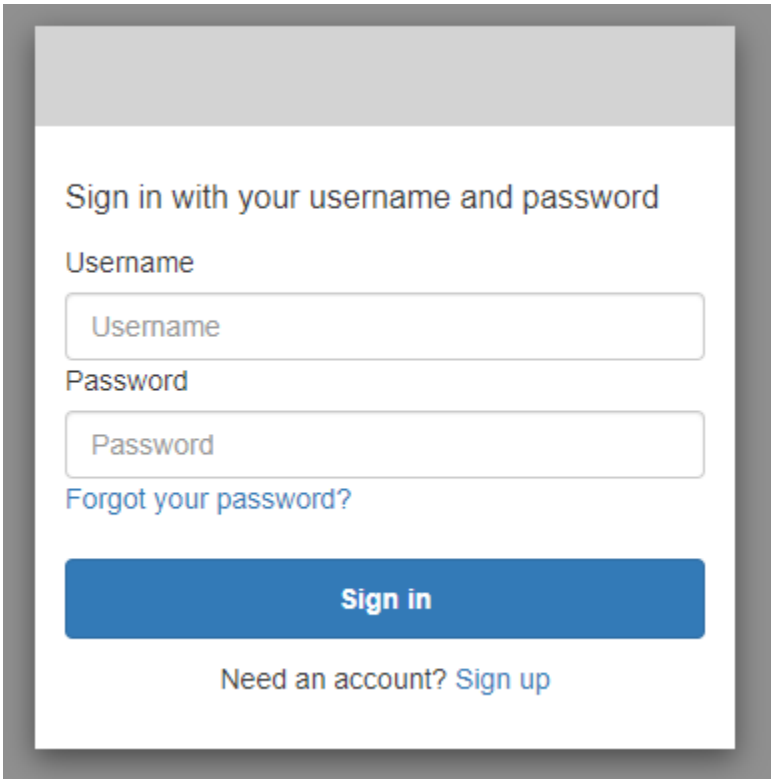
1. Según las opciones que haya elegido el propietario de la aplicación, es posible que pueda elegir los proveedores con los que iniciar sesión o que solo vea un mensaje con un nombre de usuario y una contraseña. Al iniciar sesión con un nombre de usuario y una contraseña de esta página, Amazon Cognito es su proveedor de inicio de sesión. De lo contrario, el proveedor de inicio de sesión se representa mediante el botón que elija.

   Si normalmente elige un proveedor en la página de inicio de sesión y su contraseña no funciona, siga el procedimiento para restablecerla con el proveedor. Si Amazon Cognito es su proveedor de inicio de sesión, elija Forgot your password? (¿Ha olvidado la contraseña?)

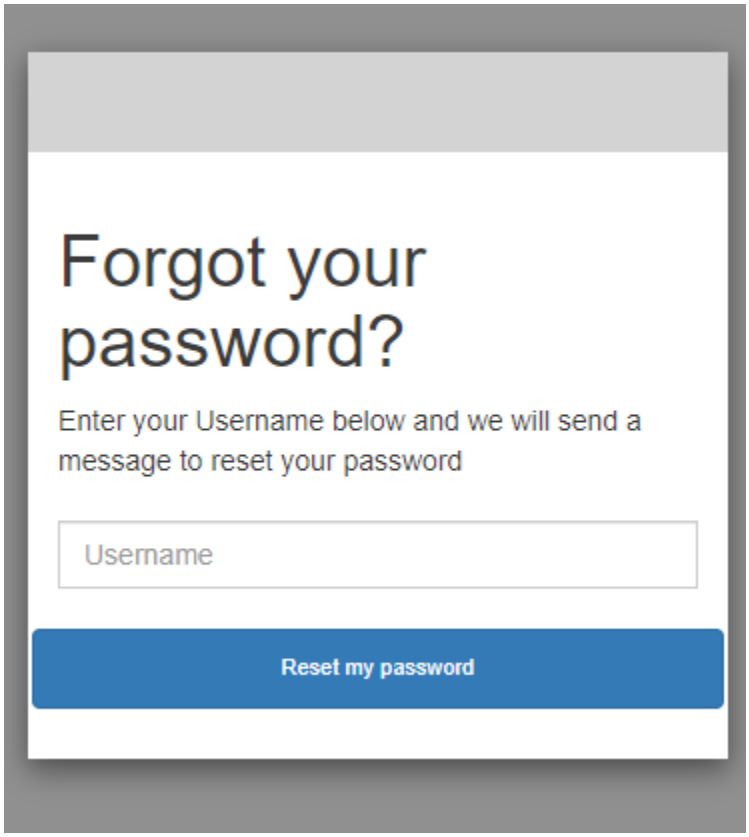
With multiple sign-in providers



With only Amazon Cognito as a sign-in provider

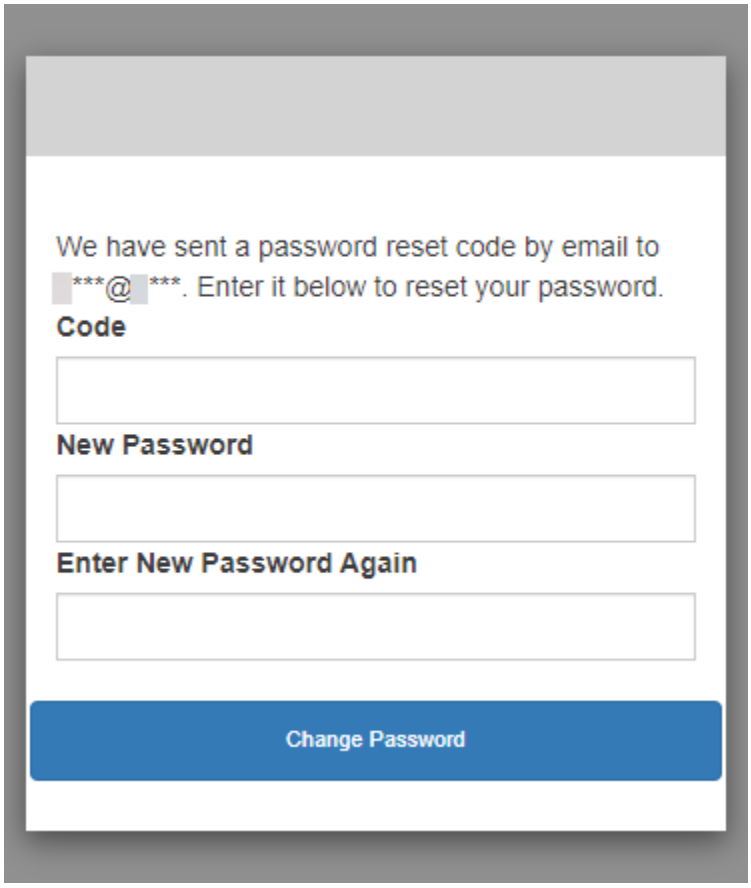


2. En la página Forgot your password? (¿Ha olvidado la contraseña?), Amazon Cognito le pide la información que utiliza para iniciar sesión. Podría ser su nombre de usuario, dirección de correo electrónico o número de teléfono.



3. Amazon Cognito le enviará un código como mensaje de correo electrónico o mensaje de texto SMS.

   Introduzca el código recibido e introduzca su nueva contraseña dos veces en los campos correspondientes. Debe ingresar el código de restablecimiento en un plazo de 8 minutos.



4. Después de cambiar la contraseña, vuelva a la página de inicio de sesión e inicie sesión con la nueva contraseña.

Autorización de alcances, M2M y API con servidores de recursos

Después de configurar un dominio para su grupo de usuarios, Amazon Cognito aprovisiona automáticamente un servidor de autorización OAuth 2.0 y una interfaz de usuario web hospedada con páginas de registro e inicio de sesión que su aplicación puede presentar a sus usuarios. Para más información, consulte Agregue un cliente de aplicaciones con la interfaz de usuario alojada. Puede elegir los ámbitos que desea que el servidor de autorización agregue a los tokens de acceso. Los ámbitos autorizan el acceso a los servidores de recursos y a los datos de los usuarios.

Un servidor de recursos es un servidor de API OAuth 2.0. Para asegurar los recursos con acceso protegido, valida que los tokens de acceso de su grupo de usuarios contengan los ámbitos que autorizan el método y la ruta solicitados en la API que protege. Verifica al emisor basándose en la firma del token, la validez en función del tiempo de caducidad del token y el nivel de acceso en función del alcance de las notificaciones de tokens. Los ámbitos del grupo de usuarios figuran en

la reclamación del token de acceso. `scope` Para obtener más información sobre las solicitudes de tokens de acceso a Amazon Cognito, consulte Uso del token de acceso.

Con Amazon Cognito, los ámbitos de los tokens de acceso pueden autorizar el acceso a las API externas o a los atributos del usuario. Puede emitir tokens de acceso a usuarios locales, usuarios federados o identidades de máquinas.

Autorización M achine-to-machine (M2M)

Amazon Cognito admite aplicaciones que acceden a los datos de la API con identidades de máquinas. Las identidades de las máquinas de los grupos de usuarios son clientes confidenciales que se ejecutan en servidores de aplicaciones y se conectan a API remotas. Su funcionamiento se lleva a cabo sin la interacción del usuario: tareas programadas, flujos de datos o actualizaciones de activos. Cuando estos clientes autorizan sus solicitudes con un token de acceso, realizan la autorización de máquina a máquina (M2M). En la autorización M2M, un secreto compartido reemplaza las credenciales de usuario en el control de acceso.

Una aplicación que accede a una API con autorización M2M debe tener un ID de cliente y un secreto de cliente. En su grupo de usuarios, debe crear un cliente de aplicaciones que admita la concesión de credenciales de cliente. Para admitir las credenciales de los clientes, el cliente de la aplicación debe tener un secreto de cliente y usted debe tener un dominio de grupo de usuarios. En este flujo, la identidad de su máquina solicita un token de acceso directamente desdePunto de conexión de token. Solo puede autorizar ámbitos personalizados de los servidores de recursos en los tokens de acceso para la concesión de credenciales de clientes. Para obtener más información sobre la configuración de los clientes de aplicaciones, consulteClientes de aplicación de grupo de usuarios.

El token de acceso que se obtiene al conceder las credenciales de un cliente es una declaración verificable de las operaciones que quieres permitir que la identidad de tu máquina solicite desde una API. Para obtener más información sobre cómo los tokens de acceso autorizan las solicitudes de API, sigue leyendo. Para ver un ejemplo de aplicación, consulte Autorización de máquina a máquina basada en Amazon Cognito y API Gateway mediante AWS CDK.

La autorización M2M tiene un modelo de facturación que difiere de la forma en que se factura a los usuarios activos mensuales (MAU). Si bien la autenticación de los usuarios conlleva un coste por usuario activo, la facturación M2M refleja las credenciales de los clientes activos, los clientes de aplicaciones y el volumen total de solicitudes de fichas. Para obtener más información, consulte Precios de Amazon Cognito. Para controlar los costes de la autorización M2M, optimice la duración de los tokens de acceso y el número de solicitudes de token que realizan sus aplicaciones. Consulte

Almacenamiento en caché de tokens para ver una forma de utilizar el almacenamiento en caché de API Gateway para reducir las solicitudes de nuevos tokens en la autorización M2M.

Para obtener información sobre cómo optimizar las operaciones de Amazon Cognito que añaden costes a su AWS factura, consulte. Administración de los costos de

Acerca de los ámbitos

Un ámbito es un nivel de acceso que una aplicación puede solicitar a un recurso. En un token de acceso de Amazon Cognito, el alcance está respaldado por la confianza que haya establecido con su grupo de usuarios: un emisor de tokens de acceso de confianza con una firma digital conocida. Los grupos de usuarios pueden generar tokens de acceso con ámbitos que demuestren que su cliente está autorizado para administrar parte o la totalidad de su propio perfil de usuario, o para recuperar datos de una API de backend. Los grupos de usuarios de Amazon Cognito emiten tokens de acceso con el ámbito de API reservado por los grupos de usuarios, los ámbitos personalizados y los ámbitos estándar.

Ámbito de API reservado para los grupos de usuarios

El ámbito `aws.cognito.signin.user.admin` autoriza a la API de grupos de usuarios de Amazon Cognito. Autoriza al portador de un token de acceso a consultar y actualizar toda la información sobre un usuario de un grupo de usuarios mediante, por ejemplo, las operaciones de la API GetUsery UpdateUserAttributesla API. Cuando autentique a su usuario con la API de grupos de usuarios de Amazon Cognito, este será el único ámbito que recibirá en el token de acceso. También es el único ámbito que necesita para leer y escribir atributos de usuario que haya autorizado que lea y escriba su cliente de aplicación. También puede solicitar este alcance en las solicitudes dirigidas al Autorizar punto de conexión. Este ámbito por sí solo no es suficiente para solicitar los atributos de usuario de Punto de conexión de UserInfo. En el caso de los tokens de acceso que autorizan la API de grupos de usuarios y las solicitudes de `userInfo` para los usuarios, debe solicitar ambos ámbitos `openid` y `aws.cognito.signin.user.admin` en una solicitud de `/oauth2/authorize`.

Ámbitos personalizados

Los ámbitos personalizados autorizan las solicitudes a las API externas que protegen los servidores de recursos. Puede solicitar ámbitos personalizados con otros tipos de ámbitos. Puede encontrar más información sobre los ámbitos personalizados en esta página.

Ámbitos estándar

Cuando autentique a los usuarios con el servidor de autorización OAuth 2.0 del grupo de usuarios, incluso con la interfaz de usuario alojada, deberá solicitar ámbitos. Puede autenticar usuarios locales de grupos de usuarios y usuarios federados de terceros en su servidor de autorización de Amazon Cognito. Los ámbitos estándar de OAuth 2.0 autorizan a su aplicación a leer la información de usuario del Punto de conexión de UserInfo de su grupo de usuarios. El modelo OAuth, por el que se consultan los atributos del usuario desde el punto de conexión de `userInfo`, puede optimizar su aplicación para un gran volumen de solicitudes de atributos del usuario. El punto de conexión de `userInfo` devuelve atributos en un nivel de permiso determinado por los ámbitos en el token de acceso. Puede autorizar al cliente de la aplicación a emitir tokens de acceso con los siguientes ámbitos estándar de OAuth 2.0.

openid

El ámbito mínimo para las consultas de OpenID Connect (OIDC). Autoriza el token de identificación, la solicitud de identificador único `sub` y la posibilidad de solicitar otros ámbitos.

> **Note**
>
> Cuando solicita el ámbito de `openid` y no otros, el token de ID del grupo de usuarios y la respuesta `userInfo` incluyen reclamaciones de todos los atributos de usuario que el cliente de la aplicación pueda leer. Cuando solicita `openid` y otros ámbitos estándar como `profile`, `email` y `phone`, el contenido del token de ID y la respuesta de userInfo se limitan a las restricciones de los ámbitos adicionales.
> Por ejemplo, una solicitud a Autorizar punto de conexión con el parámetro `scope=openid+email` devuelve un token de ID con `sub`, `email` y `email_verified`. El token de acceso de esta solicitud devuelve los mismos atributos de Punto de conexión de UserInfo. Una solicitud con un parámetro `scope=openid` devuelve todos los atributos legibles por el cliente del token de ID y de `userInfo`.

profile

Autoriza todos los atributos de usuario que el cliente de la aplicación puede leer.

email

Autoriza los atributos de usuario `email` y `email_verified`. Amazon Cognito devuelve `email_verified` si se ha establecido un conjunto de valores de forma explícita.

phone

Autoriza los atributos de usuario `phone_number` y `phone_number_verified`.

Acerca de los servidores de recursos

Una API de servidor de recursos puede conceder acceso a la información de una base de datos o controlar los recursos de TI. Un token de acceso de Amazon Cognito puede autorizar el acceso a las API compatibles con OAuth 2.0. Las API de REST de Amazon API Gateway tienen soporte integrado para obtener autorización con los tokens de acceso de Amazon Cognito. Su aplicación pasa el token de acceso de la llamada a API al servidor de recursos. El servidor de recursos inspecciona el token de acceso para determinar si debe conceder acceso.

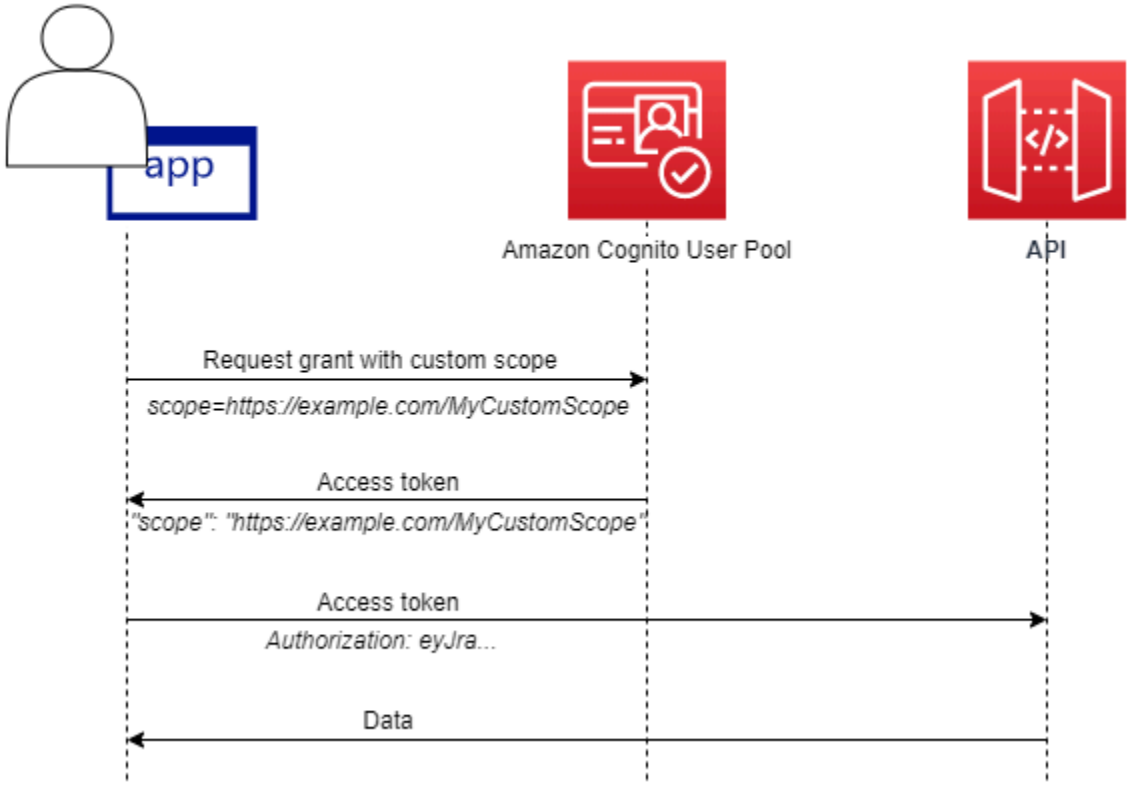
Amazon Cognito podría realizar actualizaciones futuras del esquema de tokens de acceso al grupo de usuarios. Si su aplicación analiza el contenido del token de acceso antes de pasarlo a una API, debe diseñar el código para que acepte actualizaciones del esquema.

Usted define los ámbitos personalizados, que amplían las capacidades de autorización de un grupo de usuarios para incluir fines no relacionados con la consulta y modificación de usuarios y sus atributos. Por ejemplo, si tiene un servidor de recursos para fotos, este puede definir dos ámbitos: `photos.read` para el acceso de lectura a las fotos y `photos.write` para el acceso de escritura y eliminación. Puede configurar una API para aceptar los tokens de acceso para autorización y la concesión de solicitudes de `HTTP GET` para acceder a los tokens con `photos.read` en la reclamación de `scope` y solicitudes de `HTTP POST` a tokens con `photos.write`. Estos son ámbitos personalizados.

> **Note**
>
> El servidor de recursos debe verificar la firma del token de acceso y la fecha de vencimiento antes de procesar las notificaciones del token. Para obtener más información sobre la verificación de tokens, consulte Verificación de un JSON Web Token. Para obtener más información sobre la verificación y utilización de tokens de grupos de usuarios en Amazon API Gateway, consulte el blog Integración de grupos de usuarios de Amazon Cognito con API Gateway. API Gateway es una buena opción para inspeccionar tokens de acceso y proteger sus recursos. Para obtener más información sobre los autorizadores de Lambda de API Gateway, consulte Uso de autorizadores Lambda de API Gateway.

Información general

Con Amazon Cognito, puede crear Servidores de recursos de OAuth 2.0 y asociar Ámbitos personalizados con ellos. Los ámbitos personalizados de un token de acceso autorizan acciones específicas en la API. Puede autorizar a cualquier cliente de aplicación del grupo de usuarios a emitir ámbitos personalizados desde cualquiera de los servidores de recursos. Asocie los ámbitos personalizados con un cliente de la aplicación y solicite esos ámbitos en las adjudicaciones de código de autorización OAuth2.0, las adjudicaciones implícitas y las adjudicaciones de credenciales de cliente de Punto de conexión de token. Amazon Cognito agrega ámbitos personalizados a la reclamación de `scope` en un token de acceso. Un cliente puede utilizar el token de acceso en su servidor de recursos, lo que hace que la decisión de conceder la autorización se base en los ámbitos presentes en el token. Para obtener más información acerca del ámbito de aplicación de tokens de acceso, consulte Uso de tokens con grupos de usuarios.



Para obtener un token de acceso con ámbitos personalizados, su aplicación debe enviar una solicitud al Punto de conexión de token para canjear un código de autorización o solicitar una concesión de credenciales de cliente. En la IU alojada, también puede solicitar ámbitos personalizados en un token de acceso a partir de una concesión implícita.

> **Note**
>
> Porque están diseñadas para la autenticación interactiva con personas con el grupo de usuarios como IdP InitiateAuth, AdminInitiateAuthy las solicitudes solo producen `scope` un reclamo en el token de acceso con el valor único. `aws.cognito.signin.user.admin`

Administrar el servidor de recursos y los ámbitos personalizados

Al crear un servidor de recursos, debe proporcionar un nombre y un identificador de servidor de recursos. Por cada ámbito que cree en el servidor de recursos, debe proporcionar un nombre y una descripción.

- Nombre de servidor de recursos: un nombre sencillo para el servidor de recursos, como `Solar system object tracker` o `Photo API`.
- Identificador de servidor de recursos: un identificador único para el servidor de recursos. El identificador es cualquier nombre que quiera asociar a su API, por ejemplo, `solar-system-data`. Puede configurar identificadores más largos, como `https://solar-system-data-api.example.com` como una referencia más directa a las rutas URI de la API, pero las cadenas más largas aumentan el tamaño de los tokens de acceso.
- Nombre del ámbito: el valor que quiere en las reclamaciones del `scope`. Por ejemplo, `sunproximity.read`.
- Descripción: una descripción sencilla del ámbito. Por ejemplo, `Check current proximity to sun`.

Amazon Cognito puede incluir ámbitos personalizados en los tokens de acceso para cualquier usuario, ya sea local del grupo de usuarios o federado con un proveedor de identidades de terceros. Puede elegir ámbitos para los tokens de acceso de sus usuarios durante los flujos de autenticación con el servidor de autorización OAuth 2.0 que incluye la interfaz de usuario alojada. La autenticación del usuario debe comenzar en Autorizar punto de conexión con `scope` como uno de los parámetros de la solicitud. A continuación, se presenta el formato recomendado para los servidores de recursos. Para un identificador, utilice un nombre fácil de usar para la API. Para un ámbito personalizado, utilice la acción que se autorice.

```
resourceServerIdentifier/scopeName
```

Por ejemplo, ha descubierto un nuevo asteroide en el cinturón de Kuiper y quiere registrarlo a través de su API `solar-system-data`. El ámbito que autoriza las operaciones de escritura en la base de datos de asteroides es `asteroids.add`. Cuando solicite el token de acceso que le autorizará a registrar su descubrimiento, formatee su parámetro de solicitud HTTPS `scope` como `scope=solar-system-data/asteroids.add`.

Eliminar un ámbito de un servidor de recursos no elimina su asociación con todos los clientes. En cambio, el ámbito está marcado inactivo. Amazon Cognito no agrega ámbitos inactivos para acceder a los tokens, sino que, por lo demás, continúa con normalidad si la aplicación solicita uno. Si vuelve a agregar el ámbito al servidor de recursos más adelante, Amazon Cognito lo vuelve a escribir en el token de acceso. Si solicita un ámbito que no ha asociado al cliente de la aplicación, independientemente de si lo ha eliminado del servidor de recursos del grupo de usuarios, se produce un error en la autenticación.

Puede usar la API o la AWS Management Console CLI para definir los servidores de recursos y los ámbitos de su grupo de usuarios.

Definir un servidor de recursos para el grupo de usuarios (AWS Management Console)

Puede utilizarla AWS Management Console para definir un servidor de recursos para su grupo de usuarios.

Para definir un servidor de recursos

1. Inicie sesión en la consola de Amazon Cognito.
2. En el panel de navegación, elija Manage your User Pools (Administrar sus grupos de usuarios) y elija el grupo de usuarios que desea editar.
3. Elija el iconoIntegración de aplicacionestab y localizarServidores de recursos.
4. Elija Create a resource share (Crear un recurso compartido).
5. Escriba unNombre del servidor de recursos. Por ejemplo, `Photo Server`.
6. Escriba unIdentificadores de servidores de. Por ejemplo, `com.example.photos`.
7. Ingrese los Custom scopes (Ámbitos personalizados) de sus recurso, por ejemplo, `read` y `write`.
8. Para cada Scope name (Nombre de ámbito), escriba una Description (Descripción), por ejemplo, `view your photos` y `update your photos`.
9. Seleccione Crear.

Los ámbitos personalizados se pueden revisar en elIntegración de aplicacionespestaña debajoServidores de recursos, en elÁmbitos personalizadoscolumn. Los ámbitos personalizados se pueden habilitar para clientes de aplicaciones desde elIntegración de aplicacionespestaña debajoClientes de aplicaciones. Seleccione un cliente de aplicación, localiceConfiguración de IU alojaday eligeEditar. AñadirÁmbitos personalizadosy eligeGuarde los cambios.

Definir un servidor de recursos para su grupo de usuarios (AWS CLI y AWS API)

Utilice los siguientes comandos para especificar la configuración del servidor de recursos para su grupo de usuarios.

Para crear un servidor de recursos

- AWS CLI: `aws cognito-idp create-resource-server`
- AWS API: CreateResourceServer

Para obtener información acerca de la configuración del servidor de recursos

- AWS CLI: `aws cognito-idp describe-resource-server`
- AWS API: DescribeResourceServer

Para mostrar información acerca de todos los servidores de recursos del grupo de usuarios

- AWS CLI: `aws cognito-idp list-resource-servers`
- AWS API: ListResourceServers

Para eliminar un servidor de recursos

- AWS CLI: `aws cognito-idp delete-resource-server`
- AWS API: DeleteResourceServer

Para actualizar la configuración de un servidor de recursos

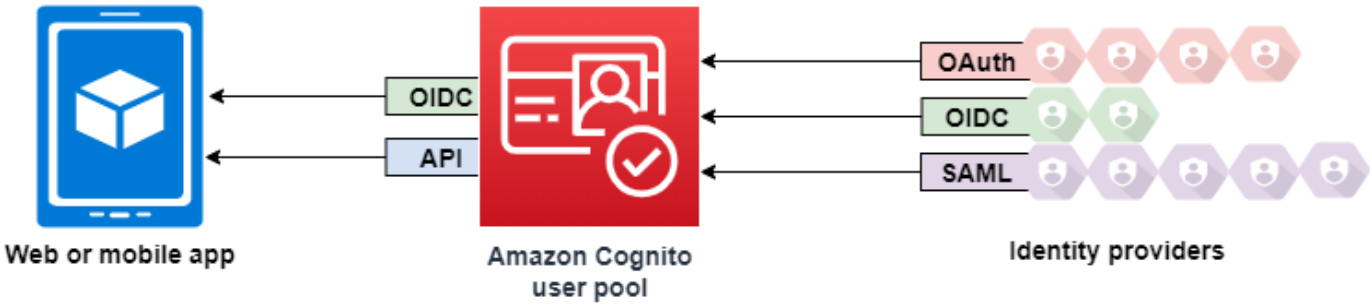
- AWS CLI: `aws cognito-idp update-resource-server`
- AWS API: UpdateResourceServer

Agregar inicio de sesión de grupo de usuarios a través de un tercero

Los usuarios de tu aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidad (IdP) externo. El grupo de usuarios gestiona la sobrecarga de gestión de los tokens que se devuelven al iniciar sesión en redes sociales a través de Facebook, Google, Amazon y Apple, y desde OpenID Connect (OIDC) y SAML. IdPs Con la interfaz de usuario web alojada integrada, Amazon Cognito permite gestionar y gestionar los tokens de todos los usuarios autenticados. IdPs De esta forma, los sistemas backend pueden estandarizar un conjunto de tokens para los grupos de usuarios.

Cómo funciona el inicio de sesión federado en los grupos de usuarios de Amazon Cognito

El inicio de sesión a través de un tercero (federación) está disponible en los grupos de usuarios de Amazon Cognito. Esta característica es independiente de la federación a través de grupos de identidades de Amazon Cognito (identidades federadas).



Amazon Cognito es un directorio de usuarios y un proveedor de identidades (IdP) de OAuth 2.0. Cuando registre usuarios locales en el directorio de Amazon Cognito, el grupo de usuarios es un IdP de la aplicación. Un usuario local existe exclusivamente en el directorio del grupo de usuarios sin federación a través de un IdP externo.

Cuando conecta Amazon Cognito a las redes sociales, SAML u OpenID Connect (OIDC IdPs), su grupo de usuarios actúa como un puente entre varios proveedores de servicios y su aplicación. Para su IdP, Amazon Cognito es un proveedor de servicios (SP). Debe IdPs pasar un token de ID de OIDC o una afirmación de SAML a Amazon Cognito. Amazon Cognito lee las afirmaciones sobre su usuario en el token o afirmación y las asigna a un nuevo perfil de usuario del directorio del grupo de usuarios.

A continuación, Amazon Cognito crea un perfil de usuario para el usuario federado en su propio directorio. Amazon Cognito agrega atributos a su usuario en función de las notificaciones de su IdP de identidad y, en el caso de OIDC y proveedores de identidad social, un punto de conexión `userinfo` operado por IDP. Los atributos de usuario cambian en el grupo de usuarios cuando cambia un atributo de IdP asignado. También puede agregar más atributos independientes de los del IdP.

Una vez que Amazon Cognito crea un perfil para el usuario federado, cambia su función y se presenta como IdP de su aplicación, que ahora es el SP. Amazon Cognito es una combinación de proveedor de identidad de OAuth 2.0 de OIDC. Genera tokens de acceso, tokens de ID y tokens de actualización. Para obtener más información acerca de los tokens, consulte Uso de tokens con grupos de usuarios.

Debe diseñar una aplicación que se integre con Amazon Cognito para autenticar y autorizar a los usuarios, federados o locales.

Las responsabilidades de una aplicación como proveedor de servicios con Amazon Cognito

Verificar y procesar la información de los tokens

En la mayoría de los casos, Amazon Cognito redirige al usuario autenticado a una URL de aplicación que agrega con un código de autorización. Su aplicación intercambia el código para tokens de acceso, ID y actualización. Entonces, debe comprobar la validez de los tokens y enviar información a su usuario en función de las afirmaciones de los tokens.

Responder a eventos de autenticación con solicitudes de API de Amazon Cognito

La aplicación debe integrarse con la API de grupos de usuarios de Amazon Cognito y lospuntos de conexión de la API de autenticación. La API de autenticación inicia y cierra sesión para el usuario y administra tokens. La API de grupos de usuarios tiene diversas operaciones que administran el grupo de usuarios, los usuarios y la seguridad del entorno de autenticación. La aplicación debe saber qué hacer a continuación cuando reciba una respuesta de Amazon Cognito.

Información que debe saber sobre los grupos de usuarios de Amazon Cognito: inicio de sesión de terceros

- Si desea que los usuarios inicien sesión con proveedores federados, debe elegir un dominio. Esto configura la interfaz de usuario alojada de Amazon Cognito y los puntos de conexión de interfaz de usuario y puntos de conexión de OIDC. Para obtener más información, consulte Uso de un dominio propio con la IU alojada.
- No puede iniciar sesión con usuarios federados con operaciones de API como y. InitiateAuthAdminInitiateAuth Los usuarios federados solo pueden iniciar sesión con el Punto de conexión Login o el Autorizar punto de conexión.
- El Autorizar punto de conexión es un punto de conexión de redirección. Si proporciona un parámetro `idp_identifier` o `identity_provider` en su solicitud, se redirige silenciosamente a su IdP, omitiendo la interfaz de usuario alojada. De lo contrario, se redirige al Punto de conexión Login de la interfaz de usuario alojada. Para ver un ejemplo, consulte Escenario de ejemplo: marcar aplicaciones de Amazon Cognito en un panel empresarial.
- Cuando la IU alojada redirige una sesión a un IdP federado, Amazon Cognito incluye el encabezado de `user-agent` `Amazon/Cognito` en la solicitud.
- Amazon Cognito deriva el atributo `username` de un perfil de usuario federado a partir de una combinación de un identificador fijo y el nombre de su IdP. Para generar un nombre de usuario que coincida con sus requisitos personalizados, cree una asignación al atributo `preferred_username`. Para obtener más información, consulte Cuestiones que debe saber acerca de los mapeos.

  Ejemplo: `MyIDP_bob@example.com`
- Amazon Cognito registra información sobre la identidad de su usuario federado en un atributo y una notificación en el token de ID, llamada `identities`. Esta notificación contiene el proveedor de su usuario y su ID exclusivo del proveedor. No se puede cambiar el atributo `identities` en un perfil de usuario directamente. Para obtener más información acerca de cómo vincular un usuario federado, consulte Vinculación de usuarios federados a un perfil de usuario existente.
- Cuando actualice su IdP en una solicitud de API UpdateIdentityProvider, los cambios pueden tardar hasta un minuto en aparecer en la interfaz de usuario alojada.
- Amazon Cognito admite hasta 20 redireccionamientos HTTP entre él y su IdP.
- Cuando el usuario inicia sesión con la interfaz de usuario alojada, el navegador almacena una cookie de inicio de sesión cifrada que registra el cliente y el proveedor con los que ha iniciado sesión. Si intentan iniciar sesión de nuevo con los mismos parámetros, la interfaz de usuario

alojada reutiliza cualquier sesión existente que no haya caducado y el usuario se autentica sin volver a proporcionar las credenciales. Si el usuario vuelve a iniciar sesión con un IdP diferente, incluido un cambio hacia o desde el inicio de sesión del grupo de usuarios local, debe proporcionar las credenciales y generar una nueva sesión de inicio de sesión.

Puedes asignar cualquier parte de tu grupo de usuarios IdPs a cualquier cliente de aplicaciones y los usuarios solo pueden iniciar sesión con un IdP que hayas asignado a su cliente de aplicaciones.

Temas

Configuración de proveedores de identidad para su grupo de usuarios

En la pestaña Experiencia de inicio de sesión, en Inicio de sesión con un proveedor de identidad federado, puede agregar proveedores de identidad (IdPs) a su grupo de usuarios. Para obtener más información, consulte Agregar inicio de sesión de grupo de usuarios a través de un tercero.

Temas

Configurar el inicio de sesión de los usuarios con un IdP de redes sociales

Puede utilizar la federación para que los grupos de usuarios de Amazon Cognito se integren en los proveedores de identidad de redes sociales, como Facebook, Google y Login with Amazon.

Para añadir un proveedor de identidad social, primero debe crear una cuenta de desarrollador con el proveedor de identidad. Después de crear la cuenta de desarrollador, registre la aplicación con

el proveedor de identidad. El proveedor de identidad crea un ID y un secreto de aplicación, y usted configura estos valores en su grupo de usuarios de Amazon Cognito.

- Google Identity Platform
- Facebook for Developers
- Login with Amazon
- Inicio de sesión con Apple

Para integrar el inicio de sesión de usuario con un IdP de redes sociales

1. Inicie sesión en la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS.
2. En el panel de navegación, elija User Pools (Grupos de usuarios) y elija el grupo de usuarios que desea editar.
3. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión) y localice Federated sign-in (Inicio de sesión federado).
4. Elija Add an identity provider (Agregar un proveedor de identidad), o elija el proveedor de identidad de Facebook,Google, Amazon o Apple que ha configurado, localice Identity provider information (Información de proveedor de identidad), y elija Edit (Editar). Para obtener más información acerca de agregar un proveedor de identidad social, consulte Usar proveedores de identidad social con un grupo de usuarios.
5. Introduzca la información de su proveedor de identidad social realizando uno de los siguientes pasos, según su elección de IdP:

   Facebook, Google y Login with Amazon

   Ingrese el ID y el secreto de aplicación que recibió al crear la aplicación de cliente.

   Inicio de sesión con Apple

   Ingrese el ID de servicio que proporcionó a Apple, así como el ID de equipo, el ID de clave y la clave privada que recibió al crear el cliente de aplicación.
6. Para Authorize scopes (Autorizar ámbitos), introduzca los nombres de los ámbitos de los proveedores de identidad social que desea asignar a los atributos del grupo de usuarios. Los ámbitos definen a qué atributos de usuario, tales como nombre y correo electrónico, desea acceder con su aplicación. Al introducir ámbitos, utilice las siguientes pautas que se basan en su elección del proveedor de identidad (IdP):

- Facebook — Ámbitos separados con comas. Por ejemplo:

  `public_profile, email`

- Google, Login with Amazon y SignInWithApple — Ámbitos separados con espacios. Por ejemplo:
  - Google: `profile email openid`
  - Login with Amazon: `profile postal_code`
  - SignInWithApple: `name email`

> **Note**
>
> Para SignInWithApple (consola), utilice las casillas de verificación para elegir ámbitos.

7. Elija Guardar cambios.
8. Desde el pestaña App client integration (Integración de clientes de aplicaciones), elija uno de los App clients (Clientes de aplicaciones) en la lista y, a continuación, elija Edit hosted UI settings (Editar la configuración de IU). Agregue el nuevo proveedor de identidad social al cliente de aplicación en Identity providers (Proveedores de identidad).
9. Elija Guardar cambios.

Para obtener más información sobre las redes sociales, consulte IdPs. Usar proveedores de identidad social con un grupo de usuarios

Configurar el inicio de sesión de usuarios con un IdP de OIDC

Puede integrar el inicio de sesión de usuarios a través de un proveedor de identidad OpenID Connect (OIDC), como Salesforce o Ping Identity.

Para agregar un proveedor OIDC a un grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS.
2. Elija User Pools (Grupos de usuarios) en el menú de navegación.
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.

4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). Localice Federated sign-in (Inicio de sesión federado) y luego seleccione Add an identity provider (Agregar un proveedor de identidad).
5. Elija un proveedor de identidad de OpenID Connect.
6. Introduzca un nombre único en Provider name (Nombre de proveedor).
7. Introduzca el ID de cliente que recibió de su proveedor en Client ID (ID de cliente).
8. Introduzca el secreto de cliente que recibió de su proveedor en Client Secret (Secreto de cliente).
9. Introduzca los Ámbitos autorizados para este proveedor. Los ámbitos definen qué grupos de atributos de usuario (tales como `name` y `email`) serán solicitados por su aplicación al proveedor. Los ámbitos deben estar separados por espacios, de acuerdo con la especificación OAuth 2.0.

   El usuario debe autorizar que se proporcionen estos atributos a su aplicación.
10. Seleccione un Attribute request method (Método de solicitud de atributo) para proporcionar a Amazon Cognito el método de HTTP (GET o POST) que usa Amazon Cognito para obtener los detalles de usuario del punto de conexión userInfo operado por su proveedor.
11. Seleccione un Setup method (Método de configuración) para recuperar los puntos de enlace de OpenID Connect con Auto fill through issuer URL (Autorrellenar mediante la URL del emisor) o Manual input (Entrada manual). Use Auto fill through issuer URL (Autorrellenar mediante la URL del emisor) cuando su proveedor tenga un punto de conexión `.well-known/openid-configuration` público en el que Amazon Cognito pueda recuperar las URL de los puntos de conexión de `authorization`, `token`, `userInfo` y `jwks_uri`.
12. Introduzca la URL del emisor o las URL de los puntos de conexión de `authorization`, `token`, `userInfo` y `jwks_uri` de su IdP.

> **Note**
>
> Solo puede utilizar los números de puerto 443 y 80 con las URL de detección, rellenado automático e ingresadas manualmente. Los inicios de sesión de usuario fallan si su proveedor de OIDC utiliza puertos TCP no estándar.
> La URL del emisor debe comenzar por `https://` y no pueden terminar con el carácter `/`. Por ejemplo, Salesforce usa esta URL: `https://login.salesforce.com`
> El documento `openid-configuration` asociado a la URL del emisor debe proporcionar URL HTTPS para los siguientes valores: `authorization_endpoint`,

`token_endpoint`, `userinfo_endpoint` y `jwks_uri`. Del mismo modo, cuando elija Manual input (Entrada manual), solo podrá ingresar URL HTTPS.

13. A la notificación OIDC sub se le asigna el atributo de grupo de usuarios Username (Nombre de usuario) de forma predeterminada. Puede asignar a las notificaciones OIDC otros atributos de grupo de usuarios. Introduzca la notificación OIDC y seleccione el atributo de grupo de usuarios correspondiente en la lista desplegable. Por ejemplo, a la notificación email (correo electrónico) se le suele asignar el atributo de grupo de usuarios Email (Correo electrónico).
14. Asigne atributos adicionales de su proveedor de identidades a su grupo de usuarios. Para obtener más información, consulte Especificación de asignaciones de atributos del proveedor de identidad para su grupo de usuarios.
15. Seleccione Crear.
16. Desde la pestaña App client integration (Integración de clientes de aplicaciones), seleccione una entre App clients (Clientes de aplicaciones) en la lista y Edit hosted UI settings (Editar configuración de IU alojadas). Agregue el nuevo proveedor de identidad OIDC al cliente de la aplicación en Identity providers (Proveedores de identidad).
17. Elija Guardar cambios.

Para obtener más información sobre el OIDC IdPs, consulte. Uso de proveedores de identidad OIDC con un grupo de usuarios

Configurar el inicio de sesión de usuario con un IdP SAML

Puede utilizar la federación de grupos de usuarios de Amazon Cognito para que se integren en un proveedor de identidad (IdP) SAML. Proporcione un documento de metadatos, ya sea cargando el archivo o escribiendo una URL de punto de enlace del documento de metadatos. Para obtener información sobre cómo obtener documentos de metadatos para el SAML IdPs de terceros, consulte. Configurar tu proveedor de identidades SAML externo

Para configurar un proveedor de identidad SAML 2.0 en su grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.

4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). Localice Federated sign-in (Inicio de sesión federado) y luego seleccione Add an identity provider (Agregar un proveedor de identidad).
5. Elija un proveedor de identidad SAML.
6. Introduzca los identificadores separados por comas. Un identificador indica a Amazon Cognito que debe comprobar la dirección de correo electrónico que introduce un usuario al iniciar sesión y, a continuación, dirigirlo al proveedor que corresponda a su dominio.
7. Elija Add sign-out flow (Añadir flujo de cierre de sesión) si desea que Amazon Cognito envíe solicitudes de cierre de sesión firmadas a su proveedor cuando un usuario cierra la sesión. Configure el proveedor de identidad SAML 2.0 para que envíe respuestas de cierre de sesión al punto de conexión `https://`*`mydomain.us-east-1.amazoncognito.com`*`/saml2/logout` que crea Amazon Cognito al configurar la IU alojada. El punto de conexión `saml2/logout` utiliza el enlace POST.

   > **Note**
   >
   > Si selecciona esta opción y el proveedor de identidad SAML espera una solicitud de cierre de sesión firmada, también deberá configurar el certificado de firma que ofrece Amazon Cognito en el IdP SAML.
   > El proveedor de identidad SAML procesará la solicitud de cierre de sesión firmada y cerrará la sesión de Amazon Cognito del usuario.

8. Seleccione un Origen de documentos de metadatos. Si su proveedor de identidad ofrece metadatos SAML en una URL pública, puede elegir Metadata document URL (URL del documento de metadatos) e introducir esa URL pública. En caso contrario, elija Upload metadata document (Cargar documento de metadatos) y seleccione un archivo de metadatos que haya descargado anteriormente de su proveedor.

   > **Note**
   >
   > Si su proveedor tiene un punto de conexión público, le recomendamos que ingrese una URL de documento de metadatos, en lugar de cargar un archivo. Si utiliza la URL, Amazon Cognito actualiza los metadatos automáticamente. Normalmente, los metadatos se actualizan cada seis horas o antes de que caduquen, lo que ocurra primero.

9. Asigne atributos entre el proveedor de SAML y la aplicación para asignar atributos de proveedor SAML al perfil de usuario de su grupo de usuarios. Incluya los atributos requeridos del grupo de usuarios en la asignación de atributos.

   Por ejemplo, cuando elige User pool attribute (Atributo grupo de usuarios) `email`, escriba el nombre de atributo SAML tal como aparece en la aserción SAML del proveedor de identidad. Es posible que su proveedor de identidades ofrezca aserciones SAML de ejemplo y como referencia. Algunos proveedores de identidad utilizan nombres sencillos, como `email`, mientras que otros utilizan nombres de atributo con formato de URL similares a este:

   ```
   http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
   ```

10. Seleccione Crear.

> **Note**
>
> Si aparece `InvalidParameterException` al crear un IdP SAML con una URL de punto de conexión de metadatos HTTPS, asegúrese de que el punto de conexión de los metadatos tenga SSL correctamente configurado y de que tenga un certificado SSL válido asociado. Un ejemplo de una excepción de este tipo sería “Error al recuperar el *<punto de conexión de los metadatos>*”.

Para configurar el proveedor de identidad SAML para añadir un certificado de firma

- Para obtener el certificado que contiene la clave pública que utiliza el IdP para comprobar la solicitud de cierre de sesión firmada, elija Mostrar certificado de firma en Proveedores SAML activos en el cuadro de diálogo SAML en Proveedores de identidad en la página de la consola Federación.

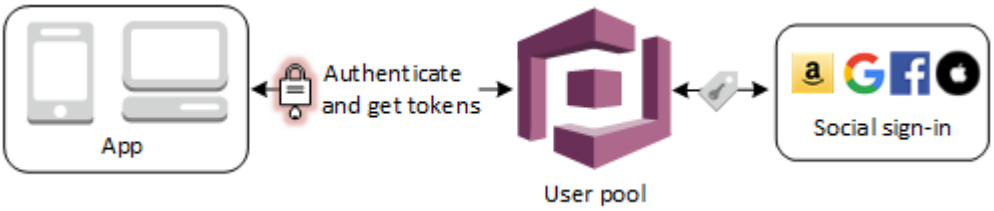
Para obtener más información sobre SAML IdPs , consulte. Uso de proveedores de identidad SAML con un grupo de usuarios

Usar proveedores de identidad social con un grupo de usuarios

Los usuarios de web y aplicaciones móviles pueden iniciar sesión a través de proveedores de identidad de redes sociales como Facebook, Google, Amazon y Apple. Con la IU web alojada e incorporada, Amazon Cognito proporciona el control y la administración de los tokens de los usuarios

autenticados por todos los proveedores de identidad. De esta forma, los sistemas backend pueden estandarizar un conjunto de tokens para los grupos de usuarios. Debe habilitar la IU alojada para que se integre con los proveedores de identidad social compatibles. Cuando Amazon Cognito crea su interfaz de usuario alojada, crea puntos de enlace de OAuth 2.0 que Amazon Cognito y su OIDC y sus redes sociales utilizan para intercambiar información. IdPs Para obtener más información, consulte la Referencia de la API de Auth para grupos de usuarios de Amazon Cognito.

Puede añadir un IDP social en la AWS CLI o la AWS Management Console API de Amazon Cognito, o bien utilizar la misma.



> **Note**
>
> El inicio de sesión a través de un tercero (federación) está disponible en los grupos de usuarios de Amazon Cognito. Esta característica es independiente de la federación a través de grupos de identidades de Amazon Cognito (identidades federadas).

Temas

- Requisitos previos
- Paso 1: Registrarse en un proveedor de identidad social
- Paso 2: Añadir un proveedor de identidad social al grupo de usuarios
- Paso 3: Probar la configuración del proveedor de identidad social

Requisitos previos

Antes de comenzar, necesitará lo siguiente:

- Un grupo de usuarios con un cliente de aplicación y un dominio de grupo de usuarios. Para obtener más información, consulte Crear un grupo de usuarios.
- Un IdP para redes sociales.

Paso 1: Registrarse en un proveedor de identidad social

Antes de crear un proveedor de identidad social con Amazon Cognito, debe registrar su aplicación en él para recibir un ID y un secreto del cliente.

Para registrar una aplicación en Facebook

1. Cree una cuenta de desarrollador con Facebook.
2. Inicie sesión con sus credenciales de Facebook.
3. En el menú My Apps (Mis aplicaciones), elija Create New App (Crear nueva aplicación).
4. Escriba un nombre para la aplicación de Facebook y, a continuación, elija Create App ID (Crear ID de aplicación).
5. En la barra de navegación de la izquierda, elija Settings (Configuración) y luego Basic (Básica).
6. Tome nota del valor de App ID (ID de aplicación) y de App Secret (Secreto de la aplicación). Los usará en la sección siguiente.
7. Elija + Add Platform (+ Agregar plataforma) en la parte inferior de la página.
8. Elija Website (Sitio web).
9. En Website (Sitio web), escriba la ruta de acceso a la página de inicio de sesión de la aplicación en Site URL (URL del sitio).

   ```
   https://mydomain.us-east-1.amazoncognito.com/login?
   response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
   ```

10. Elija Guardar cambios.
11. Ingrese la ruta de acceso a la raíz del dominio del grupo de usuarios en App Domains (Dominios de aplicación).

    ```
    https://mydomain.us-east-1.amazoncognito.com
    ```

12. Elija Guardar cambios.
13. En la barra de navegación elija Products (Productos) y, a continuación, Set up (Configurar) para el producto con Facebook Login (Inicio de sesión con Facebook).
14. En la barra de navegación elija Facebook Login (Inicio de sesión con Facebook) y, a continuación, Settings (Configuración).

Introduzca la ruta de acceso al punto de conexión `/oauth2/idpresponse` para el dominio del grupo de usuarios en Valid OAuth Redirect URIs (URI de redirección de OAuth válidas).

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

15. Elija Guardar cambios.

Para registrar una aplicación en Amazon

1. Cree una cuenta de desarrollador con Amazon.
2. Inicie sesión con las credenciales de Amazon.
3. Debe crear un perfil de seguridad de Amazon para recibir un ID y un secreto de cliente de Amazon.

   Elija Apps and Services (Aplicaciones y servicios) en la barra de navegación de la parte superior de la página y, a continuación, elija Login with Amazon (Inicio de sesión con Amazon).
4. Elija Create a Security Profile (Crear un perfil de seguridad).
5. Escriba un valor en Security Profile Name (Nombre del perfil de seguridad), en Security Profile Description (Descripción del perfil de seguridad) y en Consent Privacy Notice URL (URL del aviso sobre consentimiento de confidencialidad).
6. Seleccione Save (Guardar).
7. Elija Client ID (ID de cliente) y Client Secret (Secreto de cliente) para mostrar el ID de cliente y el secreto. Los usará en la sección siguiente.
8. Coloque el cursor sobre el engranaje, elija Web Settings (Configuración de web) y, a continuación, elija Edit (Editar).
9. Escriba el dominio del grupo de usuarios en Allowed Origins (Orígenes permitidos).

   ```
   https://mydomain.us-east-1.amazoncognito.com
   ```
10. Escriba el dominio del grupo de usuarios con el punto de conexión `/oauth2/idpresponse` en URL de devolución permitidas.

    ```
    https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
    ```
11. Seleccione Save (Guardar).

Para registrar una aplicación en Google

Para obtener más información sobre OAuth 2.0 en la plataforma de Google Cloud, consulte Más información sobre la autenticación y la autorización en la documentación de Google Workspace for Developers.

1. Cree una cuenta de desarrollador con Google.
2. Inicie sesión en la consola de Google Cloud Platform.
3. En la barra de navegación superior, elija Select a project (Seleccionar un proyecto). Si ya tiene un proyecto en la plataforma de Google, este menú muestra tu proyecto predeterminado.
4. Seleccione NEW PROJECT (NUEVO PROYECTO).
5. Escriba un nombre para su proyecto y, a continuación, elija CREATE (CREAR).
6. En la barra de navegación izquierda, elija APIs and Services (API y servicios), luego Oauth consent screen (Pantalla de consentimiento de Oauth).
7. Introduzca la información de la aplicación, un dominio de aplicaciones, dominios autorizados e información de contacto del desarrollador. Sus dominios autorizados deben incluir `amazoncognito.com` y la raíz de su dominio personalizado, por ejemplo `example.com`. Elija SAVE AND CONTINUE (GUARDAR Y CONTINUAR).
8. 1. En Scopes (Ámbitos), elija Add or remove scopes (Agregar y eliminar ámbitos) y elija, como mínimo, los siguientes ámbitos de OAuth.

    1. `.../auth/userinfo.email`
    2. `.../auth/userinfo.profile`
    3. openid
9. En Test Users (Usuarios de prueba), elija Add Users (Añadir usuarios). Introduzca su dirección de correo electrónico y cualquier otro usuario de prueba autorizado y, a continuación, elija SAVE AND CONTINUE (GUARDAR Y CONTINUAR).
10. Expanda de nuevo la barra de navegación izquierda y elija APIs and Services (API y servicios), luego Credentials (Credenciales).
11. Elija CREATE CREDENTIALS (CREAR CREDENCIALES), luego OAuth client ID (ID de cliente de OAuth).
12. Seleccione un tipo de aplicacióny asigne un nombre al cliente.
13. En JavaScript Orígenes autorizados, elija AGREGAR URI. Introduzca el dominio del grupo de usuarios.

```
https://mydomain.us-east-1.amazoncognito.com
```

14. En Authorized redirect URIs (URI de redirección autorizadas), elija ADD URI (AÑADIR URI). Introduzca la al punto de conexión `/oauth2/idpresponse` de su dominio de grupo de usuarios.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

15. Seleccione CREATE (Crear).
16. Almacene de forma segura los valores en los que muestra Google EN Your client ID (ID de tu cliente y Your client secret (Secreto de tu cliente). Proporcione estos valores a Amazon Cognito cuando agregue un proveedor de IdP Google.

Para registrar una aplicación con Apple, siga estos pasos:

Para up-to-date obtener más información sobre cómo configurar el inicio de sesión con Apple, consulta Cómo configurar tu entorno para iniciar sesión con Apple en la documentación para desarrolladores de Apple.

1. Cree una cuenta de desarrollador en Apple.
2. Inicie sesión con las credenciales de Apple.
3. En la barra de navegación de la izquierda, elija Certificates, Identifiers & Profiles (Certificados, identificadores y perfiles).
4. En la barra de navegación de la izquierda, elija Identifiers (Identificadores).
5. En la página Identifiers (Identificadores), elija el icono +.
6. En la página Register a New Identifier (Registrar un nuevo identificador), elija App IDs (ID de aplicaciones) y, a continuación, Continue (Continuar).
7. En la página Select a type (Seleccionar tipo), elija App y, a continuación, elija Continue (Continuar).
8. En la página Register an App ID (Registrar un ID de aplicación), haga lo siguiente:

    1. En Description (Descripción), introduzca una descripción.
    2. En App ID Prefix (Prefijo de ID de aplicación), introduzca un ID del paquete. Anote el valor de laPrefijo de ID de aplicación. Utilizarás este valor después de elegir Apple como proveedor de identidad enPaso 2: Añadir un proveedor de identidad social al grupo de usuarios.

3. En Capabilities (Funcionalidades), elija SignInWithApple y, a continuación, elija Edit (Editar).
4. En la página Sign in with Apple: App ID Configuration (Inicio de sesión con Apple: Configuración del ID de aplicación), elija configurar la aplicación como principal o agrupada con otros ID de aplicación y, a continuación, elija Save (Guardar).
5. Elija Continue (Continuar).

9. En la página Confirm your App ID (Confirmar ID de Apple), elija Register (Registrarse).
10. En la página Identifiers (Identificadores), elija el icono +.
11. En la página Register a New Identifier (Registrar un nuevo identificador), elija Services IDs (ID de servicios) y, a continuación, Continue (Continuar).
12. En la página Register a Services ID (Registrar un ID de servicio), haga lo siguiente:

    1. En Description (Descripción), escriba una descripción.
    2. En Identifier (Identificador), escriba un identificador. Anote el ID de servicios, ya que necesitará este valor para configurar Apple como proveedor en su grupo de identidades de Paso 2: Añadir un proveedor de identidad social al grupo de usuarios.
    3. Seleccione Continue (Continuar), a continuación, Register (Registrarse).
13. Elija el ID de servicios que acaba de crear en la página de identificadores.

    1. Seleccione SignInWithApple y, a continuación, elija Configure (Configurar).
    2. En la página Web Authentication Configuration (Configuración de autenticación web), seleccione el ID de aplicación creado anteriormente comoPrimary App ID (ID de aplicación principal).
    3. Elija el icono + situado al lado de Website URLs (URL de sitio web).
    4. En Domains and subdomains (Dominios y subdominios), introduzca el dominio del grupo de usuarios sin un prefijo `https://`.

    ```
    mydomain.us-east-1.amazoncognito.com
    ```

    5. En Return URLs (URL de devolución), introduzca la ruta al punto de conexión `/oauth2/idpresponse` del dominio del grupo de usuarios.

    ```
    https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
    ```

    6. Elija Next (Siguiente) y, a continuación, elija Done (Listo). No es necesario verificar el dominio.

7. Elija Continue (Continuar) y, a continuación, elija Save (Guardar).

14. En la barra de navegación de la izquierda, elija Keys (Claves).
15. En la página Keys (Claves), elija el icono +.
16. En la página Register a New Key (Registrar una nueva clave), haga lo siguiente:

    1. En Key Name (Nombre de clave), escriba un nombre de clave.
    2. Elija SignInWithApple y, a continuación, Configure (Configurar).
    3. En la página Configure Key (Configurar clave), seleccione el ID de aplicación creado anteriormente como Primary App ID (ID de aplicación principal). Seleccione Guardar.
    4. Seleccione Continue (Continuar) y, a continuación, Register (Registrarse).

17. En la página Download Your Key (Descargar clave), elija Download (Descargar) para descargar la clave privada, anote el Key ID (ID de la clave) y, a continuación, Done (Listo). Necesitará esta clave privada y el valor de ID de clave que se muestra en esta página después de elegir Apple como proveedor de identidad en Paso 2: Añadir un proveedor de identidad social al grupo de usuarios.

Paso 2: Añadir un proveedor de identidad social al grupo de usuarios

Para configurar el IdP social de un grupo de usuarios con el AWS Management Console

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). Localice Federated sign-in (Inicio de sesión federado) y, a continuación, seleccione Add an identity provider (Añadir un proveedor de identidad).
5. Elija un IdP para redes sociales: Facebook, Google, Login with Amazon o Apple.
6. Elija uno de los siguientes pasos, según el IdP para redes sociales que haya seleccionado:

    - Google y Login with Amazon: Escriba la ID de cliente de aplicacióny el secreto del cliente de aplicación generado en la sección anterior.
    - Facebook: escriba la ID de cliente de aplicación y el secreto del cliente de aplicación generado en la sección anterior y, a continuación, elija una versión de API (por ejemplo, la versión 2.12). Recomendamos elegir la versión más reciente disponible posible, ya que cada versión de la

API de Facebook tiene un ciclo de vida y una fecha de retirada. Los ámbitos y atributos de Facebook pueden variar según las versiones de la API. Recomendamos que pruebe su inicio de sesión de identidad social con Facebook para asegurarse de que la federación funcione según lo previsto.

- Inicio de sesión con Apple: escriba laID de servicio,ID de equipo,ID de clave, yClave privadagenerado en la sección anterior.

7. Introduzca los nombres de los ámbitos autorizados que desea utilizar. Los ámbitos definen a qué atributos de usuario (como `name` y `email`) desea acceder con su aplicación. En el caso de Facebook, deben separarse con comas. En el caso de Google y Login with Amazon, deben separarse con espacios. Para SignInWithApple, marque las casillas de verificación de los ámbitos a los que desee acceder.

Proveedor de identidad social	Ámbitos de ejemplo
Facebook	`public_profile, email`
Google	`profile email openid`
Login with Amazon	`profile postal_code`
Inicio de sesión con Apple	`email name`

Al usuario de la aplicación se le pedirá que esté de acuerdo con proporcionar estos atributos a su aplicación. Para obtener más información acerca de sus ámbitos, consulte la documentación de Google, Facebook, Login with Amazon o Inicio de sesión con Apple.

En el caso de Sign in with Apple (Inicio de sesión con Apple), estos son escenarios de usuario en los que es posible que no se devuelvan los ámbitos.

- Un usuario final se encuentra con errores después de salir de la página de inicio de sesión de Apple (puede ser un error interno de Amazon Cognito o de cualquier cosa que haya escrito el desarrollador).
- El identificador de ID de servicio se utiliza en todos los grupos de usuarios u otros servicios de autenticación.
- Un desarrollador añade ámbitos adicionales después de que el usuario final haya iniciado sesión (no se recupera ninguna información nueva).

- Un desarrollador elimina al usuario y luego el usuario vuelve a iniciar sesión sin quitar la aplicación de su perfil de ID de Apple.

8. Asigne atributos de su IdP a su grupo de usuarios. Para obtener más información, consulte Especificación de asignaciones de atributos del proveedor de identidad para su grupo de usuarios.
9. Seleccione Crear.
10. DelIntegración de clientes de aplicaciones, elija uno de los Clientes de aplicaciones en la lista y Edit hosted UI settings (Modificar la configuración de IU). Agregue el nuevo IdP social al cliente de aplicación en Identity providers (Proveedores de identidad).
11. Elija Guardar cambios.

Paso 3: Probar la configuración del proveedor de identidad social

Puede crear una URL de inicio de sesión con los elementos de las dos secciones anteriores. Úselo para probar la configuración del proveedor de identidad social.

```
https://mydomain.us-east-1.amazoncognito.com/login?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

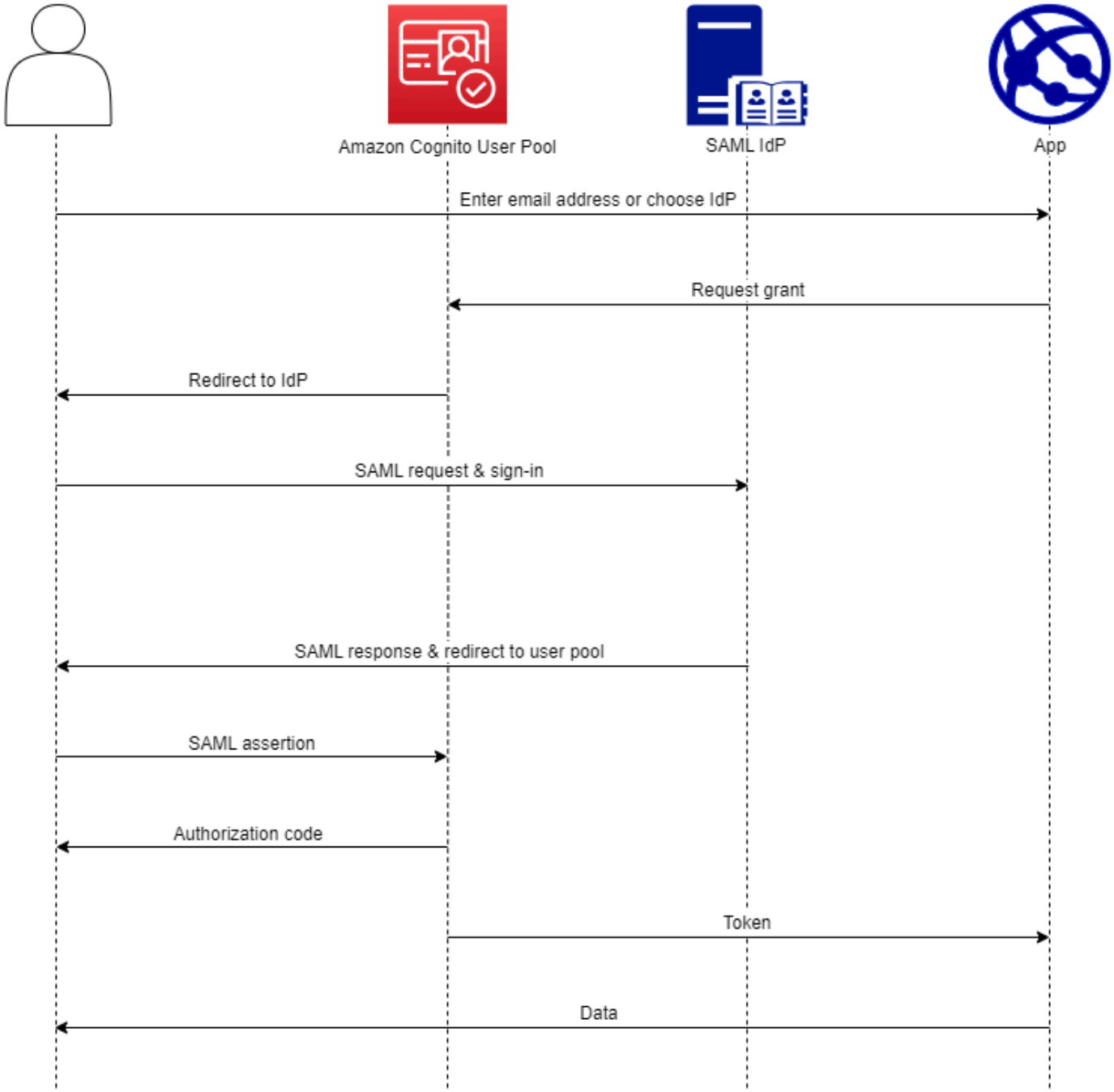
Puede encontrar el dominio en la página de la consola Domain name (Nombre de dominio) del grupo de usuarios. El valor de client_id se encuentra en la página App client settings (Configuración del cliente de aplicación). Use la URL de devolución de llamada para el parámetro redirect_uri. Esta es la URL de la página a la que se redirigirá al usuario después de una autenticación correcta.

> **Note**
>
> Amazon Cognito cancela las solicitudes de autenticación que no se completan en 5 minutos y redirige al usuario a la IU alojada. La página muestra un mensaje de error `Something went wrong`.

Uso de proveedores de identidad SAML con un grupo de usuarios

Puede elegir que los usuarios de sus aplicaciones web y móviles inicien sesión a través de un proveedor de identidad (IdP) de SAML, como Microsoft Active Directory Federation Services (ADFS) o Shibboleth. Debe elegir un IdP SAML compatible con el estándar SAML 2.0.

Con la interfaz de usuario alojada y los puntos de enlace de federación, Amazon Cognito autentica a los usuarios de IDP locales y de terceros y emite tokens web JSON (JWT). Con los tokens que emite Amazon Cognito, puede consolidar varias fuentes de identidad en un estándar universal de OpenID Connect (OIDC) en todas sus aplicaciones. Amazon Cognito puede procesar las afirmaciones de SAML de sus proveedores externos en ese estándar de SSO. Puede crear y administrar un IDP de SAML en AWS Management Console, a través de o con AWS CLI la API de grupos de usuarios de Amazon Cognito. Para crear su primer IdP de SAML en AWS Management Console, consulte. Añadir y administrar proveedores de identidad de SAML en un grupo de usuarios



Note

La federación con inicio de sesión a través de un IdP de terceros es una característica de los grupos de usuarios de Amazon Cognito. Los grupos de identidades de Amazon Cognito, también denominados identidades federadas de Amazon Cognito, son una implementación de la federación que debe configurar por separado en cada grupo de identidades. Un grupo

de usuarios puede ser un IdP de terceros para un grupo de identidades. Para obtener más información, consulte Grupos de identidades de Amazon Cognito.

Referencia rápida para la configuración del IdP

Debe configurar su IdP de SAML para aceptar solicitudes y enviar respuestas a su grupo de usuarios. La documentación de su IdP de SAML incluirá información sobre cómo añadir su grupo de usuarios como parte de confianza o aplicación para su IdP de SAML 2.0. La siguiente documentación proporciona los valores que debe proporcionar para el ID de entidad del SP y la URL del servicio al consumidor de aserciones (ACS).

Referencia rápida de valores SAML del grupo de usuarios

ID de entidad SP

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

URL ACS

```
https://Your user pool domain/saml2/idpresponse
```

Debe configurar su grupo de usuarios para que sea compatible con su proveedor de identidad. Los pasos de alto nivel para agregar un IdP SAML externo son los siguientes.

1. Descarga los metadatos de SAML de tu IDP o recupera la URL de tu punto final de metadatos. Consulte Configurar tu proveedor de identidades SAML externo.
2. Agregue un nuevo IdP a su grupo de usuarios. Cargue los metadatos de SAML o proporcione la URL de los metadatos. Consulte Añadir y administrar proveedores de identidad de SAML en un grupo de usuarios.
3. Asigne el IdP a los clientes de su aplicación. Consulte Clientes de aplicación de grupo de usuarios

Temas

- Información que debe saber sobre SAML IdPs en los grupos de usuarios de Amazon Cognito
- Distinción entre mayúsculas y minúsculas en los nombres de usuario SAML
- Añadir y administrar proveedores de identidad de SAML en un grupo de usuarios

- Inicio de sesión SAML en grupos de usuarios de Amazon Cognito
- Uso del inicio de sesión SAML iniciado por SPI
- Uso del inicio de sesión SAML iniciado por el IdP
- Flujo de cierre de sesión de SAML
- Firma y cifrado de SAML
- Nombres e identificadores de proveedores de identidad SAML
- Configurar tu proveedor de identidades SAML externo

Información que debe saber sobre SAML IdPs en los grupos de usuarios de Amazon Cognito

Amazon Cognito procesa las aserciones de SAML por usted

Los grupos de usuarios de Amazon Cognito admiten la federación SAML 2.0 con puntos de conexión POST-binding. De esta forma, se suprime la necesidad de que la aplicación recupere o analice las respuestas de aserciones SAML, ya que el grupo de usuarios recibe directamente la respuesta SAML del IdP a través de un agente de usuario. El grupo de usuarios actúa como proveedor de servicios (SP) en nombre de la aplicación. Amazon Cognito admite el inicio de sesión único (SSO) iniciado por SP e IDP, tal como se describe en las secciones 5.1.2 y 5.1.4 de la descripción técnica de SAML V2.0.

Proporcione un certificado de firma de IdP válido

El certificado de firma de los metadatos de su proveedor de SAML no debe caducar al configurar el IDP de SAML en su grupo de usuarios.

Los grupos de usuarios admiten varios certificados de firma

Cuando el IdP de SAML incluye más de un certificado de firma en los metadatos de SAML, al iniciar sesión, el grupo de usuarios determina que la afirmación de SAML es válida si coincide con algún certificado de los metadatos de SAML. Cada certificado de firma no debe tener más de 4096 caracteres.

Mantenga el parámetro de estado del relé

Amazon Cognito y el IdP SAML mantienen la información de la sesión con un parámetro `relayState`.

1. Amazon Cognito admite valores de `relayState` superiores a 80 bytes. Aunque en las especificaciones de SAML se establece que el valor de `relayState` “no debe superar

los 80 bytes de tamaño", la práctica actual del sector se desvía con frecuencia de este comportamiento. Como consecuencia, rechazar valores de `relayState` de más de 80 bytes interrumpirá muchas integraciones de proveedores SAML estándar.

2. El `relayState` token es una referencia opaca a la información estatal mantenida por Amazon Cognito. Amazon Cognito no garantiza el contenido del parámetro `relayState`. No analice el contenido de forma que la aplicación dependa del resultado. Para obtener más información, consulte la especificación de SAML 2.0.

Identifique el punto final de ACS

El proveedor de identidades SAML requiere que establezca un punto de conexión del consumidor de aserción. El IdP redirige a los usuarios a este punto de conexión con la aserción de SAML. Configure el siguiente punto de conexión en el dominio de su grupo de usuarios para enlace POST de SAML 2.0 en su proveedor de identidades SAML.

```
https://Your user pool domain/saml2/idpresponse
With an Amazon Cognito domain:
https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse
With a custom domain:
https://auth.example.com/saml2/idpresponse
```

Para obtener más información sobre los dominios del grupo de usuarios, consulte Configuración de un dominio del grupo de usuarios.

No se han reproducido afirmaciones

No puede repetir ni reproducir una aserción de SAML en el punto de conexión `saml2/idpresponse` de Amazon Cognito. Una aserción de SAML reproducida tiene un ID de aserción que duplica el ID de una respuesta de IdP anterior.

El ID del grupo de usuarios es el ID de la entidad SP

Debe proporcionar su IdP con su ID de grupo de usuarios en el proveedor de servicios (SP)`urn`, también denominado URI de audiencia o ID de entidad SP. El URI de destino del grupo de usuarios tiene el siguiente formato.

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

Encontrará el ID de su grupo de usuarios en la descripción general del grupo de usuarios de la consola de Amazon Cognito.

Mapee todos los atributos necesarios

Configure el IdP SAML para proporcionar valores para los atributos que establezca como necesarios en el grupo de usuarios. Por ejemplo, `email` es un atributo obligatorio y común para grupos de usuarios. Antes de que los usuarios puedan iniciar sesión, las aserciones del IdP SAML deben incluir una afirmación que asigne al `email` del atributo de grupo de usuarios. Para obtener más información acerca de la asignación de atributos, consulte Especificación de asignaciones de atributos del proveedor de identidad para su grupo de usuarios.

El formato de aserción tiene requisitos específicos

Su IdP de SAML debe incluir las siguientes afirmaciones en la afirmación de SAML.

1. `NameID`Una reclamación. Amazon Cognito asocia una afirmación de SAML con el usuario de destino mediante. `NameID` Si `NameID` cambia, Amazon Cognito considerará que la afirmación es para un usuario nuevo. El atributo que defina `NameID` en la configuración de su IdP debe tener un valor persistente.

```
<saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:persistent">
  carlos
</saml2:NameID>
```

2. Una reclamación `AudienceRestriction` con un valor `Audience` que establece el ID de la entidad SP del grupo de usuarios como el objetivo de la respuesta.

```
<saml:AudienceRestriction>
  <saml:Audience> urn:amazon:cognito:sp:us-east-1_EXAMPLE
</saml:AudienceRestriction>
```

3. En el caso del inicio de sesión único iniciado por un SP, un `Response` elemento con el `InResponseTo` valor del identificador de solicitud de SAML original.

```
<saml2p:Response Destination="https://mydomain.us-east-1.amazoncognito.com/
saml2/idpresponse" ID="id123" InResponseTo="_dd0a3436-bc64-4679-
a0c2-cb4454f04184" IssueInstant="Date-time stamp" Version="2.0"
 xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:xs="http://
www.w3.org/2001/XMLSchema">
```

> **Note**
>
> Las aserciones SAML iniciadas por el IdP no deben contener ningún valor. `InResponseTo`

4. Un `SubjectConfirmationData` elemento con un `Recipient` valor del `saml2/idpresponse` punto final del grupo de usuarios y, en el caso del SAML iniciado por SP, un `InResponseTo` valor que coincida con el ID de solicitud de SAML original.

```
<saml2:SubjectConfirmationData InResponseTo="_dd0a3436-bc64-4679-a0c2-
cb4454f04184" NotOnOrAfter="Date-time stamp" Recipient="https://mydomain.us-
east-1.amazoncognito.com/saml2/idpresponse"/>
```

Solicitudes de inicio de sesión iniciadas por SP

Cuando el Autorizar punto de conexión redirige a su usuario a la página de inicio de sesión de su IdP, Amazon Cognito incluye una solicitud SAML en un parámetro URL de la solicitud `HTTP GET`. Una solicitud de SAML contiene información sobre su grupo de usuarios, incluido su punto final ACS. Si lo desea, puede aplicar una firma criptográfica a estas solicitudes.

Firme las solicitudes y cifre las respuestas

Cada grupo de usuarios con un proveedor de SAML genera un key pair asimétrico y un certificado de firma para una firma digital que Amazon Cognito asigna a las solicitudes de SAML. Cada IDP de SAML externo que configure para admitir una respuesta SAML cifrada hace que Amazon Cognito genere un nuevo key pair y un certificado de cifrado para ese proveedor. Para ver y descargar los certificados con la clave pública, elija su IDP en la pestaña Experiencia de inicio de sesión de la consola de Amazon Cognito.

Para establecer la confianza con las solicitudes de SAML de su grupo de usuarios, proporcione a su IdP una copia del certificado de firma SAML 2.0 de su grupo de usuarios. Si no configura el IdP para que acepte solicitudes firmadas por su grupo de usuarios, su IdP podría ignorar las solicitudes de SAML firmadas.

1. Amazon Cognito aplica una firma digital a las solicitudes de SAML que el usuario pasa a su IdP. Su grupo de usuarios firma todas las solicitudes de cierre de sesión único (SLO) y puede configurar su grupo de usuarios para que firme las solicitudes de inicio de sesión único (SSO) para cualquier IDP externo de SAML. Al proporcionar una copia del certificado, el IdP puede comprobar la integridad de las solicitudes de SAML de los usuarios.

2. Su IdP de SAML puede cifrar las respuestas de SAML con el certificado de cifrado. Cuando configura un IdP con cifrado SAML, su IdP solo debe enviar respuestas cifradas.

Codifique caracteres no alfanuméricos

Amazon Cognito no acepta caracteres UTF-8 de 4 bytes, como # o, que su IdP pase como valor de atributo. Puede codificar el carácter en Base64 para enviarlo como texto y, después, descodificarlo en la aplicación.

En el siguiente ejemplo, no se aceptará la notificación de atributo:

```
<saml2:Attribute Name="Name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
  <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xsd:string">#</saml2:AttributeValue>
</saml2:Attribute>
```

Al contrario que en el ejemplo anterior, no se aceptará la notificación de atributo siguiente:

```
<saml2:Attribute Name="Name" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
  <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xsd:string">8J+YkA==</saml2:AttributeValue>
</saml2:Attribute>
```

El punto final de los metadatos debe tener una seguridad de capa de transporte válida

Si aparece `InvalidParameterException` al crear un IdP SAML con una URL de punto de enlace de metadatos HTTPS (por ejemplo, "Error al recuperar los metadatos del *<punto de enlace de metadatos>*)", asegúrese de que el punto de enlace de los metadatos tenga SSL correctamente configurado y de que haya un certificado SSL válido asociado. Para obtener más información sobre la validación de certificados, consulte ¿Qué es un certificado SSL/TLS? .

Los clientes de aplicaciones con SAML iniciado por el IdP solo pueden iniciar sesión con SAML

Cuando activas la compatibilidad con un IdP de SAML 2.0 que admite el inicio de sesión iniciado por el IdP en un cliente de aplicaciones, solo puedes añadir otro SAML IdPs 2.0 a ese cliente de aplicación. No puedes añadir el directorio de usuarios del grupo de usuarios ni todos los proveedores de identidad externos que no sean de SAML a un cliente de aplicaciones configurado de esta manera.

Las respuestas de cierre de sesión deben utilizar el enlace POST

El `/saml2/logout` punto final acepta `LogoutResponse` como `HTTP POST` solicitudes. Los grupos de usuarios no aceptan respuestas de cierre de sesión con `HTTP GET` carácter vinculante.

Distinción entre mayúsculas y minúsculas en los nombres de usuario SAML

Cuando un usuario federado intenta iniciar sesión, el proveedor de identidad (IdP) de SAML pasa un mensaje exclusivo `NameId` a Amazon Cognito en la afirmación de SAML del usuario. Amazon Cognito identifica a un usuario federado de SAML por su reclamación `NameId`. Independientemente de la configuración de distinción entre mayúsculas y minúsculas de su grupo de usuarios, Amazon Cognito reconoce a un usuario federado recurrente de un IDP de SAML cuando aprueba su reclamación única y distingue entre mayúsculas y minúsculas. `NameId` Si se asigna un atributo como `email` a `NameId`, y el usuario cambia la dirección de correo electrónico, no podrá iniciar sesión en la aplicación.

Asigne `NameId` en las aserciones SAML de un atributo IdP con valores que no cambian.

Por ejemplo, Carlos tiene un perfil de usuario en el grupo de usuarios que distingue mayúsculas de minúsculas de una aserción SAML de Active Directory Federation Services (ADFS) que ha pasado un valor `NameId` de `Carlos@example.com`. La siguiente vez que Carlos intente iniciar sesión, su IdP de ADFS pasa un valor `NameId` de `carlos@example.com`. Dado que `NameId` debe coincidir exactamente en mayúsculas y minúsculas, el inicio de sesión no se produce con éxito.

Si los usuarios no pueden iniciar sesión después de que cambie su `NameID`, elimine sus perfiles de usuario del grupo de usuarios. Amazon Cognito creará nuevos perfiles de usuario la siguiente vez que se inicie sesión.

Temas

- Añadir y administrar proveedores de identidad de SAML en un grupo de usuarios
- Inicio de sesión SAML en grupos de usuarios de Amazon Cognito
- Uso del inicio de sesión SAML iniciado por SPI
- Uso del inicio de sesión SAML iniciado por el IdP
- Flujo de cierre de sesión de SAML
- Firma y cifrado de SAML
- Nombres e identificadores de proveedores de identidad SAML

- Configurar tu proveedor de identidades SAML externo

Añadir y administrar proveedores de identidad de SAML en un grupo de usuarios

Los siguientes procedimientos muestran cómo crear, modificar y eliminar proveedores de SAML en un grupo de usuarios de Amazon Cognito.

AWS Management Console

Puede usarlo AWS Management Console para crear y eliminar proveedores de identidad de SAML (). IdPs

Antes de crear un IdP de SAML, debe tener el documento de metadatos de SAML que obtiene del IdP de terceros. Para obtener instrucciones sobre cómo obtener o generar el documento de metadatos de SAML necesario, consulte Configurar tu proveedor de identidades SAML externo.

Para configurar un IdP SAML 2.0 en su grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS .
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). Localice Federated sign-in (Inicio de sesión federado) y elija Add an identity provider (Añadir un proveedor de identidad).
5. Elija un IdP SAML.
6. Introduzca un nombre de proveedor. Puede pasar este nombre descriptivo en un parámetro de `identity_provider` solicitud alAutorizar punto de conexión.
7. Introduzca Identificadores separados por comas. Un identificador indica a Amazon Cognito que debe comprobar la dirección de correo electrónico que introduce un usuario al iniciar sesión y, a continuación, dirigirlo al proveedor que corresponda a su dominio.
8. Elija Add sign-out flow (Añadir flujo de cierre de sesión) si desea que Amazon Cognito envíe solicitudes de cierre de sesión firmadas a su proveedor cuando un usuario cierra la sesión. Debe configurar el IdP SAML 2.0 para enviar respuestas de cierre de sesión al punto de conexión de `https://mydomain.us-east-1.amazoncognito.com/saml2/logout` que se crea al configurar la IU alojada. El punto de conexión `saml2/logout` utiliza el enlace POST.

> **Note**
>
> Si selecciona esta opción y su IdP de SAML espera una solicitud de cierre de sesión firmada, también debe proporcionar a su IdP de SAML el certificado de firma de su grupo de usuarios.
> El proveedor de identidades (IdP) SAML procesará la solicitud de cierre de sesión firmada y cerrará la sesión de Amazon Cognito del usuario.

9. Elija la configuración de inicio de sesión SAML iniciada por el IdP. Como práctica recomendada de seguridad, elige Aceptar únicamente las aserciones SAML iniciadas por el SP. Si ha preparado su entorno para aceptar de forma segura las sesiones de inicio de sesión de SAML no solicitadas, elija Aceptar aserciones de SAML iniciadas por SP e iniciadas por IdP. Para obtener más información, consulte Inicio de sesión SAML en grupos de usuarios de Amazon Cognito.
10. Seleccione un Origen de documentos de metadatos. Si su IdP ofrece metadatos SAML en una URL pública, puede elegir Metadata document URL (URL del documento de metadatos) e introducir esa URL pública. En caso contrario, elija Upload metadata document (Cargar documento de metadatos) y seleccione un archivo de metadatos que haya descargado anteriormente de su proveedor.

> **Note**
>
> Te recomendamos que introduzcas la URL de un documento de metadatos si tu proveedor tiene un terminal público en lugar de subir un archivo. Amazon Cognito actualiza automáticamente los metadatos desde la URL de los metadatos. Normalmente, los metadatos se actualizan cada seis horas o antes de que caduquen, lo que ocurra primero.

11. Asigne los atributos entre su proveedor de SAML y su grupo de usuarios para asignar los atributos del proveedor de SAML al perfil de usuario de su grupo de usuarios. Incluya los atributos requeridos del grupo de usuarios en la asignación de atributos.

    Por ejemplo, cuando elige User pool attribute (Atributo grupo de usuarios) `email`, escriba el nombre de atributo SAML tal como aparece en la aserción SAML del IdP. Si su IdP SAML ofrece aserciones SAML de ejemplo, estas podrían servirle para encontrar el nombre. Algunos IdPs usan nombres simples, como`email`, mientras que otros usan nombres como los siguientes.

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

12. Seleccione Crear.

API/CLI

Utilice los siguientes comandos para crear y administrar un proveedor de identidades (IdP) SAML.

Para crear un IdP y cargar un documento de metadatos

- AWS CLI: `aws cognito-idp create-identity-provider`

  Ejemplo con archivo de metadatos: `aws cognito-idp create-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1 --provider-type SAML --provider-details file:///details.json --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

  Donde `details.json` contiene:

```
"ProviderDetails": {
      "MetadataFile": "<SAML metadata XML>",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

> **Note**
>
> Si <SAML metadata XML>contiene alguna instancia del personaje", debes agregar \ como personaje de escape:\".

  Ejemplo con URL de metadatos: `aws cognito-idp create-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1 --provider-type SAML --provider-details MetadataURL=https://`

myidp.example.com/sso/saml/metadata `--attribute-mapping` *email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress*

- AWS API: CreateIdentityProvider

Para cargar un nuevo documento de metadatos para un proveedor de identidades (IdP)

- AWS CLI: `aws cognito-idp update-identity-provider`

  Ejemplo con archivo de metadatos: `aws cognito-idp update-identity-provider --user-pool-id` *us-east-1_EXAMPLE* `--provider-name=`*SAML_provider_1* `--provider-details file:///details.json --attribute-mapping` *email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress*

  Donde `details.json` contiene:

```
"ProviderDetails": {
      "MetadataFile": "<SAML metadata XML>",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

> **Note**
>
> Si <SAML metadata XML>contiene alguna instancia del personaje", debes agregar \ como personaje de escape:\".

  Ejemplo con URL de metadatos: `aws cognito-idp update-identity-provider --user-pool-id` *us-east-1_EXAMPLE* `--provider-name=`*SAML_provider_1* `--provider-details MetadataURL=`*https://myidp.example.com/sso/saml/metadata* `--attribute-mapping` *email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress*

- AWS API: UpdateIdentityProvider

Para obtener información acerca de un IdP específico

- AWS CLI: `aws cognito-idp describe-identity-provider`

  ```
  aws cognito-idp describe-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1
  ```

- AWS API: DescribeIdentityProvider

Para enumerar información sobre todos IdPs

- AWS CLI: `aws cognito-idp list-identity-providers`

  Ejemplo: `aws cognito-idp list-identity-providers --user-pool-id us-east-1_EXAMPLE --max-results 3`

- AWS API: ListIdentityProviders

Para eliminar un proveedor de identidad

- AWS CLI: `aws cognito-idp delete-identity-provider`

  ```
  aws cognito-idp delete-identity-provider --user-pool-id us-east-1_EXAMPLE --provider-name=SAML_provider_1
  ```

- AWS API: DeleteIdentityProvider

Para configurar el proveedor de identidad SAML para añadir un grupo de usuarios como una relación de confianza

- El URN del proveedor del servicio de grupos de usuarios es: `urn:amazon:cognito:sp:us-east-1_EXAMPLE`. Amazon Cognito requiere un valor de restricción de audiencia que coincida con este URN en la respuesta de SAML. Configure su IdP para que utilice el siguiente punto final de enlace POST para el mensaje de respuesta de IdP a SP.

  ```
  https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse
  ```

- Su IdP de SAML debe `NameID` completar la aserción de SAML y todos los atributos necesarios para su grupo de usuarios. `NameID`se utiliza para identificar de forma exclusiva al usuario federado de SAML en el grupo de usuarios. Su IdP debe pasar el ID de nombre SAML de cada usuario en

un formato coherente y que distinga entre mayúsculas y minúsculas. Cualquier variación en el valor del ID de nombre de un usuario crea un nuevo perfil de usuario.

Para proporcionar un certificado de firma al IDP de SAML 2.0

- Para descargar una copia de la clave pública de Amazon Cognito que su IdP puede utilizar para validar las solicitudes de cierre de sesión de SAML, seleccione la pestaña Experiencia de inicio de sesión de su grupo de usuarios, seleccione su IdP y, en Ver certificado de firma, seleccione Descargar como .crt.

Puede eliminar cualquier proveedor SAML que haya configurado en su grupo de usuarios con la consola de Amazon Cognito.

Cómo eliminar un proveedor SAML

1. Inicie sesión en la consola de Amazon Cognito.
2. En el panel de navegación, elija User Pools (Grupos de usuarios), y elija el grupo de usuarios que desea editar.
3. Seleccione la pestaña Experiencia de inicio de sesión y busque el inicio de sesión del proveedor de identidad federado.
4. Selecciona el botón de radio situado junto al SAML IdPs que deseas eliminar.
5. Cuando se le pida Delete identity provider (Eliminar proveedor de identidad), ingrese el nombre del proveedor SAML para confirmar su eliminación y, a continuación, elija Delete (Eliminar).

Inicio de sesión SAML en grupos de usuarios de Amazon Cognito

Amazon Cognito admite el inicio de sesión único (SSO) iniciado por el proveedor de servicios (iniciado por el SP) y el SSO iniciado por el IdP. Como práctica recomendada de seguridad, implemente el SSO iniciado por el SP en su grupo de usuarios. En la sección 5.1.2 de SAML V2.0 Technical Overview (Información técnica general de SAML V2.0), se explica el inicio de sesión único iniciado por el proveedor de servicios. Amazon Cognito es el proveedor de identidad (IdP) para la aplicación. La aplicación es el proveedor de servicios (SP) que recupera tokens para usuarios autenticados. No obstante, cuando utiliza un IdP externo para autenticar usuarios, Amazon Cognito es el SP. Cuando los usuarios de SAML 2.0 se autentican con un flujo iniciado por un SP, siempre deben realizar primero una solicitud a Amazon Cognito y redirigirlos al IdP para su autenticación.

En algunos casos de uso empresariales, el acceso a las aplicaciones internas comienza en un marcador de un panel alojado por el IdP de la empresa. Cuando un usuario selecciona un marcador, el IdP genera una respuesta SAML y la envía al SP para autenticar al usuario con la aplicación.

Puede configurar un IdP de SAML en su grupo de usuarios para que admita el SSO iniciado por el IdP. Cuando admite la autenticación iniciada por un IdP, Amazon Cognito no puede comprobar si ha solicitado la respuesta de SAML que recibe porque Amazon Cognito no inicia la autenticación con una solicitud de SAML. En el SSO iniciado por SP, Amazon Cognito establece parámetros de estado que validan una respuesta de SAML con respecto a la solicitud original. Con el inicio de sesión iniciado por SP, también puede protegerse contra la falsificación de solicitudes entre sitios (CSRF).

Para ver un ejemplo de cómo crear un SAML iniciado por SP en un entorno en el que no desee que los usuarios interactúen con la interfaz de usuario alojada en el grupo de usuarios, consulte. Escenario de ejemplo: marcar aplicaciones de Amazon Cognito en un panel empresarial

Temas

Escenario de ejemplo: marcar aplicaciones de Amazon Cognito en un panel empresarial

Puede crear marcadores en sus paneles de IdP de SAML u OIDC que proporcionen a los grupos de usuarios de Amazon Cognito acceso SSO a las aplicaciones web. Puede enlazar con Amazon Cognito de una forma que no requiera que los usuarios inicien sesión con la IU alojada. Para ello, añada un marcador de inicio de sesión a su portal que redirija al grupo de usuarios Autorizar punto de conexión de Amazon Cognito en el siguiente formato.

```
https://mydomain.us-east-1.amazoncognito.com/authorize?
response_type=code&identity_provider=MySAMLIdP&client_id=1example23456789&redire
www.example.com
```

> **Note**
>
> También puede utilizar un parámetro `idp_identifier` en lugar de un parámetro `identity_provider` en su solicitud al punto de conexión de autorización. Un identificador de IdP es un nombre alternativo o un dominio de correo electrónico que puede configurar al crear un proveedor de identidades en su grupo de usuarios. Consulte Nombres e identificadores de proveedores de identidad SAML.

Cuando utiliza los parámetros apropiados en su solicitud a `/authorize`, Amazon Cognito inicia silenciosamente el flujo de inicio de sesión iniciado por SP y redirige al usuario para que inicie sesión con su IdP.

Para empezar, añade un IdP de SAML a tu grupo de usuarios. Cree un cliente de aplicación que utilice su IdP SAML para iniciar sesión y que tenga la URL de su aplicación como URL de devolución de llamada autorizada. Para obtener más información acerca de los clientes de aplicación, consulte Clientes de aplicación de grupo de usuarios.

Antes de implementar este acceso autenticado en su portal, pruebe el inicio de sesión iniciado por SP en su aplicación desde la interfaz de usuario alojada. Para obtener más información acerca de cómo configurar un IdP SAML en Amazon Cognito, consulte Configurar tu proveedor de identidades SAML externo.

El siguiente diagrama muestra un flujo de autenticación que emula el inicio de sesión único iniciado por IdP. Los usuarios pueden autenticarse con Amazon Cognito desde un enlace del portal de su empresa.

Una vez que cumplas con los requisitos, crea un marcador para ti Autorizar punto de conexión que incluya uno o varios parámetros. `identity_provider` `idp_identifier` La autenticación del usuario se realiza de la siguiente manera.

1. El usuario inicia sesión en el panel de IdP de SSO. Las aplicaciones empresariales para las que el usuario tiene autorización de acceso rellenan este panel de control.
2. El usuario elige el enlace a la aplicación que autentica con Amazon Cognito. En muchos portales SSO puede agregar un enlace de aplicación personalizado. Funcionará cualquier función que se pueda utilizar para crear un enlace a una URL pública en el portal SSO.
3. El enlace de aplicación personalizado en el portal SSO dirige al usuario al grupo de usuarios Autorizar punto de conexión. El enlace incluye parámetros para `response_type`, `client_id`, `redirect_uri` y `identity_provider`. El parámetro `identity_provider` es el nombre que asignó al IdP en el grupo de usuarios. También puede utilizar un parámetro `idp_identifier` en lugar del parámetro `identity_provider`. Un usuario accede al punto final de la federación desde un enlace que contiene un `identity_provider` parámetro `idp_identifier` o. Este usuario omite la página de inicio de sesión y navega directamente para autenticarse con su IdP. Para obtener más información sobre cómo asignar nombres a SAML IdPs, consulte. Nombres e identificadores de proveedores de identidad SAML

   URL de ejemplo

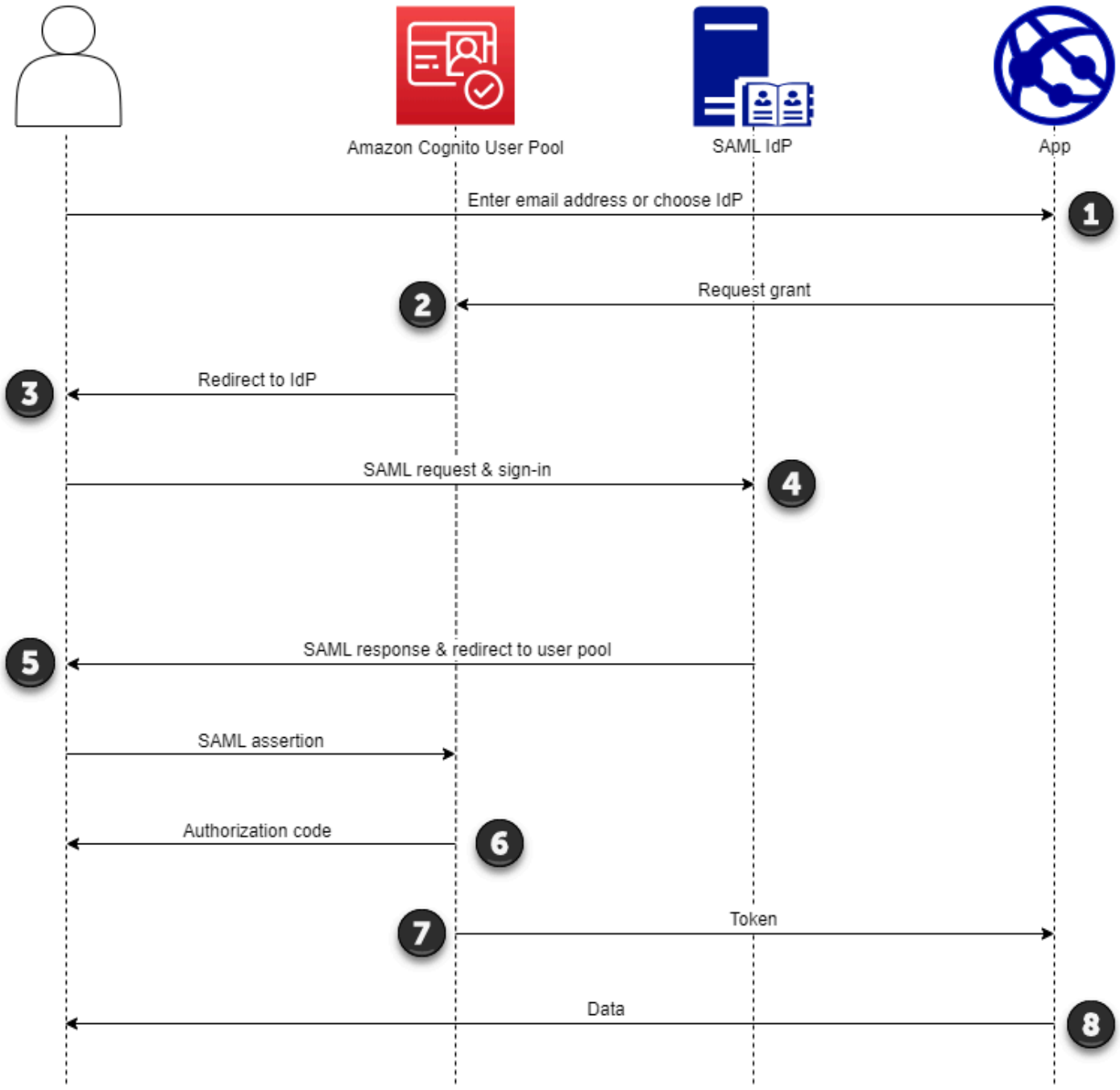
```
https://mydomain.us-east-1.amazoncognito.com/authorize?
response_type=code&
identity_provider=MySAMLIdP&
client_id=1example23456789&
redirect_uri=https://www.example.com
```

4. Amazon Cognito redirige la sesión de usuario a su IdP con una solicitud SAML.
5. Es posible que el usuario haya recibido una cookie de sesión de su IdP cuando inició sesión en el panel de control. Su IdP utiliza esta cookie para validar al usuario silenciosamente y redirigirlo al punto de conexión `idpresponse` de Amazon Cognito con una respuesta SAML. Si no hay ninguna sesión activa, el IdP vuelve a autenticar al usuario antes de publicar la respuesta SAML.
6. Amazon Cognito valida la respuesta SAML y crea o actualiza el perfil de usuario en función de la afirmación SAML.
7. Amazon Cognito redirige al usuario a su aplicación interna con un código de autorización. Ha configurado la URL interna de la aplicación como URL de redirección autorizada para su cliente de aplicación.
8. La aplicación intercambia el código de autorización de los tokens de Amazon Cognito. Para obtener más información, consulte Punto de conexión de token.

Uso del inicio de sesión SAML iniciado por SPI

Como práctica recomendada, implemente el inicio de sesión service-provider-initiated (iniciado por SP) en su grupo de usuarios. Amazon Cognito inicia la sesión del usuario y lo redirige a su IdP. Con este método, tiene el mayor control sobre quién presenta las solicitudes de inicio de sesión. También puedes permitir el inicio de sesión iniciado por un IdP en determinadas condiciones. Para obtener más información, consulte Inicio de sesión SAML en grupos de usuarios de Amazon Cognito.

El siguiente proceso muestra cómo los usuarios inician sesión en tu grupo de usuarios a través de un proveedor de SAML.



1. El usuario introduce su dirección de correo electrónico en una página de inicio de sesión. Para determinar la redirección del usuario a su IdP, puedes recopilar su dirección de correo electrónico en una aplicación personalizada o invocar la interfaz de usuario alojada en la vista web. Puedes configurar la interfaz de usuario alojada para que muestre una lista IdPs o solo pida una dirección de correo electrónico.

2. La aplicación invoca el punto final de redireccionamiento del grupo de usuarios y solicita una sesión con el ID de cliente que corresponde a la aplicación y el ID de IdP que corresponde al usuario.
3. Amazon Cognito redirige al usuario al IdP con una solicitud de SAML, firmada opcionalmente, en un elemento. `AuthnRequest`
4. El IdP autentica al usuario de forma interactiva o con una sesión recordada en una cookie del navegador.
5. El IdP redirige al usuario al punto final de respuesta SAML de su grupo de usuarios con la afirmación SAML cifrada opcionalmente en su carga útil POST.

   > **Note**
   >
   > Amazon Cognito cancela las sesiones que no reciben respuesta en un plazo de 5 minutos y redirige al usuario a la interfaz de usuario alojada. Cuando su usuario experimente este resultado, recibirá un `Something went wrong` mensaje de error.

6. Tras comprobar la afirmación de SAML y mapear los atributos de usuario de las afirmaciones de la respuesta, Amazon Cognito crea o actualiza internamente el perfil del usuario en el grupo de usuarios. Por lo general, su grupo de usuarios devuelve un código de autorización a la sesión del navegador del usuario.
7. El usuario presenta su código de autorización en la aplicación, que lo intercambia por tokens web JSON (JWT).
8. Tu aplicación acepta y procesa tu token de identificación de usuario como autenticación, genera solicitudes autorizadas a los recursos con su token de acceso y almacena su token de actualización.

Cuando un usuario se autentica y recibe una concesión de código de autorización, el grupo de usuarios devuelve los identificadores de identificación, acceso y actualización. El token de ID es un objeto de autenticación para la administración de identidades basada en el OIDC. El token de acceso es un objeto de autorización con alcances de OAuth 2.0. El token de actualización es un objeto que genera nuevos identificadores y identificadores de acceso cuando los tokens actuales de tu usuario han caducado. Puede configurar la duración de los tokens de los usuarios en el cliente de la aplicación de su grupo de usuarios.

También puede elegir la duración de los tokens de actualización. Cuando el token de actualización de un usuario caduque, este debe volver a iniciar sesión. Si se autenticaron a través de un IdP

de SAML, la duración de la sesión de sus usuarios se establece en función de la caducidad de sus tokens, no de la caducidad de la sesión con su IdP. Tu aplicación debe almacenar el token de actualización de cada usuario y renovar su sesión cuando caduque. La interfaz de usuario alojada mantiene las sesiones de los usuarios en una cookie del navegador que es válida durante 1 hora.

Uso del inicio de sesión SAML iniciado por el IdP

Al configurar su proveedor de identidad para el inicio de sesión en SAML 2.0 iniciado por el IdP, puede presentar las aserciones de SAML en el `saml2/idpresponse` punto final del dominio de su grupo de usuarios sin necesidad de iniciar la sesión en el. Autorizar punto de conexión Un grupo de usuarios con esta configuración acepta aserciones SAML iniciadas por el IdP de un proveedor de identidad externo del grupo de usuarios que admite el cliente de la aplicación solicitada. Los siguientes pasos describen el proceso general para configurar e iniciar sesión con un proveedor de SAML 2.0 iniciado por un IdP.

1. Cree o designe un grupo de usuarios y un cliente de aplicaciones.
2. Cree un IdP SAML 2.0 en su grupo de usuarios.
3. Configure su IdP para que admita el inicio del IdP. El SAML iniciado por IdP introduce consideraciones de seguridad a las que no están sujetos otros proveedores de SSO. Por este motivo, no puedes añadir aplicaciones que no sean SAML IdPs, incluido el propio grupo de usuarios, a ningún cliente de aplicaciones que utilice un proveedor de SAML con un inicio de sesión iniciado por el IdP.
4. Asocia tu proveedor de SAML iniciado por el IdP a un cliente de aplicaciones de tu grupo de usuarios.
5. Dirija al usuario a la página de inicio de sesión de su IdP de SAML y recupere una afirmación de SAML.
6. Dirija al usuario al `saml2/idpresponse` punto final de su grupo de usuarios con su afirmación de SAML.
7. Reciba tokens web JSON (JWT).

Para aceptar afirmaciones de SAML no solicitadas en tu grupo de usuarios, debes tener en cuenta su efecto en la seguridad de tu aplicación. Es probable que se produzcan intentos de suplantación de solicitudes y CSRF cuando aceptas solicitudes iniciadas por un IdP. Aunque su grupo de usuarios no puede verificar una sesión de inicio de sesión iniciada por un IdP, Amazon Cognito valida los parámetros de solicitud y las afirmaciones de SAML.

Además, su afirmación de SAML no debe contener ninguna `InResponseTo` reclamación y debe haberse emitido en los 6 minutos anteriores.

Debe enviar las solicitudes con SAML iniciado por el IdP a su. `/saml2/idpresponse` En el caso de las solicitudes de autorización de la interfaz de usuario alojada o iniciadas por el SP, debes proporcionar parámetros que identifiquen el cliente de la aplicación solicitada, los ámbitos, el URI de redireccionamiento y otros detalles como parámetros de cadena de consulta en las solicitudes. `HTTP GET` Sin embargo, en el caso de las aserciones de SAML iniciadas por el IdP, los detalles de la solicitud deben formatearse como un `RelayState` parámetro en el cuerpo de la solicitud. `HTTP POST` El cuerpo de la solicitud también debe contener la aserción de SAML como parámetro. `SAMLResponse`

El siguiente es un ejemplo de solicitud para un proveedor de SAML iniciado por un IdP.

```
POST /saml2/idpresponse HTTP/1.1
User-Agent: USER_AGENT
Accept: */*
Host: example.auth.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded

SAMLResponse=[Base64-encoded SAML assertion]&RelayState=identity_provider
%3DMySAMLIdP%26client_id%3D1example23456789%26redirect_uri%3Dhttps%3A%2F
%2Fwww.example.com%26response_type%3Dcode%26scope%3Demail%2Bopenid%2Bphone

HTTP/1.1 302 Found
Date: Wed, 06 Dec 2023 00:15:29 GMT
Content-Length: 0
x-amz-cognito-request-id: 8aba6eb5-fb54-4bc6-9368-c3878434f0fb
Location: https://www.example.com?code=[Authorization code]
```

AWS Management Console

Para configurar un IdP para el SAML iniciado por el IdP

1. Cree un grupo de usuarios, un cliente de aplicaciones y un proveedor de identidades de SAML.
2. Desvincule todos los proveedores de identidad social y de OIDC de su cliente de aplicaciones, si hay alguno asociado.
3. Ve a la pestaña Experiencia de inicio de sesión de tu grupo de usuarios.

4. En Iniciar sesión con un proveedor de identidad federado, edita o agrega un proveedor de SAML.
5. En Inicio de sesión SAML iniciado por el IdP, selecciona Aceptar aserciones SAML iniciadas por el SP e iniciadas por el IdP.
6. Elija Guardar cambios.

API/CLI

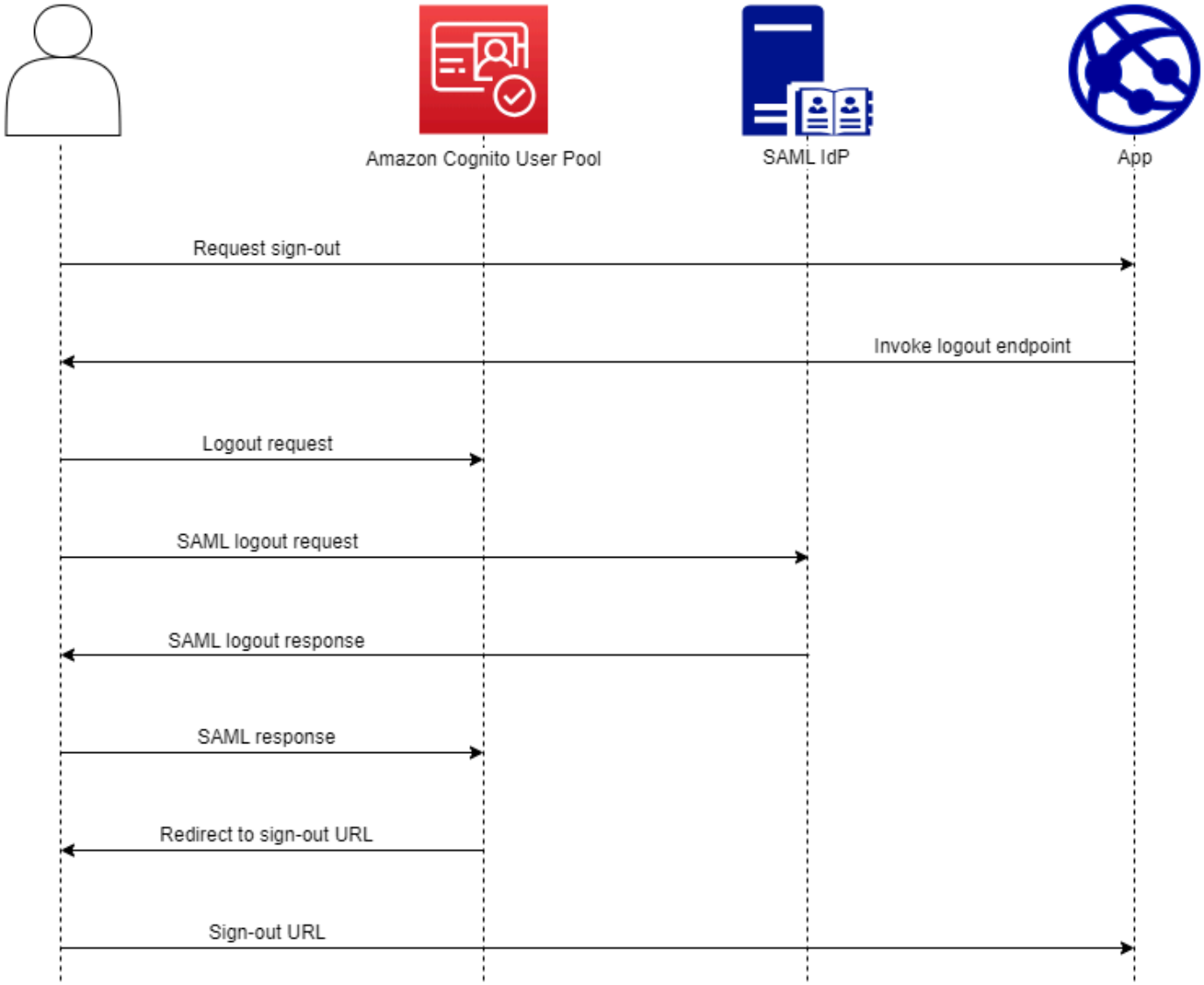
Para configurar un IdP para el SAML iniciado por el IdP

Configure el SAML iniciado por el IdP con el `IDPInit` parámetro de una solicitud de API CreateIdentityProvider. UpdateIdentityProvider El siguiente es un ejemplo `ProviderDetails` de un IdP que admite el SAML iniciado por el IdP.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Flujo de cierre de sesión de SAML

Amazon Cognito admite el cierre de sesión único con SAML 2.0. Cuando configura su IdP de SAML para que admita el flujo de cierre de sesión, Amazon Cognito redirige al usuario con una solicitud de cierre de sesión de SAML firmada a su IdP. Amazon Cognito determina la ubicación de redireccionamiento a partir de la `SingleLogoutService` URL de los metadatos del IdP. Amazon Cognito firma la solicitud de cierre de sesión con el certificado de firma del grupo de usuarios.



Cuando dirige a un usuario con una sesión de SAML al `/logout` punto de enlace de su grupo de usuarios, Amazon Cognito redirige a su usuario de SAML con la siguiente solicitud al punto de enlace de SLO que se especifica en los metadatos del IdP.

```
https://[SingleLogoutService endpoint]?
SAMLRequest=[encoded SAML request]&
RelayState=[RelayState]&
SigAlg=http://www.w3.org/2001/04/xmldsig-more#rsa-sha256&
Signature=[User pool RSA signature]
```

A continuación, el usuario regresa a su `saml2/logout` punto final con un mensaje `LogoutResponse` de su IdP. Su IdP debe enviar `LogoutResponse` la solicitud`HTTP POST`. A continuación, Amazon Cognito los redirige al destino de redireccionamiento desde su solicitud de cierre de sesión inicial.

Es posible que su proveedor de SAML envíe un mensaje `LogoutResponse` con más de uno. `AuthnStatement` La primera `sessionIndex` `AuthnStatement` de una respuesta de este tipo debe coincidir con la de la respuesta `sessionIndex` de SAML que autenticó originalmente al usuario. Si `sessionIndex` está en alguna otra`AuthnStatement`, Amazon Cognito no reconocerá la sesión y no se cerrará la sesión del usuario.

AWS Management Console

Para configurar el cierre de sesión de SAML

1. Cree un grupo de usuarios, un cliente de aplicaciones y un IdP de SAML.
2. Al crear o editar tu proveedor de identidad SAML, en Información del proveedor de identidad, marca la casilla con el título Añadir flujo de cierre de sesión.
3. En la pestaña Experiencia de inicio de sesión de su grupo de usuarios, en Inicio de sesión con un proveedor de identidad federado, elija su IDP y busque el certificado de firma.
4. Selecciona Descargar como .crt.
5. Configure su proveedor de SAML para que admita el cierre de sesión único y la firma de solicitudes de SAML, y cargue el certificado de firma del grupo de usuarios. Su IdP debe redirigirse al dominio `/saml2/logout` de su grupo de usuarios.

API/CLI

Para configurar el cierre de sesión con SAML

Configure el cierre de sesión único con el `IDPSignout` parámetro de una solicitud de API CreateIdentityProvidero UpdateIdentityProviderAPI. El siguiente es un ejemplo `ProviderDetails` de un IdP que admite el cierre de sesión único de SAML.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",,
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
```

```
}
```

Firma y cifrado de SAML

Amazon Cognito admite las solicitudes de SAML firmadas y las respuestas de SAML cifradas para iniciar y cerrar sesión. Todas las operaciones criptográficas realizadas durante las operaciones SAML del grupo de usuarios deben generar firmas y texto cifrado con las claves que genera user-pool-provided Amazon Cognito. Actualmente, no puede configurar un grupo de usuarios para que firme solicitudes o acepte afirmaciones cifradas con una clave externa.

> **Note**
>
> Los certificados de su grupo de usuarios tienen una validez de 10 años. Una vez al año, Amazon Cognito genera nuevos certificados de firma y cifrado para su grupo de usuarios. Amazon Cognito devuelve el certificado más reciente cuando solicita el certificado de firma y firma las solicitudes con el certificado de firma más reciente. Su IdP puede cifrar las afirmaciones de SAML con cualquier certificado de cifrado de grupo de usuarios que no esté caducado. Sus certificados anteriores seguirán siendo válidos durante toda su vigencia. Como práctica recomendada, actualice el certificado en la configuración de su proveedor una vez al año.

Temas

- Aceptar respuestas SAML cifradas de su IdP
- Firmar solicitudes de SAML

Aceptar respuestas SAML cifradas de su IdP

Amazon Cognito y su IdP pueden establecer la confidencialidad en las respuestas de SAML cuando los usuarios inician y cierran sesión. Amazon Cognito asigna un key pair de claves RSA público-privadas y un certificado a cada proveedor de SAML externo que configure en su grupo de usuarios. Al habilitar el cifrado de respuestas para el proveedor de SAML de su grupo de usuarios, debe cargar su certificado en un IdP que admita las respuestas SAML cifradas. La conexión del grupo de usuarios con el IdP de SAML no funciona antes de que el IdP comience a cifrar todas las afirmaciones de SAML con la clave proporcionada.

A continuación, se ofrece un resumen del flujo de un inicio de sesión cifrado con SAML.

1. El usuario comienza a iniciar sesión y elige su IDP de SAML.
2. El grupo de usuarios Autorizar punto de conexión redirige al usuario a su IDP de SAML con una solicitud de inicio de sesión de SAML. Si lo desea, su grupo de usuarios puede acompañar esta solicitud con una firma que permita la verificación de integridad por parte del IdP. Cuando desee firmar solicitudes de SAML, debe configurar su IdP para que acepte las solicitudes que su grupo de usuarios haya firmado con la clave pública del certificado de firma.
3. El IdP de SAML inicia sesión en el usuario y genera una respuesta de SAML. El IdP cifra la respuesta con la clave pública y redirige al usuario al punto final del grupo de usuarios. `/saml2/idpresponse` El IdP debe cifrar la respuesta según se define en la especificación SAML 2.0. Para obtener más información, consulte `Element <EncryptedAssertion>` Afirmaciones y protocolos del lenguaje de marcado de aserciones de seguridad (SAML) V2.0 de OASIS.
4. Su grupo de usuarios descifra el texto cifrado de la respuesta SAML con la clave privada e inicia sesión como usuario.

> **Important**
>
> Cuando habilitas el cifrado de respuestas para un IdP de SAML en tu grupo de usuarios, tu IdP debe cifrar todas las respuestas con una clave pública específica del proveedor. Amazon Cognito no acepta respuestas de SAML sin cifrar de un IDP externo de SAML que usted configure para admitir el cifrado.

Cualquier IdP SAML externo de su grupo de usuarios puede admitir el cifrado de respuesta y cada IdP recibe su propio par de claves.

AWS Management Console

Para configurar el cifrado de respuestas de SAML

1. Cree un grupo de usuarios, un cliente de aplicaciones y un IdP de SAML.
2. Al crear o editar tu proveedor de identidad de SAML, en Firmar solicitudes y cifrar respuestas, marca la casilla que lleva el título Exigir afirmaciones de SAML cifradas a este proveedor.
3. En la pestaña Experiencia de inicio de sesión de su grupo de usuarios, en Inicio de sesión con un proveedor de identidad federado, seleccione su IdP de SAML y elija Ver certificado de cifrado.

4. Selecciona Descargar como .crt y envía el archivo descargado a tu IdP de SAML. Configure su IdP de SAML para cifrar las respuestas de SAML con la clave del certificado.

API/CLI

Para configurar el cifrado de respuestas de SAML

Configure el cifrado de respuesta con el `EncryptedResponses` parámetro de una solicitud CreateIdentityProviderо una solicitud de UpdateIdentityProviderAPI. El siguiente es un ejemplo `ProviderDetails` de un IdP que admite la firma de solicitudes.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Firmar solicitudes de SAML

La capacidad de demostrar la integridad de las solicitudes de SAML 2.0 a su IDP es una ventaja de seguridad del inicio de sesión SAML iniciado por Amazon Cognito SP. Cada grupo de usuarios con un dominio recibe un certificado de firma X.509 del grupo de usuarios. Con la clave pública de este certificado, los grupos de usuarios aplican una firma criptográfica a las solicitudes de cierre de sesión que el grupo de usuarios genera cuando los usuarios seleccionan un IdP de SAML. Si lo desea, puede configurar el cliente de la aplicación para que firme las solicitudes de inicio de sesión con SAML. Al firmar las solicitudes de SAML, el IdP puede comprobar que la firma de los metadatos XML de las solicitudes coincide con la clave pública del certificado del grupo de usuarios que usted proporciona.

AWS Management Console

Para configurar la firma de solicitudes de SAML

1. Cree un grupo de usuarios, un cliente de aplicaciones y un IdP de SAML.
2. Al crear o editar tu proveedor de identidades de SAML, en Firmar solicitudes y cifrar respuestas, marca la casilla que lleva el título Firmar las solicitudes de SAML a este proveedor.

3. En la pestaña Experiencia de inicio de sesión de tu grupo de usuarios, en Inicio de sesión con un proveedor de identidad federado, selecciona Ver certificado de firma.
4. Selecciona Descargar como .crt y envía el archivo descargado a tu IdP de SAML. Configura tu IdP de SAML para verificar la firma de las solicitudes de SAML entrantes.

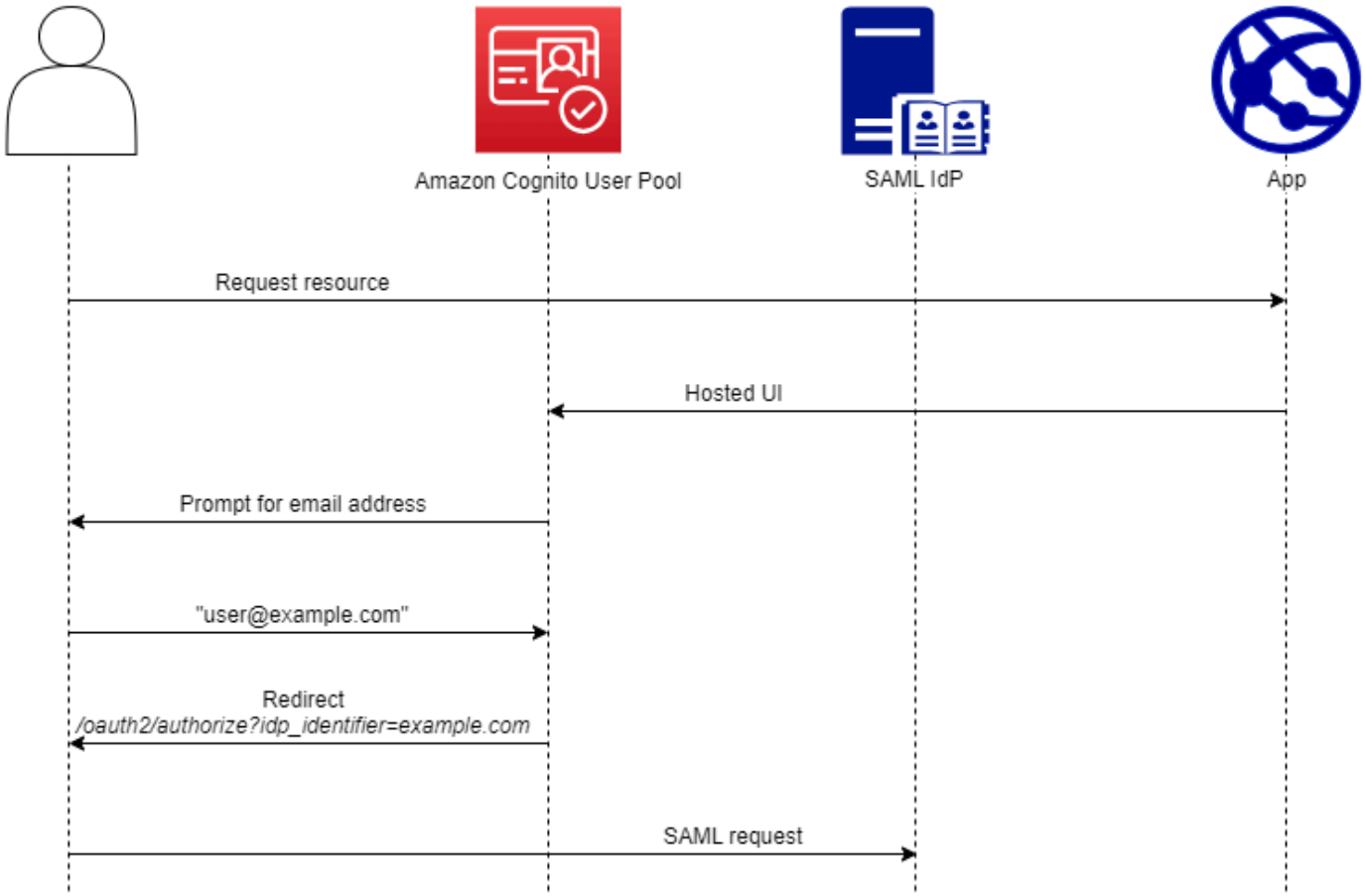
API/CLI

Para configurar la firma de solicitudes de SAML

Configura la firma de solicitudes con el `RequestSigningAlgorithm` parámetro de una solicitud CreateIdentityProviderо una solicitud de UpdateIdentityProviderAPI. El siguiente es un ejemplo `ProviderDetails` de un IdP que admite la firma de solicitudes.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Nombres e identificadores de proveedores de identidad SAML



Al asignar un nombre a los proveedores de identidad de SAML (IdPs) y asignar identificadores de IdP, puede automatizar el flujo de solicitudes de inicio y cierre de sesión iniciadas por el SP a ese proveedor. Para obtener información sobre las restricciones de cadena del nombre del proveedor, consulte la propiedad de. `ProviderName` CreateIdentityProvider

También puedes elegir hasta 50 identificadores para tus proveedores de SAML. Un identificador es un nombre descriptivo para un IdP de su grupo de usuarios y debe ser único dentro del grupo de usuarios. Si sus identificadores de SAML coinciden con los dominios de correo electrónico de sus usuarios, la interfaz de usuario alojada en Amazon Cognito solicita la dirección de correo electrónico de cada usuario, evalúa el dominio en su dirección de correo electrónico y lo redirige al IDP correspondiente a su dominio. Como la misma organización puede ser propietaria de varios dominios, un único IdP puede tener varios identificadores.

Ya sea que utilices o no identificadores de dominio de correo electrónico, puedes usar identificadores en una aplicación multiusuario para redirigir a los usuarios al IdP correcto. Si desea omitir por

completo la interfaz de usuario alojada, puede personalizar los enlaces que presenta a los usuarios para que los redirijan Autorizar punto de conexión directamente a su IdP. Para iniciar sesión en sus usuarios con un identificador y redirigirlos a su IdP, incluya el identificador en el formato `idp_identifier=myidp.example.com` en los parámetros de solicitud de su solicitud de autorización inicial.

Otro método para transferir un usuario a tu IdP consiste en rellenar el parámetro `identity_provider` con el nombre de tu IdP en el siguiente formato de URL.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
identity_provider=MySAMLIdP&
client_id=1example23456789&
redirect_uri=https://www.example.com
```

Una vez que un usuario inicia sesión con su IdP de SAML, este lo redirige con una respuesta de SAML en el cuerpo a su punto final. `HTTP POST /saml2/idpresponse` Amazon Cognito procesa la afirmación de SAML y, si las afirmaciones de la respuesta cumplen con las expectativas, la redirige a la URL de devolución de llamada del cliente de la aplicación. Una vez que el usuario haya completado la autenticación de esta manera, solo interactuará con las páginas web de su IdP y su aplicación.

Con los identificadores de IdP en formato de dominio, la interfaz de usuario alojada en Amazon Cognito solicita las direcciones de correo electrónico al iniciar sesión y, a continuación, cuando el dominio de correo electrónico coincide con un identificador de IdP, redirige a los usuarios a la página de inicio de sesión de su IdP. Por ejemplo, usted crea una aplicación que requiere que los empleados de dos empresas diferentes inicien sesión. La primera empresa, AnyCompany A, es propietaria de `exampleA.com` y`exampleA.co.uk`. La segunda empresa, AnyCompany B, es propietaria`exampleB.com`. Para este ejemplo, ha configurado dos IdPs, una para cada empresa, de la siguiente manera:

- Para el IdP A, define los identificadores `exampleA.com` y `exampleA.co.uk`.
- Para el IdP B, usted define el identificador `exampleB.com`.

En tu aplicación, invoca la interfaz de usuario alojada del cliente de la aplicación para solicitar a cada usuario que introduzca su dirección de correo electrónico. Amazon Cognito deriva el dominio de la dirección de correo electrónico, correlaciona el dominio con un IdP con un identificador de dominio y redirige al usuario al IdP correcto con una solicitud que contiene un parámetro de solicitud. Autorizar

punto de conexión `idp_identifier` Por ejemplo, si un usuario entrabob@exampleA.co.uk, la siguiente página con la que interactúa es la página de inicio de sesión del IdP en. `https://auth.exampleA.co.uk/sso/saml`

También puede implementar la misma lógica de forma independiente. En tu aplicación, puedes crear un formulario personalizado que recopile las entradas del usuario y las correlacione con el IdP correcto según tu propia lógica. Puedes generar portales de aplicaciones personalizados para cada uno de los inquilinos de la aplicación, en los que cada uno de ellos enlaza con el punto de conexión autorizado con el identificador del inquilino en los parámetros de la solicitud.

Para recopilar una dirección de correo electrónico y analizar el dominio en la interfaz de usuario alojada, asigne al menos un identificador a cada IDP de SAML que haya asignado a su cliente de aplicación. De forma predeterminada, la pantalla de inicio de sesión de la interfaz de usuario alojada muestra un botón para cada uno de los botones IdPs que hayas asignado a tu cliente de aplicación. Sin embargo, si has asignado correctamente los identificadores, la página de inicio de sesión de la interfaz de usuario alojada se parece a la siguiente imagen.

El análisis de dominios en la interfaz de usuario alojada requiere que utilices dominios como identificadores de IdP. Si asignas un identificador de cualquier tipo a cada uno de los SAML IdPs de un cliente de aplicación, la interfaz de usuario alojada de esa aplicación ya no muestra los botones de selección de IdP. Agregue identificadores de IdP para SAML cuando desee utilizar el análisis del correo electrónico o la lógica personalizada para generar redireccionamientos. Si quieres generar redireccionamientos silenciosos y también quieres que tu interfaz de usuario alojada muestre una lista de ellos IdPs, no asignes identificadores y utilices el `identity_provider` parámetro de solicitud en tus solicitudes de autorización.

- Si asigna solo un IdP SAML a su cliente de aplicación, la página de inicio de sesión de la IU alojada muestra un botón para iniciar sesión con ese IdP.
- Si asignas un identificador a cada IDP de SAML que actives para el cliente de tu aplicación, aparecerá un mensaje de usuario para que introduzca una dirección de correo electrónico en la página de inicio de sesión de la interfaz de usuario alojada.
- Si tiene varios IdPs y no les asigna un identificador a todos, la página de inicio de sesión de la interfaz de usuario alojada muestra un botón para iniciar sesión con cada IdP asignado.
- Si ha asignado identificadores a su cliente de aplicaciones IdPs y desea que su interfaz de usuario alojada muestre una selección de botones de IdP, añada un nuevo IdP que no tenga identificador a su cliente de aplicaciones o cree un nuevo cliente de aplicaciones. También puede eliminar un IdP existente y volver a añadirlo sin un identificador. Si crea un nuevo IdP, los usuarios de

SAML crearán nuevos perfiles de usuario. Esta duplicación de usuarios activos puede afectar a la facturación en el mes en que cambie la configuración de su IdP.

Para obtener más información sobre la configuración de proveedores de identidad, consulte Configuración de proveedores de identidad para su grupo de usuarios.

Configurar tu proveedor de identidades SAML externo

Para configurar soluciones de proveedores de identidad (IdP) de SAML 2.0 de terceros para que funcionen con la federación para los grupos de usuarios de Amazon Cognito, debe configurar su IdP de SAML para que se redirija a la siguiente URL de Assertion Consumer Service (ACS):. `https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse` Si su grupo de usuarios tiene un dominio de Amazon Cognito, puede encontrar la ruta del dominio del grupo de usuarios en la pestaña App Integration(Integración de aplicaciones) de su grupo de usuarios en la consola de Amazon Cognito.

Algunos tipos de SAML IdPs requieren que introduzca el formulario`urn`, también denominado URI de audiencia o ID de entidad del SP. `urn:amazon:cognito:sp:us-east-1_EXAMPLE` Encontrará el ID de su grupo de usuarios en la descripción general del grupo de usuarios de la consola de Amazon Cognito.

También debe configurar su IdP de SAML para que proporcione valores para cualquier atributo que haya designado como atributo obligatorio en su grupo de usuarios. Por lo general, `email` es un atributo obligatorio para los grupos de usuarios, en cuyo caso el IdP de SAML debe proporcionar algún tipo de `email` reclamación en su afirmación de SAML y tú debes asignar la notificación al atributo de ese proveedor.

La siguiente información de configuración para soluciones de IdP SAML 2.0 de terceros es un buen punto de partida para configurar la federación con los grupos de usuarios de Amazon Cognito. Para obtener la información más actualizada, consulte directamente la documentación de su proveedor.

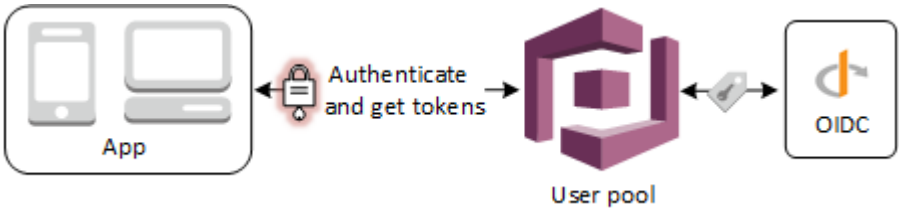
Para firmar las solicitudes de SAML, debe configurar su IdP para que confíe en las solicitudes firmadas por el certificado de firma de su grupo de usuarios. Para aceptar respuestas SAML cifradas, debe configurar su IdP para cifrar todas las respuestas SAML de su grupo de usuarios. Su proveedor dispondrá de documentación sobre la configuración de estas funciones. Para ver un ejemplo de Microsoft, consulte Configurar el cifrado del token SAML de Microsoft Entra.

> **Note**
>
> Amazon Cognito solo requiere el documento de metadatos del proveedor de identidad. Es posible que su proveedor ofrezca información de configuración para Cuenta de AWS la federación con SAML 2.0; esta información no es relevante para la integración de Amazon Cognito.

Solución	Más información
Microsoft Active Directory Federation Services (AD FS)	Explorador de metadatos de la federación
Okta	Cómo descargar los metadatos del IdP y los certificados de firma de SAML para la integración de una aplicación SAML
Auth0	Configure Auth0 como proveedor de identidad SAML
Identidad de ping () PingFederate	Exportación de metadatos de SAML desde PingFederate
JumpCloud	Notas de configuración de SAML
SecureAuth	Integración de aplicaciones SAML

Uso de proveedores de identidad OIDC con un grupo de usuarios

Puede permitir que los usuarios que ya tienen cuentas con proveedores de identidad de OpenID Connect (OIDC) se salten el paso de registro e inicien sesión en su aplicación con una cuenta existente. IdPs Con la IU web alojada e incorporada, Amazon Cognito proporciona el control y la administración de los tokens de los usuarios autenticados por todos los proveedores de identidad. De esta forma, los sistemas backend pueden estandarizar un conjunto de tokens para los grupos de usuarios.



> **Note**
>
> El inicio de sesión a través de un tercero (federación) está disponible en los grupos de usuarios de Amazon Cognito. Esta característica es independiente de la federación a través de grupos de identidades de Amazon Cognito (identidades federadas).

Puede añadir un IdP de OIDC a su grupo de usuarios mediante el método API del AWS Management Console grupo de usuarios AWS CLI, o mediante él. CreateIdentityProvider

Temas

- Requisitos previos
- Paso 1: Registrarse en un proveedor de identidad OIDC
- Paso 2: Agregar un proveedor de identidades (IdP) OIDC al grupo de usuarios
- Paso 3: Probar la configuración del proveedor de identidades (IdP) OIDC
- Flujo de autenticación de proveedores de identidad (IdP) de grupos de usuarios OIDC

Requisitos previos

Antes de comenzar, necesitará lo siguiente:

- Un grupo de usuarios con un cliente de aplicación y un dominio de grupo de usuarios. Para obtener más información, consulte Crear un grupo de usuarios.
- Un proveedor de identidad OIDC con la siguiente configuración:
  - Admite la autenticación del cliente `client_secret_post`. Amazon Cognito no comprueba la notificación de `token_endpoint_auth_methods_supported` en el punto de conexión de detección de OIDC para su IdP. Amazon Cognito no admite la autenticación del cliente `client_secret_basic`. Para obtener más información acerca de la autenticación del cliente, consulte la sección sobre autenticación del cliente en la documentación de OpenID Connect.

- Solo utiliza HTTPS para puntos de conexión de OIDC, como `openid_configuration`, `userInfo` y `jwks_uri`.
- Solo utiliza los puertos TCP 80 y 443 para puntos de conexión de OIDC.
- Solo firma tokens de ID con algoritmos HMAC-SHA, ECDSA o RSA.
- Publica una reclamación `kid` de ID clave en su `jwks_uri` e incluye una reclamación `kid` en sus tokens.

Paso 1: Registrarse en un proveedor de identidad OIDC

Antes de crear un proveedor de identidad OIDC con Amazon Cognito, debe registrar su aplicación en el proveedor de identidad OIDC para recibir un ID y un secreto de cliente.

Para registrarse en un proveedor de identidad OIDC

1. Crear una cuenta de desarrollador con el proveedor de identidad OIDC.

   Enlaces al OIDC IdPs

Proveedor de identidad OIDC	Instalación	URL de detección de OIDC
Salesforce	Instalar un proveedor de identidad Salesforce	https://login.salesforce.com
Ping Identity	Instalar un proveedor de identidad Ping Identity	https://*Dirección de dominio Ping*:9031/idp/userinfo.openid Por ejemplo: `https://pf.company.com:9031/idp/userinfo.openid`
Okta	Instalar un proveedor de identidad Okta	https://*Subdominio de Okta*.oktapreview.com o bien `https://`*`Your Okta subdomain`*`.okta.com`

Proveedor de identidad OIDC	Instalación	URL de detección de OIDC
Microsoft Azure Active Directory (Azure AD)	Instalar un proveedor de identidad Microsoft Azure AD	`https://login.microsoftonline.com/ {tenant}/v2.0`
Google	Instalar un proveedor de identidad Google	https://accounts.google.com **Note** Amazon Cognito ofrece la opción de elegir a Google como proveedor de identidad social integrado para iniciar sesión. Le recomendamos que utilice el proveedor de identidad integrado. Consulte Usar proveedores de identidad social con un grupo de usuarios.

2. Registre la URL de dominio del grupo de usuarios con el punto de enlace `/oauth2/idpresponse` en el proveedor de identidad OIDC. De este modo, se garantiza que el proveedor de identidad OIDC la aceptará cuando Amazon Cognito la presente para autenticar usuarios.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

3. Registre su URL de devolución de llamada con su grupo de usuarios de Amazon Cognito. Esta es la URL de la página a la que Amazon Cognito redirigirá al usuario después de una autenticación correcta.

```
https://www.example.com
```

4. Seleccione los ámbitos. El ámbito openid es obligatorio. El ámbito email es necesario para conceder acceso a las notificaciones email e email_verified .

5. El proveedor de identidad OIDC le proporciona un ID y un secreto de cliente. Los usará al configurar un proveedor de identidad OIDC en el grupo de usuarios.

Ejemplo: Utilizar Salesforce como un proveedor de identidad OIDC con el grupo de usuarios

Puede utilizar un IdP OIDC cuando desee establecer una relación de confianza entre un IdP compatible con OIDC como Salesforce y un grupo de usuarios.

1. Cree una cuenta en el sitio web de desarrolladores de Salesforce.
2. Inicie sesión con la cuenta de desarrollador que ha configurado en el paso anterior.
3. En la página de Salesforce, realice alguna de las operaciones siguientes:
   - Si utiliza Lightning Experience, elija el icono de engranaje de configuración y, a continuación, elija Setup Home (Inicio de configuración).
   - Si utiliza Salesforce Classic y ve la opción Setup (Configuración) en el encabezado de la interfaz de usuario, elíjala.
   - Si utiliza Salesforce Classic y no aparece la opción Setup (Configuración) en el encabezado, elija su nombre en la barra de navegación superior y elija Setup (Configuración) en la lista desplegable.
4. En la barra de navegación de la izquierda, elija Company Settings (Configuración de la empresa).
5. En la barra de navegación, elija Domain (Dominio), introduzca un dominio y elija Create (Crear).
6. En la barra de navegación izquierda, en Platform Tools (Herramientas de plataforma) y elija Apps (Aplicaciones).
7. Elija App Manager (Administrador de aplicaciones).
8. a. Elija New connected app (Nueva aplicación conectada).
   b. Rellene los campos según sea necesario.

      En Start URL (URL de inicio), ingrese una URL en el punto de conexión `/authorize` del dominio del grupo de usuarios que inicia sesión con su IdP de Salesforce. Cuando los usuarios acceden a la aplicación conectada, Salesforce los dirige a esta URL para completar el inicio de sesión. A continuación, Salesforce redirige a los usuarios a la URL de devolución de llamada que ha asociado a su cliente de aplicación.

```
https://mydomain.us-east-1.amazoncognito.com/authorize?
response_type=code&client_id=<your_client_id>&redirect_uri=https://
www.example.com&identity_provider=CorpSalesforce
```

c. Habilite OAuth settings (Configuración de OAuth) e ingrese la URL del punto de conexión `/oauth2/idpresponse` del dominio del grupo de usuarios en Callback URL (URL de devolución de llamada). Esta es la URL en la que Salesforce emite el código de autorización que Amazon Cognito intercambia por un token de OAuth.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

9. Seleccione los ámbitos. Debe incluir el valor openid de ámbito. Para conceder acceso a las notificaciones email e email_verified, añada el ámbito email. Separe los ámbitos por espacios.
10. Seleccione Crear.

    En Salesforce, el ID de cliente se denomina Consumer Key (Clave de consumidor)y el secreto de cliente se llama Consumer Secret (Secreto de consumidor). Observe los valores del ID de cliente y el secreto de cliente. Los usará en la sección siguiente.

Paso 2: Agregar un proveedor de identidades (IdP) OIDC al grupo de usuarios

En esta sección configurará el grupo de usuarios para procesar las solicitudes de autenticación basada en OIDC provenientes de un proveedor de identidad OIDC.

Para agregar un proveedor de identidad OIDC (mediante consola de Amazon Cognito), siga estos pasos:

Agregar un IdP OIDC

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus credenciales. AWS
2. Elija User Pools (Grupos de usuarios) en el menú de navegación.
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). Localice Federated sign-in (Inicio de sesión federado) y luego seleccione Add an identity provider (Agregar un proveedor de identidad).
5. Elija un IdP OpenID Connect.
6. Introduzca un nombre único en Provider name (Nombre de proveedor).

7. Introduzca el ID de cliente que recibió de su proveedor en Client ID (ID de cliente).
8. Introduzca el secreto de cliente que recibió de su proveedor en Client Secret (Secreto de cliente).
9. Introduzca los Ámbitos autorizados para este proveedor. Los ámbitos definen qué grupos de atributos de usuario (tales como `name` y `email`) serán solicitados por su aplicación al proveedor. Los ámbitos deben estar separados por espacios, de acuerdo con la especificación OAuth 2.0.

   Al usuario de la aplicación se le pedirá su consentimiento para proporcionar estos atributos a su aplicación.
10. Seleccione un Attribute request method (Método de solicitud de atributo) para proporcionar a Amazon Cognito el método de HTTP (GET o POST) que debe emplearse para obtener los detalles de usuario del punto de enlace userInfo operado por su proveedor.
11. Seleccione un Setup method (Método de configuración) para recuperar los puntos de enlace de OpenID Connect con Auto fill through issuer URL (Autorrellenar mediante la URL del emisor) o Manual input (Entrada manual). Use Auto fill through issuer URL (Autorrellenar mediante la URL del emisor) cuando su proveedor tenga un punto de conexión `.well-known/openid-configuration` público en el que Amazon Cognito pueda recuperar las URL de los puntos de conexión de `authorization`, `token`, `userInfo` y `jwks_uri`.
12. Introduzca la URL del emisor o las URL de los puntos de conexión de `authorization`, `token`, `userInfo` y `jwks_uri` de su IdP.

    > **Note**
    >
    > La URL debe comenzar por `https://` y no debe terminar con una barra `/`. Solo se pueden utilizar los números de puerto 443 y 80 con esta URL. Por ejemplo, Salesforce usa esta URL:
    > `https://login.salesforce.com`
    > Si elige autorrellenar, el documento de detección debe utilizar HTTPS para los siguientes valores: `authorization_endpoint`, `token_endpoint`, `userinfo_endpoint` y `jwks_uri`. De lo contrario, el inicio de sesión fallará.
13. A la notificación OIDC sub se le asigna el atributo de grupo de usuarios Username (Nombre de usuario) de forma predeterminada. Puede asignar a las notificaciones OIDC otros atributos de grupo de usuarios. Introduzca la notificación OIDC y elija el atributo de grupo de usuarios correspondiente en la lista desplegable. Por ejemplo, a la notificación email (correo electrónico) se le suele asignar el atributo de grupo de usuarios Email (Correo electrónico).

14. Asigne atributos de su IdP al grupo de usuarios. Para obtener más información, consulte Especificación de asignaciones de atributos del proveedor de identidad para su grupo de usuarios.
15. Seleccione Crear.
16. DelIntegración de clientes de aplicaciones, elija uno de los Clientes de aplicaciones en la lista y Edit hosted UI settings (Modificar la configuración de IU). Agregue el nuevo IdP OIDC al cliente de aplicación en Identity providers (Proveedores de identidad).
17. Elija Guardar cambios.

Para añadir un proveedor de identidad OIDC (AWS CLI)

- Consulte las descripciones de los parámetros del método CreateIdentityProviderde API.

```
aws cognito-idp create-identity-provider
--user-pool-id string
--provider-name string
--provider-type OIDC
--provider-details map

--attribute-mapping string
--idp-identifiers (list)
--cli-input-json string
--generate-cli-skeleton string
```

Utilice este mapa de detalles de proveedor:

```
{
  "client_id": "string",
  "client_secret": "string",
  "authorize_scopes": "string",
  "attributes_request_method": "string",
  "oidc_issuer": "string",

  "authorize_url": "string",
```

```
  "token_url": "string",
  "attributes_url": "string",
  "jwks_uri": "string"
}
```

Paso 3: Probar la configuración del proveedor de identidades (IdP) OIDC

Puede crear la URL de autorización con los elementos de las dos secciones anteriores y utilizarlos para probar la configuración del proveedor de identidad OIDC.

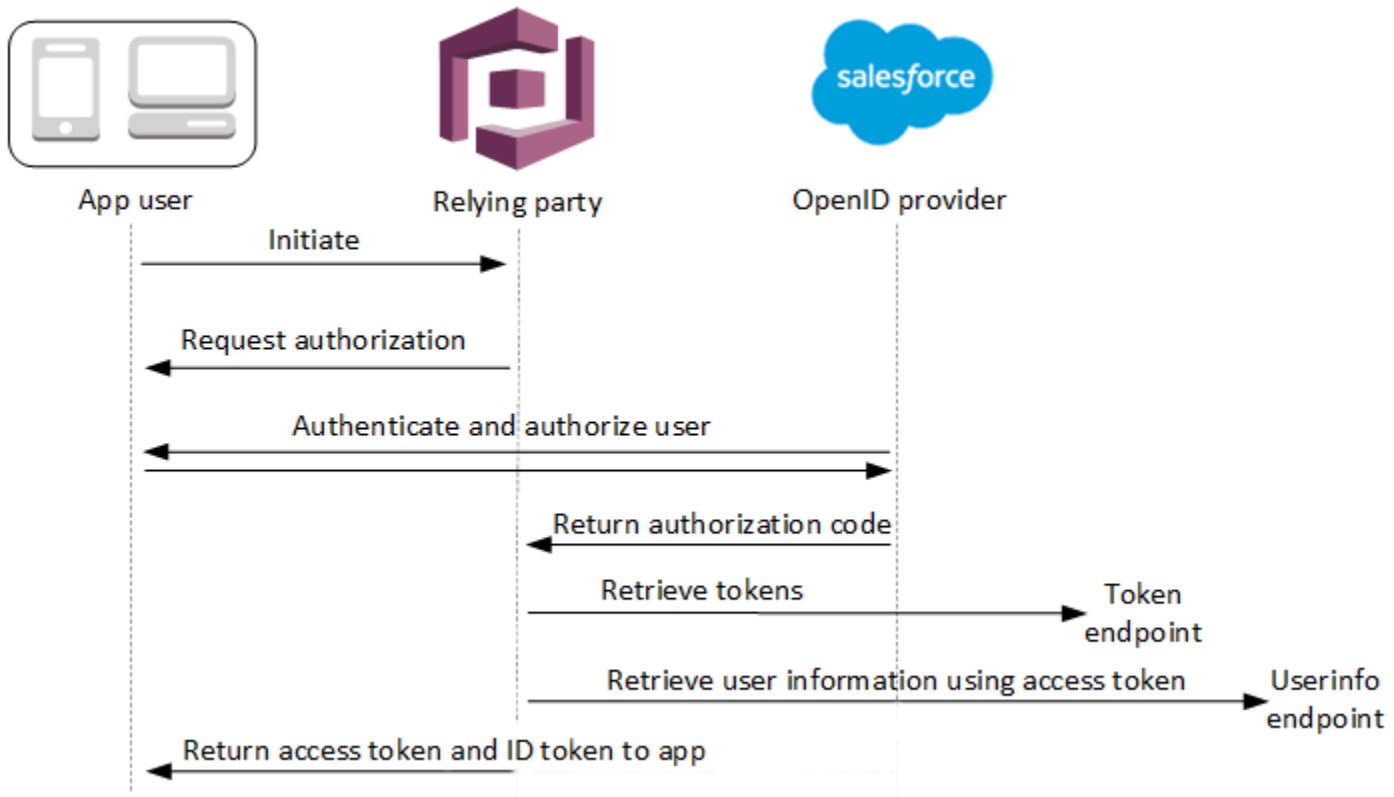
```
https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Puede encontrar el dominio en la página de la consola Domain name (Nombre de dominio) del grupo de usuarios. El client_id se encuentra en la página General settings (Configuración general). Use la URL de devolución de llamada para el parámetro redirect_uri. Esta es la URL de la página a la que se redirigirá al usuario después de una autenticación correcta.

Flujo de autenticación de proveedores de identidad (IdP) de grupos de usuarios OIDC

Cuando un usuario inicia sesión en su aplicación a través de un proveedor de identidades (IdP) OIDC, estos pasan a través del siguiente flujo de autenticación.

1. El usuario llega a la página de inicio de sesión incorporada de Amazon Cognito, donde se le ofrece la opción de iniciar sesión a través de un proveedor de identidad OIDC, como Salesforce.
2. Se redirige al usuario al punto de conexión `authorization` del IdP OIDC.
3. Una vez que el usuario está autenticado, el proveedor de identidad OIDC lo redirige a Amazon Cognito con un código de autorización.
4. Amazon Cognito intercambia el código de autorización con el proveedor de identidad OIDC para obtener un token de acceso.
5. También crea o actualiza la cuenta de usuario en el grupo de usuarios.
6. Además, otorga a la aplicación tokens de portador, que pueden incluir tokens de identidad, acceso y actualización:



> **Note**
>
> Amazon Cognito cancela las solicitudes de autenticación que no se completan en 5 minutos y redirige al usuario a la IU alojada. La página muestra un mensaje de error `Something went wrong`.

El OIDC es una capa de identidad adicional a la OAuth 2.0, que especifica los tokens de identidad con formato JSON (JWT) que emiten las aplicaciones cliente del OIDC (partes dependientes IdPs). Consulte la documentación de su proveedor de identidad OIDC para obtener información sobre cómo agregar Amazon Cognito como parte aceptante de OIDC.

Cuando un usuario se autentica con una adjudicación de código de autorización, el grupo de usuarios devuelve tokens de ID, acceso y actualización. El token de ID es un token OIDC estándar para la administración de identidades, y el token de acceso es un token OAuth 2.0 estándar. Para obtener más información sobre los tipos de adjudicaciones que puede admitir el cliente de la aplicación de grupo de usuarios, consulte Autorizar punto de conexión.

Cómo procesa un grupo de usuarios las notificaciones de un proveedor de OIDC

Cuando el usuario completa el inicio de sesión con un proveedor de OIDC externo, la interfaz de usuario alojada en Amazon Cognito recupera un código de autorización del IdP. Su grupo de usuarios intercambia el código de autorización para los tokens de acceso e identificación con el punto de conexión`token` de su IdP. Su grupo de usuarios no transfiere estos tokens a su usuario ni a su aplicación, sino que los usa para crear un perfil de usuario con los datos que presenta en las notificaciones en sus propios tokens.

Amazon Cognito no valida el token de acceso de forma independiente. En cambio, solicita información sobre los atributos del usuario al punto de conexión `userInfo` del proveedor y espera que se deniegue la solicitud si el token no es válido.

Amazon Cognito valida el token de identificación del proveedor con las siguientes comprobaciones:

1. Comprueba que el proveedor haya firmado el token con un algoritmo del siguiente conjunto: RSA, HMAC y Elliptic Curve.
2. Si el proveedor firmó el token con un algoritmo de firma asimétrico, comprueba que el identificador de clave de firma que aparece en la notificación `kid` del token aparezca en el punto de conexión `jwks_uri` del proveedor.
3. Compara la firma del token de identificación con la firma que espera en función de los metadatos del proveedor.
4. Compara la notificación `iss` con el emisor de OIDC configurado para el IdP.
5. Compara si la notificación `aud` coincide con la identificación de cliente configurada en el IdP o si contiene la identificación de cliente configurada si hay varios valores en el aviso `aud`.
6. Comprueba que la marca de tiempo de la notificación `exp` no sea anterior a la hora actual.

Su grupo de usuarios valida el token de identificación y, a continuación, intenta realizar una solicitud al punto de conexión `userInfo` del proveedor con el token de acceso del proveedor. Recupera la información del perfil de usuario que los ámbitos del token de acceso le autoricen a leer. A continuación, su grupo de usuarios busca los atributos de usuario que haya establecido como obligatorios en su grupo de usuarios. Debe crear asignaciones de atributos en la configuración del proveedor para los atributos obligatorios. Su grupo de usuarios comprueba el token de identificación del proveedor y la respuesta `userInfo`. Su grupo de usuarios escribe todas las notificaciones que coinciden con las reglas de asignación en los atributos de usuario del perfil de usuario del grupo de usuarios. Su grupo de usuarios hace caso omiso de los atributos que coinciden con una regla de asignación, pero no son obligatorios y no aparecen en las notificaciones del proveedor.

Especificación de asignaciones de atributos del proveedor de identidad para su grupo de usuarios

Puede usar la o la AWS Management Console API para especificar las AWS CLI asignaciones de atributos para el proveedor de identidades (IdP) de su grupo de usuarios.

Cuestiones que debe saber acerca de los mapeos

Antes de empezar a configurar el mapeo de atributos de usuario, revise los siguientes detalles importantes.

- Cuando un usuario federado se registra en su aplicación, debe haber una asignación para cada atributo del grupo de usuarios que su grupo de usuarios requiera. Por ejemplo, si el grupo de usuarios requiere un atributo `email` para iniciar sesión, asigne este atributo a su equivalente desde el IdP.
- De forma predeterminada, las direcciones de correo electrónico mapeadas no se verifican. No se puede verificar una dirección de correo electrónico mapeada con un código único. En su lugar, asigne un atributo desde el IdP para obtener el estado de verificación. Por ejemplo, Google y la mayoría de los proveedores de OIDC incluyen el atributo `email_verified`.
- Puede asignar tokens de proveedor de identidades (IdP) a atributos personalizados en su grupo de usuarios. Los proveedores sociales presentan un token de acceso y los proveedores de OIDC presentan un token de acceso e identificación. Para asignar un token, agregue un atributo personalizado con una longitud máxima de 2048 caracteres, otorgue al cliente de la aplicación acceso de escritura al atributo y asigne el `access_token` o el `id_token` desde el IdP al atributo personalizado.
- Para cada atributo de grupo de usuarios asignado, la longitud máxima del valor de 2048 caracteres debe ser lo suficientemente amplia para el valor que Amazon Cognito obtiene del IdP. De lo contrario, Amazon Cognito comunica un error cuando los usuarios inician sesión en la aplicación. Amazon Cognito no admite la asignación de tokens de IdP a atributos personalizados cuando los tokens tienen más de 2048 caracteres.
- Amazon Cognito deriva el `username` atributo del perfil de un usuario federado a partir de notificaciones específicas que aprueba su IdP federado, como se muestra en la siguiente tabla. Amazon Cognito antepone este valor de atributo al nombre de su IdP, por ejemplo. `MyOIDCIdP_[sub]` Si desea que sus usuarios federados tengan un atributo que coincida exactamente con un atributo de su directorio de usuarios externo, asigne ese atributo a un atributo de inicio de sesión de Amazon Cognito, como. `preferred_username`

Proveedor de identidad	Atributo de origen de **`username`**
Facebook	`id`
Google	`sub`
Login with Amazon	`user_id`
Inicio de sesión con Apple	`sub`
Proveedores SAML	`NameID`
Proveedores de OpenID Connect (OIDC)	`sub`

- Amazon Cognito debe poder actualizar los atributos del grupo de usuarios mapeados cuando los usuarios inician sesión en la aplicación. Cuando un usuario inicia sesión a través de un IdP, Amazon Cognito actualiza los atributos asignados con la información más reciente del IdP. Amazon Cognito actualiza cada atributo mapeado incluso si su valor actual ya coincide con la información más reciente. Para asegurarse de que Amazon Cognito pueda actualizar los atributos, consulte los siguientes requisitos:
  - Todos los atributos personalizados del grupo de usuarios que asigne desde su IdP deben ser mutables. Puede actualizar los atributos personalizados mutables en cualquier momento. Por el contrario, solo puede establecer un valor para el atributo personalizado inmutable de un usuario cuando cree por primera vez el perfil de usuario. Para crear un atributo personalizado mutable en la consola de Amazon Cognito, active la casilla de verificación Mutable del atributo que agregue al seleccionar Add custom attributes (Añadir atributos personalizados) en la pestaña Sign-up experience (Experiencia de registro). O bien, si crea su grupo de usuarios mediante la operación de CreateUserPoolAPI, puede establecer el `Mutable` parámetro para cada uno de estos atributos en. `true` Si su IDP envía un valor para un atributo inmutable asignado, Amazon Cognito devuelve un error y se produce un error al iniciar sesión.
  - En la configuración del cliente de la aplicación, los atributos asignados deben ser de escritura. Puede definir los atributos que se pueden escribir en la página App clients (Clientes de aplicaciones) en la consola de Amazon Cognito. O bien, si crea el cliente de aplicación mediante la operación `CreateUserPoolClient` de la API, puede agregar estos atributos a la matriz `WriteAttributes`. Si su IdP envía un valor para un atributo mapeado que no se puede escribir, Amazon Cognito no establece el valor del atributo y procede a la autenticación.

- Cuando los atributos del IdP contienen varios valores, Amazon Cognito aplana todos los valores en una sola cadena delimitada por comas y codifica en forma de URL los valores que contienen caracteres no alfanuméricos (excepto los caracteres ", ',' y `.` "). `-` `*` `_` Debe descodificar y analizar los valores individuales antes de usarlos en la aplicación.

Especificación de asignaciones de atributos del proveedor de identidad para su grupo de usuarios (AWS Management Console)

Puede usarlo AWS Management Console para especificar las asignaciones de atributos para el IdP de su grupo de usuarios.

> **Note**
>
> Amazon Cognito mapeará las notificaciones entrantes a los atributos del grupo de usuarios solo si las notificaciones existen en el token de entrada. Si una notificación asignada anteriormente ya no existe en el token de entrada, no cambiará ni se eliminará. Si la aplicación requiere la asignación de notificaciones eliminadas, puede usar el desencadenador de Lambda de autenticación previa para eliminar el atributo personalizado durante la autenticación y permitir que estos atributos vuelvan a rellenarse desde el token de entrada.

Para especificar una asignación de atributo de IdP social

1. Inicie sesión en la consola de Amazon Cognito. Si se le solicita, introduzca sus credenciales. AWS
2. En el panel de navegación, elija User Pools (Grupos de usuarios) y elija el grupo de usuarios que desea editar.
3. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión) y localice Federated sign-in (Inicio de sesión federado).
4. Elija Add an identity provider (Agregar un proveedor de identidad), o elija el IdP de Facebook, Google, Amazon o Apple que haya configurado. Localice Attribute mapping (Asignación de atributos) y elija Edit (Editar).

   Para obtener más información acerca de cómo agregar un IdP social, consulte Usar proveedores de identidad social con un grupo de usuarios.
5. Para cada atributo que necesite asignar, complete los pasos siguientes:

a. Seleccione un atributo de la columna User pool attribute (Atributo de grupo de usuarios). Este es el atributo que se asigna al perfil de usuario de su grupo de usuarios. Los atributos personalizados se enumeran después de los atributos estándar.

b. Seleccione un atributo de la columna attribute (atributo) de ***<provider>***. Este será el atributo que se pasa desde el directorio de proveedores. Los atributos conocidos del proveedor social se proporcionan en una lista desplegable.

c. Para asignar atributos adicionales entre su IdP y Amazon Cognito, elija Add another attribute (Agregar otro atributo).

6. Elija Guardar cambios.

Para especificar un mapeo de atributo de proveedor SAML

1. Inicie sesión en la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. En el panel de navegación, elija User Pools (Grupos de usuarios) y elija el grupo de usuarios que desea editar.
3. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión) y localice Federated sign-in (Inicio de sesión federado).
4. Elija Add an identity provider (Agregar un proveedor de identidad) o elija el IdP SAML que ha configurado. Localice Attribute mapping (Asignación de atributos) y elija Edit (Editar). Para obtener más información acerca de cómo agregar un IdP SAML, consulte Uso de proveedores de identidad SAML con un grupo de usuarios.
5. Para cada atributo que necesite asignar, complete los pasos siguientes:

   a. Seleccione un atributo de la columna User pool attribute (Atributo de grupo de usuarios). Este es el atributo que se asigna al perfil de usuario de su grupo de usuarios. Los atributos personalizados se enumeran después de los atributos estándar.

   b. Seleccione un atributo de la columna SAML attribute (Atributo de SAML). Este será el atributo que se pasa desde el directorio de proveedores.

      Es posible que su IdP ofrezca aserciones SAML como referencia. Algunos IdPs usan nombres simples, como`email`, mientras que otros usan nombres de atributos con formato URL similares a los siguientes:

      ```
      http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
      ```

c. Para asignar atributos adicionales entre su IdP y Amazon Cognito, elija Add another attribute (Agregar otro atributo).

6. Elija Guardar cambios.

Especificar las asignaciones de atributos de los proveedores de identidad para su grupo de usuarios (y API)AWS CLIAWS

Utilice los siguientes comandos para especificar asignaciones de atributos del IdP para su grupo de usuarios.

Para especificar asignaciones de atributos en el momento de crear el proveedor

- AWS CLI: `aws cognito-idp create-identity-provider`

  Ejemplo con archivo de metadatos: `aws cognito-idp create-identity-provider --user-pool-id <user_pool_id> --provider-name=SAML_provider_1 --provider-type SAML --provider-details file:///details.json --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`

  Donde `details.json` contiene:

```
{
    "MetadataFile": "<SAML metadata XML>"
}
```

> **Note**
>
> Si el *<XML de metadatos de SAML>* contiene comillas ("), se debe incluir un carácter de escape (\").

  Ejemplo con URL de metadatos:

```
aws cognito-idp create-identity-provider \
--user-pool-id us-east-1_EXAMPLE \
--provider-name=SAML_provider_1 \
--provider-type SAML \
--provider-details MetadataURL=https://myidp.example.com/saml/metadata \
```

```
--attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
emailaddress
```

- AWS API: CreateIdentityProvider

Para especificar asignaciones de atributo de un IdP existente

- AWS CLI: `aws cognito-idp update-identity-provider`

  Ejemplo: `aws cognito-idp update-identity-provider --user-pool-id <user_pool_id> --provider-name <provider_name> --attribute-mapping email=http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`
- AWS API: UpdateIdentityProvider

Para obtener información sobre la asignación de atributos para un IdP específico

- AWS CLI: `aws cognito-idp describe-identity-provider`

  Ejemplo: `aws cognito-idp describe-identity-provider --user-pool-id <user_pool_id> --provider-name <provider_name>`
- AWS API: DescribeIdentityProvider

Vinculación de usuarios federados a un perfil de usuario existente

A menudo, el mismo usuario tiene un perfil con varios proveedores de identidad (IdPs) que usted ha conectado a su grupo de usuarios. Amazon Cognito puede vincular cada aparición de un usuario al mismo perfil de usuario de su directorio. De esta forma, una persona que tenga varios usuarios de IdP puede tener una experiencia coherente en su aplicación. AdminLinkProviderForUserindica a Amazon Cognito que reconozca el identificador único de un usuario en su directorio federado como usuario del grupo de usuarios. Un usuario de su grupo de usuarios cuenta como un usuario activo mensual (MAU) a efectos de facturación si tiene cero o más identidades federadas asociadas al perfil de usuario.

Cuando un usuario federado inicia sesión en su grupo de usuarios por primera vez, Amazon Cognito busca un perfil local que usted haya vinculado a su identidad. Si no existe ningún perfil vinculado, su grupo de usuarios crea uno nuevo. Puede crear un perfil local y vincularlo a su usuario federado en cualquier momento antes de que inicie sesión por primera vez, mediante una solicitud de `AdminLinkProviderForUser` API, ya sea en una tarea de preconfiguración

planificada o en una. Desencadenador de Lambda de prerregistro. Después de que su usuario inicie sesión y Amazon Cognito detecte un perfil local vinculado, su grupo de usuarios lee las solicitudes de su usuario y las compara con las reglas de asignación del IdP. A continuación, su grupo de usuarios actualiza el perfil local vinculado con las reclamaciones asignadas desde su inicio de sesión. De esta forma, puede configurar el perfil local con las solicitudes de acceso y conservar sus solicitudes de identidad en poder de su proveedor up-to-date . Después de que Amazon Cognito haga coincidir su usuario federado con un perfil vinculado, este siempre iniciará sesión en ese perfil. A continuación, podrá vincular más identidades de proveedores de sus usuarios al mismo perfil, lo que proporcionará a un cliente una experiencia coherente en su aplicación. Para vincular a un usuario federado que haya iniciado sesión anteriormente, primero debe eliminar su perfil existente. Puede identificar los perfiles existentes por su formato: `[Provider name]_identifier`. Por ejemplo, `LoginWithAmazon_amzn1.account.AFAEXAMPLE`. Un usuario que ha creado y, a continuación, ha vinculado a una identidad de usuario de terceros tiene el nombre de usuario con el que se creó y un `identities` atributo que contiene los detalles de sus identidades vinculadas.

> **Important**
>
> Dado `AdminLinkProviderForUser` que permite a un usuario con una identidad federada externa iniciar sesión como un usuario existente en el grupo de usuarios, es fundamental que solo se utilice con atributos externos IdPs y de proveedor en los que el propietario de la aplicación confíe.

Por ejemplo, si es un proveedor de servicios administrados (MSP) con una aplicación que comparte con varios clientes. Cada uno de los clientes inicia sesión en su aplicación a través de Active Directory Federation Services (ADFS). Su administrador de TI, Carlos, tiene una cuenta en cada uno de los dominios de sus clientes. Quiere que Carlos sea reconocido como administrador de aplicaciones cada vez que inicie sesión, independientemente del IdP.

Su ADFS IdPs presenta la dirección de correo electrónico de Carlos `msp_carlos@example.com` en la `email` reclamación de las afirmaciones de SAML de Carlos a Amazon Cognito. Cree un usuario en su grupo de usuarios con el nombre de usuario `Carlos`. Los siguientes comandos AWS Command Line Interface (AWS CLI) vinculan las identidades de Carlos desde ADFS1, ADFS2 y ADFS3. IdPs

> **Note**
>
> Puede vincular a un usuario en función de reivindicaciones de atributos específicas. Esta capacidad es exclusiva de OIDC y SAML. IdPs Para otros tipos de proveedores, debe vincular en función de un atributo de origen fijo. Para obtener más información, consulte. AdminLinkProviderForUser Debe establecer `ProviderAttributeName` en `Cognito_Subject` al vincular un IdP social a un perfil de usuario. `ProviderAttributeValue` debe ser el identificador único del usuario con el IdP.

```
aws cognito-idp admin-link-provider-for-user \
--user-pool-id us-east-1_EXAMPLE \
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \
--source-user
 ProviderName=ADFS1,ProviderAttributeName=email,ProviderAttributeValue=msp_carlos@example.com

aws cognito-idp admin-link-provider-for-user \
--user-pool-id us-east-1_EXAMPLE \
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \
--source-user
 ProviderName=ADFS2,ProviderAttributeName=email,ProviderAttributeValue=msp_carlos@example.com

aws cognito-idp admin-link-provider-for-user \
--user-pool-id us-east-1_EXAMPLE \
--destination-user ProviderAttributeValue=Carlos,ProviderName=Cognito \
--source-user
 ProviderName=ADFS3,ProviderAttributeName=email,ProviderAttributeValue=msp_carlos@example.com
```

El perfil de usuario `Carlos`en el grupo de usuarios tiene ahora lo siguiente: atributo `identities`.

```
[{
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS1",
    "providerType": "SAML",
    "issuer": "http://auth.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}, {
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS2",
```

```
    "providerType": "SAML",
    "issuer": "http://auth2.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}, {
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS3",
    "providerType": "SAML",
    "issuer": "http://auth3.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}]
```

Cuestiones que debe saber acerca de la vinculación de usuarios federados

- Puede vincular hasta cinco usuarios federados a cada perfil de usuario.
- Puede vincular usuarios federados a un perfil de usuario federado existente o a un usuario local.
- No puede vincular los proveedores a los perfiles de usuario del AWS Management Console.
- El token de ID de usuario contiene todos sus proveedores asociados en el`identities`notificación.
- Puedes establecer una contraseña para el perfil de usuario federado creado automáticamente en una solicitud de API. AdminSetUserPassword A continuación, el estado de ese usuario cambia de `EXTERNAL_PROVIDER` a `CONFIRMED`. Un usuario en este estado puede iniciar sesión como usuario federado e iniciar flujos de autenticación en la API como un usuario local vinculado. También pueden modificar su contraseña y sus atributos en las solicitudes de API autenticadas mediante token, como y. ChangePasswordUpdateUserAttributes Como práctica de seguridad recomendada y para mantener a los usuarios sincronizados con su IdP externo, no establezca contraseñas en los perfiles de usuarios federados. En su lugar, enlace a los usuarios a perfiles locales con `AdminLinkProviderForUser`.
- Amazon Cognito rellena los atributos del usuario en un perfil de usuario local vinculado cuando el usuario inicia sesión a través del IdP. Amazon Cognito procesa las reclamaciones de identidad en el token de ID de un IdP OIDC y también comprueba el punto de conexión de `userInfo` de los proveedores de OAuth 2.0 y OIDC. Amazon Cognito prioriza la información de un token de ID frente a la información de `userInfo`.

Cuando sepa que su usuario ya no utiliza una cuenta de usuario externa que haya vinculado a su perfil, puede desvincular esa cuenta de usuario de su grupo de usuarios. Cuando vinculó

su usuario, suministró el nombre del atributo del usuario, el valor del atributo y el nombre del proveedor en la solicitud. Para eliminar un perfil que tu usuario ya no necesite, realiza una solicitud de AdminDisableProviderForUserAPI con parámetros equivalentes.

Consulte AdminLinkProviderForUserpara ver ejemplos y sintaxis de comandos adicionales en los AWS SDK.

Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda

Amazon Cognito trabaja con funciones de AWS Lambda para modificar el comportamiento de autenticación de su grupo de usuarios. Puede configurar su grupo de usuarios para que invoque automáticamente funciones de Lambda antes de su primer registro, después de que completen la autenticación y en varias etapas intermedias. Sus funciones pueden modificar el comportamiento predeterminado de su flujo de autenticación, realizar solicitudes a la API para modificar su grupo de usuarios u otros recursos de AWS y comunicarse con sistemas externos. El código de las funciones de Lambda es suyo. Amazon Cognito envía datos de eventos a su función, espera a que la función procese los datos y, en la mayoría de los casos, anticipa un evento de respuesta que refleja cualquier cambio que desee realizar en la sesión.

Dentro del sistema de eventos de solicitud y respuesta, puede introducir sus propios retos de autenticación, migrar usuarios entre su grupo de usuarios y otro almacén de identidades, personalizar mensajes y modificar tokens web JSON (JWT).

Los disparadores de Lambda pueden personalizar la respuesta que Amazon Cognito ofrece al usuario después de iniciar una acción en su grupo de usuarios. Por ejemplo, puede impedir el inicio de sesión de un usuario que, de otro modo, lo conseguiría. El usuario también puede realizar operaciones en tiempo de ejecución en su entorno de AWS, API externas, bases de datos o almacenes de identidades. El disparador de migración de usuarios, por ejemplo, puede combinar una acción externa con un cambio en Amazon Cognito: puede buscar la información del usuario en un directorio externo y, a continuación, establecer los atributos de un nuevo usuario en función de esa información externa.

Cuando tiene un disparador de Lambda asignado a su grupo de usuarios, Amazon Cognito interrumpe su flujo predeterminado para solicitar información a su función. Amazon Cognito genera un evento de JSON y lo pasa a la función. El evento contiene información sobre la solicitud del usuario para crear una cuenta de usuario, iniciar sesión, restablecer una contraseña o actualizar un

atributo. La función tendrá entonces la oportunidad de realizar una acción o de enviar de vuelta el evento sin modificarlo.

En la siguiente tabla se resumen algunas formas de utilizar los desencadenadores de Lambda para personalizar las operaciones del grupo de usuarios:

Flujo del grupo de usuarios	Operación	Descripción
Flujo de autenticación personalizado	Definición de desafíos de autenticación	Determina el siguiente desafío en un flujo de autenticación personalizado
	Creación de desafíos de autenticación	Crea un desafío en un flujo de autenticación personalizado
	Verificación de la respuesta al desafío de autenticación	Determina si una respuesta es correcta en un flujo de autenticación personalizado
Eventos de autenticación	the section called “Desencadenador de Lambda anterior a la autenticación”	Validación personalizada para aceptar o denegar la solicitud de inicio de sesión
	the section called “Desencadenador de Lambda posterior a la autenticación”	Registra eventos para los análisis personalizados
	the section called “Desencadenador de Lambda de pregeneración de tokens.”	Aumenta o suprime las notificaciones de tokens
Registro	the section called “Desencadenador de Lambda de prerregistro.”	Realiza una validación personalizada que acepta o rechaza la solicitud de inscripción
	the section called “Desencadenador de Lambda de posconfirmación.”	Agrega mensajes de bienvenida personalizados o

Flujo del grupo de usuarios	Operación	Descripción
		el registro de eventos para los análisis personalizados
	the section called “Migración del desencadenador de Lambda del usuario”	Migra un usuario desde un directorio de usuarios existente a los grupos de usuarios
Mensajes	the section called “Desencadenador de Lambda para mensajes personalizados”	Realiza una personalización avanzada y localiza mensajes
Creación de tokens	the section called “Desencadenador de Lambda de pregeneración de tokens.”	Agrega o elimina atributos en tokens de identificación
Proveedores externos de correo electrónico y SMS	the section called “Desencadenadores de Lambda para remitentes personalizados”	Usa un proveedor de terceros para enviar mensajes SMS y de correo electrónico

Temas

- Consideraciones importantes
- Adición de un desencadenador de Lambda a un grupo de usuarios
- Evento desencadenador de Lambda para un grupo de usuarios
- Parámetros comunes del desencadenador de Lambda para un grupo de usuarios
- Conexión de las operaciones de la API a los disparadores de Lambda
- Conexión de disparadores de Lambda a las operaciones funcionales del grupo de usuarios
- Desencadenador de Lambda de prerregistro.
- Desencadenador de Lambda de posconfirmación.
- Desencadenador de Lambda anterior a la autenticación
- Desencadenador de Lambda posterior a la autenticación.
- Desencadenadores de Lambda de desafío de autenticación personalizado
- Desencadenador de Lambda anterior a la generación del token

- Migración del desencadenador de Lambda del usuario
- Desencadenador de Lambda para mensajes personalizados
- Desencadenadores de Lambda para remitentes personalizados

Consideraciones importantes

Al preparar sus grupos de usuarios para funciones de Lambda, tenga en cuenta lo siguiente:

- Es posible que los eventos que Amazon Cognito envía a los desencadenadores de Lambda cambien con las nuevas características. Es posible que cambien las posiciones de los elementos de respuesta y solicitud en la jerarquía JSON o que se agreguen los nombres de los elementos. En la función de Lambda, puede esperar recibir los pares clave-valor del elemento de entrada que se describen en esta guía, pero una validación de entrada más estricta puede provocar errores en las funciones.
- Puede elegir una de las múltiples versiones de los eventos que Amazon Cognito envía a algunos desencadenadores. Es posible que algunas versiones requieran que acepte un cambio en los precios de Amazon Cognito. Para obtener más información acerca de los precios, consulte Precios de Amazon Cognito. Para personalizar los tokens de acceso en Desencadenador de Lambda anterior a la generación del token, debe configurar el grupo de usuarios con características de seguridad avanzadas y actualizar la configuración de los desencadenadores de Lambda para usar la versión 2 del evento.
- Excepto por Desencadenadores de Lambda para remitentes personalizados, Amazon Cognito invoca funciones de Lambda de forma sincrónica. Cuando Amazon Cognito llama a la función de Lambda, esta debe responder en un plazo de 5 segundos. Si no es así y si se puede volver a intentar la llamada, Amazon Cognito vuelve a intentar la llamada. Después de tres intentos fallidos, la función agota el tiempo de espera. No puede cambiar ese valor de tiempo de espera de cinco segundos. Para obtener más información, consulte Modelo de programación de Lambda en la guía para desarrolladores de AWS Lambda.

  Amazon Cognito no reintenta las llamadas a funciones que devuelven un Error de invocación con un código de estado HTTP de 500-599. Estos códigos indican un problema de configuración que hace que Lambda no pueda lanzar la función. Para obtener más información, consulte Control de errores y reintentos automáticos en AWS Lambda.
- No puede declarar una versión de función en la configuración de su desencadenador de Lambda. Los grupos de usuarios de Amazon Cognito invocan la última versión de su función de forma predeterminada. No obstante, puede asociar una versión de función a un alias y establecer

su desencadenador `LambdaArn` al ARN del alias en una solicitud a la API CreateUserPool o UpdateUserPool. Esta opción no está disponible en la AWS Management Console. Para obtener más información acerca de los alias, consulte Alias de función de Lambda en la Guía para desarrolladores de AWS Lambda.

- Si elimina un desencadenador de Lambda, deberá actualizar el desencadenador correspondiente en el grupo de usuarios. Por ejemplo, si elimina el desencadenador posterior a la autenticación, deberá establecer el desencadenador Posterior a la autenticación del grupo de usuarios correspondiente en none (ninguno).
- Si la función de Lambda no devuelve los parámetros de solicitud y respuesta a Amazon Cognito o devuelve un error, el evento de autenticación no se realiza correctamente. Puede devolver un error en la función para impedir que un usuario se registre, autentique, genere el token o cualquier otra etapa del flujo de autenticación que invoque un desencadenador de Lambda.

  La interfaz de usuario alojada en Amazon Cognito devuelve los errores que los desencadenadores de Lambda generan como texto de error sobre la solicitud de inicio de sesión. La API de los grupos de usuarios de Amazon Cognito devuelve los errores de activación en formato *`[trigger]`* `failed with error` *`[error text from response]`*. Como práctica recomendada, en las funciones de Lambda solo genere errores que quiera que vean los usuarios. Utilice métodos de salida como `print()` para registrar cualquier información confidencial o de depuración en CloudWatch Logs. Para ver un ejemplo, consulte Ejemplo de antes de registrarse: denegar el registro si el nombre de usuario tiene menos de cinco caracteres.
- Puede agregar una función de Lambda en otra Cuenta de AWS como desencadenador del grupo de usuarios. Debe agregar desencadenadores multicuentas con las operaciones de la API CreateUserPool y UpdateUserPool o LOS equivalentes en AWS CloudFormation y AWS CLI. No puede agregar funciones para varias cuentas en la AWS Management Console.
- Al agregar un desencadenador de Lambda en la consola de Amazon Cognito, Amazon Cognito agrega una política basada en recursos a la función que permite al grupo de usuarios invocar la función. Cuando crea un desencadenador de Lambda fuera de la consola de Amazon Cognito, incluida una función entre cuentas, debe agregar permisos a la política basada en recursos de la función de Lambda. Los permisos agregados deben permitir a Amazon Cognito invocar la función en nombre del grupo de usuarios. Puede agregar permisos desde la consola de Lambda o usar la operación de la API AddPermission de Lambda.

  Ejemplo de política basada en recursos de Lambda

  En el siguiente ejemplo de política basada en recursos de Lambda otorga a Amazon Cognito una capacidad limitada para invocar una función Lambda. Amazon Cognito solo puede invocar la

función cuando lo hace en nombre del grupo de usuarios en la condición `aws:SourceArn` y en la cuenta en la condición `aws:SourceAccount`.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "lambda-allow-cognito",
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-idp.amazonaws.com"
            },
            "Action": "lambda:InvokeFunction",
            "Resource": "<your Lambda function ARN>",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "<your account number>"
                },
                "ArnLike": {
                    "AWS:SourceArn": "<your user pool ARN>"
                }
            }
        }
    ]
}
```

Adición de un desencadenador de Lambda a un grupo de usuarios

Para agregar un desencadenador de Lambda a un grupo de usuarios con la consola, siga estos pasos:

1. Use la consola de Lambda para crear una función de Lambda. Para obtener más información sobre las funciones de Lambda, consulte la Guía para desarrolladores de AWS Lambda.
2. Diríjase a la consola de Amazon Cognito y luego elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña User pool properties (Propiedades del grupo de usuarios) y localice Lambda triggers (Desencadenadores Lambda).
5. Elija Add a Lambda trigger (Agregar un desencadenador Lambda).

6. Seleccione una Category (Categoría) de desencadenador de Lambda en función de la fase de autenticación que desee personalizar.
7. Seleccione Assign Lambda function (Asignar función Lambda) y seleccione una función en la misma Región de AWS que el grupo de usuarios.

> **Note**
>
> Si las credenciales de AWS Identity and Access Management (IAM) tienen permiso para actualizar la función de Lambda, Amazon Cognito agrega una política basada en recursos de Lambda. Con esta política, Amazon Cognito puede llamar a la función que seleccione. Si las credenciales de sesión iniciada no tienen permisos de IAM suficientes, debe actualizar la política basada en recursos por separado. Para obtener más información, consulte the section called “Consideraciones importantes”.

8. Elija Save changes (Guardar cambios).
9. En la consola de Lambda, puede registrar la función de Lambda con CloudWatch. Para obtener más información, consulte Acceso a CloudWatch Logs para Lambda.

Evento desencadenador de Lambda para un grupo de usuarios

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función de Lambda devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. Con este evento, se muestran los parámetros comunes del desencadenador de Lambda.

JSON

```
{
    "version": "string",
    "triggerSource": "string",
    "region": AWSRegion,
    "userPoolId": "string",
    "userName": "string",
    "callerContext":
        {
            "awsSdkVersion": "string",
            "clientId": "string"
        },
    "request":
        {
```

```
            "userAttributes": {
                "string": "string",
                ....
            }
        },
    "response": {}
}
```

Parámetros comunes del desencadenador de Lambda para un grupo de usuarios

versión

El número de versión de la función de Lambda.

triggerSource

El nombre del evento que desencadenó la función de Lambda. Consulte Conexión de disparadores de Lambda a las operaciones funcionales del grupo de usuarios para ver una descripción del origen de cada disparador (triggerSource).

región

Región de AWS como instancia `AWSRegion`.

userPoolId

El ID del grupo de usuarios.

userName

El nombre de usuario del usuario actual.

callerContext

Metadatos sobre la solicitud y el entorno del código. Contiene los campos awsSdkVersion y clientId.

awsSdkVersion

La versión del SDK de AWS que generó la solicitud.

clientId

El ID de cliente de la aplicación del grupo de usuarios.

request

Detalles de la solicitud de API de su usuario. Incluye los siguientes campos y cualquier parámetro de solicitud que sea específico del disparador. Por ejemplo, un evento que Amazon Cognito envía a un desencadenador de autenticación previa también contendrá un parámetro `userNotFound`. Puede procesar el valor de este parámetro para realizar una acción personalizada cuando el usuario intente iniciar sesión con un nombre de usuario no registrado.

userAttributes

Uno o varios pares de clave-valor de nombres y valores de atributos de usuario, por ejemplo `"email": "john@example.com"`.

respuesta

Este parámetro no contiene ninguna información en la solicitud original. La función de Lambda debe devolver el evento completo a Amazon Cognito y añadir los parámetros de devolución a `response`. Para ver qué parámetros de devolución puede incluir la función, consulte la documentación del disparador que desee utilizar.

Conexión de las operaciones de la API a los disparadores de Lambda

En las siguientes secciones, se describen los disparadores de Lambda a los que invoca Amazon Cognito a partir de la actividad de su grupo de usuarios.

Cuando la aplicación inicia la sesión de los usuarios a través de la API de los grupos de usuarios, la interfaz de usuario alojada o los puntos de conexión de grupo de usuarios de Amazon Cognito, Amazon Cognito invoca las funciones de Lambda en función del contexto de la sesión. Para obtener más información sobre la API de los grupos de usuarios de Amazon Cognito y los puntos de conexión del grupo de usuarios, consulte Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios. En las tablas de las secciones siguientes, se describen los eventos que hacen que Amazon Cognito invoque una función y la cadena `triggerSource` que Amazon Cognito incluye en la solicitud.

Temas

- Disparadores de Lambda en la API de Amazon Cognito
- Se activa Lambda para los usuarios locales de Amazon Cognito en la interfaz de usuario alojada
- Desencadenadores de Lambda para usuarios federados

Disparadores de Lambda en la API de Amazon Cognito

En la siguiente tabla, se describen las cadenas de origen de los disparadores de Lambda que Amazon Cognito puede invocar cuando la aplicación crea, inicia sesión o actualiza a un usuario local.

Orígenes de desencadenadores de usuarios locales en la API de Amazon Cognito

Operación de la API	Disparador de Lambda	Origen del disparador
AdminCreateUser	Anterior a la inscripción	`PreSignUp_AdminCreateUser`
	Anterior a la generación del token	`TokenGeneration_NewPasswordChallenge`
	Mensaje personalizado	`CustomMessage_AdminCreateUser`
	Remitente de correo electrónico personalizado	`CustomEmailSender_AdminCreateUser`
	Remitente de SMS personalizado	`CustomSMSSender_AdminCreateUser`
SignUp	Anterior a la inscripción	`PreSignUp_SignUp`
	Mensaje personalizado	`CustomMessage_SignUp`
	Remitente de correo electrónico personalizado	`CustomEmailSender_SignUp`
	Remitente de SMS personalizado	`CustomSMSSender_SignUp`
ConfirmSignUp AdminConfirmSignUp	Posterior a la confirmación	`PostConfirmation_ConfirmSignUp`
InitiateAuth AdminInitiateAuth	Anterior a la autenticación	`PreAuthentication_Authentication`

Operación de la API	Disparador de Lambda	Origen del disparador
	Definición de desafíos de autenticación	`DefineAuthChallenge_Authentication`
	Creación de desafíos de autenticación	`CreateAuthChallenge_Authentication`

	Anterior a la generación del token	`TokenGeneration_Authentication`
		`TokenGeneration_AuthenticateDevice`
		`TokenGeneration_RefreshTokens`

	Migración de usuarios	`UserMigration_Authentication`
	Mensaje personalizado	`CustomMessage_Authentication`
	Remitente de correo electrónico personalizado	`CustomEmailSender_AccountTakeOverNotification`
	Remitente de SMS personalizado	`CustomSMSSender_Authentication`
ForgotPassword	Migración de usuarios	`UserMigration_ForgotPassword`
	Mensaje personalizado	`CustomMessage_ForgotPassword`
	Remitente de correo electrónico personalizado	`CustomEmailSender_ForgotPassword`

Operación de la API	Disparador de Lambda	Origen del disparador
	Remitente de SMS personalizado	`CustomSMSSender_ForgotPassword`
ConfirmForgotPassword	Posterior a la confirmación	`PostConfirmation_ConfirmForgotPassword`
UpdateUserAttributes AdminUpdateUserAttributes	Mensaje personalizado	`CustomMessage_UpdateUserAttribute`
	Remitente de correo electrónico personalizado	`CustomEmailSender_UpdateUserAttribute`
	Remitente de SMS personalizado	`CustomSMSSender_UpdateUserAttribute`
VerifyUserAttributes	Mensaje personalizado	`CustomMessage_VerifyUserAttribute`
	Remitente de correo electrónico personalizado	`CustomEmailSender_VerifyUserAttribute`
	Remitente de SMS personalizado	`CustomSMSSender_VerifyUserAttribute`

Se activa Lambda para los usuarios locales de Amazon Cognito en la interfaz de usuario alojada

En la siguiente tabla, se describen las cadenas de origen de los disparadores de Lambda que Amazon Cognito puede invocar cuando un usuario local inicia sesión en el grupo de usuarios con la interfaz de usuario alojada.

Orígenes de los desencadenadores de usuarios locales en la interfaz de usuario alojada

URI de UI alojada	Disparador de Lambda	Origen del disparador
`/signup`	Anterior a la inscripción	`PreSignUp_SignUp`

URI de UI alojada	Disparador de Lambda	Origen del disparador
	Mensaje personalizado	`CustomMessage_SignUp`
	Remitente de correo electrónico personalizado	`CustomEmailSender_SignUp`
	Remitente de SMS personalizado	`CustomSMSSender_SignUp`
`/confirmuser`	Posterior a la confirmación	`PostConfirmation_ConfirmSignUp`
`/login`	Anterior a la autenticación	`PreAuthentication_Authentication`
	Definición de desafíos de autenticación	`DefineAuthChallenge_Authentication`
	Creación de desafíos de autenticación	`CreateAuthChallenge_Authentication`

	Anterior a la generación del token	`TokenGeneration_Authentication`
		`TokenGeneration_AuthenticateDevice`
		`TokenGeneration_RefreshTokens`

	Migración de usuarios	`UserMigration_Authentication`
	Mensaje personalizado	`CustomMessage_Authentication`
	Remitente de correo electrónico personalizado	`CustomEmailSender_AccountTakeOverNotification`

URI de UI alojada	Disparador de Lambda	Origen del disparador
	Remitente de SMS personalizado	`CustomSMSSender_Authentication`
`/forgotpassword`	Migración de usuarios	`UserMigration_ForgotPassword`
	Mensaje personalizado	`CustomMessage_ForgotPassword`
	Remitente de correo electrónico personalizado	`CustomEmailSender_ForgotPassword`
	Remitente de SMS personalizado	`CustomSMSSender_ForgotPassword`
`/confirmforgotpassword`	Posterior a la confirmación	`PostConfirmation_ConfirmForgotPassword`

Desencadenadores de Lambda para usuarios federados

Puede utilizar los siguientes desencadenadores de Lambda para personalizar los flujos de trabajo del grupo de usuarios para los usuarios que inician sesión con un proveedor federado.

> **Note**
>
> Los usuarios federados pueden usar la UI alojada en Amazon Cognito para iniciar sesión o pueden generar una solicitud a Autorizar punto de conexión que los redirija de forma silenciosa a la página de inicio de sesión de su proveedor de identidad. No puede iniciar la sesión de usuarios federados con la API de grupos de usuarios de Amazon Cognito.

Orígenes de los desencadenadores de usuarios federados

Evento de inicio de sesión	Disparador de Lambda	Origen del disparador
Primer inicio de sesión	Anterior a la inscripción	`PreSignUp_ExternalProvider`
	Posterior a la confirmación	`PostConfirmation_ConfirmSignUp`
	Anterior a la generación del token	`TokenGeneration_HostedAuth`
Inicios de sesión posteriores	Anterior a la autenticación	`PreAuthentication_Authentication`
	Posterior a la autenticación	`PostAuthentication_Authentication`
	Anterior a la generación del token	`TokenGeneration_HostedAuth`

El inicio de sesión federado no llama a ningún Desencadenadores de Lambda de desafío de autenticación personalizado, Migración del desencadenador de Lambda del usuario, Desencadenador de Lambda para mensajes personalizados o Desencadenadores de Lambda para remitentes personalizados en el grupo de usuarios.

Conexión de disparadores de Lambda a las operaciones funcionales del grupo de usuarios

Cada disparador de Lambda cumple un rol funcional en su grupo de usuarios. Por ejemplo, un disparador puede modificar su flujo de registro o añadir un desafío de autenticación personalizado. El evento que Amazon Cognito envía a una función de Lambda puede reflejar una de las múltiples acciones que componen ese rol funcional. Por ejemplo, Amazon Cognito invoca un disparador previo al registro cuando el usuario se registra y cuando crea un usuario. Cada uno de estos casos para el mismo rol funcional tiene su propio valor de `triggerSource`. La función de Lambda puede procesar los eventos entrantes de forma diferente según la operación que la haya invocado.

Amazon Cognito también invoca todas las funciones asignadas cuando un evento se corresponde con el origen de un disparador. Por ejemplo, cuando un usuario inicia sesión en un grupo de usuarios al que ha asignado los disparadores de migración de usuario y autenticación previa, activa ambos.

Disparadores de inscripción, confirmación e inicio de sesión (autenticación)

Desencadenador	Valor de triggerSource	Evento
Anterior a la inscripción	`PreSignUp_SignUp`	Anterior a la inscripción.
Anterior a la inscripción	`PreSignUp_AdminCreateUser`	Anterior a la inscripción cuando un administrador crea un nuevo usuario.
Anterior a la inscripción	`PreSignUp_ExternalProvider`	Prerregistro para proveedores de identidad externos.
Posterior a la confirmación	`PostConfirmation_ConfirmSignUp`	Posterior a la confirmación de la inscripción.
Posterior a la confirmación	`PostConfirmation_ConfirmForgotPassword`	Posterior a la confirmación de la contraseña olvidada.
Anterior a la autenticación	`PreAuthentication_Authentication`	Anterior a la autenticación.
Posterior a la autenticación	`PostAuthentication_Authentication`	Posterior a la autenticación.

Disparadores de desafío de autenticación personalizados

Desencadenador	Valor de triggerSource	Evento
Definición de desafíos de autenticación	`DefineAuthChallenge_Authentication`	Definición de desafíos de autenticación.
Creación de desafíos de autenticación	`CreateAuthChallenge_Authentication`	Creación de desafíos de autenticación.

Desencadenador	Valor de triggerSource	Evento
Verificación de desafío de autenticación	`VerifyAuthChallengeResponse_Authentication`	Verificación de la respuesta a los desafíos de autenticación.

Disparadores anteriores a la generación del token

Desencadenador	Valor de triggerSource	Evento
Anterior a la generación del token	`TokenGeneration_HostedAuth`	Amazon Cognito autentica el usuario desde su página de inicio de sesión de la IU alojada.
Anterior a la generación del token	`TokenGeneration_Authentication`	Flujos de autenticación de usuarios completos.
Anterior a la generación del token	`TokenGeneration_NewPasswordChallenge`	El administrador crea el usuario. Amazon Cognito lo llama cuando el usuario debe cambiar una contraseña temporal.
Anterior a la generación del token	`TokenGeneration_AuthenticateDevice`	Fin de la autenticación de un dispositivo de usuario.
Anterior a la generación del token	`TokenGeneration_RefreshTokens`	Un usuario intenta actualizar los tokens de identidad y acceso.

Disparadores de migración de usuarios

Desencadenador	Valor de triggerSource	Evento
Migración de usuario	`UserMigration_Authentication`	Migración de usuarios durante el inicio de sesión.

Desencadenador	Valor de triggerSource	Evento
Migración de usuario	`UserMigration_ForgotPassword`	Migración de usuarios durante el flujo de recuperación de contraseñas olvidadas.

Disparadores de mensaje personalizado

Desencadenador	Valor de triggerSource	Evento
Mensaje personalizado	`CustomMessage_SignUp`	Mensaje personalizado cuando un usuario se registra en el grupo de usuarios.
Mensaje personalizado	`CustomMessage_AdminCreateUser`	Mensaje personalizado al crear un usuario como administrador y Amazon Cognito le envía una contraseña temporal.
Mensaje personalizado	`CustomMessage_ResendCode`	Mensaje personalizado cuando el usuario actual solicita un nuevo código de confirmación.
Mensaje personalizado	`CustomMessage_ForgotPassword`	Mensaje personalizado cuando el usuario solicita un restablecimiento de contraseña.
Mensaje personalizado	`CustomMessage_UpdateUserAttribute`	Mensaje personalizado cuando un usuario cambia su dirección de correo electrónico o número de teléfono y Amazon Cognito envía un código de verificación.

Desencadenador	Valor de triggerSource	Evento
Mensaje personalizado	`CustomMessage_VerifyUserAttribute`	Mensaje personalizado cuando un usuario agrega una dirección de correo electrónico o un número de teléfono y Amazon Cognito envía un código de verificación.
Mensaje personalizado	`CustomMessage_Authentication`	Mensaje personalizado cuando un usuario que ha configurado la MFA por SMS inicia sesión.

Desencadenador de Lambda de prerregistro.

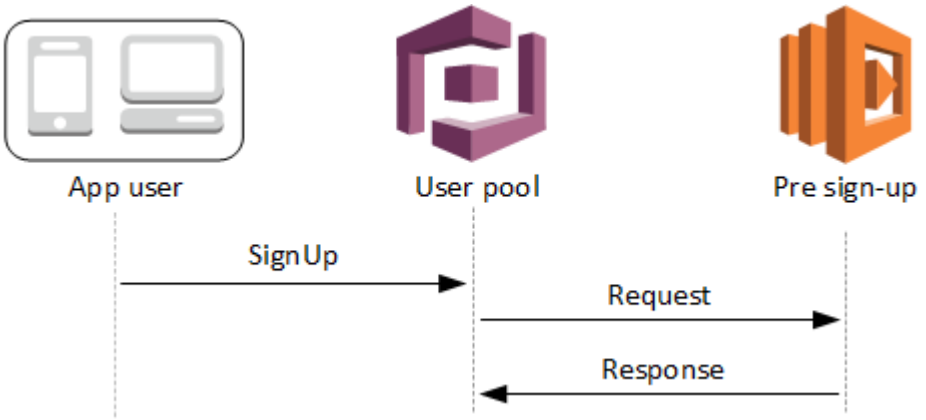
Poco antes de que Amazon Cognito registre un nuevo usuario, activa el la función AWS Lambda de registro previo. Como parte del proceso de registro puede utilizar esta función para hacer una validación personalizada y, en función de los resultados de esta, aceptar o denegar la solicitud de registro.

Temas

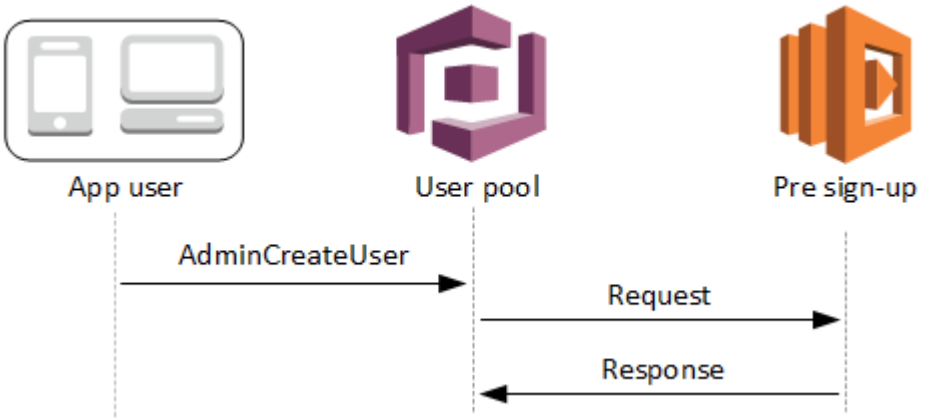
- Flujos de Lambda de prerregistro.
- Parámetros del desencadenador de Lambda de prerregistro
- Tutoriales sobre inscripción
- Ejemplo anterior a la inscripción: Confirmación automática de los usuarios de un dominio registrado
- Ejemplo de invocación anterior a la inscripción: Confirmación y verificación automáticas de todos los usuarios
- Ejemplo de antes de registrarse: denegar el registro si el nombre de usuario tiene menos de cinco caracteres

Flujos de Lambda de prerregistro.

Flujo de inscripción del cliente.



Flujo de inscripción del servidor



La solicitud contiene datos de validación del cliente. Estos datos provienen de los `ValidationData` valores que se pasan al grupo de usuarios SignUp y a los métodos de la AdminCreateUser API.

Parámetros del desencadenador de Lambda de prerregistro

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
```

```
            . . .
        },
        "validationData": {
            "string": "string",
            . . .
         },
        "clientMetadata": {
            "string": "string",
            . . .
         }
    },

    "response": {
        "autoConfirmUser": "boolean",
        "autoVerifyPhone": "boolean",
        "autoVerifyEmail": "boolean"
    }
}
```

Parámetros de la solicitud anteriores a la inscripción

userAttributes

Uno o varios pares de nombre y valor que representan atributos de usuario. Los nombres de atributo son las claves.

validationData

Uno o varios pares clave-valor con datos de atributos de usuario que su aplicación pasó a Amazon Cognito en la solicitud de creación de un nuevo usuario. Envíe esta información a su función Lambda en el ValidationData parámetro de su solicitud AdminCreateUsero de SignUpAPI.

Amazon Cognito no establece sus ValidationData datos como atributos del usuario que cree. ValidationData es información de usuario temporal que usted proporciona para su activador Lambda previo al registro.

clientMetadata

Uno o varios pares clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda que especifica para el desencadenador de prerregistro. Puede pasar estos datos a la función Lambda mediante el ClientMetadata parámetro de las siguientes acciones de la API: AdminCreateUser, AdminRespondToAuthChallengeForgotPassword, y. SignUp

Parámetros de la respuesta anterior a la inscripción

En la respuesta, puede establecer `autoConfirmUser` en `true` si desea confirmar automáticamente al usuario. Puede establecer `autoVerifyEmail` en `true` para verificar automáticamente el correo electrónico del usuario. Puede establecer `autoVerifyPhone` en `true` para verificar automáticamente el número de teléfono del usuario.

> **Note**
>
> Amazon Cognito ignora los parámetros de respuesta `autoVerifyPhone`, `autoVerifyEmail` y `autoConfirmUser` cuando la API `AdminCreateUser` desencadena la función de Lambda de registro previo.

autoConfirmUser

Establezca este parámetro en `true` para confirmar automáticamente al usuario, o en `false` en caso contrario.

autoVerifyEmail

Si se establece en `true`, se verifica la dirección de correo electrónico de un usuario registrado; en caso contrario, `false`. Si `autoVerifyEmail` está establecido en `true`, el atributo `email` debe ser un valor válido distinto de null. De lo contrario, se producirá un error y el usuario no podrá completar la inscripción.

Si el atributo `email` se selecciona como un alias, se creará un alias de la dirección de correo electrónico del usuario cuando se establezca `autoVerifyEmail`. Si ya existe un alias con esa dirección de correo electrónico, el alias se moverá al usuario nuevo y la dirección de correo electrónico del usuario anterior se marcará como no verificada. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

autoVerifyPhone

Si se establece en `true`, se verifica el número de teléfono de un usuario registrado; en caso contrario, `false`. Si `autoVerifyPhone` está establecido en `true`, el atributo `phone_number` debe ser un valor válido distinto de null. De lo contrario, se producirá un error y el usuario no podrá completar la inscripción.

Si el atributo `phone_number` se selecciona como un alias, se creará un alias de número de teléfono del usuario cuando se establezca `autoVerifyPhone`. Si ya existe un alias con ese

número de teléfono, el alias se moverá al número de teléfono del usuario nuevo y anterior y se marcará como no verificado. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

Tutoriales sobre inscripción

La función de Lambda de prerregistro se desencadena antes del registro del usuario. Consulte estos tutoriales de registro de Amazon Cognito para Android JavaScript e iOS.

Plataforma	Tutorial
JavaScript SDK de identidad	Registra a los usuarios con JavaScript
SDK de identidad para Android	Inscripción de usuarios con Android
SDK de identidad para iOS	Inscripción de usuarios con iOS

Ejemplo anterior a la inscripción: Confirmación automática de los usuarios de un dominio registrado

Puede usar el desencadenador de Lambda de prerregistro para agregar lógica personalizada que valide el registro de usuarios nuevos en un grupo de usuarios. Este es un JavaScript programa de ejemplo que muestra cómo registrar un nuevo usuario. Este invoca a un desencadenador de Lambda de prerregistro como parte de la autenticación.

JavaScript

```
var attributeList = [];
var dataEmail = {
  Name: "email",
  Value: "...", // your email here
};
var dataPhoneNumber = {
  Name: "phone_number",
  Value: "...", // your phone number here with +country code and no delimiters in
 front
```

```
};

var dataEmailDomain = {
  Name: "custom:domain",
  Value: "example.com",
};
var attributeEmail = new AmazonCognitoIdentity.CognitoUserAttribute(dataEmail);
var attributePhoneNumber = new AmazonCognitoIdentity.CognitoUserAttribute(
  dataPhoneNumber
);
var attributeEmailDomain = new AmazonCognitoIdentity.CognitoUserAttribute(
  dataEmailDomain
);

attributeList.push(attributeEmail);
attributeList.push(attributePhoneNumber);
attributeList.push(attributeEmailDomain);

var cognitoUser;
userPool.signUp(
  "username",
  "password",
  attributeList,
  null,
  function (err, result) {
    if (err) {
      alert(err);
      return;
    }
    cognitoUser = result.user;
    console.log("user name is " + cognitoUser.getUsername());
  }
);
```

Se trata de un desencadenador de Lambda de ejemplo al que se llama en el momento previo al registro en el grupo de usuarios mediante el desencadenador de Lambda de prerregistro. Utiliza un atributo personalizado, custom:domain, para confirmar automáticamente a los usuarios nuevos de un determinado dominio de correo electrónico. Los usuarios nuevos que no pertenezcan al dominio personalizado se añadirán al grupo de usuarios, pero no se confirmarán automáticamente.

Node.js

```
exports.handler = (event, context, callback) => {
  // Set the user pool autoConfirmUser flag after validating the email domain
  event.response.autoConfirmUser = false;

  // Split the email address so we can compare domains
  var address = event.request.userAttributes.email.split("@");

  // This example uses a custom attribute "custom:domain"
  if (event.request.userAttributes.hasOwnProperty("custom:domain")) {
    if (event.request.userAttributes["custom:domain"] === address[1]) {
      event.response.autoConfirmUser = true;
    }
  }

  // Return to Amazon Cognito
  callback(null, event);
};
```

Python

```
def lambda_handler(event, context):
    # It sets the user pool autoConfirmUser flag after validating the email domain
    event['response']['autoConfirmUser'] = False

    # Split the email address so we can compare domains
    address = event['request']['userAttributes']['email'].split('@')

    # This example uses a custom attribute 'custom:domain'
    if 'custom:domain' in event['request']['userAttributes']:
        if event['request']['userAttributes']['custom:domain'] == address[1]:
            event['response']['autoConfirmUser'] = True

    # Return to Amazon Cognito
    return event
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
    "request": {
        "userAttributes": {
            "email": "testuser@example.com",
            "custom:domain": "example.com"
        }
    },
    "response": {}
}
```

Ejemplo de invocación anterior a la inscripción: Confirmación y verificación automáticas de todos los usuarios

En este ejemplo se confirman todos los usuarios y se establece la verificación de los atributos `email` y `phone_number` de cada usuario si se especifican. Además, si están habilitados los alias, se crearán alias automáticamente para `phone_number` y `email` cuando esté habilitada la verificación automática.

> **Note**
>
> Si ya existe un alias con el mismo número de teléfono, el alias se moverá al número de teléfono del usuario nuevo y el atributo `phone_number` del usuario anterior se marcará como no verificado. Lo mismo sucede con las direcciones de correo electrónico. Para evitar que esto suceda, puedes usar la ListUsers API de grupos de usuarios para ver si hay un usuario existente que ya esté usando el número de teléfono o la dirección de correo electrónico del nuevo usuario como alias.

Node.js

```
const handler = async (event) => {
  // Confirm the user
  event.response.autoConfirmUser = true;
  // Set the email as verified if it is in the request
  if (event.request.userAttributes.hasOwnProperty("email")) {
    event.response.autoVerifyEmail = true;
  }
```

```
  // Set the phone number as verified if it is in the request
  if (event.request.userAttributes.hasOwnProperty("phone_number")) {
    event.response.autoVerifyPhone = true;
  }

  return event;
};

export { handler };
```

Python

```
def lambda_handler(event, context):
    # Confirm the user
    event['response']['autoConfirmUser'] = True

    # Set the email as verified if it is in the request
    if 'email' in event['request']['userAttributes']:
        event['response']['autoVerifyEmail'] = True

    # Set the phone number as verified if it is in the request
    if 'phone_number' in event['request']['userAttributes']:
        event['response']['autoVerifyPhone'] = True

    # Return to Amazon Cognito
    return event
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
  "request": {
    "userAttributes": {
      "email": "user@example.com",
      "phone_number": "+12065550100"
    }
```

```
  },
  "response": {}
}
```

Ejemplo de antes de registrarse: denegar el registro si el nombre de usuario tiene menos de cinco caracteres

En este ejemplo se comprueba la longitud del nombre de usuario de una solicitud de registro. El ejemplo devuelve un error si el usuario ha ingresado un nombre de menos de cinco caracteres de longitud.

Node.js

```
exports.handler = (event, context, callback) => {
    // Impose a condition that the minimum length of the username is 5 is imposed on
 all user pools.
    if (event.userName.length < 5) {
        var error = new Error("Cannot register users with username less than the
 minimum length of 5");
        // Return error to Amazon Cognito
        callback(error, event);
    }
    // Return to Amazon Cognito
    callback(null, event);
};
```

Python

```
def lambda_handler(event, context):
    if len(event['userName']) < 5:
        raise Exception("Cannot register users with username less than the minimum
 length of 5")
    # Return to Amazon Cognito
    return event
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
  "userName": "rroe",
  "response": {}
}
```

Desencadenador de Lambda de posconfirmación.

Amazon Cognito invoca este desencadenador después de que un usuario registrado confirme su cuenta de usuario. En la función de Lambda posterior a la confirmación, puede enviar mensajes personalizados o agregar solicitudes de API personalizadas. Por ejemplo, puede consultar un sistema externo y rellenar atributos adicionales para el usuario. Amazon Cognito invoca este desencadenador solo para los usuarios que se registran en el grupo de usuarios, no para las cuentas de usuario que crea con las credenciales de administrador.

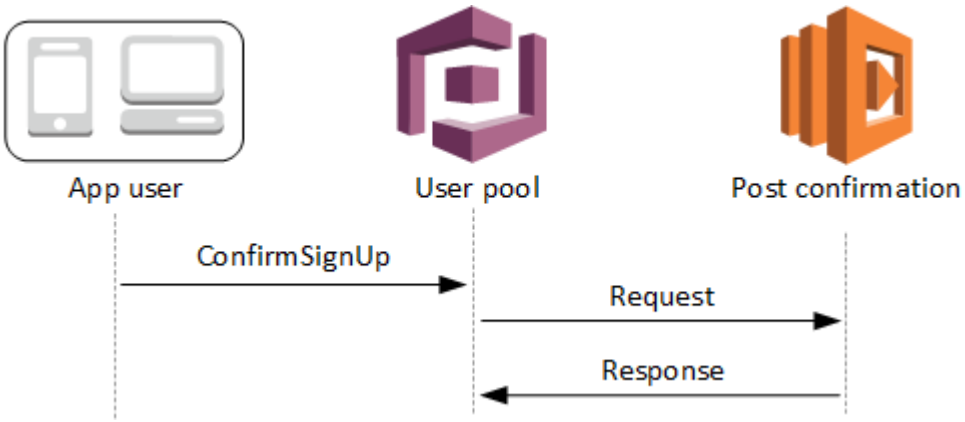
La solicitud contiene los atributos actuales del usuario confirmado.

Temas

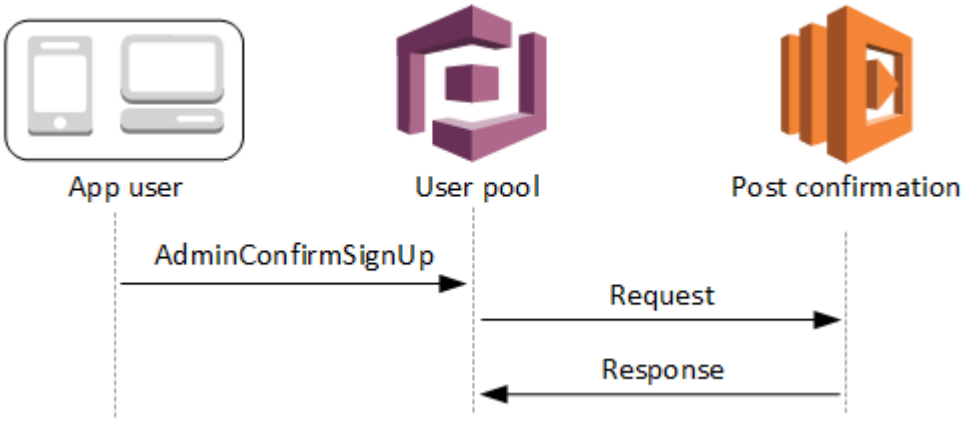
- Flujos de Lambda de posconfirmación
- Parámetros del desencadenador de Lambda de posconfirmación
- Tutoriales de confirmación del usuario
- Ejemplo de invocación posterior a la confirmación

Flujos de Lambda de posconfirmación

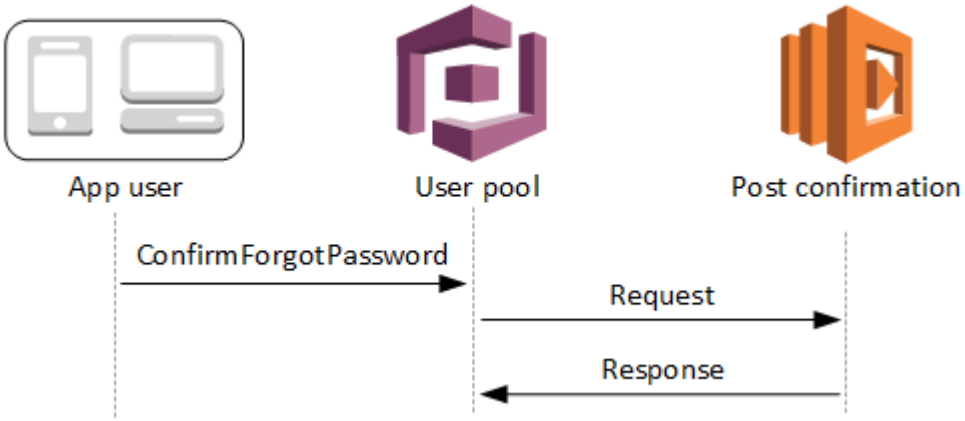
Flujo de confirmación de inscripción del cliente



Flujo de confirmación de inscripción del servidor.



Flujo de confirmación de contraseña olvidada



Parámetros del desencadenador de Lambda de posconfirmación

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
            "userAttributes": {
                "string": "string",
                . . .
            },
            "clientMetadata": {
             "string": "string",
             . . .
```

```
            }
        },
    "response": {}
}
```

Parámetros de solicitud posterior a la confirmación

userAttributes

Uno o varios pares de clave-valor que representan atributos de usuario.

clientMetadata

Uno o varios pares de clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda que especifique para el desencadenador de posconfirmación. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata de las siguientes acciones de la API: AdminConfirmSignUp, ConfirmForgotPassword, ConfirmSignUp y SignUp.

Parámetros de la respuesta posterior a la confirmación

No se espera que la respuesta contenga información adicional.

Tutoriales de confirmación del usuario

La función de Lambda de posconfirmación se desencadena de inmediato en el momento posterior a que Amazon Cognito confirme un nuevo usuario. Consulte estos tutoriales de confirmación de usuario en JavaScript, Android e iOS.

Plataforma	Tutorial
SDK de identidad para JavaScript	Confirmación de usuarios con JavaScript
SDK de identidad para Android	Confirmación de usuarios con Android
SDK de identidad para iOS	Confirmación de usuarios con iOS

Ejemplo de invocación posterior a la confirmación

Mediante este función de Lambda de ejemplo, se envía un mensaje de correo electrónico de confirmación al usuario con Amazon SES. Para obtener más información, consulte la Guía para desarrolladores de Amazon Simple Email Service.

Node.js

```
// Import required AWS SDK clients and commands for Node.js. Note that this requires
// the `@aws-sdk/client-ses` module to be either bundled with this code or included
// as a Lambda layer.
import { SES, SendEmailCommand } from "@aws-sdk/client-ses";
const ses = new SES();

const handler = async (event) => {
  if (event.request.userAttributes.email) {
    await sendTheEmail(
      event.request.userAttributes.email,
      `Congratulations ${event.userName}, you have been confirmed.`
    );
  }
  return event;
};

const sendTheEmail = async (to, body) => {
  const eParams = {
    Destination: {
      ToAddresses: [to],
    },
    Message: {
      Body: {
        Text: {
          Data: body,
        },
      },
      Subject: {
        Data: "Cognito Identity Provider registration completed",
      },
    },
    // Replace source_email with your SES validated email address
    Source: "<source_email>",
  };
  try {
```

```
    await ses.send(new SendEmailCommand(eParams));
  } catch (err) {
    console.log(err);
  }
};

export { handler };
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
    "request": {
        "userAttributes": {
            "email": "user@example.com",
            "email_verified": true
        }
    },
    "response": {}
}
```

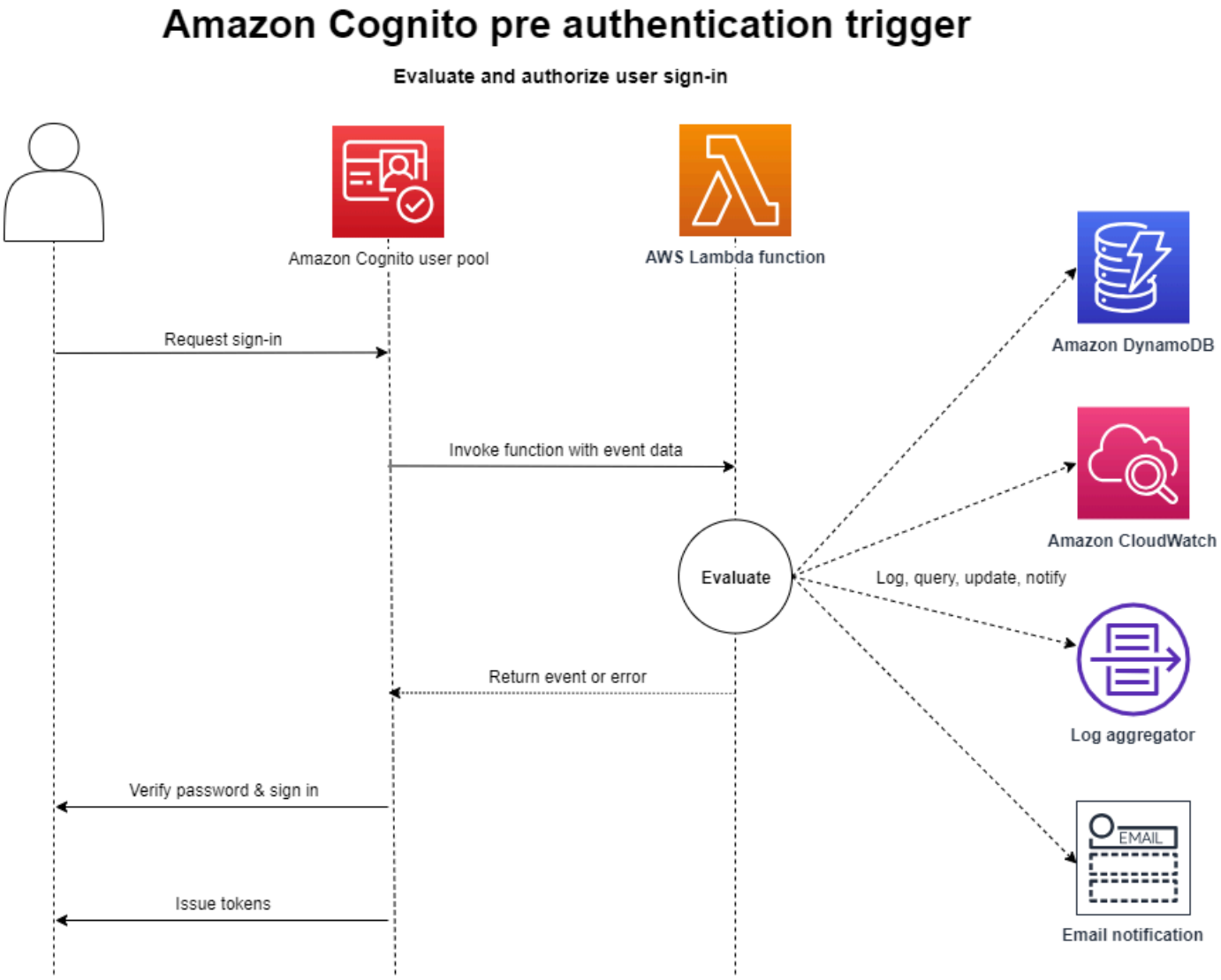
Desencadenador de Lambda anterior a la autenticación

Amazon Cognito invoca a este desencadenador cuando un usuario intenta iniciar sesión, lo que le permite crear una validación personalizada que realiza acciones preparatorias. Por ejemplo, puede denegar la solicitud de autenticación o registrar los datos de sesión en un sistema externo.

> **Note**
>
> Este desencadenador de Lambda no se activa cuando un usuario no existe o ya tiene una sesión en el grupo de usuarios. Si la configuración `PreventUserExistenceErrors` de un cliente de aplicaciones de grupo de usuarios está establecida a `ENABLED`, se activará el desencadenador de Lambda.

Temas

Información general sobre el flujo de autenticación



La solicitud contiene datos de validación del cliente de los valores `ClientMetadata` transferidos por la aplicación a las operaciones de la API `InitiateAuth` y `AdminInitiateAuth` del grupo de usuarios.

Para obtener más información, consulte Flujo de autenticación de los grupos de usuarios.

Parámetros del desencadenador de Lambda de preautenticación

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "validationData": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {}
}
```

Parámetros de la solicitud anterior a la autenticación

userAttributes

Uno o varios pares de nombre-valor que representan atributos de usuario.

userNotFound

Al establecer `PreventUserExistenceErrors` en `ENABLED` para el cliente del grupo de usuarios, Amazon Cognito rellena este booleano.

validationData

Uno o varios pares de clave-valor que contienen los datos de validación de la solicitud de inicio de sesión del usuario. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata en las acciones de la API InitiateAuth y AdminInitiateAuth.

Parámetros de la respuesta anterior a la autenticación

Amazon Cognito no espera ninguna información de devolución adicional en la respuesta. La función puede devolver un error para rechazar el intento de inicio de sesión o utilizar operaciones de la API para consultar y modificar los recursos.

Ejemplo invocación anterior a la autenticación

Esta función de ejemplo impide que los usuarios inicien sesión en el grupo de usuarios con un cliente de aplicación específico. Como la función de Lambda de autenticación previa no se invoca cuando el usuario tiene una sesión existente, esta función solo impide sesiones nuevas con el ID de cliente de la aplicación que desea bloquear.

Node.js

```
const handler = async (event) => {
  if (
    event.callerContext.clientId === "user-pool-app-client-id-to-be-blocked"
  ) {
    throw new Error("Cannot authenticate users from this user pool app client");
  }

  return event;
};

export { handler };
```

Python

```
def lambda_handler(event, context):
    if event['callerContext']['clientId'] == "<user pool app client id to be
 blocked>":
        raise Exception("Cannot authenticate users from this user pool app client")

    # Return to Amazon Cognito
    return event
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
    "callerContext": {
        "clientId": "<user pool app client id to be blocked>"
    },
    "response": {}
}
```

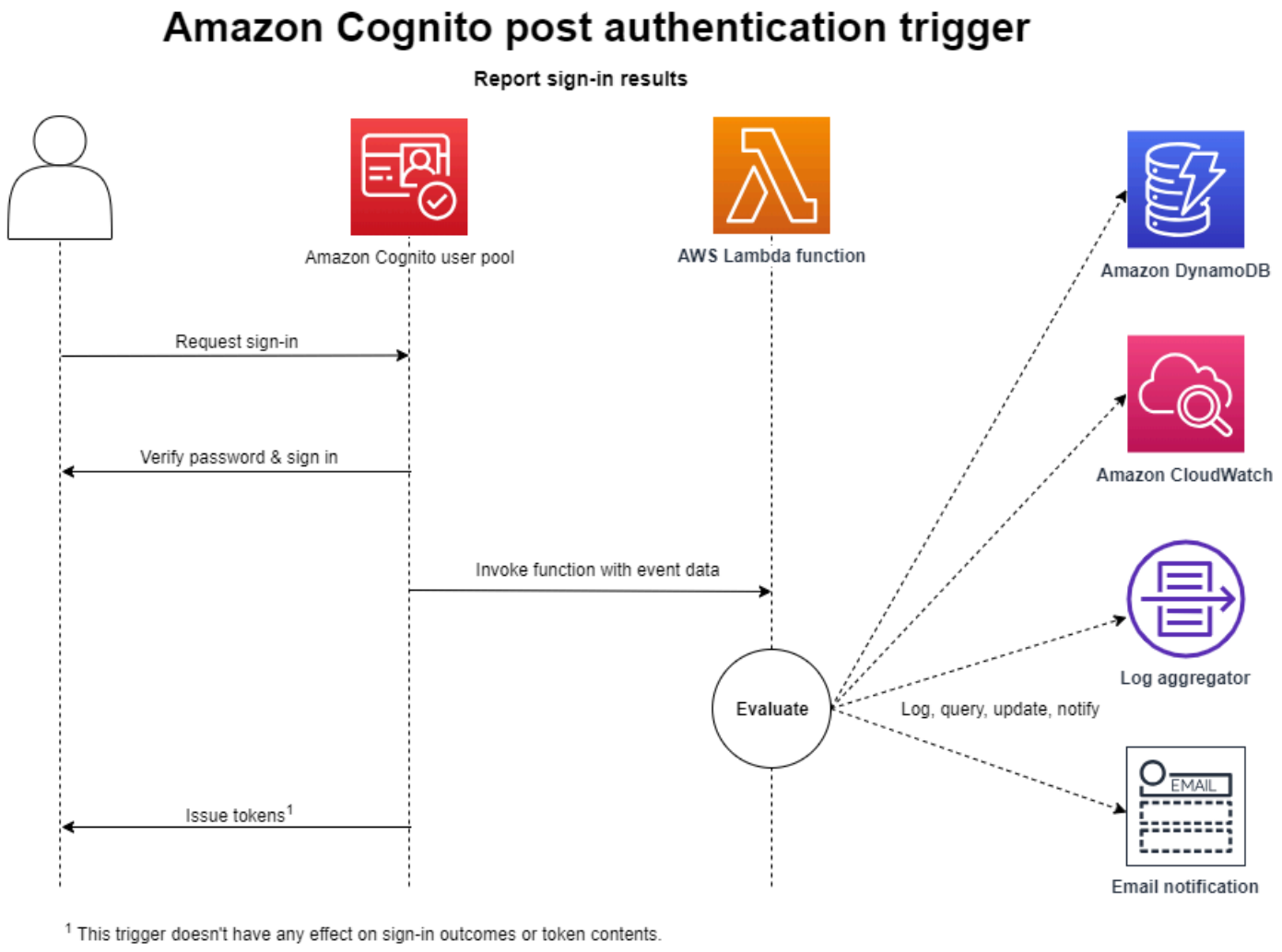
Desencadenador de Lambda posterior a la autenticación.

Amazon Cognito invoca este desencadenador después de que un usuario inicie sesión, lo que le permite agregar lógica personalizada después de que Amazon Cognito autentique al usuario.

Temas

- Información general sobre el flujo de autenticación
- Parámetros del desencadenador de Lambda de posautenticación
- Tutoriales de autenticación
- Ejemplo de invocación posterior a la autenticación

Información general sobre el flujo de autenticación



Para obtener más información, consulte Flujo de autenticación de los grupos de usuarios.

Parámetros del desencadenador de Lambda de posautenticación

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
```

```
        "userAttributes": {
             "string": "string",
             . . .
         },
         "newDeviceUsed": boolean,
         "clientMetadata": {
             "string": "string",
             . . .
            }
        },
    "response": {}
}
```

Parámetros de la solicitud posterior a la autenticación

newDeviceUsed

Este indicador señala si el usuario ha iniciado sesión en un nuevo dispositivo. Amazon Cognito solo establece esta marca si el valor de los dispositivos recordados del grupo de usuarios es `Always` o `User Opt-In`.

userAttributes

Uno o varios pares de nombre y valor que representan atributos de usuario.

clientMetadata

Uno o varios pares de clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda que especifica para el desencadenador de posautenticación. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata en las acciones de la API AdminRespondToAuthChallenge y RespondToAuthChallenge. Amazon Cognito no incluye los datos transferidos del parámetro ClientMetadata en las operaciones de la API AdminInitiateAuth y InitiateAuth en la solicitud que transfiere a la función de autenticación posterior.

Parámetros de la respuesta posterior a la autenticación

Amazon Cognito no espera ninguna información de devolución adicional en la respuesta. La función puede utilizar operaciones de la API para consultar y modificar los recursos o registrar metadatos de eventos en un sistema externo.

Tutoriales de autenticación

Inmediatamente después de que Amazon Cognito inicie la sesión de un usuario, activa la función de Lambda de autenticación posterior. Consulte estos tutoriales de inicio de sesión en JavaScript, Android e iOS.

Plataforma	Tutorial
SDK de identidad para JavaScript	Inicio de sesión de usuarios con JavaScript
SDK de identidad para Android	Inicio de sesión de usuarios con Android
SDK de identidad para iOS	Inicio de sesión de usuarios con iOS

Ejemplo de invocación posterior a la autenticación

Mediante esta función de Lambda de posautenticación de ejemplo, se envían datos de un inicio de sesión correcto a CloudWatch Logs.

Node.js

```
const handler = async (event) => {
  // Send post authentication data to Amazon CloudWatch logs
  console.log("Authentication successful");
  console.log("Trigger function =", event.triggerSource);
  console.log("User pool = ", event.userPoolId);
  console.log("App client ID = ", event.callerContext.clientId);
  console.log("User ID = ", event.userName);

  return event;
};

export { handler }
```

Python

```
import os
```

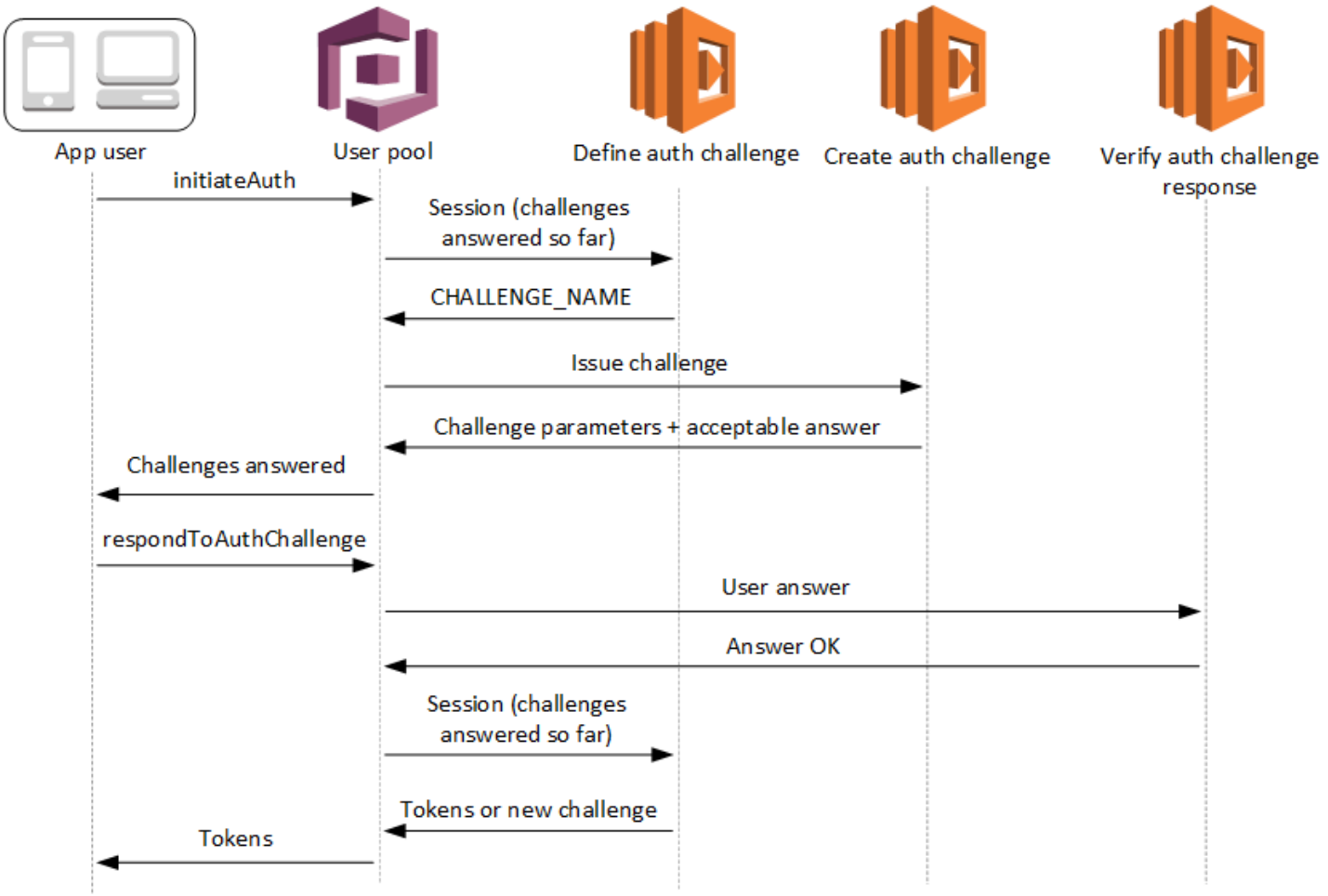
```
def lambda_handler(event, context):

    # Send post authentication data to Cloudwatch logs
    print ("Authentication successful")
    print ("Trigger function =", event['triggerSource'])
    print ("User pool = ", event['userPoolId'])
    print ("App client ID = ", event['callerContext']['clientId'])
    print ("User ID = ", event['userName'])

    # Return to Amazon Cognito
    return event
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
  "triggerSource": "testTrigger",
  "userPoolId": "testPool",
  "userName": "testName",
  "callerContext": {
      "clientId": "12345"
  },
  "response": {}
}
```

Desencadenadores de Lambda de desafío de autenticación personalizado



Estos desencadenadores de Lambda emiten y verifican sus propios desafíos como parte de un flujo de autenticación personalizado para un grupo de usuarios.

Definición de desafíos de autenticación

Amazon Cognito invoca este desencadenador para iniciar el flujo de autenticación personalizado.

Creación de desafíos de autenticación

Amazon Cognito invoca este desencadenador después de Define Auth Challenge (Definir desafío de autenticación) para crear un desafío personalizado.

Verificación de la respuesta a los desafíos de autenticación

Amazon Cognito invoca este desencadenador para verificar si la respuesta del usuario final a un desafío personalizado es válida.

Puede incorporar nuevos tipos de desafío con estos desencadenadores de Lambda desafío. Por ejemplo, estos tipos de desafío podrían incluir CAPTCHA o preguntas de desafío dinámicas.

Puede generalizar la autenticación en dos pasos comunes con los métodos de API InitiateAuth y RespondToAuthChallenge del grupo de usuarios.

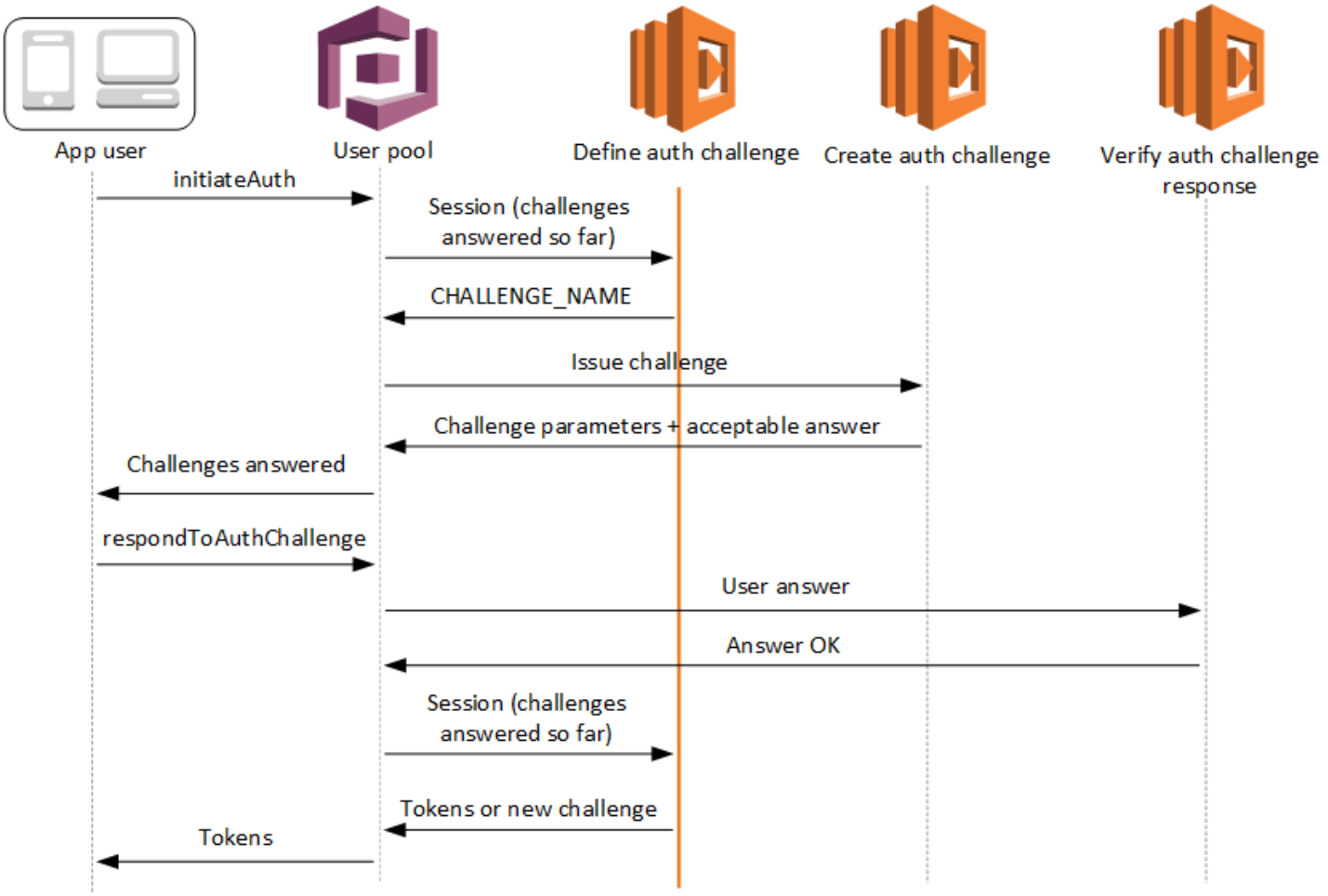
En este flujo, un usuario se autentica respondiendo a desafíos sucesivos hasta que se produce un error de autenticación o se emiten tokens para el usuario. Estas dos llamadas a la API se puede repetir para incluir desafíos distintos.

> **Note**
>
> La interfaz de usuario alojada de Amazon Cognito no admite autenticación personalizada con desencadenadores de Lambda de desafío de autenticación personalizados.

Temas

- Desencadenador de Lambda para definir el desafío de autenticación
- Desencadenador de Lambda para definir el desafío de autenticación
- Desencadenador de Lambda para verificar la respuesta al desafío de autenticación

Desencadenador de Lambda para definir el desafío de autenticación



Definición de desafíos de autenticación

Amazon Cognito invoca este desencadenador para iniciar el flujo de autenticación personalizado.

La solicitud de este desencadenador de Lambda contiene `session`. El parámetro `session` es una matriz que cuenta con todos los desafíos que se presentan al usuario durante el proceso de autenticación actual. La solicitud también incluye el resultado correspondiente. La matriz `session` almacena los detalles del desafío (`ChallengeResult`) en orden cronológico. El desafío `session[0]` representa el primer desafío que recibe el usuario.

Puede hacer que Amazon Cognito verifique las contraseñas de los usuarios antes de que emita los desafíos personalizados. Los desencadenadores de Lambda asociados a la categoría de autenticación de las cuotas de recursos de solicitudes se ejecutarán al realizar la autenticación SRP en un flujo de desafío personalizado. Le presentamos la información general sobre el proceso:

1. La aplicación inicia sesión llamando a `InitiateAuth` o `AdminInitiateAuth` con el mapa `AuthParameters`. Los parámetros deben incluir `CHALLENGE_NAME: SRP_A,` y valores para `SRP_A` y `USERNAME`.
2. Amazon Cognito invoca su desencadenador de Lambda definición de desafío de autenticación con una sesión inicial que contiene `challengeName: SRP_A` y `challengeResult: true`.
3. Después de recibir estos datos de entrada, la función de Lambda responde con `challengeName: PASSWORD_VERIFIER`, `issueTokens: false`, `failAuthentication: false`.
4. Si la verificación de la contraseña se realiza de manera correcta, Amazon Cognito llama a la función de Lambda con una nueva sesión que contiene `challengeName: PASSWORD_VERIFIER` y `challengeResult: true`.
5. Para iniciar los desafíos personalizados, la función de Lambda responde con `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` y `failAuthentication: false`. Si no desea comenzar el flujo de autenticación personalizado con la verificación de la contraseña, puede iniciar sesión con el mapa `AuthParameters`, que incluye `CHALLENGE_NAME: CUSTOM_CHALLENGE`.
6. El bucle de desafíos se repite hasta que todos los desafíos tengan respuesta.

Temas

- Parámetros del desencadenador de Lambda para definir el desafío de autenticación
- Ejemplo de definición de desafíos de autenticación

Parámetros del desencadenador de Lambda para definir el desafío de autenticación

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
                . . .
        },
        "session": [
            ChallengeResult,
```

```
            . . .
        ],
        "clientMetadata": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "challengeName": "string",
        "issueTokens": boolean,
        "failAuthentication": boolean
    }
}
```

Parámetros de la solicitud para definir desafíos de autenticación

Al llamar a la función Lambda, Amazon Cognito proporciona los siguientes parámetros:

userAttributes

Uno o varios pares de nombre-valor que representan atributos de usuario.

userNotFound

Valor booleano que rellena Amazon Cognito cuando `PreventUserExistenceErrors` se establece como `ENABLED` en el cliente del grupo de usuarios. Un valor de `true` significa que el ID de usuario (nombre de usuario, dirección de correo electrónico, etc.) no coincide con ningún usuario existente. Cuando `PreventUserExistenceErrors` se establece en `ENABLED`, el servicio no informa a la aplicación de la inexistencia de usuarios. Recomendamos que las funciones de Lambda mantengan la misma experiencia del usuario y tengan en cuenta la latencia. De esta forma, la persona que realiza la llamada no podrá detectar un comportamiento diferente si el usuario existe o no existe.

sesión

Matriz de `ChallengeResult` elementos. Cada matriz contiene los siguientes elementos:

challengeName

Uno de los siguientes tipos de desafío: `CUSTOM_CHALLENGE`, `SRP_A`, `PASSWORD_VERIFIER`, `SMS_MFA`, `DEVICE_SRP_AUTH`, `DEVICE_PASSWORD_VERIFIER` o bien `ADMIN_NO_SRP_AUTH`.

Cuando la función de desafío de autenticación definida emite un desafío `PASSWORD_VERIFIER` para un usuario que ha configurado la autenticación multifactorial, Amazon Cognito lo sigue con un desafío `SMS_MFA`. En la función, incluya la gestión de los eventos de entrada de los desafíos de `SMS_MFA`. No necesita invocar el desafío de `SMS_MFA` desde la función de desafío de autenticación definida.

> **Important**
>
> Cuando la función determine si un usuario se ha autenticado de forma satisfactoria y deba emitirle tokens, compruebe siempre `challengeName` en la función de desafío de autenticación de definición y si coincide el valor esperado.

challengeResult

Establezca este parámetro en `true` si el usuario ha respondido correctamente al desafío o en `false`, en caso contrario.

challengeMetadata

El nombre del desafío personalizado. Solo se usa si `challengeName` es `CUSTOM_CHALLENGE`.

clientMetadata

Uno o varios pares de clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda que especifica destinada al desencadenador de Lambda para definir el desafío de autenticación. Puede transferir estos datos a la función de Lambda mediante el parámetro `ClientMetadata` de las operaciones de la API AdminRespondToAuthChallenge y RespondToAuthChallenge. La solicitud que llama a la función de definición de desafíos de autenticación no incluye los datos transferidos en el parámetro ClientMetadata en las operaciones de la API AdminInitiateAuth y InitiateAuth.

Parámetros de la respuesta a la definición de desafíos de autenticación

En la respuesta puede devolver la etapa siguiente del proceso de autenticación.

challengeName

Cadena que contiene el nombre del siguiente desafío. Si quiere plantear un nuevo desafío al usuario, especifique aquí el nombre de dicho desafío.

issueTokens

Establezca este parámetro en `true` si cree que el usuario se ha autenticado suficientemente respondiendo a los desafíos. Si el usuario no ha respondido suficientemente a los desafíos, establézcalo en `false`.

failAuthentication

Establezca este parámetro en `true` si desea finalizar el proceso de autenticación en curso. Para continuar el proceso de autenticación actual, establézcalo en `false`.

Ejemplo de definición de desafíos de autenticación

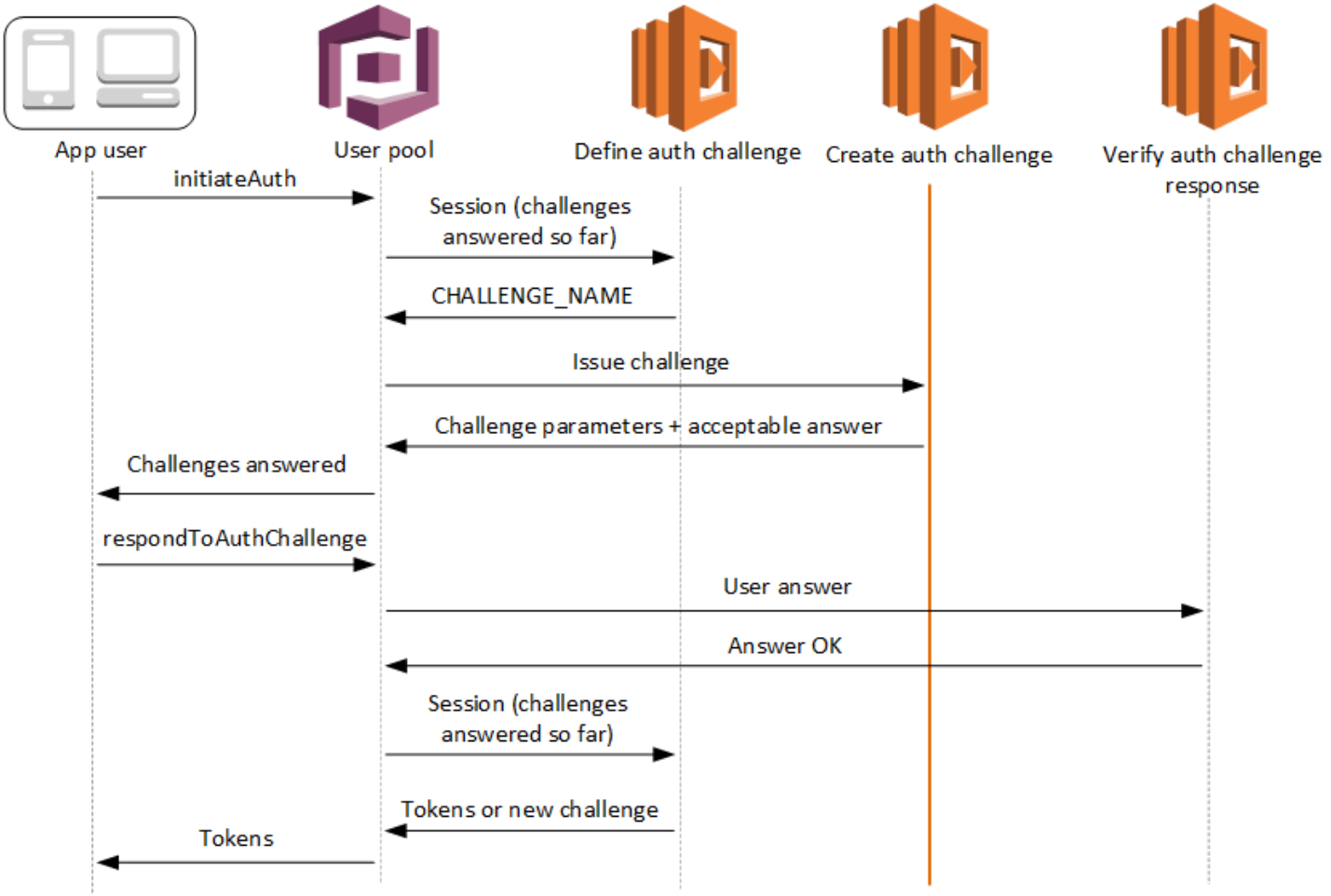
En este ejemplo se definen una serie de desafíos de autenticación y se emiten tokens solo si el usuario ha completado correctamente todos los desafíos.

Node.js

```
const handler = async (event) => {
  if (
    event.request.session.length == 1 &&
    event.request.session[0].challengeName == "SRP_A"
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "PASSWORD_VERIFIER";
  } else if (
    event.request.session.length == 2 &&
    event.request.session[1].challengeName == "PASSWORD_VERIFIER" &&
    event.request.session[1].challengeResult == true
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "CUSTOM_CHALLENGE";
  } else if (
    event.request.session.length == 3 &&
    event.request.session[2].challengeName == "CUSTOM_CHALLENGE" &&
    event.request.session[2].challengeResult == true
  ) {
    event.response.issueTokens = false;
    event.response.failAuthentication = false;
    event.response.challengeName = "CUSTOM_CHALLENGE";
  } else if (
```

```
    event.request.session.length == 4 &&
    event.request.session[3].challengeName == "CUSTOM_CHALLENGE" &&
    event.request.session[3].challengeResult == true
  ) {
    event.response.issueTokens = true;
    event.response.failAuthentication = false;
  } else {
    event.response.issueTokens = false;
    event.response.failAuthentication = true;
  }

  return event;
};

export { handler }
```

Desencadenador de Lambda para definir el desafío de autenticación



Creación de desafíos de autenticación

Amazon Cognito invoca este desencadenador después de Define Aut Challenge (Definir desafío de autenticación) si se ha especificado un desafío personalizado como parte del desencadenador Define Auth Challenge (Definir desafío de autenticación). Crea un flujo de autenticación personalizado.

Este desencadenador de Lambda se invoca para crear un desafío que se presenta al usuario. La solicitud de este desencadenador de Lambda incluye los parámetros `challengeName` y `session`. `challengeName` es una cadena y es el nombre del siguiente desafío al usuario. El valor de este atributo se establece en el desencadenador de Lambda para definir el desafío de autenticación.

El bucle de desafíos se repetirá hasta que todos los desafíos tengan respuesta.

Temas

- Parámetros del desencadenador de Lambda para crear el desafío de autenticación
- Ejemplo de creación de desafíos de autenticación

Parámetros del desencadenador de Lambda para crear el desafío de autenticación

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "challengeName": "string",
        "session": [
            ChallengeResult,
            . . .
        ],
        "clientMetadata": {
            "string": "string",
```

```
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "publicChallengeParameters": {
            "string": "string",
            . . .
        },
        "privateChallengeParameters": {
            "string": "string",
            . . .
        },
        "challengeMetadata": "string"
    }
}
```

Parámetros de la solicitud para crear desafíos de autenticación

userAttributes

Uno o varios pares de nombre y valor que representan atributos de usuario.

userNotFound

Este valor booleano se rellena cuando `PreventUserExistenceErrors` se establece como `ENABLED` en el cliente del grupo de usuarios.

challengeName

El nombre del nuevo desafío.

session

El elemento session consiste en una matriz de elementos de `ChallengeResult` que contienen, cada uno, los elementos siguientes:

challengeName

El tipo de desafío. Puede ser uno de los siguientes: `"CUSTOM_CHALLENGE"`, `"PASSWORD_VERIFIER"`, `"SMS_MFA"`, `"DEVICE_SRP_AUTH"`, `"DEVICE_PASSWORD_VERIFIER"` o `"ADMIN_NO_SRP_AUTH"`.

challengeResult

Establezca este parámetro en `true` si el usuario ha respondido correctamente al desafío o en `false`, en caso contrario.

challengeMetadata

El nombre del desafío personalizado. Solo se usa si `challengeName` es `"CUSTOM_CHALLENGE"`.

clientMetadata

Uno o varios pares de clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda que especifica destinada al desencadenador para crear desafíos de autenticación. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata de las acciones de la API AdminRespondToAuthChallenge y RespondToAuthChallenge. La solicitud que llama a la función de definición de desafíos de autenticación no incluye los datos transferidos en el parámetro ClientMetadata en las operaciones de la API AdminInitiateAuth y InitiateAuth.

Parámetros de la respuesta para crear desafíos de autenticación

publicChallengeParameters

Uno o varios pares de clave y valor para que la aplicación cliente los use en el desafío que se va a presentar al usuario. Este parámetro debe contener toda la información necesaria para que el desafío que se presente al usuario sea preciso.

privateChallengeParameters

Solo el desencadenador de Lambda para verificar la respuesta al desafío de autenticación utiliza este parámetro. Debe contener toda la información necesaria para validar la respuesta del usuario al desafío. Dicho de otro modo, el parámetro `publicChallengeParameters` contiene la pregunta que se formula al usuario y `privateChallengeParameters` contiene las respuestas válidas a la pregunta.

challengeMetadata

El nombre del desafío personalizado, si se trata de uno.

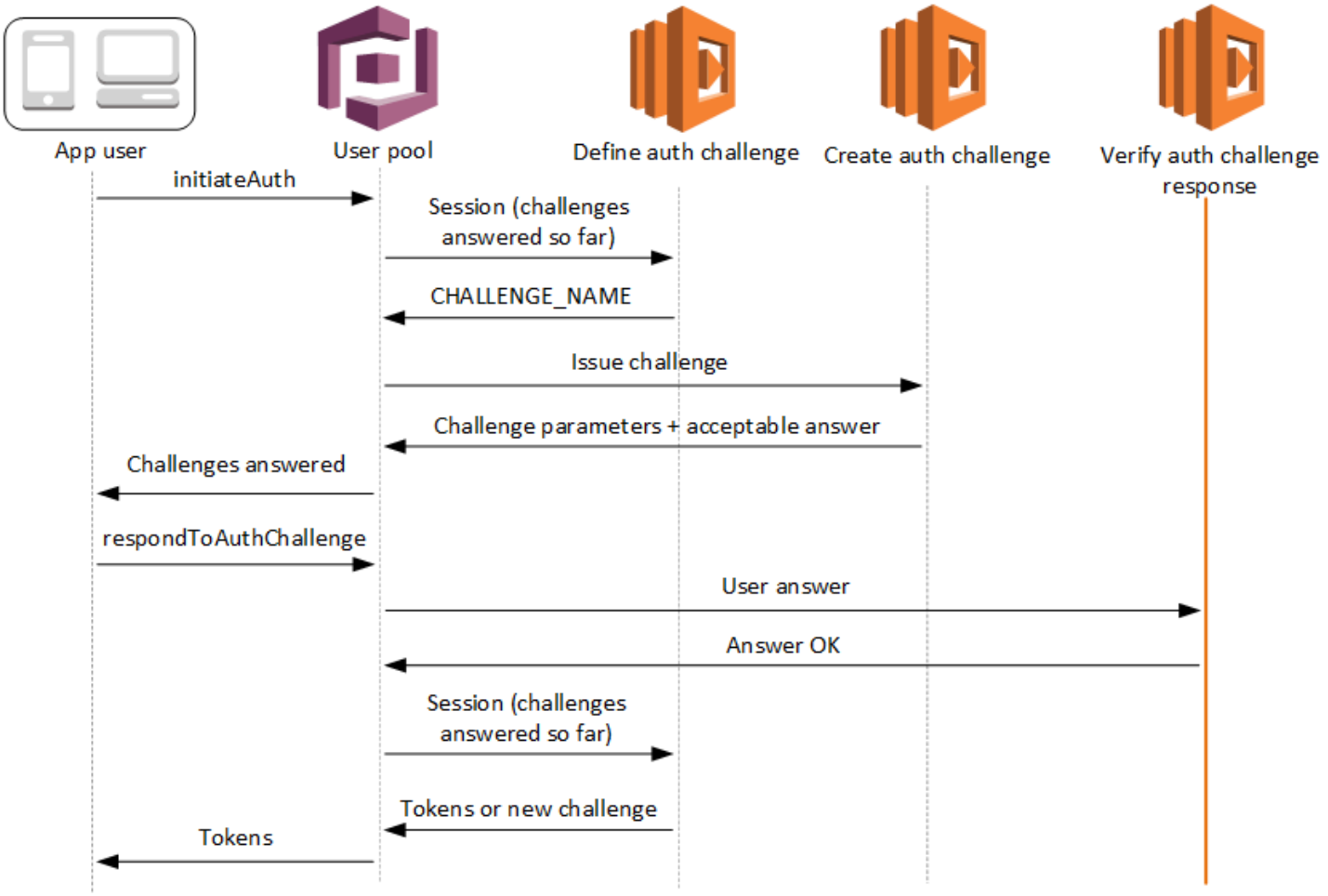
Ejemplo de creación de desafíos de autenticación

Se crea un CAPTCHA como desafío para el usuario. La URL de la imagen del CAPTCHA se añade a los parámetros de desafío públicos como "`captchaUrl`", mientras que la respuesta esperada se añade a los parámetros de desafío privados.

Node.js

```
const handler = async (event) => {
  if (event.request.challengeName !== "CUSTOM_CHALLENGE") {
    return event;
  }

  if (event.request.session.length === 2) {
    event.response.publicChallengeParameters = {};
    event.response.privateChallengeParameters = {};
    event.response.publicChallengeParameters.captchaUrl = "url/123.jpg";
    event.response.privateChallengeParameters.answer = "5";
  }

  if (event.request.session.length === 3) {
    event.response.publicChallengeParameters = {};
    event.response.privateChallengeParameters = {};
    event.response.publicChallengeParameters.securityQuestion =
      "Who is your favorite team mascot?";
    event.response.privateChallengeParameters.answer = "Peccy";
  }

  return event;
};

export { handler }
```

Desencadenador de Lambda para verificar la respuesta al desafío de autenticación



Verificación de la respuesta a los desafíos de autenticación

> Amazon Cognito lama a este desencadenador para verificar si la respuesta del usuario a un desafío de autenticación personalizado es o no válida. Forma parte del flujo de autenticación personalizado de un grupo de usuarios.

La solicitud de este disparador contiene los parámetros `privateChallengeParameters` y `challengeAnswer`. El desencadenador de Lambda para definir el desafío de autenticación devuelve los valores de `privateChallengeParameters`, que contienen la respuesta esperada del usuario. El parámetro `challengeAnswer` contiene la respuesta del usuario al desafío.

La respuesta contiene el atributo `answerCorrect`. Si el usuario finaliza correctamente el desafío, Amazon Cognito establece el valor del atributo en `true`. Si el usuario no finaliza correctamente el desafío, Amazon Cognito establece el valor del atributo en `false`.

El bucle de desafíos se repite hasta que los usuarios respondan a todos los desafíos.

Temas

- Parámetros del desencadenador de Lambda para verificar el desafío de autenticación
- Ejemplo de verificación de la respuesta a los desafíos de autenticación

Parámetros del desencadenador de Lambda para verificar el desafío de autenticación

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "privateChallengeParameters": {
            "string": "string",
            . . .
        },
        "challengeAnswer": "string",
        "clientMetadata": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "answerCorrect": boolean
    }
}
```

Parámetros de la solicitud para verificar desafíos de autenticación

userAttributes

Este parámetro contiene uno o varios pares de nombre-valor que representan atributos de usuario.

userNotFound

Cuando Amazon Cognito establece `PreventUserExistenceErrors` en `ENABLED` para su cliente de grupo de usuarios, Amazon Cognito rellena este booleano.

privateChallengeParameters

Este parámetro proviene del desencadenador para definir el desafío de autenticación. Para determinar si el usuario ha superado un desafío, Amazon Cognito compara los parámetros con la challengeAnswer de un usuario.

Este parámetro contiene toda la información necesaria para validar la respuesta del usuario al desafío. Esta información incluye la pregunta que Amazon Cognito presenta al usuario (`publicChallengeParameters`) y las respuestas válidas a la pregunta (`privateChallengeParameters`). Solo el desencadenador de Lambda de verificación de la respuesta al desafío de autenticación utiliza este parámetro.

challengeAnswer

Este valor de parámetro es la respuesta del usuario al desafío.

clientMetadata

Este parámetro contiene uno o varios pares de clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda para verificar el desencadenador del desafío de autenticación. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata en las operaciones de la API AdminRespondToAuthChallenge y RespondToAuthChallenge. Amazon Cognito no incluye los datos transferidos del parámetro ClientMetadata en las operaciones de la API AdminInitiateAuth y InitiateAuth en la solicitud que transfiere a la función de verificación de la autenticación.

Parámetros de la respuesta para verificar desafíos de autenticación

answerCorrect

Si el usuario finaliza correctamente el desafío, Amazon Cognito establece este parámetro en `true`. Si el usuario no finaliza correctamente el desafío, Amazon Cognito establece el parámetro en `false`.

Ejemplo de verificación de la respuesta a los desafíos de autenticación

En este ejemplo, la función de Lambda comprueba si la respuesta del usuario a un desafío coincide con la respuesta esperada. Amazon Cognito establece el parámetro `answerCorrect` en `true` si la respuesta del usuario coincide con la respuesta esperada.

Node.js

```
const handler = async (event) => {
  if (
    event.request.privateChallengeParameters.answer ==
    event.request.challengeAnswer
  ) {
    event.response.answerCorrect = true;
  } else {
    event.response.answerCorrect = false;
  }

  return event;
};

export { handler };
```

Desencadenador de Lambda anterior a la generación del token

Dado que Amazon Cognito invoca este desencadenador antes de que se genere el token, puede personalizar las notificaciones de los tokens del grupo de usuarios. Con las Características básicas de la versión dos o del evento desencadenante previo a la generación del token `V1_0`, puede personalizar el token de identidad (ID). En los grupos de usuarios con características de seguridad avanzadas activas, puede generar la versión 2 o evento desencadenante `V2_0` con la personalización del token de acceso.

Amazon Cognito envía un evento `V1_0` como una solicitud a la función con datos que escribiría en el token de ID. Un evento `V2_0` es una solicitud única con los datos que Amazon Cognito escribiría en los tokens tanto de de identidad como de acceso. Para personalizar ambos tokens, debe actualizar la función para usar la versión del desencadenador más reciente y enviar los datos de ambos tokens en la misma respuesta.

Este desencadenador Lambda puede añadir, eliminar y modificar algunas notificaciones en los tokens de identidad y acceso antes de que Amazon Cognito las envíe a su aplicación. Para utilizar esta característica, asocie una función de Lambda desde la consola del grupos de usuarios de Amazon Cognito o actualice su grupo de usuarios `LambdaConfig` a través de la (AWS Command Line Interface)AWS CLI.

Versiones de eventos

Su grupo de usuarios puede entregar diferentes versiones de un evento desencadenante previo a la generación del token a su función Lambda. Un `V1_0` disparador proporciona los parámetros para la modificación de los tokens de identificación. Un `V2_0` disparador proporciona los parámetros siguientes.

1. Las funciones de un `V1_0` disparador.
2. La posibilidad de personalizar los tokens de acceso.
3. La capacidad de transferir tipos de datos complejos a los valores declarados de los identificadores y los identificadores de acceso:
   - Cadena
   - Número
   - Booleano
   - Matriz de cadenas, números, valores booleanos o una combinación de cualquiera de estos
   - JSON

> **Note**
>
> En el token de identificación, puede rellenar objetos complejos con los valores de las reclamacionesphone_number_verified, excepto,email_verified, `updated_at` y. `address`

Los grupos de usuarios proporcionan `V1_0` eventos de forma predeterminada. Para configurar su grupo de usuarios para enviar un `V2_0` evento, elija una versión de evento desencadenante de Funciones básicas y personalización del token de acceso al configurar el activador en la consola de Amazon Cognito. También puede establecer el valor de `LambdaVersion` en los LambdaConfigparámetros de una solicitud UpdateUserPool o de CreateUserPool API. Se aplican costes adicionales a la personalización de los tokens de acceso con `V2_0` los eventos. Para obtener más información, consulte Precios de Amazon Cognito.

Reclamaciones y ámbitos excluidos

Amazon Cognito limita las reclamaciones y los ámbitos que puede agregar, modificar o suprimir en los tokens de acceso e identidad. Si la función de Lambda intenta establecer un valor para cualquiera de estas afirmaciones, Amazon Cognito emite un token con el valor de la reclamación original, si había alguno en la solicitud.

Reclamaciones compartidas

- `acr`
- `amr`
- `at_hash`
- `auth_time`
- `azp`
- `exp`
- `iat`
- `iss`
- `jti`
- `nbf`
- `nonce`
- `origin_jti`
- `sub`
- `token_use`

Reclamaciones del token del ID

- `identities`

- `aud`
- `cognito:username`

Reclamaciones del token de acceso

- `username`
- `client_id`
- `scope`

> **Note**
>
> Puede cambiar los alcances de un token de acceso con `scopesToAdd` y los valores de respuesta `scopesToSuppress`, pero no puede modificar la reclamación de `scope` directamente. No puede agregar ámbitos que comiencen por `aws.cognito`, como el ámbito reservado del grupo de usuarios `aws.cognito.signin.user.admin`.

- `device_key`
- `event_id`
- `version`

No puede agregar ni invalidar reclamaciones con los siguientes prefijos, pero puede suprimirlas o impedir que aparezcan en el token.

- `dev:`
- `cognito:`

Puede agregar una reclamación de `aud` para acceder a los tokens, pero el valor debe coincidir con el ID de cliente de la aplicación de la sesión actual. Puede derivar el ID de cliente en el evento de solicitud de `event.callerContext.clientId`.

Personalización del token de identidad

Con el desencadenador de Lambda previo a la generación de tokens, puede personalizar el contenido de un token de identidad (ID) del grupo de usuarios. El token de ID proporciona los atributos de usuario de un origen de identidades fiable para iniciar sesión en una aplicación web o móvil. Para obtener más información acerca de los tokens de ID, consulte Uso del token de ID.

Los usos del desencadenador de Lambda previo a la generación de tokens con un token de ID incluyen los siguientes.

- Realice un cambio en el tiempo de ejecución en el rol de IAM que el usuario solicita de un grupo de identidades.
- Agregue atributos de usuario desde un origen externo.
- Agregue o sustituya los valores de los atributos de usuario existentes.
- Suprima la divulgación de los atributos de usuario que, debido a los ámbitos autorizados del usuario y al acceso de lectura a los atributos que ha concedido al cliente de la aplicación, se transferirían a la aplicación.

Personalización del token de acceso

Con el desencadenador de Lambda previo a la generación de tokens, puede personalizar el contenido de un token de acceso del grupo de usuarios. El token de acceso autoriza a los usuarios a recuperar información de recursos de acceso protegido, como las operaciones de API autorizadas por el token de Amazon Cognito y las API de terceros. Si bien puede generar tokens de acceso para la autorización machine-to-machine (M2M) con Amazon Cognito con una concesión de credenciales de cliente, las solicitudes M2M no invocan la función de activación previa a la generación del token y no pueden emitir tokens de acceso personalizados. Para obtener más información acerca de los tokens de acceso, consulte Uso del token de acceso.

Los usos del desencadenador de Lambda previo a la generación de tokens con un token de acceso incluyen los siguientes.

- Agregue o suprima los ámbitos de OAuth 2.0 en la reclamación de `scope`. Por ejemplo, puede agregar ámbitos a un token de acceso resultante de la autenticación de la API de grupos de usuarios de Amazon Cognito, que solo asigna el ámbito `aws.cognito.signin.user.admin`.
- Cambie la suscripción de un usuario en los grupos de usuarios.
- Agregue notificaciones que aún no estén presentes en un token de acceso de Amazon Cognito.
- Suprima la divulgación de las reclamaciones que, de otro modo, se transferirían a la aplicación.

Para poder personalizar el acceso al grupo de usuarios, debe configurar el grupo de usuarios para que genere una versión actualizada de la solicitud de desencadenador. Actualice el grupo de usuarios como se muestra en el siguiente procedimiento.

AWS Management Console

Para admitir la personalización del token de acceso en un desencadenador de Lambda previo a la generación de tokens

1. Diríjase a la consola de Amazon Cognito y luego elija User Pools (Grupos de usuarios).
2. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
3. Si aún no lo ha hecho, active las características de seguridad avanzadas desde la pestaña Integración de la aplicación.
4. Elija la pestaña User pool properties (Propiedades del grupo de usuarios) y localice Lambda triggers (Desencadenadores Lambda).
5. Agregue o edite un desencadenador previo a la generación de tokens.
6. Elija una función de Lambda en Asignar función de Lambda.
7. Elija una Versión del evento del desencadenador de las Características básicas y personalización del token de acceso. Esta configuración actualiza los parámetros de solicitud que Amazon Cognito envía a la función para incluir campos para la personalización del token de acceso.

User pools API

Para admitir la personalización del token de acceso en un desencadenador de Lambda previo a la generación de tokens

Genere una CreateUserPoolsolicitud de API o API. UpdateUserPool Debe especificar un valor para todos los parámetros que no desee establecer en un valor predeterminado. Para obtener más información, consulte Actualización de la configuración del grupo de usuarios.

Incluya el siguiente contenido en el parámetro `LambdaVersion` de la solicitud. Un valor `LambdaVersion` de `V2_0` hace que el grupo de usuarios agregue parámetros para la personalización del token de acceso. Para invocar una versión de función específica, utilice el ARN de una función de Lambda con una versión de función como el valor de `LambdaArn`.

```
"PreTokenGenerationConfig": {
   "LambdaArn": "arn:aws:lambda:us-west-2:123456789012:function:MyFunction",
   "LambdaVersion": "V2_0"
},
```

Temas

Fuentes del desencadenador de Lambda de pregeneración de tokens

Valor de triggerSource	Evento
`TokenGeneration_HostedAuth`	Se llama durante la autenticación desde la página de inicio de sesión de la IU alojada de Amazon Cognito.
`TokenGeneration_Authentication`	Se llama después de que se hayan completado los flujos de autenticación.
`TokenGeneration_NewPasswordChallenge`	Se llama después de que un administrador cree al usuario. Este flujo se invoca cuando el usuario tiene que cambiar una contraseña temporal.
`TokenGeneration_AuthenticateDevice`	Se llama al final de la autenticación de un dispositivo de usuario.
`TokenGeneration_RefreshTokens`	Se llama cuando un usuario intenta actualizar los tokens de identidad y acceso.

Parámetros del desencadenador de Lambda de pregeneración de tokens

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes. Al agregar un desencadenador de Lambda previo a la generación de tokens al grupo de usuarios, puede elegir una versión de desencadenador. Esta versión determina si Amazon Cognito transfiere una solicitud a la función de Lambda con parámetros adicionales para la personalización del token de acceso.

Version 1

El token de la versión 1 puede establecer la pertenencia a grupos, las funciones de IAM y nuevos reclamos en los tokens de identificación.

```
{
    "request": {
        "userAttributes": {"string": "string"},
        "groupConfiguration": {
                "groupsToOverride": [
                    "string",
                    "string"
                ],
                "iamRolesToOverride": [
                    "string",
                    "string"
                ],
                "preferredRole": "string"
        },
        "clientMetadata": {"string": "string"}
    },
    "response": {
        "claimsOverrideDetails": {
            "claimsToAddOrOverride": {"string": "string"},
            "claimsToSuppress": [
                "string",
                "string"
            ],
            "groupOverrideDetails": {
                "groupsToOverride": [
                    "string",
                    "string"
                ],
```

```
                "iamRolesToOverride": [
                    "string",
                    "string"
                ],
                "preferredRole": "string"
            }
        }
    }
}
```

Version 2

El evento de solicitud de la versión 2 añade campos que personalizan el token de acceso. También añade compatibilidad con tipos de `claimsToOverride` datos complejos en el objeto de respuesta. La función Lambda puede devolver los siguientes tipos de datos con el valor de: `claimsToOverride`

- Cadena
- Número
- Booleano
- Matriz de cadenas, números, valores booleanos o una combinación de cualquiera de estos
- JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string"
        },
        "scopes": ["string", "string"],
        "groupConfiguration": {
            "groupsToOverride": ["string", "string"],
            "iamRolesToOverride": ["string", "string"],
            "preferredRole": "string"
        },
        "clientMetadata": {
            "string": "string"
        }
    },
    "response": {
        "claimsAndScopeOverrideDetails": {
```

```
            "idTokenGeneration": {
                "claimsToAddOrOverride": {
                    "string": [accepted datatype]
                },
                "claimsToSuppress": ["string", "string"]
            },
            "accessTokenGeneration": {
                "claimsToAddOrOverride": {
                    "string": [accepted datatype]
                },
                "claimsToSuppress": ["string", "string"],
                "scopesToAdd": ["string", "string"],
                "scopesToSuppress": ["string", "string"]
            },
            "groupOverrideDetails": {
                "groupsToOverride": ["string", "string"],
                "iamRolesToOverride": ["string", "string"],
                "preferredRole": "string"
            }
        }
    }
}
```

Parámetros de la solicitud anterior a la generación del token

Nombre	Descripción	Versión mínima del evento del desencadenador
userAttributes	Los atributos del perfil de usuario en el grupo de usuarios.	1
groupConfiguration	Objeto de entrada que contiene la configuración de grupo actual. El objeto incluye `groupsToOverride` , `iamRolesToOverride` y `preferredRole` .	1
groupsToOverride	Los grupos del grupo de usuarios de los que es miembro su usuario.	1
iamRolesToAnular	Puede asociar un grupo de grupos de usuarios a un rol AWS Identity and Access Management (IAM). Este	1

Nombre	Descripción	Versión mínima del evento del desencadenador
	elemento es una lista de todos los roles de IAM de los grupos a los que pertenece su usuario.	
preferredRole	Puede establecer una prioridad para los grupos del grupo de usuarios. Este elemento contiene el nombre del rol de IAM del grupo con la mayor prioridad en el elemento `groupsToOverride` .	1
clientMetadata	Uno o varios pares clave-valor que puede especificar y proporcionar como datos de entrada personalizados a la función de Lambda para el desencadenador anterior a la generación del token. Para pasar estos datos a la función Lambda, utilice el ClientMetadata parámetro en las operaciones AdminRespondToAuthChallengey RespondToAuthChallengeAPI. Amazon Cognito no incluye datos del `ClientMetadata` parámetro ni de las operaciones de la InitiateAuthAPI en la solicitud que transfiere a la función de generación previa del token. AdminInitiateAuth	1
alcances	Los ámbitos de OAuth 2.0 del usuario. Los ámbitos que están presentes en un token de acceso son los ámbitos estándar y personalizados del grupo de usuarios que el usuario ha solicitado y que usted ha autorizado emitir al cliente de la aplicación.	2

Parámetros de la respuesta anterior a la generación del token

Nombre	Descripción	Versión mínima del evento del desencadenador
claimsOverrideDetails	Un contenedor para todos los elementos de un evento desencadenante `V1_0`.	1
claimsAndScopeOverrideDetails	Un contenedor para todos los elementos de un evento desencadenante `V2_0`.	2
idTokenGeneration	Las reclamaciones que desea invalidar, agregar o suprimir en el token del ID de usuario. Estos valores de personalización del token principal al ID aparecen solo en los eventos de la versión 2, pero los elementos secundarios aparecen en los eventos de la versión 1.	2
accessTokenGeneration	Las reclamaciones y ámbitos que desea invalidar, agregar o suprimir en el token de acceso del usuario. Este elemento principal para acceder a los valores de personalización del token solo aparece en los eventos de la versión 2.	2
claimsToAddOrOverride	Un mapa de una o más reclamaciones y los valores que desee agregar o modificar. Para las reclamaciones relacionadas con el grupo, utilice `groupOverrideDetails` en su lugar. En los eventos de la versión 2, este elemento aparece en `accessTokenGeneration` y `idTokenGeneration`.	1 *
claimsToSuppress	Una lista de reclamaciones que quiere que Amazon Cognito suprima. Si tu función suprime y reemplaza un valor de notificación, Amazon Cognito suprime la notificación.	1

Nombre	Descripción	Versión mínima del evento del desencadenador
	En los eventos de la versión 2, este elemento aparece en `accessTokenGeneration` y `idTokenGeneration`.	
groupOverrideDetails	Objeto de salida que contiene la configuración de grupo actual. El objeto incluye `groupsToOverride`, `iamRolesToOverride` y `preferredRole`. La función sustituye el objeto `groupOverrideDetails` por el objeto que proporcione. Si proporciona un objeto vacío o nulo en la respuesta, entonces Amazon Cognito suprimirá los grupos. Para dejar la configuración de grupos existente tal como está, copie el valor del objeto `groupConfiguration` de la solicitud en el objeto `groupOverrideDetails` de la respuesta. Luego transfiéralo de nuevo al servicio. Los tokens de ID y de acceso de Amazon Cognito contienen la notificación `cognito:groups`. El objeto `groupOverrideDetails` sustituye la reclamación de `cognito:groups` en tokens de acceso y tokens de ID.	1
scopesToAdd	Una lista de ámbitos de OAuth 2.0 que quiere agregar a la reclamación de `scope` en el token de acceso del usuario. No puede agregar valores de ámbito que contengan uno o más caracteres de espacio en blanco.	2
scopesToSuppress	Una lista de ámbitos de OAuth 2.0 que quiere eliminar de la reclamación de `scope` en el token de acceso del usuario.	2

* Los objetos de respuesta a los eventos de la versión 1 pueden devolver cadenas. Los objetos de respuesta a los eventos de la versión 2 pueden devolver objetos complejos.

Ejemplo de la segunda versión de un evento desencadenante previo al token: añadir y suprimir notificaciones, ámbitos y grupos

En este ejemplo, se realizan las siguientes modificaciones a los tokens de un usuario.

1. Establece su `family_name` como `Doe` en el token de ID.
2. Impide que las notificaciones `email` y `phone_number` aparezcan en el token de ID.
3. Establece su notificación `cognito:roles` de token de ID a `"arn:aws:iam::123456789012:role\/sns_callerA","arn:aws:iam::123456789012:role\/sns_callerC","arn:aws:iam::123456789012:role\/sns_callerB"`.
4. Establece su notificación `cognito:preferred_role` de token de ID a `arn:aws:iam::123456789012:role/sns_caller`.
5. Añade los ámbitos `openid`, `email` y `solar-system-data/asteroids.add` al token de acceso.
6. Suprime el ámbito `phone_number` y `aws.cognito.signin.user.admin` del token de acceso. La eliminación de `phone_number` impide la recuperación del número de teléfono del usuario de `userInfo`. La eliminación de `aws.cognito.signin.user.admin` impide las solicitudes de la API por el usuario para leer y modificar su propio perfil con la API de grupos de usuarios de Amazon Cognito.

   > **Note**
   >
   > La eliminación de `phone_number` de los ámbitos solo impide la recuperación del número de teléfono de un usuario si los ámbitos restantes del token de acceso incluyen `openid` y al menos un ámbito estándar más. Para obtener más información, consulte Acerca de los ámbitos.

7. Establece su ID y notificación `cognito:groups` de token de acceso en `"new-group-A","new-group-B","new-group-C"`.

JavaScript

```
export const handler = function(event, context) {
  event.response = {
    "claimsAndScopeOverrideDetails": {
      "idTokenGeneration": {
```

```
        "claimsToAddOrOverride": {
          "family_name": "Doe"
    },
        "claimsToSuppress": [
          "email",
          "phone_number"
    ]
      },
      "accessTokenGeneration": {
        "scopesToAdd": [
          "openid",
          "email",
          "solar-system-data/asteroids.add"
        ],
        "scopesToSuppress": [
          "phone_number",
          "aws.cognito.signin.user.admin"
        ]
      },
      "groupOverrideDetails": {
        "groupsToOverride": [
    "new-group-A",
    "new-group-B",
    "new-group-C"
  ],
        "iamRolesToOverride": [
          "arn:aws:iam::123456789012:role/new_roleA",
          "arn:aws:iam::123456789012:role/new_roleB",
          "arn:aws:iam::123456789012:role/new_roleC"
        ],
        "preferredRole": "arn:aws:iam::123456789012:role/new_role",
      }
    }
  };
  // Return to Amazon Cognito
  context.done(null, event);
};
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
    "version": "2",
    "triggerSource": "TokenGeneration_Authentication",
    "region": "us-east-1",
    "userPoolId": "us-east-1_EXAMPLE",
    "userName": "JaneDoe",
    "callerContext": {
        "awsSdkVersion": "aws-sdk-unknown-unknown",
        "clientId": "1example23456789"
    },
    "request": {
        "userAttributes": {
            "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "cognito:user_status": "CONFIRMED",
            "email_verified": "true",
            "phone_number_verified": "true",
            "phone_number": "+12065551212",
            "family_name": "Zoe",
            "email": "Jane.Doe@example.com"
        },
        "groupConfiguration": {
            "groupsToOverride": ["group-1", "group-2", "group-3"],
            "iamRolesToOverride": ["arn:aws:iam::123456789012:role/sns_caller1",
 "arn:aws:iam::123456789012:role/sns_caller2", "arn:aws:iam::123456789012:role/
sns_caller3"],
            "preferredRole": ["arn:aws:iam::123456789012:role/sns_caller"]
        },
        "scopes": [
            "aws.cognito.signin.user.admin", "openid", "email", "phone"
        ]
    },
    "response": {
        "claimsAndScopeOverrideDetails": []
    }
}
```

Ejemplo de la segunda versión del evento previo a la generación de fichas: añadir reclamaciones con objetos complejos

En este ejemplo, se realizan las siguientes modificaciones a los tokens de un usuario.

1. Agrega afirmaciones de los tipos numérico, de cadena, booleano y JSON al token de ID. Este es el único cambio que los eventos desencadenantes de la segunda versión ponen a disposición del token de ID.
2. Añade notificaciones de los tipos numérico, de cadena, booleano y JSON al token de acceso.
3. Agrega tres ámbitos al token de acceso.
4. Suprime los `sub` reclamos `email` y atributos en los identificadores de acceso y de identificación.
5. Suprime el `aws.cognito.signin.user.admin` alcance del token de acceso.

JavaScript

```
export const handler = function(event, context) {

    var scopes = ["MyAPI.read", "MyAPI.write", "MyAPI.admin"]
    var claims = {}
    claims["aud"]= event.callerContext.clientId;
    claims["booleanTest"] = false;
    claims["longTest"] = 9223372036854775807;
    claims["exponentTest"] = 1.7976931348623157E308;
    claims["ArrayTest"] = ["test", 9223372036854775807, 1.7976931348623157E308,
 true];
    claims["longStringTest"] = "\{\
        \"first_json_block\": \{\
            \"key_A\": \"value_A\",\
            \"key_B\": \"value_B\"\
        \},\
        \"second_json_block\": \{\
            \"key_C\": \{\
                \"subkey_D\": [\
                    \"value_D\",\
                    \"value_E\"\
                ],\
                \"subkey_F\": \"value_F\"\
            \},\
            \"key_G\": \"value_G\"\
        \}\
    \}";
    claims["jsonTest"] = {
     "first_json_block": {
      "key_A": "value_A",
      "key_B": "value_B"
     },
```

```
     "second_json_block": {
      "key_C": {
       "subkey_D": [
        "value_D",
        "value_E"
       ],
       "subkey_F": "value_F"
      },
      "key_G": "value_G"
     }
    };
    event.response = {
        "claimsAndScopeOverrideDetails": {
            "idTokenGeneration": {
                "claimsToAddOrOverride": claims,
                "claimsToSuppress": ["email","sub"]
            },
            "accessTokenGeneration": {
                "claimsToAddOrOverride": claims,
                "claimsToSuppress": ["email","sub"],
                "scopesToAdd": scopes,
                "scopesToSuppress": ["aws.cognito.signin.user.admin"]
            }
        }
    };
    console.info("EVENT response\n" + JSON.stringify(event, (_, v) => typeof v ===
 'bigint' ? v.toString() : v, 2))
    console.info("EVENT response size\n" + JSON.stringify(event, (_, v) => typeof v
 === 'bigint' ? v.toString() : v).length)
    // Return to Amazon Cognito
    context.done(null, event);
};
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
    "version": "2",
```

```
    "triggerSource": "TokenGeneration_HostedAuth",
    "region": "us-west-2",
    "userPoolId": "us-west-2_EXAMPLE",
    "userName": "JaneDoe",
    "callerContext": {
        "awsSdkVersion": "aws-sdk-unknown-unknown",
        "clientId": "1example23456789"
    },
    "request": {
        "userAttributes": {
            "sub": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "cognito:user_status": "CONFIRMED"
            "email_verified": "true",
            "phone_number_verified": "true",
            "phone_number": "+12065551212",
            "email": "Jane.Doe@example.com"
        },
        "groupConfiguration": {
            "groupsToOverride": ["group-1", "group-2", "group-3"],
            "iamRolesToOverride": ["arn:aws:iam::123456789012:role/sns_caller1"],
            "preferredRole": ["arn:aws:iam::123456789012:role/sns_caller1"]
        },
        "scopes": [
            "aws.cognito.signin.user.admin",
            "phone",
            "openid",
            "profile",
            "email"
        ]
    },
    "response": {
        "claimsAndScopeOverrideDetails": []
    }
}
```

Ejemplo uno de versión de evento de generación anterior al token: Agregar una notificación nueva y suprimir otra existente

En este ejemplo, se utiliza el evento de desencadenador versión 1 con una función de Lambda anterior a la generación del token para agregar una reclamación nueva y suprimir una existente.

Node.js

```
const handler = async (event) => {
  event.response = {
    claimsOverrideDetails: {
      claimsToAddOrOverride: {
        my_first_attribute: "first_value",
        my_second_attribute: "second_value",
      },
      claimsToSuppress: ["email"],
    },
  };

  return event;
};

export { handler };
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo: puesto que el ejemplo de código no procesa ningún parámetro de solicitud, puede utilizar un evento de prueba con una solicitud vacía. Para obtener más información sobre los parámetros de solicitud habituales, consulte Evento desencadenador de Lambda para un grupo de usuarios.

JSON

```
{
  "request": {},
  "response": {}
}
```

Ejemplo uno de versión de evento de generación anterior al token: Modificar la pertenencia de un usuario a un grupo

En este ejemplo, se utiliza el evento de desencadenador versión 1 con una función de Lambda anterior a la generación del token para modificar la suscripción de un grupo de usuarios.

Node.js

```
const handler = async (event) => {
  event.response = {
    claimsOverrideDetails: {
      groupOverrideDetails: {
        groupsToOverride: ["group-A", "group-B", "group-C"],
        iamRolesToOverride: [
          "arn:aws:iam::XXXXXXXXXXX:role/sns_callerA",
          "arn:aws:iam::XXXXXXXXX:role/sns_callerB",
          "arn:aws:iam::XXXXXXXXXX:role/sns_callerC",
        ],
        preferredRole: "arn:aws:iam::XXXXXXXXXXX:role/sns_caller",
      },
    },
  };

  return event;
};

export { handler };
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
  "request": {},
  "response": {}
}
```

Migración del desencadenador de Lambda del usuario

Amazon Cognito llama a este desencadenador si un usuario no aparece en el grupo de usuarios en el momento del inicio de sesión con una contraseña o en el flujo de recuperación de contraseñas olvidadas. Cuando la función de Lambda finaliza responde con éxito, Amazon Cognito crea el

usuario en el grupo de usuarios. Para obtener información detallada sobre el flujo de autenticación con el desencadenador de Lambda para migrar usuarios, consulte Importación de usuarios a grupos de usuarios con un desencadenador de Lambda para la migración de usuarios.

Con este desencadenador de Lambda, se pueden migrar usuarios desde el directorio de usuarios actual a grupos de usuarios de Amazon Cognito en el momento del inicio de sesión o durante el flujo de recuperación de contraseñas olvidadas.

Temas

Fuentes del desencadenador de Lambda para migrar usuarios

Valor de triggerSource	Evento
`UserMigration_Authentication`	Migración de usuarios al iniciar sesión.
`UserMigration_ForgotPassword`	Migración de usuarios durante el flujo de recuperación de contraseñas olvidadas.

Parámetros del desencadenador de Lambda para migrar usuarios

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "userName": "string",
    "request": {
        "password": "string",
        "validationData": {
            "string": "string",
            . . .
```

```
        },
        "clientMetadata": {
            "string": "string",
       . . .
        }
    },
    "response": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "finalUserStatus": "string",
        "messageAction": "string",
        "desiredDeliveryMediums": [ "string", . . .],
        "forceAliasCreation": boolean,
        "enableSMSMFA": boolean
    }
}
```

Parámetros de la solicitud para la migración de usuarios

userName

Nombre de usuario que ingresa el usuario al iniciar sesión.

password

Contraseña que ingresa el usuario al iniciar sesión. Amazon Cognito no envía este valor en una solicitud iniciada por un flujo de recuperación de contraseñas olvidadas.

validationData

Uno o varios pares de clave-valor que contienen los datos de validación de la solicitud de inicio de sesión del usuario. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata en las acciones de la API InitiateAuth y AdminInitiateAuth.

clientMetadata

Uno o varios pares de clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda para el desencadenador para migrar usuarios. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata en las acciones de la API AdminRespondToAuthChallenge y ForgotPassword.

Parámetros de la respuesta para la migración de usuarios

userAttributes

Este campo es obligatorio.

Este campo debe contener uno o varios pares de nombre-valor que Amazon Cognito almacena en el perfil del usuario del grupo de usuarios y utiliza como atributos del usuario. Puede incluir atributos de usuario estándar y personalizados. Los atributos personalizados deben tener el prefijo `custom:` para distinguirlos de los atributos estándar. Para obtener más información, consulte Atributos personalizados.

> **Note**
>
> Para que los usuarios puedan restablecer sus contraseñas en el flujo de recuperación de contraseñas olvidadas, deben disponer de una dirección de correo electrónico o un número de teléfono verificados. Amazon Cognito envía un mensaje con el código de restablecimiento de la contraseña a la dirección de correo electrónico o al número de teléfono de los atributos del usuario.

Atributos	Requisito
Todos los atributos que ha marcado como obligatorios al crear el grupo de usuarios	Si faltan atributos obligatorios durante la migración, Amazon Cognito usará los valores predeterminados.
`username`	Es obligatorio si ha configurado el grupo de usuarios con atributos de alias y el nombre de usuario para iniciar sesión, y si el usuario ha ingresado un valor de alias válido para iniciar sesión. Este valor del alias puede ser una dirección de correo electrónico, un nombre de usuario preferido o un número de teléfono. Si la solicitud y el grupo de usuarios cumplen los requisitos de alias, la respuesta de la función debe asignar el parámetro `username` que recibió a un atributo de alias. Además, la respuesta debe asignar su propio valor al atributo `username`. Si el grupo

Atributos	Requisito
	de usuarios no cumple las condiciones requeridas para asignar el grupo recibido `username` a un alias, entonces el parámetro `username` en la respuesta debe coincidir exactamente con la solicitud u omitirse. **Note** El `username` debe ser único en el grupo de usuarios.

finalUserStatus

Puede establecer este parámetro en `CONFIRMED` para confirmar automáticamente a los usuarios que pueden iniciar sesión con sus contraseñas anteriores. Al configurar un usuario en `CONFIRMED`, este no debe tomar medidas adicionales para poder iniciar sesión. Si no establece este atributo en `CONFIRMED`, se establece en `RESET_REQUIRED`.

Un `finalUserStatus` de `RESET_REQUIRED` significa que el usuario debe cambiar su contraseña inmediatamente después de la migración al iniciar sesión y que la aplicación cliente debe gestionar la `PasswordResetRequiredException` durante el flujo de autenticación.

Note

Amazon Cognito no aplica la política de seguridad de contraseñas que configuró para el grupo de usuarios durante la migración mediante el desencadenador de Lambda. Si la contraseña no cumple con la política de contraseñas configurada, Amazon Cognito seguirá aceptando la contraseña para que pueda seguir migrando al usuario. Para aplicar la política de seguridad de la contraseña y rechazar contraseñas que no cumplan con la política, valide la seguridad de la contraseña del código. A continuación, establezca FinalUserStatus en `RESET_REQUIRED` si la contraseña no cumple con la política.

messageAction

Puede establecer este parámetro en `SUPPRESS` para rechazar el envío del mensaje de bienvenida que Amazon Cognito suele enviar a los usuarios nuevos. Si la función no devuelve este parámetro, Amazon Cognito envía el mensaje de bienvenida.

desiredDeliveryMediums

Este parámetro puede establecerse en `EMAIL` para enviar el mensaje de bienvenida por correo electrónico o en `SMS` para enviar el mensaje de bienvenida por SMS. Si la función no devuelve este parámetro, Amazon Cognito envía el mensaje de bienvenida por SMS.

forceAliasCreation

Si establece este parámetro en `TRUE` y el número de teléfono o la dirección de correo electrónico que se han especificado en el parámetro UserAttributes ya existen como alias para otro usuario, la llamada a la API migrará el alias del usuario anterior al usuario recién creado. El usuario anterior ya no podrá iniciar sesión con ese alias.

Si define este parámetro en `FALSE` y el alias existe, Amazon Cognito no migrará al usuario y devolverá un error a la aplicación cliente.

Si no devuelve este parámetro, Amazon Cognito asume que su valor es “false”.

enableSMSMFA

Establezca este parámetro en `true` para solicitar que el usuario migrado complete la autenticación multifactor (MFA) de mensajes de texto SMS para iniciar sesión. El grupo de usuarios debe tener habilitada la MFA. Los atributos del usuario en los parámetros de la solicitud deben incluir un número de teléfono o, de lo contrario, la migración de ese usuario producirá un error.

Ejemplo: Migrar un usuario con una contraseña existente

Con esta función de Lambda de ejemplo, se migra el usuario con una contraseña existente y se suprime el mensaje de bienvenida de Amazon Cognito.

Node.js

```
const validUsers = {
  belladonna: { password: "Test123", emailAddress: "bella@example.com" },
};

// Replace this mock with a call to a real authentication service.
const authenticateUser = (username, password) => {
  if (validUsers[username] && validUsers[username].password === password) {
    return validUsers[username];
  } else {
```

```
    return null;
  }
};

const lookupUser = (username) => {
  const user = validUsers[username];

  if (user) {
    return { emailAddress: user.emailAddress };
  } else {
    return null;
  }
};

const handler = async (event) => {
  if (event.triggerSource == "UserMigration_Authentication") {
    // Authenticate the user with your existing user directory service
    const user = authenticateUser(event.userName, event.request.password);
    if (user) {
      event.response.userAttributes = {
        email: user.emailAddress,
        email_verified: "true",
      };
      event.response.finalUserStatus = "CONFIRMED";
      event.response.messageAction = "SUPPRESS";
    }
  } else if (event.triggerSource == "UserMigration_ForgotPassword") {
    // Look up the user in your existing user directory service
    const user = lookupUser(event.userName);
    if (user) {
      event.response.userAttributes = {
        email: user.emailAddress,
        // Required to enable password-reset code to be sent to user
        email_verified: "true",
      };
      event.response.messageAction = "SUPPRESS";
    }
  }

  return event;
};

export { handler };
```

Desencadenador de Lambda para mensajes personalizados

Amazon Cognito llama a este desencadenador antes de enviar un mensaje de verificación por teléfono o correo electrónico, o un código de autenticación multifactor (MFA, por sus siglas en inglés). Puede personalizar el mensaje dinámicamente con el desencadenador de mensajes personalizado. Los mensajes personalizados estáticos se pueden editar en la pestaña Message Customizations (Personalizaciones de mensajes) de la consola de Amazon Cognito original.

La solicitud incluye `codeParameter`. Esta es una cadena que actúa de marcador de posición del código que Amazon Cognito entrega al usuario. Especifique la cadena `codeParameter` en el cuerpo del mensaje, en el lugar donde desea que se inserte el código de verificación. Cuando Amazon Cognito recibe esta respuesta, reemplaza la cadena `codeParameter` por el código de verificación real.

> **Note**
>
> Una función de Lambda para mensajes personalizados con el desencadenador de origen `CustomMessage_AdminCreateUser` devuelve un nombre de usuario y un código de verificación. Como un usuario creado por un administrador debe recibir tanto su nombre de usuario como su código, la respuesta de la función debe incluir tanto `request.usernameParameter` como `request.codeParameter`.

Temas

Fuentes de desencadenadores de Lambda para mensajes personalizados

Valor de triggerSource	Evento
`CustomMessage_SignUp`	Mensaje personalizado para enviar el código de confirmación posterior a la inscripción.

Valor de triggerSource	Evento
`CustomMessage_AdminCreateUser`	Mensaje personalizado para enviar la contraseña temporal a un usuario nuevo.
`CustomMessage_ResendCode`	Mensaje personalizado para volver a enviar el código de confirmación a un usuario ya existente.
`CustomMessage_ForgotPassword`	Mensaje personalizado para enviar el código de confirmación a una solicitud de contraseña olvidada.
`CustomMessage_UpdateUserAttribute`	Mensaje personalizado: cuando el correo electrónico o el número de teléfono de un usuario cambia, este disparador envía automáticamente un código de verificación al usuario. No se puede utilizar para otros atributos.
`CustomMessage_VerifyUserAttribute`	Mensaje personalizado: este disparador envía un código de verificación al usuario cuando este lo solicita manualmente para un correo electrónico o un número de teléfono nuevo.
`CustomMessage_Authentication`	Mensaje personalizado para enviar código de MFA durante la autenticación.

Parámetros de desencadenadores de Lambda para mensajes personalizados

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "userAttributes": {
```

```
            "string": "string",
            . . .
        }
        "codeParameter": "####",
        "usernameParameter": "string",
        "clientMetadata": {
            "string": "string",
            . . .
        }
    },
    "response": {
        "smsMessage": "string",
        "emailMessage": "string",
        "emailSubject": "string"
    }
}
```

Parámetros de la solicitud para mensajes personalizados

userAttributes

Uno o varios pares de nombre y valor que representan atributos de usuario.

codeParameter

Cadena que se usa como marcador de posición del código de verificación en los mensajes personalizados.

Parámetrousername

El nombre de usuario. Amazon Cognito incluye este parámetro en las solicitudes que provienen de los usuarios creados por el administrador.

clientMetadata

Uno o varios pares de clave-valor que puede proporcionar como datos de entrada personalizados a la función de Lambda que especifica destinada al desencadenador para mensajes personalizados. La solicitud que invoca una función de mensaje personalizada no incluye los datos transferidos en el ClientMetadata parámetro en AdminInitiateAuthlas operaciones de InitiateAuthAPI. Para pasar estos datos a la función Lambda, puede usar el ClientMetadata parámetro en las siguientes acciones de la API:

- AdminResetUserPassword

- AdminRespondToAuthChallenge
- AdminUpdateUserAttributes
- ForgotPassword
- GetUserAttributeVerificationCode
- ResendConfirmationCode
- SignUp
- UpdateUserAttributes

Parámetros de la respuesta para mensajes personalizados

En la respuesta, especifique el texto personalizado que usará en los mensajes a los usuarios. Para ver las restricciones de cadena que Amazon Cognito aplica a estos parámetros, consulte. MessageTemplateType

smsMessage

El mensaje de texto SMS personalizado que se envía a los usuarios. Debe incluir el valor `codeParameter` recibido en la solicitud.

emailMessage

Mensaje de correo electrónico personalizado que se envía a los usuarios. Puede utilizar el formato HTML en el parámetro `emailMessage`. Debe incluir el valor `codeParameter` que ha recibido en la solicitud como variable `{####}`. Amazon Cognito puede utilizar el parámetro `emailMessage` solo si el atributo `EmailSendingAccount` del grupo de usuarios es `DEVELOPER`. Si el atributo `EmailSendingAccount` del grupo de usuarios no es `DEVELOPER` y se devuelve un parámetro `emailMessage`, Amazon Cognito genera un código de error 400 `com.amazonaws.cognito.identity.idp.model.InvalidLambdaResponseException`. El atributo `EmailSendingAccount` de un grupo de usuarios es `DEVELOPER` cuando elige utilizar Amazon Simple Email Service (Amazon SES) para enviar mensajes de correo electrónico. De lo contrario, el valor es `COGNITO_DEFAULT`.

emailSubject

La línea de asunto del mensaje personalizado. Solo puede usar el `emailSubject` parámetro si el EmailSendingAccount atributo del grupo de usuarios es`DEVELOPER`. Si el atributo `EmailSendingAccount` del grupo de usuarios no es `DEVELOPER` y Amazon Cognito devuelve un parámetro `emailSubject`, Amazon Cognito genera un código de error 400 `com.amazonaws.cognito.identity.idp.model.InvalidLambdaResponseException`.

El atributo `EmailSendingAccount` de un grupo de usuarios es `DEVELOPER` cuando elige utilizar Amazon Simple Email Service (Amazon SES) para enviar mensajes de correo electrónico. De lo contrario, el valor es `COGNITO_DEFAULT`.

Ejemplo de mensaje personalizado para registrarse

Esta función de Lambda personaliza un mensaje de correo electrónico o SMS cuando el servicio necesita que una aplicación envíe un código de verificación al usuario.

Amazon Cognito puede llamar a un desencadenador de Lambda en varios eventos: después del registro, al reenviar un código de verificación, para recuperar una contraseña olvidada o al verificar un atributo de usuario. La respuesta contiene mensajes tanto para SMS como para correo electrónico. El mensaje debe incluir el parámetro de código `"####"`. Este parámetro es el marcador de posición del código de verificación que recibe el usuario.

La longitud máxima de un mensaje de correo electrónico es de 20 000 caracteres UTF-8. Esta longitud incluye el código de verificación. Puede utilizar etiquetas HTML en estos mensajes de correo electrónico.

La longitud máxima de los mensaje SMS es 140 caracteres UTF-8. Esta longitud incluye el código de verificación.

Node.js

```
const handler = async (event) => {
  if (event.triggerSource === "CustomMessage_SignUp") {
    const message = `Thank you for signing up. Your confirmation code is
 ${event.request.codeParameter}.`;
    event.response.smsMessage = message;
    event.response.emailMessage = message;
    event.response.emailSubject = "Welcome to the service.";
  }
  return event;
};

export { handler };
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta.

En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
  "version": 1,
  "triggerSource": "CustomMessage_SignUp/CustomMessage_ResendCode/
CustomMessage_ForgotPassword/CustomMessage_VerifyUserAttribute",
  "region": "<region>",
  "userPoolId": "<userPoolId>",
  "userName": "<userName>",
  "callerContext": {
      "awsSdk": "<calling aws sdk with version>",
      "clientId": "<apps client id>",
      ...
  },
  "request": {
      "userAttributes": {
          "phone_number_verified": false,
          "email_verified": true,
           ...
      },
      "codeParameter": "####"
  },
  "response": {
      "smsMessage": "<custom message to be sent in the message with code parameter>"
      "emailMessage": "<custom message to be sent in the message with code
 parameter>"
      "emailSubject": "<custom email subject>"
  }
}
```

Ejemplo de mensaje personalizado para la creación de usuarios por parte del administrador

La solicitud que Amazon Cognito envió a este ejemplo de función Lambda de mensajes personalizados tiene un `triggerSource` valor de, un nombre de usuario `CustomMessage_AdminCreateUser` y una contraseña temporal. La función se completa con la contraseña temporal `${event.request.codeParameter}` de la solicitud y con el nombre de usuario `${event.request.usernameParameter}` de la solicitud.

Los mensajes personalizados deben insertar los valores de `codeParameter` `smsMessage` y `usernameParameter` dentro del objeto `emailMessage` de respuesta. En este ejemplo, la función escribe el mismo mensaje en los campos de respuesta `event.response.smsMessage` y`event.response.emailMessage`.

La longitud máxima de un mensaje de correo electrónico es de 20 000 caracteres UTF-8. Esta longitud incluye el código de verificación. Puede usar etiquetas HTML en estos correos electrónicos. La longitud máxima de los mensaje SMS es 140 caracteres UTF-8. Esta longitud incluye el código de verificación.

La respuesta contiene mensajes tanto para SMS como para correo electrónico.

Node.js

```
const handler = async (event) => {
  if (event.triggerSource === "CustomMessage_AdminCreateUser") {
    const message = `Welcome to the service. Your user name is
 ${event.request.usernameParameter}. Your temporary password is
 ${event.request.codeParameter}`;
    event.response.smsMessage = message;
    event.response.emailMessage = message;
    event.response.emailSubject = "Welcome to the service";
  }
  return event;
};

export { handler }
```

Amazon Cognito transfiere la información del evento a la función de Lambda. A continuación, la función devuelve el mismo objeto de evento a Amazon Cognito con los cambios en la respuesta. En la consola de Lambda puede configurar un evento de prueba con los datos relevantes para el desencadenador de Lambda. El siguiente es un evento de prueba para este código de ejemplo:

JSON

```
{
  "version": 1,
  "triggerSource": "CustomMessage_AdminCreateUser",
  "region": "<region>",
  "userPoolId": "<userPoolId>",
```

```
  "userName": "<userName>",
  "callerContext": {
      "awsSdk": "<calling aws sdk with version>",
      "clientId": "<apps client id>",
      ...
  },
  "request": {
      "userAttributes": {
          "phone_number_verified": false,
          "email_verified": true,
           ...
      },
      "codeParameter": "####",
      "usernameParameter": "username"
  },
  "response": {
      "smsMessage": "<custom message to be sent in the message with code parameter
 and username parameter>"
      "emailMessage": "<custom message to be sent in the message with code parameter
 and username parameter>"
      "emailSubject": "<custom email subject>"
  }
}
```

Desencadenadores de Lambda para remitentes personalizados

Los grupos de usuarios de Amazon Cognito proporcionan los desencadenadores de Lambda `CustomEmailSender` y `CustomSMSSender` para activar las notificaciones de correo electrónico y SMS de terceros. Puede elegir proveedores de SMS y correo electrónico para que envíen las notificaciones a los usuarios desde su código de función de Lambda. Cuando Amazon Cognito debe enviar notificaciones como códigos de confirmación, códigos de verificación o contraseñas temporales, los eventos activan las funciones de Lambda configuradas. Amazon Cognito envía el código y las contraseñas temporales (secretos) a sus funciones de Lambda activadas. Amazon Cognito cifra estos secretos con una clave administrada por el cliente AWS KMS y el AWS Encryption SDK. La AWS Encryption SDK es una biblioteca de cifrado del lado del cliente que le ayuda a cifrar y descifrar datos genéricos.

> **Note**
>
> Puede utilizar la AWS CLI o el SDK para configurar sus grupos de usuarios con el fin de utilizar estos desencadenadores de Lambda. Estas configuraciones no están disponibles en la consola de Amazon Cognito.

CustomEmailSender

Amazon Cognito invoca este desencadenador para enviar notificaciones por correo electrónico a los usuarios.

CustomSMSSender

Amazon Cognito invoca este desencadenador para enviar notificaciones por SMS a los usuarios.

Recursos

Los siguientes recursos pueden ayudarle a utilizar los desencadenadores `CustomEmailSender` y `CustomSMSSender`.

AWS KMS

AWS KMS es un servicio administrado para crear y controlar claves de AWS KMS. Estas claves cifran los datos. Para obtener más información, consulte ¿Qué es AWS Key Management Service?.

Clave KMS

Una clave KMS es una representación lógica de una clave criptográfica. La clave de KMS incluye metadatos, como el ID de clave, la fecha de creación, la descripción y el estado de la clave. La clave de KMS también contiene el material de claves utilizado para cifrar y descifrar datos. Para obtener más información, consulte AWS claves KMS.

Claves KMS simétricas

Una clave KMS simétrica es una clave de cifrado de 256 bits que no sale de AWS KMS sin cifrar. Para utilizar una clave KMS simétrica, tiene que llamar a AWS KMS. Amazon Cognito utiliza claves simétricas. La misma clave cifra y descifra. Para obtener más información, consulte Claves KMS simétricas.

Desencadenador de Lambda para remitentes de correos electrónicos personalizados

Al asignar un desencadenador de envío de correo electrónico personalizado al grupo de usuarios, Amazon Cognito invoca una función de Lambda en lugar de su comportamiento predeterminado cuando un evento de usuario requiere que envíe un mensaje de correo electrónico. Con un desencadenador de remitente personalizado, la función de AWS Lambda puede enviar notificaciones por correo electrónico a los usuarios a través del método y el proveedor que elija. El código personalizado de la función debe procesar y entregar todos los mensajes de correo electrónico del grupo de usuarios.

> **Note**
>
> Actualmente, no puede asignar desencadenadores de remitente personalizados en la consola de Amazon Cognito. Puede asignar un desencadenador con el parámetro `LambdaConfig` en una solicitud de API `CreateUserPool` o `UpdateUserPool`.

Para usar este desencadenador, siga estos pasos:

1. Cree una clave de cifrado simétrica en AWS Key Management Service (AWS KMS). Amazon Cognito genera secretos (contraseñas temporales, códigos de verificación y confirmación) y después utiliza esta clave KMS para cifrarlos. A continuación, puede usar la operación de la API Descifrar en la función de Lambda para descifrar los secretos y enviarlos al usuario como texto sin formato. El AWS Encryption SDK es una herramienta útil para las operaciones de AWS KMS en la función.
2. Cree una función de Lambda que desee asignar como desencadenador de remitente personalizado. Conceda permisos `kms:Decrypt` para la clave KMS al rol de la función de Lambda.
3. Conceda el acceso `cognito-idp.amazonaws.com` a la entidad principal del servicio de Amazon Cognito para llamar a la función de Lambda.
4. Escriba el código de función de Lambda que dirige sus mensajes a métodos de entrega personalizados o proveedores externos. Para entregar el código de verificación o confirmación del usuario, Base64 descodifica y descifra el valor del parámetro `code` de la solicitud. Esta operación genera un código o contraseña en texto plano que debe incluir en el mensaje.
5. Actualice el grupo de usuarios para que utilice un desencadenador Lambda de remitente personalizado. La entidad principal de IAM que actualiza o crea un grupo de usuarios con un desencadenador de remitente personalizado debe tener permiso para crear una concesión para

la clave de KMS. El fragmento de `LambdaConfig` siguiente asigna funciones personalizadas de envío de SMS y correo electrónico.

```
"LambdaConfig": {
   "KMSKeyID": "arn:aws:kms:us-
east-1:123456789012:key/a6c4f8e2-0c45-47db-925f-87854bc9e357",
   "CustomEmailSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
      "LambdaVersion": "V1_0"
   },
   "CustomSMSSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
      "LambdaVersion": "V1_0"
   }
```

Parámetros de desencadenador de Lambda para remitente de correo electrónico personalizado

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "type": "customEmailSenderRequestV1",
        "code": "string",
        "clientMetadata": {
            "string": "string",
             . . .
            },
        "userAttributes": {
            "string": "string",
            . . .
         }
}
```

Parámetros de solicitudes de remitente de correo electrónico personalizado

type

La versión de la solicitud. Para un evento de remitente de correo electrónico personalizado, el valor de esta cadena es siempre `customEmailSenderRequestV1`.

code

El código cifrado que su función puede descifrar y enviar al usuario.

clientMetadata

Uno o varios pares clave-valor que puede proporcionar como datos de entrada personalizados al desencadenador de la función de Lambda de remitente de correo electrónico personalizado. Puede transferir estos datos a la función de Lambda mediante el parámetro ClientMetadata en las acciones de la API AdminRespondToAuthChallenge y RespondToAuthChallenge. Amazon Cognito no incluye los datos transferidos del parámetro ClientMetadata en las operaciones de la API AdminInitiateAuth y InitiateAuth en la solicitud que transfiere a la función de autenticación posterior.

userAttributes

Uno o varios pares clave-valor que representan atributos de usuario.

Parámetros de respuesta de remitente de correo electrónico personalizado

Amazon Cognito no espera ninguna información de devolución adicional en la respuesta de remitente de correo electrónico personalizado. La función puede utilizar operaciones de la API para consultar y modificar los recursos o registrar metadatos de eventos en un sistema externo.

Activación del desencadenador de Lambda para remitente de correo electrónico personalizado

Para configurar un desencadenador de remitente de correo electrónico personalizado que utilice lógica personalizada para enviar mensajes de correo electrónico a su grupo de usuarios, active el desencadenador de la siguiente manera. En el procedimiento siguiente se asigna un desencadenador de correo electrónico personalizado, un desencadenador de SMS personalizado o ambos a su grupo de usuarios. Después de agregar su desencadenador de remitente de correo electrónico personalizado, Amazon Cognito siempre envía los atributos de usuario, como la dirección de correo electrónico, y el código de un solo uso a su función de Lambda en lugar de enviar de forma predeterminada un mensaje de correo electrónico con Amazon Simple Email Service.

> **Important**
>
> Amazon Cognito aplica códigos de escape HTML a caracteres reservados como < (`<`) y > (`>`) en la contraseña temporal de su usuario. Estos caracteres pueden aparecer en las contraseñas temporales que Amazon Cognito envía a su función de remitente de correo electrónico personalizado, pero no en los códigos de verificación temporales. Para enviar contraseñas temporales, su función de Lambda debe anular los códigos de escape de estos caracteres después de descifrar la contraseña y antes de enviar el mensaje a su usuario.

1. Cree una clave de cifrado en AWS KMS. Esta clave se utiliza para cifrar contraseñas temporales y códigos de autorización que genera Amazon Cognito. A continuación, puede descifrar estos secretos con la función de Lambda de remitente personalizado para enviarlos a su usuario como texto sin formato.
2. Conceda a la entidad principal `cognito-idp.amazonaws.com` del servicio Amazon Cognito acceso para cifrar códigos con la clave KMS.

   Aplique la siguiente política basada en recursos a su clave KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cognito-idp.amazonaws.com"
        },
        "Action": "kms:CreateGrant",
        "Resource": "arn:aws:kms:us-
west-2:111222333444:key/1example-2222-3333-4444-999example",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "111222333444"
            },
            "ArnLike": {
                "aws:SourceArn": "arn:aws:cognito-idp:us-
west-2:111222333444:userpool/us-east-1_EXAMPLE"
            }
        }
    }]
}
```

3. Cree una función de Lambda para el desencadenador de remitente personalizado. Amazon Cognito utiliza el SDK de cifrado de AWS para cifrar los secretos, las contraseñas temporales y los códigos que autorizan las solicitudes de API de sus usuarios.

   - Asigne un rol de IAM a su función de Lambda que tenga, como mínimo, permisos `kms:Decrypt` para la clave KMS.

4. Conceda el acceso `cognito-idp.amazonaws.com` a la entidad principal del servicio de Amazon Cognito para llamar a la función de Lambda.

   El siguiente comando de la AWS CLI concede a Amazon Cognito permiso para invocar su función de Lambda:

   ```
   aws lambda add-permission --function-name lambda_arn --statement-id
    "CognitoLambdaInvokeAccess" --action lambda:InvokeFunction --principal cognito-
   idp.amazonaws.com
   ```

5. Elabore el código de su función de Lambda para enviar sus mensajes. Amazon Cognito utiliza AWS Encryption SDK para cifrar secretos antes de enviarlos a la función de Lambda de remitente personalizado. En su función, descifre el secreto y procese los metadatos pertinentes. A continuación, envíe el código, su propio mensaje personalizado y el número de teléfono de destino a la API personalizada que entrega el mensaje.
6. Agregue el AWS Encryption SDK a su función de Lambda. Para obtener más información, consulte Lenguajes de programación del SDK de cifrado de AWS. Complete los siguientes pasos para actualizar el paquete de Lambda.

   a. Exporte su función de Lambda como un archivo .zip en la AWS Management Console.
   b. Abra la función y agregue el AWS Encryption SDK. Para obtener más información y enlaces de descarga, consulte Lenguajes de programación de AWS Encryption SDK en la Guía para desarrolladores de AWS Encryption SDK.
   c. Comprima su función con sus dependencias del SDK y cargue la función en Lambda. Para obtener más información, consulte Implementación de funciones de Lambda como archivos .zip en la Guía para desarrolladores de AWS Lambda.

7. Actualice el grupo de usuarios para agregar desencadenadores de Lambda de remitente personalizado. Incluya un parámetro `CustomSMSSender` o `CustomEmailSender` en una solicitud de API `UpdateUserPool`. La operación de API `UpdateUserPool` requiere todos los parámetros de su grupo de usuarios y los parámetros que desea modificar. Si no proporciona

todos los parámetros relevantes, Amazon Cognito establece los valores de los parámetros que faltan en sus valores predeterminados. Como se demuestra en el ejemplo siguiente, incluya entradas para todas las funciones de Lambda que desee agregar o mantener en su grupo de usuarios. Para obtener más información, consulte Actualización de la configuración del grupo de usuarios.

```
   #Send this parameter in an 'aws cognito-idp update-user-pool' CLI command,
including any existing
   #user pool configurations.

     --lambda-config "PreSignUp=lambda-arn, \
                      CustomSMSSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
                      CustomEmailSender={LambdaVersion=V1_0,LambdaArn=lambda-arn},
\
                      KMSKeyID=key-id"
```

Para eliminar un desencadenador de Lambda de remitente personalizado con un comando `update-user-pool` de la AWS CLI, omita el parámetro `CustomSMSSender` o `CustomEmailSender` de `--lambda-config` e incluya todos los demás desencadenadores que desee usar con su grupo de usuarios.

Para eliminar un desencadenador de Lambda de remitente personalizado con una solicitud de API `UpdateUserPool`, omita el parámetro `CustomSMSSender` o `CustomEmailSender` del cuerpo de la solicitud que contiene el resto de la configuración del grupo de usuarios.

Ejemplo de código

En el siguiente ejemplo de Node.js se procesa un evento de mensaje de correo electrónico en la función de Lambda de remitentes de correo electrónico personalizado. En este ejemplo se supone que la función tiene dos variables de entorno definidas.

`KEY_ALIAS`

El alias de la clave de KMS que desea utilizar para cifrar y descifrar los códigos de sus usuarios.

`KEY_ARN`

El nombre de recurso de Amazon (ARN) de la clave de KMS que desea utilizar para cifrar y descifrar los códigos de sus usuarios.

```
const AWS = require('aws-sdk');
const b64 = require('base64-js');
const encryptionSdk = require('@aws-crypto/client-node');
//Configure the encryption SDK client with the KMS key from the environment variables.
const { encrypt, decrypt } =
 encryptionSdk.buildClient(encryptionSdk.CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT);
const generatorKeyId = process.env.KEY_ALIAS;
const keyIds = [ process.env.KEY_ARN ];
const keyring = new encryptionSdk.KmsKeyringNode({ generatorKeyId, keyIds })
exports.handler = async (event) => {
 //Decrypt the secret code using encryption SDK.
 let plainTextCode;
 if(event.request.code){
  const { plaintext, messageHeader } = await decrypt(keyring,
 b64.toByteArray(event.request.code));
  plainTextCode = plaintext
 }
 //PlainTextCode now contains the decrypted secret.
 if(event.triggerSource == 'CustomEmailSender_SignUp'){
  //Send an email message to your user via a custom provider.
  //Include the temporary password in the message.
 }
 else if(event.triggerSource == 'CustomEmailSender_ResendCode'){
 }
 else if(event.triggerSource == 'CustomEmailSender_ForgotPassword'){
 }
 else if(event.triggerSource == 'CustomEmailSender_UpdateUserAttribute'){
 }
 else if(event.triggerSource == 'CustomEmailSender_VerifyUserAttribute'){
 }
 else if(event.triggerSource == 'CustomEmailSender_AdminCreateUser'){
 }
 else if(event.triggerSource == 'CustomEmailSender_AccountTakeOverNotification'){
 }
 return;
};
```

Fuentes del desencadenador de Lambda para remitentes de correos electrónicos personalizados

En la siguiente tabla se muestra el evento desencadenante de las fuentes del desencadenador para correos electrónicos personalizados en el código de Lambda.

TriggerSource value	Evento
CustomEmailSender_SignUp	Un usuario se registra y Amazon Cognito envía un mensaje de bienvenida.
CustomEmailSender_ForgotPassword	Un usuario solicita un código para restablecer su contraseña.
CustomEmailSender_ResendCode	Un usuario solicita un código de sustitución para restablecer su contraseña.
CustomEmailSender_UpdateUserAttribute	Un usuario actualiza una dirección de correo electrónico o un atributo de número de teléfono y Amazon Cognito envía un código para verificar el atributo.
CustomEmailSender_VerifyUserAttribute	Un usuario crea una dirección de correo electrónico nueva o un atributo de número de teléfono y Amazon Cognito envía un código para verificar el atributo.
CustomEmailSender_AdminCreateUser	Crea un nuevo usuario en su grupo de usuarios y Amazon Cognito le envía una contraseña temporal.
CustomEmailSender_AccountTakeOverNotification	Amazon Cognito detecta un intento de asumir una cuenta de usuario y envía una notificación al usuario.

Desencadenador de Lambda para remitentes personalizados de SMS

Al asignar un desencadenador de envío de SMS personalizado al grupo de usuarios, Amazon Cognito invoca una función de Lambda en lugar de su comportamiento predeterminado cuando un evento de usuario requiere que envíe un mensaje SMS. Con un activador de remitente personalizado, tu AWS Lambda función puede enviar notificaciones por SMS a tus usuarios a través del método y el proveedor que elijas. El código personalizado de la función debe procesar y entregar todos los mensajes SMS del grupo de usuarios.

> **Note**
>
> Actualmente, no puede asignar desencadenadores de remitente personalizados en la consola de Amazon Cognito. Puede asignar un desencadenador con el parámetro `LambdaConfig` en una solicitud de API `CreateUserPool` o `UpdateUserPool`.

Para usar este desencadenador, siga estos pasos:

1. Crea una clave de cifrado simétrica en AWS Key Management Service (AWS KMS). Amazon Cognito genera secretos (contraseñas temporales, códigos de verificación y confirmación) y después utiliza esta clave KMS para cifrarlos. A continuación, puede usar la operación de la API Descifrar en la función de Lambda para descifrar los secretos y enviarlos al usuario como texto sin formato. AWS Encryption SDKEs una herramienta útil para AWS KMS las operaciones de su función.
2. Cree una función de Lambda que desee asignar como desencadenador de remitente personalizado. Conceda permisos `kms:Decrypt` para la clave KMS al rol de la función de Lambda.
3. Conceda el acceso `cognito-idp.amazonaws.com` a la entidad principal del servicio de Amazon Cognito para llamar a la función de Lambda.
4. Escriba el código de función de Lambda que dirige sus mensajes a métodos de entrega personalizados o proveedores externos. Para entregar el código de verificación o confirmación del usuario, Base64 descodifica y descifra el valor del parámetro `code` de la solicitud. Esta operación genera un código o contraseña en texto plano que debe incluir en el mensaje.
5. Actualice el grupo de usuarios para que utilice un desencadenador Lambda de remitente personalizado. La entidad principal de IAM que actualiza o crea un grupo de usuarios con un desencadenador de remitente personalizado debe tener permiso para crear una concesión para la clave de KMS. El fragmento de `LambdaConfig` siguiente asigna funciones personalizadas de envío de SMS y correo electrónico.

```
"LambdaConfig": {
   "KMSKeyID": "arn:aws:kms:us-
east-1:123456789012:key/a6c4f8e2-0c45-47db-925f-87854bc9e357",
   "CustomEmailSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
      "LambdaVersion": "V1_0"
   },
```

```
   "CustomSMSSender": {
      "LambdaArn": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
      "LambdaVersion": "V1_0"
   }
```

Parámetros de desencadenador de Lambda para remitente de SMS personalizado

La solicitud que Amazon Cognito envía a esta función de Lambda es una combinación de los parámetros que se indican a continuación y los parámetros comunes que Amazon Cognito agrega a todas las solicitudes.

JSON

```
{
    "request": {
        "type": "customSMSSenderRequestV1",
        "code": "string",
        "clientMetadata": {
            "string": "string",
             . . .
            },
        "userAttributes": {
            "string": "string",
            . . .
         }
}
```

Parámetros de solicitudes de remitente de SMS personalizado

type

La versión de la solicitud. Para un evento de remitente de SMS personalizado, el valor de esta cadena es siempre `customSMSSenderRequestV1`.

code

El código cifrado que su función puede descifrar y enviar al usuario.

clientMetadata

Uno o varios pares clave-valor que puede proporcionar como datos de entrada personalizados al desencadenador de la función de Lambda de remitente de SMS personalizado. Para pasar

estos datos a la función Lambda, puede usar el ClientMetadata parámetro en las acciones AdminRespondToAuthChallengey de la RespondToAuthChallengeAPI. Amazon Cognito no incluye datos del ClientMetadata parámetro ni de las operaciones de InitiateAuthAPI en la solicitud que transfiere a la función de autenticación posterior. AdminInitiateAuth

userAttributes

Uno o varios pares clave-valor que representan atributos de usuario.

Parámetros de respuesta de remitente de SMS personalizado

Amazon Cognito no espera ninguna información de devolución adicional en la respuesta. La función puede utilizar operaciones de la API para consultar y modificar los recursos o registrar metadatos de eventos en un sistema externo.

Activación del desencadenador de Lambda para remitente de SMS personalizado

Puede configurar un desencadenador de remitente personalizado que utilice lógica personalizada para enviar mensajes SMS a su grupo de usuarios. En el procedimiento siguiente se asigna un desencadenador de SMS personalizado, un desencadenador de correo electrónico personalizado o ambos a su grupo de usuarios. Después de agregar su desencadenador de remitente de SMS personalizado, Amazon Cognito siempre envía los atributos de usuario, como el número de teléfono, y el código de un solo uso a su función de Lambda en lugar de enviar de forma predeterminada un mensaje SMS con Amazon Simple Notification Service.

> **Important**
>
> Amazon Cognito aplica códigos de escape HTML a caracteres reservados como < (`<`) y > (`>`) en la contraseña temporal de su usuario. Estos caracteres pueden aparecer en las contraseñas temporales que Amazon Cognito envía a su función de remitente de correo electrónico personalizado, pero no en los códigos de verificación temporales. Para enviar contraseñas temporales, su función de Lambda debe anular los códigos de escape de estos caracteres después de descifrar la contraseña y antes de enviar el mensaje a su usuario.

1. Cree una clave de cifrado en AWS KMS. Esta clave se utiliza para cifrar contraseñas temporales y códigos de autorización que genera Amazon Cognito. A continuación, puede descifrar estos secretos con la función de Lambda de remitente personalizado para enviarlos a su usuario como texto sin formato.

2. Conceda a la entidad principal `cognito-idp.amazonaws.com` del servicio Amazon Cognito acceso para cifrar códigos con la clave KMS.

   Aplique la siguiente política basada en recursos a su clave KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cognito-idp.amazonaws.com"
        },
        "Action": "kms:CreateGrant",
        "Resource": "arn:aws:kms:us-
west-2:111222333444:key/1example-2222-3333-4444-999example",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "111222333444"
            },
            "ArnLike": {
                "aws:SourceArn": "arn:aws:cognito-idp:us-
west-2:111222333444:userpool/us-east-1_EXAMPLE"
            }
        }
    }]
}
```

3. Cree una función de Lambda para el desencadenador de remitente personalizado. Amazon Cognito utiliza el SDK de cifrado de AWS para cifrar los secretos, las contraseñas temporales y los códigos que autorizan las solicitudes de API de sus usuarios.

   - Asigne un rol de IAM a su función de Lambda que tenga, como mínimo, permisos `kms:Decrypt` para la clave KMS.

4. Conceda el acceso `cognito-idp.amazonaws.com` a la entidad principal del servicio de Amazon Cognito para llamar a la función de Lambda.

   El siguiente AWS CLI comando otorga permiso a Amazon Cognito para invocar la función Lambda:

```
        aws lambda add-permission --function-name lambda_arn --statement-id
 "CognitoLambdaInvokeAccess" --action lambda:InvokeFunction --principal cognito-
idp.amazonaws.com
```

5. Elabore el código de su función de Lambda para enviar sus mensajes. Amazon Cognito se utiliza AWS Encryption SDK para cifrar los secretos antes de que Amazon Cognito los envíe al remitente personalizado, función Lambda. En su función, descifre el secreto y procese los metadatos pertinentes. A continuación, envíe el código, su propio mensaje personalizado y el número de teléfono de destino a la API personalizada que entrega el mensaje.
6. Añada el AWS Encryption SDK a su función Lambda. Para obtener más información, consulte Lenguajes de programación del SDK de cifrado de AWS. Complete los siguientes pasos para actualizar el paquete de Lambda.

   a. Exporte su función de Lambda como un archivo .zip en la AWS Management Console.

   b. Abra la función y añada la AWS Encryption SDK. Para obtener más información y enlaces de descarga, consulte Lenguajes de programación de AWS Encryption SDK en la Guía para desarrolladores de AWS Encryption SDK .

   c. Comprima su función con sus dependencias del SDK y cargue la función en Lambda. Para obtener más información, consulte Implementación de funciones de Lambda como archivos .zip en la Guía para desarrolladores de AWS Lambda .
7. Actualice el grupo de usuarios para agregar desencadenadores de Lambda de remitente personalizado. Incluya un parámetro `CustomSMSSender` o `CustomEmailSender` en una solicitud de API `UpdateUserPool`. La operación de API `UpdateUserPool` requiere todos los parámetros de su grupo de usuarios y los parámetros que desea modificar. Si no proporciona todos los parámetros relevantes, Amazon Cognito establece los valores de los parámetros que faltan en sus valores predeterminados. Como se demuestra en el ejemplo siguiente, incluya entradas para todas las funciones de Lambda que desee agregar o mantener en su grupo de usuarios. Para obtener más información, consulte Actualización de la configuración del grupo de usuarios.

```
   #Send this parameter in an 'aws cognito-idp update-user-pool' CLI command,
including any existing
   #user pool configurations.

     --lambda-config "PreSignUp=lambda-arn, \
                     CustomSMSSender={LambdaVersion=V1_0,LambdaArn=lambda-arn}, \
```

```
                    CustomEmailSender={LambdaVersion=V1_0,LambdaArn=lambda-arn},
\
                    KMSKeyID=key-id"
```

Para eliminar un disparador Lambda de remitente personalizado con un `update-user-pool` AWS CLI, omita el `CustomEmailSender` parámetro `CustomSMSSender` o e incluya todos los demás activadores que desee usar con su grupo de usuarios. `--lambda-config`

Para eliminar un desencadenador de Lambda de remitente personalizado con una solicitud de API `UpdateUserPool`, omita el parámetro `CustomSMSSender` o `CustomEmailSender` del cuerpo de la solicitud que contiene el resto de la configuración del grupo de usuarios.

Ejemplo de código

En el siguiente ejemplo de Node.js se procesa un evento de mensaje SMS en la función de Lambda de remitente de SMS personalizado. En este ejemplo se supone que la función tiene dos variables de entorno definidas.

KEY_ALIAS

El alias de la clave de KMS que desea utilizar para cifrar y descifrar los códigos de sus usuarios.

KEY_ARN

El nombre de recurso de Amazon (ARN) de la clave de KMS que desea utilizar para cifrar y descifrar los códigos de sus usuarios.

```
const AWS = require('aws-sdk');
const b64 = require('base64-js');
const encryptionSdk = require('@aws-crypto/client-node');
//Configure the encryption SDK client with the KMS key from the environment variables.

const { encrypt, decrypt } =
 encryptionSdk.buildClient(encryptionSdk.CommitmentPolicy.REQUIRE_ENCRYPT_ALLOW_DECRYPT);
const generatorKeyId = process.env.KEY_ALIAS;
const keyIds = [ process.env.KEY_ARN ];
const keyring = new encryptionSdk.KmsKeyringNode({ generatorKeyId, keyIds })
exports.handler = async (event) => {
 //Decrypt the secret code using encryption SDK.
 let plainTextCode;
```

```
 if(event.request.code){
  const { plaintext, messageHeader } = await decrypt(keyring,
 b64.toByteArray(event.request.code));
  plainTextCode = plaintext
 }
 //PlainTextCode now contains the decrypted secret.
 if(event.triggerSource == 'CustomSMSSender_SignUp'){
  //Send an SMS message to your user via a custom provider.
  //Include the temporary password in the message.
 }
 else if(event.triggerSource == 'CustomSMSSender_ResendCode'){
 }
 else if(event.triggerSource == 'CustomSMSSender_ForgotPassword'){
 }
 else if(event.triggerSource == 'CustomSMSSender_UpdateUserAttribute'){
 }
 else if(event.triggerSource == 'CustomSMSSender_VerifyUserAttribute'){
 }
 else if(event.triggerSource == 'CustomSMSSender_AdminCreateUser'){
 }
 else if(event.triggerSource == 'CustomSMSSender_AccountTakeOverNotification'){
 }
 return;
};
```

Temas

- Evaluar las capacidades de los mensajes SMS con una función de remitente de SMS personalizado
- Fuentes del desencadenador de Lambda para remitentes personalizados de SMS

Evaluar las capacidades de los mensajes SMS con una función de remitente de SMS personalizado

La función Lambda de remitente de SMS personalizado acepta los mensajes SMS que enviaría el grupo de usuarios y la función entrega el contenido según su lógica personalizada. Amazon Cognito envía el Parámetros de desencadenador de Lambda para remitente de SMS personalizado a su función. Su función puede hacer lo que desee con esta información. Por ejemplo, puede enviar el código a un tema de Amazon Simple Notification Service (Amazon SNS). Un suscriptor de temas de Amazon SNS puede ser un mensaje SMS, un punto de conexión HTTPS o una dirección de correo electrónico.

Para crear un entorno de prueba para la mensajería SMS de Amazon Cognito con una función Lambda de remitente de SMS personalizada, consulte amazon-cognito-user-pool- development-and-testing-with - sms-redirected-to-email en la biblioteca aws-samples de. GitHub El repositorio contiene AWS CloudFormation plantillas que pueden crear un nuevo grupo de usuarios o funcionar con un grupo de usuarios del que ya disponga. Estas plantillas crean funciones de Lambda y un tema de Amazon SNS. La función de Lambda que la plantilla asigna como desencadenador de remitente SMS personalizado, redirige los mensajes SMS a los suscriptores al tema de Amazon SNS.

Cuando implementa esta solución en un grupo de usuarios, todos los mensajes que Amazon Cognito suele enviar a través de mensajería SMS, la función de Lambda los envía en su lugar a una dirección de correo electrónico central. Utilice esta solución para personalizar y obtener una vista previa de los mensajes SMS y para probar los eventos del grupo de usuarios que hacen que Amazon Cognito envíe un mensaje SMS. Tras completar las pruebas, revierta la CloudFormation pila o elimine la asignación de funciones de envío de SMS personalizada de su grupo de usuarios.

> **Important**
>
> No utilice las plantillas de amazon-cognito-user-pool- development-and-testing-with - sms-redirected-to-email para crear un entorno de producción. La función de Lambda del remitente de SMS personalizado en la solución simula mensajes SMS, pero la función de Lambda los envía a una sola dirección de correo electrónico central. Para poder enviar mensajes SMS en un grupo de usuarios de Amazon Cognito de producción, debe completar los requisitos que se muestran en Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito.

Fuentes del desencadenador de Lambda para remitentes personalizados de SMS

En la siguiente tabla, se muestra el evento desencadenante de las fuentes del desencadenador de SMS personalizado en el código de Lambda.

`TriggerSource value`	Evento
`CustomSMSSender_SignUp`	Un usuario se registra y Amazon Cognito envía un mensaje de bienvenida.
`CustomSMSSender_ForgotPassword`	Un usuario solicita un código para restablecer su contraseña.

TriggerSource value	Evento
CustomSMSSender_ResendCode	Un usuario solicita un código nuevo para confirmar su registro.
CustomSMSSender_VerifyUserAttribute	Un usuario crea una dirección de correo electrónico nueva o un atributo de número de teléfono y Amazon Cognito envía un código para verificar el atributo.
CustomSMSSender_UpdateUserAttribute	Un usuario actualiza una dirección de correo electrónico o un atributo de número de teléfono y Amazon Cognito envía un código para verificar el atributo.
CustomSMSSender_Authentication	Un usuario configurado con autenticación multifactor (MFA) por SMS inicia sesión.
CustomSMSSender_AdminCreateUser	Crea un nuevo usuario en su grupo de usuarios y Amazon Cognito le envía una contraseña temporal.

Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito

Los grupos de usuarios de Amazon Cognito se integran con Amazon Pinpoint para proporcionar análisis de dichos grupos y enriquecer los datos de los usuarios para las campañas de Amazon Pinpoint. Con Amazon Pinpoint, se ofrecen análisis y campañas dirigidas a públicos específicos para mejorar la interacción de los usuarios con las aplicaciones móviles mediante notificaciones push. Gracias a la compatibilidad de los análisis de Amazon Pinpoint con los grupos de usuarios de Amazon Cognito, puede realizar seguimiento de los registros de los grupos de usuarios, los inicios de sesión, las autenticaciones fallidas, los usuarios activos diarios (DAU) y los usuarios activos mensuales (MAU) desde la consola de Amazon Pinpoint. Puede analizar los datos por intervalo de fechas o por atributos como plataforma del dispositivo, idioma del dispositivo o versión de la aplicación.

También puede configurar atributos personalizados para su aplicación. Estos atributos pueden usarse posteriormente para segmentar los usuarios en Amazon Pinpoint y enviarles notificaciones push específicas. Si selecciona Share user attribute data with Amazon Pinpoint (Compartir datos de atributos del usuario con Amazon Pinpoint) en la pestaña Analytics (Análisis) de la consola de Amazon Cognito, Amazon Pinpoint crea puntos de conexión adicionales para las direcciones de correo electrónico y los números de teléfono.

Al activar los análisis de Amazon Pinpoint en el grupo de usuarios con la consola de Amazon Cognito, también crea un rol vinculado a un servicio que Amazon Cognito asume cuando realiza una solicitud a la API a Amazon Pinpoint para el grupo de usuarios. La entidad principal de IAM que agrega la configuración de análisis debe tener permisos CreateServiceLinkedRole. El rol vinculado al servicio es AWSServiceRoleForAmazonCognitoIdp. Para obtener más información, consulte Uso de roles vinculados a servicios para Amazon Cognito.

Cuando aplique `AnalyticsConfiguration` al cliente de la aplicación en la API de Amazon Cognito, puede asignar un rol de IAM personalizado para Amazon Pinpoint y un ID externo para asumir el rol. El rol debe confiar en la entidad principal del servicio `cognito-idp` y, si la política de confianza del rol requiere un ID externo, debe coincidir con `AnalyticsConfiguration`. Debe conceder los permisos `cognito-idp:Describe*` del rol y los siguientes permisos para el proyecto de Amazon Pinpoint.

- `mobiletargeting:UpdateEndpoint`
- `mobiletargeting:PutEvents`

Disponibilidad de regiones de Amazon Cognito y Amazon Pinpoint

En la siguiente tabla se muestran las asignaciones de Región de AWS entre Amazon Cognito y Amazon Pinpoint que cumplen una de las siguientes condiciones.

- Solo puede utilizar un proyecto de Amazon Pinpoint en la región de Este de EE. UU. (Norte de Virginia) (us-east-1).
- Puede utilizar un proyecto de Amazon Pinpoint en la misma región o en la región de Este de EE. UU. (Norte de Virginia) (us-east-1)

De forma predeterminada, Amazon Cognito solo puede enviar análisis a un proyecto de Amazon Pinpoint en la misma Región de AWS. Las excepciones a esta regla son las regiones de la tabla siguiente y las regiones en las que Amazon Pinpoint no está disponible.

Amazon Pinpoint ya no está disponible en las siguientes regiones. Los grupos de usuarios de Amazon Cognito de estas regiones no admiten análisis.

- Europe (Milan)
- Middle East (Bahrain)
- Asia-Pacífico (Osaka)
- Israel (Tel Aviv)
- África (Ciudad del Cabo)
- Asia-Pacífico (Yakarta)

En la tabla se muestra la relación entre la región en la que creó su grupo de usuarios de Amazon Cognito y la región correspondiente en Amazon Pinpoint. Debe configurar su proyecto de Amazon Pinpoint en una región disponible para integrarlo con Amazon Cognito.

Región del grupo de usuarios de Amazon Cognito	Región del proyecto de Amazon Pinpoint
ap-northeast-1	us-east-1
ap-northeast-2	us-east-1
ap-south-1	us-east-1, ap-south-1
ap-southeast-1	us-east-1
ap-southeast-2	us-east-1, ap-southeast-2
ca-central-1	us-east-1
eu-central-1	us-east-1, eu-central-1
eu-west-1	us-east-1, eu-west-1
eu-west-2	us-east-1
us-east-1	us-east-1
us-east-2	us-east-1

Región del grupo de usuarios de Amazon Cognito	Región del proyecto de Amazon Pinpoint
us-west-2	us-east-1, us-west-2

Ejemplos de mapeo de regiones

- Si crea un grupo de usuarios en ap-northeast-1, podrá crear su proyecto de Amazon Pinpoint en us-east-1.
- Si crea un grupo de usuarios en ap-south-1, puede crear el proyecto de Amazon Pinpoint en us-east-1 o ap-south-1.

> **Note**
>
> Para todas las Regiones de AWS, excepto las de la tabla anterior, Amazon Cognito solo puede utilizar un proyecto de Amazon Pinpoint en la misma región que su grupo de usuarios. Si Amazon Pinpoint no está disponible en la región en la que ha creado su grupo de usuarios y no aparece en la tabla, significa que Amazon Cognito no es compatible con los análisis de Amazon Pinpoint en esa región. Para obtener información detallada sobre las Región de AWS, consulte Amazon Pinpoint endpoints and quotas (Puntos de conexión y cuotas de Amazon Pinpoint).

Especificación de la configuración del análisis de Amazon Pinpoint (AWS Management Console)

Puede configurar su grupo de usuarios de Amazon Cognito para enviar datos de análisis a Amazon Pinpoint. Amazon Cognito solo envía datos de análisis a Amazon Pinpoint para los usuarios locales. Después de configurar su grupo de usuarios para asociarlo a un proyecto de Amazon Pinpoint, deberá incluir `AnalyticsMetadata` en sus solicitudes de API. Para obtener más información, consulte Integración de su aplicación con Amazon Pinpoint.

Para definir los ajustes de análisis

1. Diríjase a la consola de Amazon Cognito. Es posible que se le soliciten sus credenciales de AWS.

2. Seleccione User Pools (Grupos de usuarios) y elija un grupo de usuarios existente de la lista.
3. Elija la pestaña App integration (Integración de aplicaciones).
4. En App clients and analytics (Clientes y análisis de aplicaciones), elija un valor de App client name (Nombre de cliente de aplicación) existente de la lista.
5. En Pinpoint analytics (Análisis de Pinpoint), elija Enable (Activar).
6. Elija un valor de Pinpoint Region (Región de Pinpoint).
7. Elija un valor de Amazon Pinpoint project (Proyecto de Amazon Pinpoint) o seleccione Create Amazon Pinpoint project (Crear proyecto de Amazon Pinpoint).

> **Note**
>
> El ID de proyecto de Amazon Pinpoint es una cadena de 32 caracteres única para cada proyecto de Amazon Pinpoint. Este aparece en la consola de Amazon Pinpoint.
> Puede mapear varias aplicaciones de Amazon Cognito a un único proyecto de Amazon Pinpoint. Sin embargo, no puede mapear una aplicación de Amazon Cognito a más de un proyecto de Amazon Pinpoint.
> En Amazon Pinpoint, cada proyecto debe ser una sola aplicación. Por ejemplo, si un desarrollador de juegos tiene dos juegos, cada uno debe ser un proyecto de Amazon Pinpoint distinto, incluso si en ambos juegos se utiliza el mismo grupo de usuarios de Amazon Cognito. Para obtener más información sobre los proyectos de Amazon Pinpoint, consulte Creación de un proyecto en Amazon Pinpoint.

8. En User data sharing (Uso compartido de datos de usuario), elija Share user data with Amazon Pinpoint (Compartir datos de usuario con Amazon Pinpoint) si desea que Amazon Cognito envíe direcciones de correo electrónico y números de teléfono a Amazon Pinpoint y cree puntos de conexión adicionales para los usuarios. Después de que sus usuarios verifiquen su dirección de correo electrónico y su número de teléfono, Amazon Cognito solo los comparte con Amazon Pinpoint si están disponibles en la cuenta de usuario.

> **Note**
>
> Con el punto de enlace, se identifica de forma exclusiva el dispositivo de un usuario al que puede enviar notificaciones push con Amazon Pinpoint. Para obtener más información sobre los puntos de enlace, consulte Adición de puntos de enlace en la Guía para desarrolladores de Amazon Pinpoint.

9. Elija Guardar cambios.

Especificación de la configuración del análisis de Amazon Pinpoint (AWS CLI y la API de AWS)

Utilice los siguientes comandos con el fin de especificar la configuración del análisis de Amazon Pinpoint para su grupo de usuarios.

Para especificar la configuración de análisis para la aplicación cliente existente del grupo de usuarios en momento de crear dicha aplicación

- AWS CLI: `aws cognito-idp create-user-pool-client`
- API de AWS: CreateUserPoolClient

Para actualizar la configuración de análisis para la aplicación cliente existente del grupo de usuarios

- AWS CLI: `aws cognito-idp update-user-pool-client`
- API de AWS: UpdateUserPoolClient

> **Note**
>
> Amazon Cognito admite integraciones dentro de las regiones cuando se utiliza `ApplicationArn`

Integración de su aplicación con Amazon Pinpoint

Puede publicar metadatos de análisis en Amazon Pinpoint para usuarios locales de Amazon Cognito en la API del grupo de usuarios.

Usuarios locales

: Los usuarios que se registraron para crear una cuenta o que se crearon en su grupo de usuarios en lugar de iniciar sesión mediante un proveedor de identidades (IdP) externo.

API de grupos de usuarios

: Las operaciones que puede integrar con un SDK de AWS, mediante una aplicación con una interfaz de usuario (UI) personalizada. No puede transferir los metadatos de análisis de los

usuarios federados o locales que inician sesión a través de la interfaz de usuario alojada. Consulte la Referencia de la API de Amazon Cognito para una lista de las operaciones de la API de los grupos de usuarios.

Tras configurar su grupo de usuarios para publicar en una campaña, Amazon Cognito pasa los metadatos a Amazon Pinpoint para las siguientes operaciones de la API.

- `AdminInitiateAuth`
- `AdminRespondToAuthChallenge`
- `ConfirmForgotPassword`
- `ConfirmSignUp`
- `ForgotPassword`
- `InitiateAuth`
- `ResendConfirmationCode`
- `RespondToAuthChallenge`
- `SignUp`

Para transferir metadatos sobre la sesión de su usuario a su campaña de Amazon Pinpoint, incluya un valor `AnalyticsEndpointId` en el parámetro `AnalyticsMetadata` de tu solicitud de API. Para ver un ejemplo de JavaScript, consulte ¿Por qué no aparecen los análisis de mi grupo de usuarios de Amazon Cognito en mi panel de Amazon Pinpoint? en el AWSCentro de conocimientos.

Configuración de análisis de grupo de usuarios

Con el análisis de Amazon Pinpoint, puede realizar seguimiento de los registros, inicios de sesión y errores de autenticación de los grupos de usuario de Amazon Cognito, así como de los usuarios activos diarios (DAU) y mensuales (MAU). También puede utilizar AWS Mobile SDK for Android o AWS Mobile SDK for iOS para configurar atributos de usuario que sean específicos de su aplicación. Estos atributos pueden usarse posteriormente para segmentar los usuarios en Amazon Pinpoint y enviarles notificaciones push específicas.

En la pestaña Integración de aplicaciones en Clientes de aplicaciones y análisis, puede ir a un cliente de aplicaciones existente o crear uno nuevo. En la configuración del cliente de aplicaciones, puede especificar un proyecto de Amazon Pinpoint que desee usar con la aplicación. Para obtener más información, consulte Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito.

> **Note**
>
> Amazon Pinpoint está disponible en varias regiones de AWS en América del Norte, Europa, Asia y Oceanía. Las regiones de Amazon Pinpoint incluyen la API de Amazon Pinpoint. Si Amazon Cognito admite una región de Amazon Pinpoint, enviará eventos a proyectos de Amazon Pinpoint dentro de la misma región de Amazon Pinpoint. Si una región no es compatible con Amazon Pinpoint, Amazon Cognito solo admitirá el envío de eventos en us-east-1. Para obtener información detallada sobre la región de Amazon Pinpoint, consulte Cuotas y puntos de enlace de Amazon Pinpoint y Uso de Amazon Pinpoint Analytics con grupos de usuarios de Amazon Cognito.

Para añadir análisis y campañas

1. Elija Add analytics and campaigns (Añadir análisis y campañas).
2. Elija un valor de Cognito app client (Cliente de aplicación de Cognito) en la lista.
3. Para mapear la aplicación de Amazon Cognito a un proyecto de Amazon Pinpoint, elija el proyecto de Amazon Pinpoint de la lista.

   > **Note**
   >
   > El ID de proyecto de Amazon Pinpoint es una cadena de 32 caracteres única para cada proyecto de Amazon Pinpoint. Aparece en la consola de Amazon Pinpoint.
   > Puede mapear varias aplicaciones de Amazon Cognito a un único proyecto de Amazon Pinpoint. Sin embargo, no puede mapear una aplicación de Amazon Cognito a más de un proyecto de Amazon Pinpoint.
   > En Amazon Pinpoint, cada proyecto debe ser una sola aplicación. Por ejemplo, si un desarrollador de juegos tiene dos juegos, cada uno debe ser un proyecto de Amazon Pinpoint diferente, incluso si en ambos juegos se utiliza el mismo grupo de usuarios de Amazon Cognito.

4. Elija Share user attribute data with Amazon Pinpoint (Compartir datos de atributos de usuario con Amazon Pinpoint) si desea que Amazon Cognito envíe las direcciones de correo electrónico y los números de teléfono a Amazon Pinpoint a fin de crear otros puntos de enlace para los usuarios.

> **Note**
>
> Un punto de conexión identifica de forma exclusiva un dispositivo de un usuario al que puede enviar notificaciones push con Amazon Pinpoint. Para obtener más información sobre los puntos de conexión, consulte Adición de puntos de conexión en la Guía para desarrolladores de Amazon Pinpoint.

5. Ingrese un rol de IAM que ya haya creado o elija Create new role (Crear nuevo rol) para crear uno nuevo en la consola de IAM.
6. Elija Guardar cambios.
7. Para definir asignaciones de aplicación adicionales, elija Add app mapping (Añadir asignación de aplicación).
8. Elija Guardar cambios.

Administración de usuarios en el grupo de usuarios

Después de crear un grupo de usuarios, puede crear, confirmar y administrar cuentas de usuarios. Con los grupos de usuarios de Amazon Cognito, puede administrar sus usuarios y su acceso a los recursos mediante el mapeo de roles de IAM a los grupos.

Puede importar los usuarios a un grupo de usuarios mediante un desencadenador de Lambda para la migración de usuarios. Este enfoque permite la migración fluida de usuarios desde su directorio de usuarios existente a grupos de usuarios al iniciar sesión en el grupo de usuarios por primera vez.

Temas

- Configuración de políticas para la creación de usuarios
- Inscripción y confirmación de cuentas de usuario
- Creación de cuentas de usuario como administrador
- Agregar grupos a un grupo de usuarios
- Gestión y búsqueda de cuentas de usuario
- Recuperación de cuentas de usuario
- Importación de usuarios a un grupo de usuarios
- Custom pool attributes (
- Adición de requisitos de contraseña para los grupos de usuarios

Configuración de políticas para la creación de usuarios

Su grupo de usuarios puede permitir que los usuarios se registren o puede crearlos como administrador. También puede controlar qué parte del proceso de comprobación y confirmación tras el registro queda en manos de sus usuarios. Por ejemplo, es posible que desee revisar los registros y aceptarlos en función de un proceso de validación externo. Esta configuración, o política de creación de usuarios por parte del administrador, también establece el tiempo que pasará antes de que un usuario ya no pueda confirmar su cuenta de usuario.

Amazon Cognito puede satisfacer las necesidades de sus clientes públicos como plataforma de gestión de acceso e identidad de los clientes (CIAM) para su software. Un grupo de usuarios que acepta el registro y tiene un cliente de aplicaciones, con o sin una interfaz de usuario alojada, crea un perfil de usuario para cualquier usuario de Internet que conozca su ID de cliente de aplicación, visible públicamente, y solicite registrarse. Un perfil de usuario registrado puede recibir tokens de identidad y acceso, así como acceder a los recursos que haya autorizado para su aplicación. Antes de activar el registro en su grupo de usuarios, revise sus opciones y asegúrese de que la configuración cumpla con sus estándares de seguridad. Configure con cuidado Habilitar el registro automático y `AllowAdminCreateUserOnly`, tal como se describe en los siguientes procedimientos.

AWS Management Console

La pestaña Experiencia de inicio de sesión de su grupo de usuarios y el paso Configurar la experiencia de registro del asistente de creación de grupos de usuarios contienen algunos de los ajustes para el registro y la creación administrativa de usuarios en su grupo de usuarios.

Para configurar la experiencia de registro

1. En Verificación y confirmación asistidas por Cognito, elija si desea Permitir que Cognito envíe mensajes automáticamente para verificar y confirmar. Con esta configuración habilitada, Amazon Cognito envía un mensaje de correo electrónico o mensaje SMS a los nuevos usuarios con un código que deben presentar a su grupo de usuarios. De este modo, se confirma que son los propietarios de la dirección de correo electrónico o el número de teléfono, se establece el atributo equivalente como verificado y se confirma la cuenta de usuario para el inicio de sesión. Los Atributos para verificar que elija determinan los métodos de entrega y los destinos de los mensajes de verificación.
2. La verificación de los cambios en los atributos no es importante a la hora de crear usuarios, pero está relacionada con la verificación de los atributos. Puede permitir que los usuarios que hayan cambiado sus atributos de inicio de sesión, pero aún no los hayan verificado,

continúen iniciando sesión con su nuevo valor de atributo o con el original. Para obtener más información, consulte Verificación al cambiar los usuarios su correo electrónico o su número de teléfono.

3. Los atributos obligatorios muestran los atributos a los que se debe proporcionar un valor para que un usuario pueda registrarse o se puede crear un usuario. Solo puede establecer los atributos necesarios en el asistente de creación de grupos de usuarios.
4. Los atributos personalizados son importantes para el proceso de creación y registro de usuarios, ya que solo se puede establecer un valor para los atributos personalizados inmutables al crear un usuario por primera vez. Para obtener más información sobre atributos personalizados, consulte Custom attributes (Atributos personalizados).
5. En Registro de autoservicio, seleccione Permitir el registro automático si desea que los usuarios puedan generar una nueva cuenta con la API de `SignUp` no autenticada. Si deshabilita el registro automático, solo podrá crear nuevos usuarios como administrador, en la consola de Amazon Cognito o con solicitudes de API de AdminCreateUser. En un grupo de usuarios en el que el registro automático está inactivo, las solicitudes de la API de SignUp devuelven `NotAuthorizedException` y la UI alojada no muestra ningún Registro.

En el caso de los grupos de usuarios en los que planea crear usuarios como administrador, puede configurar la duración de sus contraseñas temporales en la pestaña Experiencia de inicio de sesión, en Contraseñas temporales establecidas por los administradores que caducan en.

Otro elemento importante de la creación de usuarios como administrador es el mensaje de invitación. Cuando crea un usuario nuevo, Amazon Cognito le envía un mensaje con un enlace a su aplicación para que pueda iniciar sesión por primera vez. Personalice esta plantilla de mensaje en la pestaña Mensajería, en Plantillas de mensaje.

Puede configurar clientes de aplicaciones confidenciales, normalmente aplicaciones web, con un secreto de cliente que impida el registro sin el secreto de cliente de la aplicación. Como práctica recomendada de seguridad, no distribuya los secretos de los clientes de aplicaciones en clientes de aplicaciones públicos, normalmente aplicaciones móviles. Puede crear clientes de aplicaciones con secretos de cliente en la pestaña Integración de aplicaciones de la consola de Amazon Cognito.

Amazon Cognito user pools API

Puede configurar mediante programación los parámetros para la creación de usuarios en un grupo de usuarios en una solicitud de la API CreateUserPool o UpdateUserPool.

El elemento AdminCreateUserConfig establece los valores de las siguientes propiedades de un grupo de usuarios.

1. Habilitación de registro de autoservicio
2. El mensaje de invitación que se envía a los nuevos usuarios creados por el administrador

El siguiente ejemplo, cuando se añade a un cuerpo completo de solicitud de la API, establece un grupo de usuarios con el registro de autoservicio inactivo y un correo electrónico de invitación básico.

```
"AdminCreateUserConfig": {
      "AllowAdminCreateUserOnly": true,
      "InviteMessageTemplate": {
         "EmailMessage": "Your username is {username} and temporary password is
 {####}.",
         "EmailSubject": "Welcome to ExampleApp",
         "SMSMessage": "Your username is {username} and temporary password is
 {####}."
      }
   }
```

Los siguientes parámetros adicionales de una solicitud de API CreateUserPool o UpdateUserPool rigen la creación de nuevos usuarios.

AutoVerifiedAttributes

Los atributos, direcciones de correo electrónico o números de teléfono a los que desea enviar automáticamente un mensaje al registrar un nuevo usuario.

Políticas

La política de contraseñas del grupo de usuarios.

Esquema

Los atributos personalizados del grupo de usuarios. Son importantes para el proceso de creación y registro de usuarios, ya que solo se puede establecer un valor para los atributos personalizados inmutables al crear un usuario por primera vez.

Este parámetro también establece los atributos necesarios para el grupo de usuarios. El texto siguiente, cuando se inserta en el elemento `Schema` de un cuerpo completo de solicitud de API, establece el atributo `email` según sea necesario.

```
{
            "Name": "email",
            "Required": true
}
```

Inscripción y confirmación de cuentas de usuario

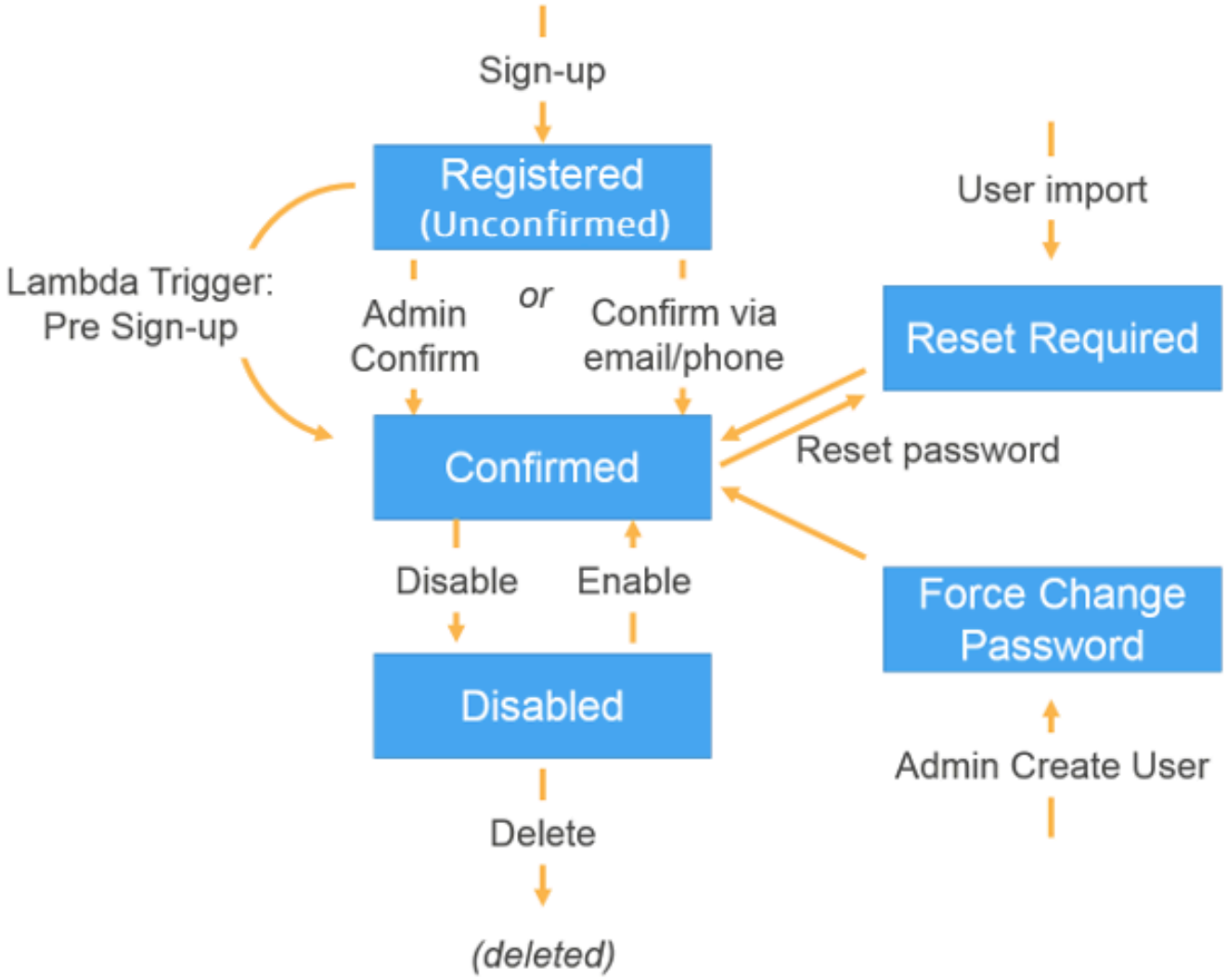
Las cuentas de usuario se añaden al grupo de usuarios siguiendo una de las formas siguientes:

- El usuario se suscribe a la aplicación cliente del grupo de usuarios. Puede ser una aplicación móvil o web.
- Puede importar la cuenta de usuario al grupo de usuarios. Para obtener más información, consulte Importación de usuarios en grupos de usuarios desde un archivo CSV.
- Puede crear la cuenta de usuario en el grupo de usuarios e invitar al usuario a iniciar sesión. Para obtener más información, consulte Creación de cuentas de usuario como administrador.

Los usuarios que se inscriben se deben confirmar antes de poder iniciar sesión. Los usuarios importados y creados ya están confirmados, pero deben crear su contraseña la primera vez que inicien sesión. En la sección siguiente se explica el proceso de confirmación y la verificación de teléfono y correo electrónico.

Información general sobre la confirmación de una cuenta de usuario

En el diagrama siguiente se ilustra el proceso de confirmación:



Una cuenta de usuario puede tener cualquiera de los estados siguientes:

Registrada (sin confirmar)

El usuario se ha inscrito correctamente, pero no puede iniciar sesión hasta que la cuenta de usuario esté confirmada. En este estado, el usuario está habilitado, pero no confirmado.

Los nuevos usuarios que se inscriben empiezan con este estado.

Confirmada

La cuenta de usuario está confirmada y el usuario puede iniciar sesión. Cuando un usuario introduce un código o sigue un enlace de correo electrónico para confirmar su cuenta de usuario, dicho correo electrónico o número de teléfono se verifica automáticamente. El código o enlace es válido durante 24 horas.

Si el administrador o un disparador Lambda de preinscripción ha confirmado la cuenta de usuario, es posible que no haya un correo electrónico o un número de teléfono asociado a la cuenta.

Restablecimiento de contraseña requerido

La cuenta de usuario está confirmada, pero el usuario debe solicitar un código y restablecer su contraseña para poder iniciar sesión.

Las cuentas de usuario que el administrador o el desarrollador importan empiezan con este estado.

Obligar a cambiar la contraseña

La cuenta de usuario está confirmada y el usuario puede iniciar sesión con una contraseña temporal, pero la primera vez que inicie sesión, el usuario debe cambiar la contraseña para poder hacer cualquier cosa.

Las cuentas de usuario que el administrador o el desarrollador crean empiezan con este estado.

Deshabilitad

Para poder eliminar una cuenta de usuario, debe deshabilitar el acceso de inicio de sesión para ese usuario.

Verificación de la información de contacto durante el registro

Tal vez desee que, cuando se registren nuevos usuarios en la aplicación, proporcionen al menos un método de contacto. Por ejemplo, con la información de contacto de los usuarios, podría:

- Enviar una contraseña temporal cuando un usuario decida restablecer su contraseña.
- Avisar a los usuarios cuando se actualicen sus datos personales o financieros.
- Enviar mensajes promocionales, como ofertas o descuentos especiales.
- Enviar resúmenes de cuenta o recordatorios de facturación.

En casos de uso como estos, es importante que envíe sus mensajes a un destino verificado. De lo contrario, los mensajes podrían enviarse a una dirección de correo electrónico o un número de teléfono no válidos que se hayan especificado de forma incorrecta. O lo que es peor, podría enviarse información confidencial a agentes malintencionados que se hagan pasar por los usuarios.

A fin de garantizar que los mensajes se envíen solo a las personas indicadas, configure el grupo de usuarios de Amazon Cognito para que los usuarios tengan que proporcionar la siguiente información al registrarse:

a. Una dirección de correo electrónico o un número de teléfono.
b. Un código de verificación que Amazon Cognito envía a esa dirección de correo electrónico o número de teléfono. Si han pasado 24 horas y el código o enlace de tu usuario ya no es válido, llama a la operación de ResendConfirmationCodeAPI para generar y enviar un código o enlace nuevo.

Al proporcionar el código de verificación, el usuario demuestra que tiene acceso a la bandeja de correo o al teléfono donde se recibió el código. Cuando el usuario proporciona el código, Amazon Cognito actualiza la información sobre él en el grupo de usuarios del modo siguiente:

- Estableciendo el estado del usuario en `CONFIRMED`.
- Actualizando los atributos del usuario para indicar que la dirección de correo electrónico o el número de teléfono se han verificado.

Para ver esta información, puede utilizar la consola de Amazon Cognito. O bien, puedes usar la operación de la `AdminGetUser` API AWS CLI, el `admin-get-user` comando que contiene o la acción correspondiente en uno de los AWS SDK.

Si un usuario tiene un método de contacto verificado, Amazon Cognito le envía de manera automática un mensaje cuando solicita restablecer la contraseña.

Para configurar el grupo de usuarios de forma que se solicite la verificación del correo electrónico o del teléfono

Cuando se verifican las direcciones de correo electrónico y los números de teléfono de los usuarios, se asegura de que puede ponerse en contacto con ellos. Complete los siguientes pasos AWS Management Console para configurar su grupo de usuarios y solicitar que los usuarios confirmen sus direcciones de correo electrónico o números de teléfono.

> **Note**
>
> Si todavía no tiene ningún grupo de usuarios en la cuenta, consulte Introducción a los grupos de usuarios.

Para configurar el grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. En el panel de navegación, seleccione Users (Usuarios). Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
3. Elija la pestaña Sign-up experience (Experiencia de registro) y localice Attribute verification and user account confirmation (Verificación de atributos y confirmación de cuenta de usuario). Elija Edit (Editar).
4. En Verificación y confirmación asistidas por Cognito, elija si desea Permitir que Cognito envíe mensajes automáticamente para verificar y confirmar. Con esta configuración habilitada, Amazon Cognito envía mensajes a los atributos de contacto del usuario que elija cuando un usuario se registre o cuando cree un perfil de usuario. Para verificar los atributos y confirmar los perfiles de usuario para iniciar sesión, Amazon Cognito envía un código o un enlace en los mensajes a los usuarios. A continuación, los usuarios deben introducir el código en la IU para que la aplicación pueda confirmarlo en una solicitud de la API `ConfirmSignUp` o `AdminConfirmSignUp`.

> **Note**
>
> También se puede deshabilitar Cognito-assisted verification and confirmation (Verificación y confirmación asistidas por Cognito) y emplear acciones de API autenticadas o desencadenadores de Lambda para verificar atributos y confirmar usuarios.
> Si elige esta opción, Amazon Cognito no enviará códigos de verificación cuando el usuario se registre. Elija esta opción si utiliza un flujo de autenticación personalizado con el que se verifica, al menos, un método de contacto sin utilizar códigos de verificación de Amazon Cognito. Por ejemplo, es posible que desee utilizar un desencadenador Lambda previo al registro que verifique automáticamente las direcciones de correo electrónico que pertenecen a un dominio específico.
> Si no verifica la información de contacto de los usuarios, es posible que no pueda utilizar la aplicación. Recuerde que los usuarios necesitan tener verificada la información de contacto para:
>
> - Restablecer sus contraseñas: Cuando un usuario elije una opción en la aplicación con la que se llama a la acción `ForgotPassword` de la API, Amazon Cognito envía una contraseña temporal a la dirección de correo electrónico o al número de teléfono del

usuario. Amazon Cognito envía esta contraseña solo si el usuario tiene, al menos, un método de contacto verificado.

- Iniciar sesión utilizando una dirección de correo electrónico o un número de teléfono como alias: Si configura el grupo de usuarios de forma que estos alias estén permitidos, los usuarios solamente podrán iniciar sesión con un alias si dicho alias se ha verificado. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

5. Elija Attributes to verify (Atributos para verificar):

Enviar un mensaje SMS, verificar el número de teléfono

Amazon Cognito envía un mensaje SMS con un código de verificación cuando el usuario se registra. Elija esta opción si normalmente se comunica con los usuarios a través de mensajes SMS. Por ejemplo, conviene utilizar números de teléfono verificados si envía notificaciones de entrega, confirmaciones de citas o alertas. Los números de teléfono de los usuarios serán el atributo verificado cuando se confirmen las cuentas; se deben tomar medidas adicionales para verificar y comunicarse con las direcciones de correo electrónico de los usuarios.

Enviar un mensaje de correo electrónico, verificar la dirección de correo electrónico

Amazon Cognito envía un mensaje de correo electrónico con un código de verificación cuando el usuario se registra. Elija esta opción si normalmente se comunica con los usuarios a través del correo electrónico. Por ejemplo, conviene utilizar direcciones de correo electrónico verificadas para enviar facturas, resúmenes de pedidos u ofertas especiales. Las direcciones de correo electrónico de los usuarios serán el atributo verificado cuando se confirmen las cuentas; se deben tomar medidas adicionales para verificar y comunicarse con los números de teléfono de los usuarios.

Enviar un mensaje SMS si hay un número de teléfono disponible; de lo contrario, enviar un mensaje de correo electrónico

Elija esta opción si no quiere que todos los usuarios tengan el mismo método de contacto verificado. En este caso, la página de registro de la aplicación podría pedir a los usuarios que verifiquen únicamente el método de contacto preferido. Cuando Amazon Cognito envía un código de verificación, lo envía mediante el método de contacto especificado en la solicitud `SignUp` de la aplicación. Si un usuario proporciona una dirección de correo electrónico y un número de teléfono y se especifican los dos métodos de contacto en la solicitud `SignUp` de la aplicación, Amazon Cognito solo envía el código de verificación al número de teléfono.

Si solicita a los usuarios que verifiquen la dirección de correo electrónico y el número de teléfono, elija esta opción. Amazon Cognito verificará uno de los métodos de contacto cuando el usuario se registre, mientras que la aplicación deberá verificar el otro cuando el usuario inicie sesión. Para obtener más información, consulte Si solicita a los usuarios que confirmen tanto el correo electrónico como el número de teléfono.

6. Elija Save changes (Guardar cambios).

Flujo de autenticación con la verificación del correo electrónico o el teléfono

Si el grupo de usuarios obliga a los usuarios a verificar los datos de contacto, la aplicación debe facilitar lo siguiente cuando el usuario se registre:

1. Un usuario inicia sesión en su aplicación introduciendo un nombre de usuario, un número de teléfono o una dirección de correo electrónico y, posiblemente, otros atributos.
2. El servicio de Amazon Cognito recibe la solicitud de registro de la aplicación. Después de verificar que la solicitud contiene todos los atributos necesarios para la inscripción, el servicio completa el proceso de inscripción y envía un código de confirmación al teléfono (en un mensaje SMS) o al correo electrónico del usuario. El código es válido durante 24 horas.
3. El servicio indica a la aplicación que la inscripción se ha completado y que la cuenta de usuario está pendiente de confirmación. La respuesta contiene información acerca de dónde se ha enviado el código de confirmación. En este momento, la cuenta de usuario está sin confirmar y la dirección de correo electrónico y el número de teléfono del usuario están sin verificar.
4. Ahora, la aplicación puede instar al usuario a que introduzca el código de confirmación. No es necesario que el usuario introduzca el código de inmediato. Sin embargo, no podrá iniciar sesión hasta después de introducir el código de confirmación.
5. El usuario introduce el código de confirmación en la aplicación.
6. La aplicación llama a `ConfirmSignUp` para enviar el código al servicio de Amazon Cognito que lo verifica y, si es correcto, establece la cuenta del usuario en el estado confirmado. Después de confirmar con éxito la cuenta del usuario, el servicio de Amazon Cognito marca de forma automática el atributo que se utilizó para confirmar (dirección de correo electrónico o número de teléfono) como verificado. A menos que el valor de este atributo cambie, el usuario no tendrá que volver a verificarlo.
7. En este punto, la cuenta de usuario se encuentra en estado confirmado y el usuario puede iniciar sesión.

Si solicita a los usuarios que confirmen tanto el correo electrónico como el número de teléfono

Amazon Cognito solo verificará uno de los métodos de contacto cuando el usuario se registre. En los casos en que Amazon Cognito deba elegir entre la verificación por dirección de correo electrónico o número de teléfono, elegirá el número de teléfono y enviará un código de verificación por mensaje SMS. Por ejemplo, si configura el grupo de usuarios de forma que los usuarios puedan verificarse por dirección de correo electrónico o número de teléfono, y la aplicación proporciona estos atributos después del registro, Amazon Cognito solo verificará el número de teléfono. Una vez que un usuario verifica el número de teléfono, Amazon Cognito establece el estado del usuario en `CONFIRMED`, por lo que el usuario tiene permiso para iniciar sesión en la aplicación.

Una vez que el usuario inicia sesión, la aplicación puede dar la opción de verificar el método de contacto que no se ha verificado durante el registro. Para verificar este segundo método, la aplicación llama a la acción `VerifyUserAttribute` de la API. Tenga en cuenta que para esta acción se requiere un parámetro `AccessToken` y que Amazon Cognito solo proporciona tokens de acceso a los usuarios autenticados. Por lo tanto, solamente puede verificar el segundo método de contacto una vez que el usuario ha iniciado sesión.

Si necesita que los usuarios verifiquen la dirección de correo electrónico y el número de teléfono, haga lo siguiente:

1. Configure el grupo de usuarios para que permita a los usuarios verificar la dirección de correo electrónico o el número de teléfono.
2. En el flujo de registro de la aplicación, pida a los usuarios que proporcionen una dirección de correo electrónico y un número de teléfono. Llame a la acción `SignUp` de la API y proporcione la dirección de correo electrónico y el número de teléfono en el parámetro `UserAttributes`. En ese momento, Amazon Cognito envía un código de verificación al teléfono del usuario.
3. En la interfaz de la aplicación, muestre una página de confirmación en la que el usuario pueda especificar el código de verificación. Confirme el usuario llamando a la acción `ConfirmSignUp` de la API. En ese momento, el estado del usuario es `CONFIRMED` y el número de teléfono del usuario está verificado, aunque la dirección de correo electrónico no lo está.
4. Muestre la página de inicio de sesión y autentique el usuario llamando a la acción `InitiateAuth` de la API. Cuando el usuario esté autenticado, Amazon Cognito devolverá un token de acceso a la aplicación.
5. Llame a la acción `GetUserAttributeVerificationCode` de la API. Especifique los siguientes parámetros en la solicitud:

- `AccessToken`: es el token de acceso que devuelve Amazon Cognito una vez que el usuario inicia sesión.
- `AttributeName`: especifique `"email"` como el valor del atributo.

Amazon Cognito envía un código de verificación a la dirección de correo electrónico del usuario.

6. Muestre una página de confirmación en la que el usuario pueda especificar el código de verificación. Cuando el usuario envíe el código, llame a la acción `VerifyUserAttribute` de la API. Especifique los siguientes parámetros en la solicitud:

   - `AccessToken`: es el token de acceso que devuelve Amazon Cognito una vez que el usuario inicia sesión.
   - `AttributeName`: especifique `"email"` como el valor del atributo.
   - `Code`: es el código de verificación que proporciona el usuario.

   En este momento, se verifica la dirección de correo electrónico.

Permitir que los usuarios se registren en la aplicación, pero con confirmación del administrador del grupo de usuarios

Es posible que no desees que el grupo de usuarios envíe automáticamente mensajes de verificación al grupo de usuarios, pero aun así quieras permitir que cualquier persona se registre para obtener una cuenta. Este modelo deja espacio, por ejemplo, para la revisión humana de las nuevas solicitudes de registro y para la validación y el procesamiento por lotes de los registros. Puede confirmar las nuevas cuentas de usuario en la consola de Amazon Cognito o mediante la operación de API autenticada por IAM. AdminConfirmSignUp Puede confirmar las cuentas de usuario como administrador si el grupo de usuarios envía mensajes de verificación o no.

Solo puede confirmar el registro de un usuario en el autoservicio con esta técnica. Para confirmar un usuario que cree como administrador, cree una solicitud de AdminSetUserPasswordAPI con el valor establecido en. `Permanent True`

1. Un usuario inicia sesión en su aplicación introduciendo un nombre de usuario, un número de teléfono o una dirección de correo electrónico y, posiblemente, otros atributos.
2. El servicio de Amazon Cognito recibe la solicitud de registro de la aplicación. Después de verificar que la solicitud contiene todos los atributos necesarios para la inscripción, el servicio

completa el proceso de inscripción e indica a la aplicación que la inscripción está completa y pendiente de confirmación. En este punto, el estado de la cuenta del usuario es no confirmado. El usuario solo podrá iniciar sesión cuando la cuenta esté confirmada.

3. Confirme la cuenta del usuario. Debes iniciar sesión en la solicitud de API AWS Management Console o firmar tu solicitud con AWS credenciales para confirmar la cuenta.

   a. Para confirmar un usuario en la consola de Amazon Cognito, vaya a la pestaña Usuarios, elija el usuario que desea confirmar y, en el menú Acciones, seleccione Confirmar.

   b. Para confirmar un usuario en la AWS API o la CLI, cree una solicitud de AdminConfirmSignUpAPI o admin-confirm-sign-upen AWS CLI.

4. En este punto, la cuenta de usuario se encuentra en estado confirmado y el usuario puede iniciar sesión.

Cálculo de los valores de hash secretos

Como práctica recomendada, asigne un secreto de cliente a su cliente de aplicaciones confidenciales. Cuando asigne un secreto de cliente a su cliente de aplicación, las solicitudes de API de los grupos de usuarios de Amazon Cognito deberán incorporar un hash que incluya el secreto de cliente en el cuerpo de la solicitud. Para validar su conocimiento del secreto de cliente para las operaciones de la API de las listas siguientes, concatene el secreto de cliente con el ID del cliente de aplicación y el nombre de usuario del usuario y, a continuación, codifique en base64 esa cadena.

Cuando su aplicación inicie sesión con los usuarios en un cliente que tiene un hash secreto, puede usar el valor de cualquier atributo de inicio de sesión de grupo de usuarios como elemento de nombre de usuario del hash secreto. Cuando su aplicación solicita tokens nuevos en una operación de autenticación con `REFRESH_TOKEN_AUTH`, el valor del elemento del nombre de usuario depende de sus atributos de inicio de sesión. Si su grupo de usuarios no tiene `username` como atributo de inicio de sesión, establezca el valor secreto de nombre de usuario de hash de la reclamación de `sub` del usuario a partir de su token de ID o acceso. Cuando `username` es un atributo de inicio de sesión, establezca el valor de nombre de usuario de hash de secreto que aparece en la reclamación de `username`.

Las siguientes API de grupos de usuarios de Amazon Cognito aceptan un valor de hash secreto de cliente en un parámetro `SecretHash`.

- ConfirmForgotPassword
- ConfirmSignUp

- ForgotPassword
- ResendConfirmationCode
- SignUp

Además, las siguientes API aceptan un valor de hash de secreto de cliente en un parámetro `SECRET_HASH`, ya sea en parámetros de autenticación o en una respuesta de desafío.

Operación de la API	Parámetro principal para SECRET_HASH
InitiateAuth	AuthParameters
AdminInitiateAuth	AuthParameters
RespondToAuthChallenge	ChallengeResponses
AdminRespondToAuthChallenge	ChallengeResponses

El valor de hash secreto es un código de autenticación de mensajes mediante algoritmos hash con clave (HMAC) codificados en Base64 que se calcula con la clave secreta de un cliente de grupo de usuarios y un nombre de usuario más el ID de cliente en el mensaje. El pseudocódigo siguiente muestra cómo se calcula este valor. En este pseudocódigo, + indica una concatenación, `HMAC_SHA256` representa una función que genera un valor HMAC utilizando HmacSHA256 y `Base64` representa una función que genera una versión con codificación Base64 de la salida del hash.

```
Base64 ( HMAC_SHA256 ( "Client Secret Key", "Username" + "Client Id" ) )
```

Para obtener información general detallada sobre cómo calcular y usar el `SecretHash` parámetro, consulte ¿Cómo soluciono los errores «No se puede verificar el hash secreto del cliente» de mi API de grupos de usuarios de Amazon Cognito<client-id>? en el Centro de AWS conocimiento.

Puede utilizar los siguientes ejemplos de código en el código de su aplicación en el servidor.

Shell

```
echo -n "[username][app client ID]" | openssl dgst -sha256 -hmac [app client secret]
 -binary | openssl enc -base64
```

Java

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public static String calculateSecretHash(String userPoolClientId, String
 userPoolClientSecret, String userName) {
    final String HMAC_SHA256_ALGORITHM = "HmacSHA256";

    SecretKeySpec signingKey = new SecretKeySpec(
            userPoolClientSecret.getBytes(StandardCharsets.UTF_8),
            HMAC_SHA256_ALGORITHM);
    try {
        Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
        mac.init(signingKey);
        mac.update(userName.getBytes(StandardCharsets.UTF_8));
        byte[] rawHmac =
 mac.doFinal(userPoolClientId.getBytes(StandardCharsets.UTF_8));
        return Base64.getEncoder().encodeToString(rawHmac);
    } catch (Exception e) {
        throw new RuntimeException("Error while calculating ");
    }
}
```

Python

```
import sys
import hmac, hashlib, base64
username = sys.argv[1]
app_client_id = sys.argv[2]
key = sys.argv[3]
message = bytes(sys.argv[1]+sys.argv[2],'utf-8')
key = bytes(sys.argv[3],'utf-8')
secret_hash = base64.b64encode(hmac.new(key, message,
 digestmod=hashlib.sha256).digest()).decode()
print("SECRET HASH:",secret_hash)
```

Confirmación de cuentas de usuario sin verificar el correo electrónico o el número de teléfono

El desencadenador de Lambda de prerregistro se puede usar para confirmar de manera automática las cuentas de usuario en el registro, sin tener que requerir un código de confirmación ni verificar el correo electrónico o el número de teléfono. Los usuarios que se confirmen de esta forma pueden iniciar sesión de forma inmediata sin tener que recibir un código.

Con este disparador, también puede marcar un correo electrónico o un número de teléfono del usuario como verificado.

> **Note**
>
> Aunque este enfoque es práctico para los usuarios cuando están empezando, le recomendamos que compruebe automáticamente al menos el correo electrónico o el número de teléfono. De no ser así, el usuario puede quedarse sin poder recuperar la contraseña si la olvida.

Si no exige que el usuario reciba e ingrese un código de confirmación al registrarse y no verifica de manera automática el correo electrónico ni el número de teléfono en el desencadenador de Lambda de prerregistro, corre el riesgo de no tener una dirección de correo electrónico ni un número de teléfono verificados para esta cuenta de usuario. El usuario puede verificar la dirección de correo electrónico o el número de teléfono en otro momento. No obstante, si el usuario se olvida de su contraseña y no cuenta con una dirección de correo electrónico o un número de teléfono verificado, el usuario estará bloqueado fuera de la cuenta, ya que el flujo de contraseña olvidada requiere un correo electrónico o un número de teléfono verificado para enviar un código de verificación al usuario.

Verificación al cambiar los usuarios su correo electrónico o su número de teléfono

Cuando un usuario actualiza su dirección de correo electrónico o número de teléfono en la aplicación, Amazon Cognito envía inmediatamente un mensaje con un código de verificación a un usuario si configuró su grupo de usuarios para verificar automáticamente ese atributo. A continuación, el usuario debe proporcionar el código del mensaje de verificación a la aplicación. A continuación, tu aplicación envía el código en una solicitud de VerifyUserAttributeAPI para completar la verificación del nuevo valor del atributo.

Si su grupo de usuarios no requiere que los usuarios verifiquen una dirección de correo electrónico o un número de teléfono actualizados, Amazon Cognito cambia inmediatamente el valor de un atributo `email` o `phone_number` actualizado y marca el atributo como no verificado. El usuario no puede iniciar sesión con un correo electrónico o número de teléfono no verificados. Debe completar la verificación del valor actualizado antes de poder utilizar dicho atributo como alias de inicio de sesión.

Si el grupo de usuarios requiere que los usuarios verifiquen una dirección de correo electrónico o un número de teléfono actualizados, Amazon Cognito mantiene el atributo verificado y establecido en su valor original hasta que el usuario verifique el nuevo valor de atributo. Si el atributo es un alias para iniciar sesión, el usuario puede iniciar sesión con el valor del atributo original hasta que la verificación cambie el atributo por el nuevo valor. Para obtener más información acerca de cómo configurar el grupo de usuarios para exigir a los usuarios que verifiquen atributos actualizados, consulte Configuración de la verificación del correo electrónico o del teléfono.

Puede utilizar un desencadenador de Lambda de mensaje personalizado para personalizar el mensaje de verificación. Para obtener más información, consulte Desencadenador de Lambda para mensajes personalizados. Cuando la dirección de correo electrónico o el número de teléfono de un usuario no están verificados, su aplicación debe informar al usuario de que debe verificar el atributo, y proporcionar un botón o enlace para que los usuarios verifiquen su nueva dirección de correo electrónico o número de teléfono.

Procesos de confirmación y verificación para las cuentas de usuario creadas por administradores o desarrolladores

Las cuentas de usuario que un administrador o un desarrollador crean ya tienen el estado confirmado, por lo que los usuarios no tienen que introducir ningún código de confirmación. El mensaje de invitación que el servicio de Amazon Cognito envía a estos usuarios incluye el nombre de usuario y una contraseña temporal. Se pide al usuario que cambie la contraseña antes de iniciar sesión. Para obtener más información, consulte la Personalizar mensajes de correo electrónico y SMS en Creación de cuentas de usuario como administrador y el disparador para mensajes personalizados en Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda.

Procesos de confirmación y verificación para las cuentas de usuario importadas

Las cuentas de usuario que se crean mediante la función de importación de usuarios en la AWS Management Console CLI o la API (consulteImportación de usuarios en grupos de usuarios desde un archivo CSV) ya están confirmadas, por lo que los usuarios no tienen que introducir

un código de confirmación. No se envía ningún mensaje de invitación. Sin embargo, las cuentas de usuario importadas requieren que los usuarios soliciten primero un código llamando al API `ForgotPassword` y que después creen una contraseña utilizando el código entregado llamando al API `ConfirmForgotPassword` antes de iniciar sesión. Para obtener más información, consulte Obligación de que los usuarios importados restablezcan sus contraseñas.

O bien el correo electrónico o el número de teléfono del usuario deben marcarse como verificados cuando se importa la cuenta de usuario, con lo que no es necesaria ninguna verificación cuando el usuario inicia sesión.

Envío de mensajes de correo electrónico para probar la aplicación

Amazon Cognito envía mensajes de correo electrónico a los usuarios cuando crean y administran sus cuentas en la aplicación cliente del grupo de usuarios. Si configura el grupo de usuarios de forma que se exija la verificación por correo electrónico, Amazon Cognito enviará un correo electrónico cuando:

- Un usuario se registre.
- Un usuario actualice su dirección de correo electrónico.
- Un usuario realice una operación que llame a la acción `ForgotPassword` de la API.
- Usted cree una cuenta de usuario como administrador.

En función de la acción que inicie el correo electrónico, el correo electrónico contendrá un código de verificación o una contraseña temporal. Es necesario que los usuarios reciban estos correos electrónicos y comprendan el mensaje. De lo contrario, tal vez no puedan iniciar sesión ni utilizar la aplicación.

Para asegurarse de que los correos electrónicos se envíen de manera adecuada y de que el mensaje aparezca como corresponde, pruebe en la aplicación estas acciones con las que se inicia el envío de correos electrónicos desde Amazon Cognito. Por ejemplo, si utiliza la página de registro de la aplicación o la acción `SignUp` de la API, puede activar el envío de un correo electrónico registrándose con una dirección de correo electrónico de prueba. Cuando realice este tipo de pruebas, recuerde lo siguiente:

> **Importante**
>
> Cuando utilice una dirección de correo electrónico para probar acciones con las que se activa el envío de correos electrónicos desde Amazon Cognito, no utilice una dirección de

correo electrónico falsa (una que no tenga buzón de correo). Utilice una dirección de correo electrónico real que pueda recibir el correo electrónico de Amazon Cognito y que no genere un rechazo permanente.
Los rechazos permanentes se producen cuando Amazon Cognito no puede entregar el correo electrónico en el buzón del destinatario, lo que siempre sucede si el buzón de correo no existe.
Amazon Cognito limita la cantidad de correos electrónicos que pueden enviar AWS las cuentas que sufren rebotes forzosos de forma persistente.

Cuando realice acciones de prueba que inicien correos electrónicos, utilice una de las siguientes direcciones de correo electrónico para impedir que se produzcan rebotes permanentes:

- La dirección de una cuenta de correo electrónico de su propiedad y que utilice para realizar pruebas. Si utiliza su propia dirección de correo electrónico, recibirá el correo electrónico que envía Amazon Cognito. Con este correo electrónico, podrá utilizar el código de verificación para probar la experiencia de registro en la aplicación. Si ha personalizado el mensaje de correo electrónico para su grupo de usuarios, podrá comprobar que el contenido personalizado tiene el aspecto deseado.
- La dirección del simulador de bandeja de correo: success@simulator.amazonses.com. Si utiliza la dirección del simulador, Amazon Cognito enviará el correo electrónico de forma correcta, pero usted no podrá verlo. Esta opción resulta útil cuando no es necesario utilizar el código de verificación ni comprobar el mensaje de correo electrónico.
- La dirección del simulador de buzón de correo con la incorporación de una etiqueta arbitraria, p. ej., success+user1@simulator.amazonses.com o success+user2@simulator.amazonses.com. Amazon Cognito envía correos electrónicos con éxito a estas direcciones, pero no puede ver los correos que envía. Esta opción resulta útil si desea probar el proceso de registro agregando varios usuarios de prueba al grupo de usuarios y cada usuario de prueba tiene una dirección de correo electrónico diferente.

Configuración de la verificación del correo electrónico o del teléfono

Puede elegir la configuración de verificación del correo electrónico o del teléfono en la pestaña Mensajería. Para obtener más información sobre la autenticación multifactor (MFA), consulte MFA por mensaje de texto SMS.

Amazon Cognito utiliza Amazon SNS para enviar mensajes SMS. Si no ha enviado ningún mensaje SMS desde Amazon Cognito o desde ningún otro Servicio de AWS sitio, Amazon SNS podría

colocar su cuenta en el entorno limitado de SMS. Le recomendamos que envíe un mensaje de texto de prueba a un número de teléfono verificado antes de retirar la cuenta del entorno aislado de producción. Además, si tiene previsto enviar mensajes SMS a números de teléfono de destino de EE. UU., debe obtener un ID de remitente o de origen de Amazon Pinpoint. Para configurar el grupo de usuarios de Amazon Cognito para mensajes SMS, consulte Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito.

Amazon Cognito puede verificar de manera automática direcciones de correo electrónico o números de teléfono. Para realizar esta verificación, Amazon Cognito envía un código de verificación o un enlace de verificación. Para las direcciones de correo electrónico, Amazon Cognito envía un código o un enlace en un mensaje de correo electrónico. Puede elegir un Tipo de verificación de código o enlace al editar la plantilla del Mensaje de verificación en la pestaña Mensajería de la consola de Amazon Cognito. Para obtener más información, consulte Personalización de los mensajes de verificación de correo electrónico.

En el caso de los números de teléfono, Amazon Cognito envía un código en un mensaje SMS.

Amazon Cognito debe verificar un número de teléfono o una dirección de correo electrónico para confirmar a los usuarios y ayudarles a recuperar contraseñas olvidadas. Como alternativa, puede confirmar automáticamente a los usuarios con el activador Lambda previo al registro o utilizar AdminConfirmSignUpla operación de API. Para obtener más información, consulte Inscripción y confirmación de cuentas de usuario.

El código o enlace de verificación es válido durante 24 horas.

Si elige solicitar la verificación de una dirección de correo electrónico o número de teléfono, Amazon Cognito envía automáticamente el código o enlace de verificación cuando un usuario inicia sesión. Si el grupo de usuarios tiene configurado un Desencadenador de Lambda para remitentes personalizados de SMS o Desencadenador de Lambda para remitentes de correos electrónicos personalizados, se llama esa función en su lugar.

> **Notas**
>
> - El uso de mensajes de texto SMS para verificar números de teléfono se cobra por separado en Amazon SNS. No se aplica ningún cargo por el envío de mensajes de correo electrónico. Para obtener información sobre los precios de Amazon SNS, consulte Precios de SMS en todo el mundo. Para ver la lista actual de los países en los que los mensajes SMS están disponibles, consulte Regiones y países admitidos.

- Cuando realice acciones de prueba en la aplicación que generen mensajes de correo electrónico de Amazon Cognito, utilice una dirección de correo electrónico real para que Amazon Cognito pueda enviar estos mensajes sin recibir rechazos permanentes. Para obtener más información, consulte the section called “Envío de mensajes de correo electrónico para probar la aplicación”.
- El proceso de recuperación de contraseñas olvidadas requiere que el usuario verifique su correo electrónico o número de teléfono.

Important

Si un usuario inicia sesión con un número de teléfono y una dirección de correo electrónico, y la configuración del grupo de usuarios exige la verificación de ambos atributos, Amazon Cognito envía un código de verificación por mensaje SMS al número de teléfono. Amazon Cognito aún no ha verificado la dirección de correo electrónico, por lo que la aplicación debe llamar GetUserpara comprobar si hay alguna dirección de correo electrónico pendiente de verificación. Si requiere verificación, la aplicación debe llamar GetUserAttributeVerificationCodepara iniciar el flujo de verificación del correo electrónico. Luego, debe enviar el código de verificación llamando VerifyUserAttribute.

Puede ajustar su cuota de gasto en mensajes SMS para un mensaje individual Cuenta de AWS y uno solo. Los límites se aplican únicamente al precio de envío de mensajes SMS. Para obtener más información, consulte la sección ¿Qué son las cuotas de gasto a nivel de cuenta o mensaje y cómo se utilizan? en Preguntas frecuentes sobre Amazon SNS.

Amazon Cognito envía mensajes SMS mediante los recursos de Amazon SNS en Región de AWS el lugar donde creó el grupo de usuarios o en una región alternativa de Amazon SNS antigua de la siguiente tabla. La excepción son los grupos de usuarios de Amazon Cognito de la región Asia-Pacífico (Seúl). Estos grupos de usuarios utilizan su configuración de Amazon SNS en la región Asia-Pacífico (Tokio). Para obtener más información, consulte Elija el Región de AWS para los mensajes SMS de Amazon SNS.

Región de Amazon Cognito	Región alternativa de Amazon SNS heredada
US East (Ohio)	Este de EE. UU. (Norte de Virginia)

Región de Amazon Cognito	Región alternativa de Amazon SNS heredada
Asia-Pacífico (Mumbai)	Asia-Pacífico (Singapur)
Asia-Pacífico (Seúl)	Asia-Pacífico (Tokio)
Canadá (centro)	Este de EE. UU. (Norte de Virginia)
Europa (Fráncfort)	Europa (Irlanda)
Europa (Londres)	Europa (Irlanda)

Ejemplo: Si su grupo de usuarios de Amazon Cognito se encuentra en la región de Asia-Pacífico (Bombay) y ha aumentado el límite de gastos en ap-southeast-1, es posible que no quiera solicitar un aumento por separado de ap-south-1. En su lugar, puede utilizar los recursos de Amazon SNS en Asia-Pacífico (Singapur).

Verificación de actualizaciones de direcciones de correo electrónico y números de teléfono

Un atributo de dirección de correo electrónico o de número de teléfono pueden activarse y no verificarse inmediatamente después de que el usuario cambie su valor. Amazon Cognito también puede exigir que el usuario verifique el nuevo valor antes de que Amazon Cognito actualice el atributo. Cuando requiera que se verifique primero el nuevo valor, los usuarios pueden utilizar el valor original para iniciar sesión y recibir mensajes hasta que verifiquen el nuevo valor.

Cuando los usuarios pueden utilizar su dirección de correo electrónico o número de teléfono como alias de inicio de sesión en el grupo de usuarios, su nombre de inicio de sesión para un atributo actualizado depende de si necesita verificar los atributos actualizados. Cuando requiera que se verifique un atributo actualizado, el usuario puede iniciar sesión con el valor del atributo original hasta que verifique el nuevo valor. Cuando no requiera que se verifique un atributo actualizado, el usuario no puede iniciar sesión ni recibir mensajes en el valor de atributo nuevo u original hasta que verifique el nuevo valor.

Por ejemplo, el grupo de usuarios permite iniciar sesión con un alias de dirección de correo electrónico y exige que los usuarios verifiquen su dirección de correo electrónico cuando se actualice. Sue, que inicia sesión como `sue@example.com`, quiere cambiar su dirección de correo electrónico a `sue2@example.com`, pero entra accidentalmente a **s**`sue2@example.com`. Sue no recibe el correo electrónico de verificación, por lo que no puede verificar `ssue2@example.com`. Sue inicia sesión como `sue@example.com` y vuelve a enviar el formulario de su aplicación

para actualizar su dirección de correo electrónico a `sue2@example.com`. Recibe este correo electrónico, proporciona el código de verificación a su aplicación y comienza el inicio de sesión como `sue2@example.com`.

Cuando un usuario actualiza un atributo y el grupo de usuarios verifica los nuevos valores de los atributos

- Pueden iniciar sesión con el valor del atributo original antes de confirmar el código para verificar el nuevo valor.
- Pueden iniciar sesión solo con el valor del atributo nuevo después de haber confirmado el código para verificar el nuevo valor.
- Si configuras `email_verified` o `phone_number_verified` `true` incluyes una solicitud de AdminUpdateUserAttributesAPI, pueden iniciar sesión antes de confirmar el código que les envió Amazon Cognito.

Cuando un usuario actualiza un atributo y el grupo de usuarios no verifica los nuevos valores del atributo

- No pueden iniciar sesión con el valor del atributo original ni recibir mensajes con él.
- No pueden iniciar sesión con el nuevo valor de atributo ni recibir mensajes que no sean un código de confirmación en él antes de confirmar el código para comprobar el nuevo valor.
- Si configuras `email_verified` o `phone_number_verified` `true` incluyes una solicitud de AdminUpdateUserAttributesAPI, pueden iniciar sesión antes de confirmar el código que les envió Amazon Cognito.

Para requerir la verificación de atributos cuando los usuarios actualizan su dirección de correo electrónico o número de teléfono

1. Inicie sesión en la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. En el panel de navegación, elija User Pools (Grupos de usuarios) y elija el grupo de usuarios que desea editar.
3. Elija la pestaña Sign-up experience (Experiencia de registro), elija Edit (Editar) en Attribute verification and user account confirmation (Verificación de atributos y confirmación de cuenta de usuario).

4. Elija Keep original attribute value active when an update is pending (Mantener activo el valor del atributo original cuando hay una actualización pendiente).
5. En Active attribute values when an update is pending (Valores de atributos activos cuando hay una actualización pendiente), elija los atributos que desea que los usuarios verifiquen antes de que Amazon Cognito actualice el valor.
6. Elija Guardar cambios.

Para requerir la verificación de la actualización de atributos con la API de Amazon Cognito, puede configurar el `AttributesRequireVerificationBeforeUpdate` parámetro en una UpdateUserPoolsolicitud.

Autorización de Amazon Cognito para enviar mensajes SMS en su nombre.

Para enviar mensajes SMS a los usuarios en su nombre, Amazon Cognito necesita su permiso. Para conceder ese permiso, puede crear un rol AWS Identity and Access Management (de IAM). En la pestaña Mensajería de la consola de Amazon Cognito, en SMS, elija Editar para configurar un rol.

Configuración de los mensajes de verificación de correo electrónico y SMS, así como mensajes de invitación al usuario

Amazon Cognito le permite personalizar los mensajes de verificación por SMS y correo electrónico, así como los mensajes de invitación de los usuarios, para mejorar la seguridad y la experiencia de usuario de su aplicación. Con Amazon Cognito, puede elegir entre verificaciones de enlaces basadas en código o con un solo clic para adaptarse a las necesidades de su aplicación. En este tema se explica cómo puede personalizar la autenticación multifactor (MFA) y las comunicaciones de verificación en la consola de Amazon Cognito.

En la pestaña Mensajería, en Plantillas de mensajes, puede personalizar:

- Su mensaje de autenticación multifactor (MFA) del mensaje de texto SMS
- Los mensajes de verificación de SMS y correo electrónico
- El tipo de verificación por correo electrónico: código o enlace
- Los mensajes de invitación al usuario
- Las direcciones de correo electrónico del remitente (FROM) y del receptor (REPLY-TO) de los correos electrónicos del grupo de usuarios

> **Note**
>
> Las plantillas de mensajes de verificación por SMS y correo electrónico solo aparecerán si ha elegido exigir la verificación de número de teléfono y de correo electrónico en la pestaña Verifications (Verificaciones). Del mismo modo, la plantilla de mensajes de MFA de SMS solo aparece si el valor en la configuración de la MFA está en required (obligatorio) u optional (opcional).

Temas

- Plantillas de mensaje
- Personalización del mensaje SMS
- Personalización de los mensajes de verificación de correo electrónico
- Personalización de los mensajes de invitación a usuarios
- Personalización de la dirección de correo electrónico
- Autorización de Amazon Cognito para enviar correos electrónicos de Amazon SES en su nombre (desde una dirección de correo electrónico FROM personalizada)

Plantillas de mensaje

Las plantillas de mensaje le permiten insertar campos en los mensajes con marcadores de posición que se sustituirán por los valores correspondientes.

Marcadores de posición de las plantillas

Descripción	Token
Código de verificación	`{####}`
Contraseña temporal	`{####}`
Nombre de usuario	`{username}`

> **Note**
>
> No puede usar el marcador de posición `{username}` en mensajes de correo electrónico de verificación. Puede usar el `{username}` marcador de posición en los mensajes de correo

electrónico de invitación que genere con la operación. AdminCreateUser Estos mensajes de correo electrónico de invitación utilizan dos marcadores de posición: el nombre de usuario, como `{username}`, y la contraseña temporal, como `{####}`.

Puede utilizar los marcadores de posición de las plantillas de seguridad avanzadas para:

- Incluir detalles específicos sobre un evento, como la dirección IP, la ciudad, el país, la hora de inicio de sesión y el nombre del dispositivo. Las funciones de seguridad avanzadas de Amazon Cognito pueden analizar estos detalles.
- Verificar si un enlace de un clic es válido.
- Usar un ID de evento, el token de comentarios y el nombre de usuario para diseñar su propio enlace de un solo clic.

Note

Para generar enlaces de un solo clic y utilizar los marcadores de posición `{one-click-link-valid}` y `{one-click-link-invalid}` en plantillas de correo electrónico de seguridad avanzadas, ya debe tener un dominio configurado para el grupo de usuarios.

Marcadores de posición de las plantillas de seguridad avanzadas

Descripción	Token
Dirección IP	`{ip-address}`
Ciudad	`{city}`
País	`{country}`
Hora de inicio de sesión	`{login-time}`
Nombre del dispositivo	`{device-name}`
El enlace de un solo clic es válido	`{one-click-link-valid}`
El enlace de un solo clic no es válido	`{one-click-link-invalid}`

Descripción	Token
ID de evento	`{event-id}`
Token de comentarios	`{feedback-token}`

Personalización del mensaje SMS

> **Note**
>
> En la nueva experiencia de consola de Amazon Cognito, puede personalizar los mensajes SMS

Puede personalizar el mensaje SMS para la autenticación multifactor (MFA) en la pestaña Mensajería debajo del encabezado Plantillas de mensajes.

> **Important**
>
> El mensaje personalizado debe contener el marcador de posición `{####}`. Este marcador de posición se sustituye por el código de autenticación antes de enviar el mensaje.

Amazon Cognito impone una longitud máxima de 140 caracteres UTF-8 para los mensajes SMS, incluido el código de autenticación.

Personalización de los mensajes de verificación por SMS

Puede personalizar el mensaje SMS para las verificaciones de número de teléfono editando la plantilla incluida en el encabezado Do you want to customize your SMS verification messages? (¿Desea personalizar sus mensajes de verificación de SMS?).

> **Important**
>
> El mensaje personalizado debe contener el marcador de posición `{####}`. Este marcador de posición se sustituye por el código de verificación antes de enviar el mensaje.

La longitud máxima del mensaje es de 140 caracteres UTF-8, incluido el código de verificación.

Personalización de los mensajes de verificación de correo electrónico

Para verificar la dirección de correo electrónico de un usuario de su grupo de usuarios con Amazon Cognito, puede enviarle un mensaje de correo electrónico con un enlace que puede seleccionar o enviarle un código que puede ingresar.

Para personalizar el asunto del correo electrónico y el contenido del mensaje para mensajes de verificación de direcciones de correo electrónico, edite la plantilla de Mensaje de verificación en la pestaña Mensajería del grupo de usuarios. Puede elegir un Tipo de verificación de código o enlace al editar la plantilla del Mensaje de verificación.

Si elige Código como el tipo de verificación, el mensaje personalizado debe contener el marcador de posición `{####}`. Al enviar el mensaje, el código de verificación reemplaza este marcador de posición.

Si elige Enlace como el tipo de verificación, el mensaje personalizado deberá contener un marcador de posición con el formato `{##Verify Your Email##}`. Puede cambiar la cadena de texto entre los caracteres del marcador de posición, por ejemplo `{##Click here##}`. Un enlace de verificación titulado Verify Your Email (Verificar correo electrónico) reemplaza a este marcador de posición.

El enlace de un mensaje de verificación de correo electrónico dirige al usuario a una URL como en el ejemplo siguiente.

```
https://<your user pool domain>/confirmUser/?
client_id=abcdefg12345678&user_name=emailtest&confirmation_code=123456
```

La longitud máxima del mensaje es de 20 000 caracteres UTF-8, incluido el código de verificación (de haberlo). Puede utilizar etiquetas HTML en este mensaje para dar formato al contenido.

Personalización de los mensajes de invitación a usuarios

Puede personalizar el mensaje de invitación del usuario que Amazon Cognito envía a los nuevos usuarios mediante SMS o mensaje de correo electrónico editando la plantilla de Mensajes de invitación en la pestaña Mensajería.

> **Important**
>
> El mensaje personalizado debe contener los marcadores de posición `{username}` y `{####}`. Cuando Amazon Cognito envía el mensaje de invitación, reemplaza estos marcadores de posición por el nombre de usuario y la contraseña de su usuario.

La longitud máxima de un mensaje SMS, incluido el código de verificación, es de 140 caracteres UTF-8. La longitud máxima de un mensaje de correo electrónico, incluido el código de verificación, es de 20 000 caracteres UTF-8. Puede utilizar etiquetas HTML en sus mensajes de correo electrónico para dar formato al contenido.

Personalización de la dirección de correo electrónico

De forma predeterminada, los mensajes de correo electrónico que Amazon Cognito envía a los usuarios de los grupos de usuarios provienen de no-reply@verificationemail.com. Puede optar por especificar las direcciones de correo electrónico personalizadas del remitente (FROM) y de respuesta (REPLY-TO) que reemplazarán a no-reply@verificationemail.com.

Para personalizar las direcciones de correo electrónico FROM y REPLY-TO

1. Vaya a la consola de Amazon Cognito y elija User Pools (Grupos de usuarios).
2. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
3. Elija la pestaña Messaging (Mensajería). En Email (Correo electrónico), elija Edit (Editar).
4. Elija una SES Region (Región SES).
5. Elija una dirección en FROM email address (Dirección de correo electrónico DE ORIGEN) en la lista de direcciones de correo electrónico que ha verificado con Amazon SES en la región SES Region (Región SES) que haya seleccionado antes. Para utilizar una dirección de correo electrónico de un dominio verificado, configure los ajustes de correo electrónico en el AWS Command Line Interface o en la API de AWS. Para obtener más información, consulte Verificación de direcciones de correo electrónico y dominios en Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service.
6. Elija un Configuration set (Conjunto de configuración) de la lista de conjuntos de configuración en la SES Region (Región SES) elegida.
7. Introduzca un FROM sender name (Nombre de remitente FROM) descriptivo para sus mensajes de correo electrónico y en el formato `John Stiles <johnstiles@example.com>`.

8. Para personalizar la dirección de correo electrónico REPLY-TO, introduzca una dirección de correo electrónico válida en el campo Dirección de correo electrónico REPLY-TO.

Autorización de Amazon Cognito para enviar correos electrónicos de Amazon SES en su nombre (desde una dirección de correo electrónico FROM personalizada)

Puede configurar Amazon Cognito para que envíe correo electrónico desde una dirección de correo electrónico FROM personalizada en lugar de su dirección predeterminada. Para utilizar una dirección personalizada, debe conceder permiso a Amazon Cognito para enviar mensajes de correo electrónico desde una identidad verificada de Amazon SES. En la mayoría de los casos, puede conceder este permiso si crea una política de autorización de envíos. Para obtener más información, consulte Uso de la autorización de envío con Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service.

Al configurar un grupo de usuarios para utilizar Amazon SES para los mensajes de correo electrónico, Amazon Cognito crea el rol `AWSServiceRoleForAmazonCognitoIdpEmailService` en su cuenta para conceder acceso a Amazon SES. No se necesita ninguna política de autorización de envío cuando se usa el rol de servicio vinculado de `AWSServiceRoleForAmazonCognitoIdpEmailService`. Solo necesita agregar una política de autorización de envío cuando utiliza la funcionalidad de correo electrónico predeterminada en el grupo de usuarios y una identidad de Amazon SES verificada como dirección FROM.

Para obtener más información acerca del rol vinculado al servicio que crea Amazon Cognito, consulte Uso de roles vinculados a servicios para Amazon Cognito.

En el siguiente ejemplo, la política de envío de autorización otorga a Amazon Cognito la capacidad limitada de utilizar una identidad verificada de Amazon SES. Amazon Cognito solo puede enviar mensajes de correo electrónico cuando lo hace en nombre del grupo de usuarios en la condición `aws:SourceArn` y la cuenta en la condición `aws:SourceAccount`. Para ver más ejemplos, consulte Ejemplos de la política de autorización de envío con Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service.

> **Note**
>
> En este ejemplo, el valor "Sid" es una cadena arbitraria que identifica de forma única la declaración. Para obtener más información sobre la sintaxis de la política, consulte Políticas autorización de envío con Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "stmnt1234567891234",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "email.cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "<your SES identity ARN>",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<your account number>"
                },
                "ArnLike": {
                    "aws:SourceArn": "<your user pool ARN>"
                }
            }
        }
    ]
}
```

La consola de Amazon Cognito agrega una política similar en su nombre cuando selecciona una identidad de Amazon SES desde el menú desplegable. Si utiliza la CLI o la API para configurar el grupo de usuarios, debe adjuntar una política estructurada, al igual que en el ejemplo anterior, a su identidad de Amazon SES.

Creación de cuentas de usuario como administrador

Después de crear el grupo de usuarios, puede crear usuarios con la AWS Management Console, así como el AWS Command Line Interface o la API de Amazon Cognito. Puede crear un perfil para un usuario nuevo de un grupo de usuarios y enviar un mensaje de bienvenida con instrucciones de inscripción al usuario a través de SMS o correo electrónico.

Los desarrolladores y los administradores pueden realizar las siguientes tareas:

- Crear un perfil de usuario nuevo usando la AWS Management Console o llamando a la API `AdminCreateUser`.
- Establezca los valores de los atributos de usuario.
- Cree atributos personalizados.
- Establezca el valor de los atributos personalizados inmutables en las solicitudes de API de `AdminCreateUser`. Esta característica no está disponible en la consola de Amazon Cognito.
- Especifique la contraseña temporal o permita que Amazon Cognito genere una de manera automática.
- Especificar si las direcciones de correo electrónico y los números de teléfono proporcionados se marcan como verificados para los nuevos usuarios.
- Especificar mensajes de invitación por SMS y correo electrónico personalizados para los nuevos usuarios mediante la AWS Management Console o un disparador Lambda de mensaje personalizado. Para obtener más información, consulte Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda.
- Especificar si los mensajes de la invitación se envían mediante SMS, correo electrónico o ambos.
- Volver a enviar el mensaje de bienvenida a un usuario existente llamando al API `AdminCreateUser` y especificando `RESEND` para el parámetro `MessageAction`.

> **Note**
>
> Esta acción no se puede realizar actualmente con la AWS Management Console.

- Suprimir el envío del mensaje de invitación cuando se crea el usuario.
- Especificar un plazo de vencimiento para la cuenta de usuario (máximo 90 días).
- Permitir a los usuarios inscribirse o requerir que solo el administrador añada a los usuarios nuevos.

Flujo de autenticación para usuarios creados por los administradores o los desarrolladores

El flujo de autenticación para estos usuarios incluye el paso adicional de enviar la nueva contraseña y proporcionar todos los valores que falten a los atributos obligatorios. Los pasos se indican a continuación; los pasos 5, 6 y 7 son específicos para dicho usuarios.

1. El usuario inicia la sesión por primera vez introduciendo su nombre de usuario y su contraseña.

2. El SDK llama a `InitiateAuth(Username, USER_SRP_AUTH)`.
3. Amazon Cognito devuelve el desafío `PASSWORD_VERIFIER` con un bloque de sal y secreto.
4. El SDK realiza los cálculos de SRP y llama a `RespondToAuthChallenge(Username, <SRP variables>, PASSWORD_VERIFIER)`.
5. Amazon Cognito devuelve el desafío `NEW_PASSWORD_REQUIRED`. El cuerpo de este desafío incluye los atributos actuales del usuario y cualquier atributo requerido en su grupo de usuarios que no tenga actualmente un valor en el perfil del usuario. Para obtener más información, consulte RespondToAuthChallenge.
6. Se indica al usuario que especifique una nueva contraseña y todos los valores que faltan para los atributos obligatorios.
7. El SDK llama a `RespondToAuthChallenge(Username, <New password>, <User attributes>)`.
8. Si el usuario necesita otro factor para la MFA, Amazon Cognito devuelve el desafío SMS_MFA y se envía el código.
9. Una vez que el usuario haya cambiado correctamente su contraseña y, opcionalmente, haya proporcionado los valores de atributo o completado el MFA, se inicia sesión para el usuario y se emiten los tokens.

Cuando el usuario haya respondido a todos los desafíos, el servicio de Amazon Cognito marcará al usuario como confirmado y generará los tokens de ID, acceso y actualización del usuario. Para obtener más información, consulte Uso de tokens con grupos de usuarios.

Creación de un usuario nuevo en la AWS Management Console

Puede establecer requisitos de contraseña de usuario, configurar los mensajes de invitación y verificación enviados a los usuarios y agregar nuevos usuarios con la consola de Amazon Cognito.

Establecer una política de contraseñas y habilitar el autorregistro

Puede configurar los ajustes para que la complejidad de las contraseñas sea mínima y para que los usuarios puedan registrarse mediante API públicas en su grupo de usuarios.

Configurar una política de contraseñas

1. Vaya a la consola de Amazon Cognito y elija User Pools (Grupos de usuarios).
2. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.

3. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión) y localice Password policy (Política de contraseñas). Elija Edit (Editar).
4. Elija un Password policy mode (Modo de política de contraseñas) de Custom (Personalizado).
5. Elija una Password minimum length (Longitud mínima de la contraseña). Para conocer los límites del requisito de longitud de la contraseña, consulte Cuotas de recursos de grupos de usuarios.
6. Elija un requisito de Password complexity (Complejidad de la contraseña).
7. Elija durante cuánto tiempo debe ser válida la contraseña establecida por los administradores.
8. Elija Save changes (Guardar cambios).

Permitir registro de autoservicio

1. Vaya a la consola de Amazon Cognito y elija User Pools (Grupos de usuarios).
2. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
3. Elija la pestaña Sign-up experience (Experiencia de registro) y localice Self-service sign-up (Registro de autoservicio). Seleccione Edit (Editar).
4. Elija si desea activar la opción Enable self-registration (Habilitar el autorregistro). El autorregistro se utiliza normalmente con clientes de aplicaciones públicas que necesitan registrar nuevos usuarios en el grupo de usuarios sin distribuir un secreto de cliente o credenciales de API AWS Identity and Access Management (IAM).

   > **Desactivación del autorregistro**
   >
   > Si no se habilita el autorregistro, se deben crear nuevos usuarios mediante acciones de API administrativas con credenciales API de AMI o iniciando sesión con proveedores federados.

5. Elija Guardar cambios.

Personalizar mensajes de correo electrónico y SMS

Personalizar mensajes de usuario

Puede personalizar los mensajes que Amazon Cognito envía a los usuarios cuando los invita a iniciar sesión, cuando se registran para obtener una cuenta de usuario o cuando inician sesión y se les solicita autenticación multifactor (MFA).

> **Note**
>
> Se envía un Invitation message (Mensaje de invitación) al crear un usuario en el grupo de usuarios e invitarlo a iniciar sesión. Amazon Cognito envía información de inicio de sesión inicial a la dirección de correo electrónico o el número de teléfono del usuario.
> Se envía un Verification message (Mensaje de verificación) cuando un usuario se registra para obtener una cuenta de usuario en el grupo de usuarios. Amazon Cognito envía un código al usuario. Cuando el usuario proporciona el código a Amazon Cognito, verifica su información de contacto y confirma la cuenta para iniciar sesión. Los códigos de verificación son válidos durante 24 horas.
> Se envía un MFA message (Mensaje de MFA) cuando se habilita la MFA por SMS en el grupo de usuarios, y un usuario que ha configurado MFA por SMS inicia sesión y se le solicita MFA.

1. Vaya a la consola de Amazon Cognito y elija User pools (Grupos de usuarios).
2. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
3. Elija la pestaña Messaging (Mensajería) y localice Message templates (Plantillas de mensaje). Seleccione Verification messages (Mensajes de verificación),Invitation messages (Mensajes de invitación) o MFA messages (Mensajes MFA), y elija Edit (Editar).
4. Personalice los mensajes en función del tipo de mensaje elegido.

   > **Note**
   >
   > Todas las variables de las plantillas de mensajes deben incluirse al personalizar el mensaje. Si, por ejemplo, no se incluye la variable {####}, el usuario no tendrá información suficiente para completar la acción de mensaje.
   > Para obtener más información, consulte Plantillas de mensaje.

5. a. Verification messages (Mensajes de verificación)

      i. Elija un Verification type (Tipo de verificación) para mensajes de Email (Correo electrónico). Una verificación de Code (Código) envía un código numérico que el usuario debe ingresar. Una verificación por Link (Enlace) envía un enlace en el que el usuario puede hacer clic para verificar su información de contacto. El texto de la variable de un mensaje de Link (Enlace) se muestra como texto de hipervínculo. Por

ejemplo, una plantilla de mensaje que utiliza la variable {##Click here##} se mostrará como Click here (Haga clic aquí) en el mensaje de correo electrónico.

ii. Ingrese un Email subject (Asunto del correo electrónico) para los mensajes de Email (Correo electrónico).

iii. Ingrese una plantilla personalizada de Email message (Mensaje de correo electrónico) para los mensajes de Email (Correo electrónico). Puede personalizar esta plantilla con HTML.

iv. Ingrese una plantilla personalizada de SMS message (Mensaje SMS) para los SMS.

v. Elija Guardar cambios.

b. Invitation messages (Mensajes de invitación)

i. Ingrese un Email subject (Asunto del correo electrónico) para los mensajes de Email (Correo electrónico).

ii. Ingrese una plantilla personalizada de Email message (Mensaje de correo electrónico) para los mensajes de Email (Correo electrónico). Puede personalizar esta plantilla con HTML.

iii. Ingrese una plantilla personalizada de SMS message (Mensaje SMS) para los SMS.

iv. Elija Guardar cambios.

c. MFA messages (Mensajes MFA)

i. Ingrese una plantilla personalizada de SMS message (Mensaje SMS) para los SMS.

ii. Elija Guardar cambios.

Crear un usuario

Crear un usuario

Puede crear nuevos usuarios para su grupo de usuarios desde la consola de Amazon Cognito. Normalmente, los usuarios pueden iniciar sesión después de haber establecido una contraseña. Para iniciar sesión con una dirección de correo electrónico, el usuario debe verificar el atributo `email`. Para iniciar sesión con un número de teléfono, el usuario debe verificar el atributo `phone_number`. Para confirmar cuentas como administrador puede usar la AWS CLI o la API, o crear perfiles de usuario con un proveedor de identidades federado. Para obtener más información, consulte la sección de referencia de API de Amazon Cognito.

1. Vaya a la consola de Amazon Cognito y elija User pools (Grupos de usuarios).

2. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
3. Elija la pestaña Users (Usuarios) y, a continuación, elija Create a user (Crear un usuario).
4. Revise User pool sign-in and security requirements (Requisitos de seguridad e inicio de sesión del grupo de usuarios) para obtener información sobre los requisitos de contraseña, los métodos de recuperación de cuentas disponibles y los atributos de alias para el grupo de usuarios.
5. Elija cómo desea enviar un Invitation message (Mensaje de invitación). Elija entre mensaje SMS, mensaje de correo electrónico o ambas opciones.

   > **Note**
   >
   > Para poder enviar mensajes de invitación debe configurar un remitente y una Región de AWS con Amazon Simple Notification Service y Amazon Simple Email Service en la pestaña Messaging (Mensajería) de su grupo de usuarios. Se aplican tarifas de mensajes y de datos al destinatario. Amazon SES le factura por los mensajes de correo electrónico por separado, así como Amazon SNS le factura por los mensajes SMS por separado.

6. Elija un Username (Nombre de usuario) para el nuevo usuario.
7. Elija Create a password (Crear una contraseña) o bien que lo haga Amazon Cognito seleccionando Generate a password (Generar una contraseña). Cualquier contraseña temporal debe cumplir con la política de contraseñas del grupo de usuarios.
8. Seleccione Crear.
9. Elija la pestaña Users (Usuarios) y luego elija la entrada User name (Nombre de usuario) para el usuario. Agregue y edite User attributes (Atributos de usuario) y Group memberships (Miembros de grupos). Consulta User event history (Historial de eventos del usuario).

Agregar grupos a un grupo de usuarios

Gracias a la compatibilidad entre los grupos y los grupos de usuarios de Amazon Cognito, se pueden crear y administrar grupos y agregar o eliminar usuarios de grupos. Utilice los grupos para crear recopilaciones de usuarios para administrar sus permisos o representar diferentes tipos de usuarios. Puede asignar una función AWS Identity and Access Management (de IAM) a un grupo para definir los permisos de los miembros de un grupo.

Puede usar grupos para crear un conjunto de usuarios dentro de un grupo de usuarios, cosa que suele hacerse a menudo para establecer los permisos para dichos usuarios. Por ejemplo, puede

crear grupos diferentes para los usuarios que son lectores, colaboradores o editores de su sitio web y su aplicación. Con el rol de IAM asociado a un grupo, también puede configurar diferentes permisos para esos grupos distintos con el fin de que solo los colaboradores puedan ingresar contenido en Amazon S3 y que solo los editores puedan publicar contenido mediante una API en Amazon API Gateway.

Puede crear y administrar grupos en un grupo de usuarios desde las AWS Management Console API y la CLI. Como desarrollador (con AWS credenciales), puede crear, leer, actualizar, eliminar y enumerar los grupos de un grupo de usuarios. También puede añadir usuarios y eliminarlos de los grupos.

No se aplica ningún cargo adicional por usar grupos dentro de un grupo de usuarios. Para obtener más información, consulte Precios de Amazon Cognito.

Asignación de roles de IAM a grupos

Puede utilizar grupos para controlar los permisos de los recursos mediante un rol de IAM. Los roles de IAM incluyen políticas de confianza y políticas de permisos. La política de confianza del rol especifica quién puede usar el rol. Las políticas de permisos especifican las acciones y los recursos a los que los miembros del grupo pueden tener acceso. Al crear un rol de IAM, configure la política de confianza del rol para permitir que los usuarios del grupo asuman el rol. En las políticas de permisos del rol, especifique los permisos que desea que tenga el grupo.

Cuando se crea un grupo en Amazon Cognito, se especifica un rol de IAM proporcionando el ARN del rol. Cuando los miembros del grupo inician sesión con Amazon Cognito, pueden recibir credenciales temporales de los grupos de identidades. Sus permisos están determinados por el rol de IAM asociado.

Los usuarios individuales pueden pertenecer a varios grupos. En su calidad de desarrollador, tiene a disposición las opciones siguientes para elegir de forma automática el rol de IAM cuando un usuario pertenece a varios grupos:

- Puede asignar valores de prioridad a cada grupo. Se elegirá el grupo que tenga la mejor prioridad (inferior) y se aplicará el rol de IAM que tenga asociado.
- La aplicación también puede elegir entre las funciones disponibles al solicitar AWS credenciales para un usuario a través de un grupo de identidades, especificando un ARN de función en el GetCredentialsForIdentity`CustomRoleARN`parámetro. El rol de IAM especificado debe coincidir con un rol que esté disponible para el usuario.

Asignación de valores de prioridad a los grupos

Un usuario puede pertenecer a más de un grupo. En los tokens de ID y acceso del usuario, la reclamación `cognito:groups` contiene la lista de todos los grupos a los que pertenece el usuario. La notificación `cognito:roles` contiene la lista de los roles correspondientes a los grupos.

Dado que un usuario puede pertenecer a más de un grupo, se puede asignar a cada grupo un nivel de prioridad. Se trata de un número que no es negativo y que indica la prioridad del grupo en relación con los demás grupos a los que el usuario pertenece en el grupo de usuarios. Cero es la máxima prioridad. Los grupos con los valores de prioridad más bajos prevalecen sobre los grupos con los valores de prioridad más altos o nulos. Si un usuario pertenece a dos o más grupos, se aplica el rol de IAM del grupo con el valor de prioridad más bajo a la reclamación `cognito:preferred_role` del token de ID de usuario.

Dos grupos pueden tener la misma prioridad. Si esto ocurre, ningún grupo prevalece sobre el otro. Si dos grupos con el mismo valor de prioridad tienen el mismo ARN de rol, ese rol se utiliza en la notificación `cognito:preferred_role` en tokens de ID para los usuarios de cada grupo. Si los dos grupos tienen ARN de roles diferentes, la notificación `cognito:preferred_role` no se establece en los tokens de ID de los usuarios.

Uso de grupos para controlar el permiso con Amazon API Gateway

Puede utilizar los grupos de un grupo de usuarios para controlar los permisos con Amazon API Gateway. Los grupos a los que pertenece un usuario están incluidos en el token de ID y el token de acceso de un grupo de usuarios en la reclamación `cognito:groups`. Puede enviar tokens de ID o de acceso con solicitudes a Amazon API Gateway y utilizar un autorizador de grupos de usuarios de Amazon Cognito para una API REST. Para obtener más información, consulte Control del acceso a una API de REST con grupos de usuarios de Amazon Cognito como autorizador en la Guía para desarrolladores de API Gateway.

También puede autorizar el acceso a una API HTTP de Amazon API Gateway con un autorizador JWT personalizado. Para obtener más información, consulte Control del acceso a las API HTTP con autorizadores de JWT en la Guía para desarrolladores de API Gateway.

Limitaciones aplicadas a los grupos

Los grupos de usuarios están sujetos a las siguientes limitaciones:

- El número de grupos que puede crear está limitado por las cuotas de servicio de Amazon Cognito.

- Los grupos no pueden estar anidados.
- No puede buscar usuarios en un grupo.
- No se pueden buscar los grupos por nombre, aunque sí puede obtener una lista de ellos.

Creación de un grupo nuevo en la AWS Management Console

Utilice el siguiente procedimiento para crear un grupo nuevo.

Para crear un grupo nuevo

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un grupo de usuarios existente.
4. Elija la pestaña Groups (Grupos) y, a continuación, elija Create group (Crear un grupo).
5. En la página Create a group (Crear un grupo), en Group name (Nombre del grupo), escriba un nombre sencillo para el grupo nuevo.
6. Opcionalmente, puede incluir información adicional sobre este grupo en cualquiera de los siguientes campos:
   - Description (Descripción): Introduzca detalles sobre para qué se utilizará este nuevo grupo.
   - Precedence (Prioridad): Amazon Cognito evalúa y aplica todos los permisos de grupo para un usuario determinado en función de a qué grupo de aquellos a los que pertenece tiene un valor de prioridad inferior. Se elegirá el grupo que tenga la prioridad más baja y se aplicará el rol de IAM que tenga asociado. Para obtener más información, consulte Asignación de valores de prioridad a los grupos.
   - IAM role (Rol de IAM): Puede asignar un rol de IAM a su grupo cuando necesite controlar los permisos de los recursos. Si va a integrar un grupo de usuarios en un grupo de identidades, el ajuste IAM role (Rol de IAM) determina qué rol se asigna en el token de ID del usuario si el grupo de identidades está configurado para elegir el rol a partir del token. Para obtener más información, consulte Asignación de roles de IAM a grupos.
   - Add users to this group (Añadir usuarios a este grupo): Agregue usuarios existentes como miembros de este grupo después de crearlo.
7. Elija Create (Crear) para confirmar.

Gestión y búsqueda de cuentas de usuario

Una vez que haya creado el grupo de usuarios, puede ver y administrar los usuarios con la AWS Management Console, la AWS Command Line Interface o la API de Amazon Cognito. En este tema se describe cómo puede ver y buscar usuarios con la AWS Management Console.

Visualización de atributos de los usuarios

Utilice el siguiente procedimiento para ver los atributos de los usuarios en la consola de Amazon Cognito.

Para ver los atributos de los usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS.
2. ElegirUser Pools (Grupos de usuarios).
3. Elija en la lista un grupo de usuarios existente.
4. Elija la pestaña Users (Usuarios) y, a continuación, seleccione un usuario de la lista.
5. En la página de detalles de los usuarios, en User attributes (Atributos de usuario), puede ver qué atributos están asociados al usuario.

Restablecimiento de la contraseña de un usuario

Utilice el siguiente procedimiento para restablecer la contraseña de un usuario en la consola de Amazon Cognito.

Para restablecer la contraseña de un usuario

1. Vaya a la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS.
2. ElegirUser Pools (Grupos de usuarios).
3. Elija en la lista un grupo de usuarios existente.
4. Elija la pestaña Users (Usuarios) y, a continuación, seleccione un usuario de la lista.
5. En la página de detalles de los usuarios, elija Actions (Acciones), Reset password (Restablecer contraseña).
6. En el cuadro de diálogo Reset password (Restablecer contraseña), compruebe la información y, cuando esté listo, elija Reset (Restablecer).

Esta acción produce el envío inmediato de un código de confirmación al usuario y deshabilita la contraseña actual de este cambiando el estado del usuario a `RESET_REQUIRED`. El código de Reset password (Restablecer contraseña) es válido durante una hora.

Búsqueda de atributos de usuario

Si ya ha creado un grupo de usuarios, puede realizar búsquedas desde el panel Users (Usuarios) de la AWS Management Console. También puede utilizar la API ListUsers de Amazon Cognito, que acepta un parámetro Filter (Filtro).

Puede buscar cualquiera de los siguientes atributos estándar. No se pueden buscar los atributos personalizados.

- username (distingue entre mayúsculas y minúsculas)
- email
- phone_number
- nombre
- given_name
- family_name
- preferred_username
- cognito:user_status (denominado Status (Estado) en la consola) (no distingue entre mayúsculas y minúsculas)
- status (denominado Enabled (Habilitado) en la consola) (distingue entre mayúsculas y minúsculas)
- sub

> **Note**
>
> También puede listar usuarios con un filtro del lado del cliente. El filtro del lado del servidor no coincide con más de 1 atributo. Para la búsqueda avanzada, utilice un filtro del lado del cliente con el parámetro `--query` de la acción `list-users` en el AWS Command Line Interface. Cuando se utiliza un filtro del lado del cliente, ListUsers devuelve una lista paginada de cero o más usuarios. Puede recibir varias páginas seguidas sin resultados. Repita la consulta con cada token de paginación devuelto hasta recibir un valor de token de paginación nulo y, a continuación, revise el resultado combinado.

Para obtener más información acerca del filtrado del lado del servidor y del lado del cliente, consulte Filtrado de salida de la AWS CLI en la Guía del usuario de AWS Command Line Interface.

Búsqueda de usuarios mediante la AWS Management Console

Si ya ha creado un grupo de usuarios, puede realizar búsquedas desde el panel Users (Usuarios) de la AWS Management Console.

Las búsquedas de la AWS Management Console son siempre búsquedas de prefijo ("comienza con").

Para buscar un usuario en la consola de Amazon Cognito

1. Diríjase a la consola de Amazon Cognito. Es posible que se le soliciten sus credenciales de AWS.
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un grupo de usuarios existente.
4. Elija la pestaña Users (Usuarios) y, a continuación, ingrese el nombre de usuario del usuario en el campo de búsqueda. Tenga en cuenta que algunos valores de atributo distinguen entre mayúsculas y minúsculas, por ejemplo Username (Nombre de usuario).

   También puede encontrar usuarios ajustando el filtro de búsqueda para restringir el ámbito a otras propiedades de usuario, como Email (Correo electrónico), Phone number (Número de teléfono) o Last name (Apellido).

Búsqueda de usuarios mediante la API **`ListUsers`**

Para buscar usuarios de su aplicación, utilice la API ListUsers de Amazon Cognito. Esta API utiliza los parámetros siguientes:

- `AttributesToGet`: serie de cadenas, donde cada cadena es el nombre de un atributo de usuario que debe devolverse por cada usuario en los resultados de búsquedas. Para recuperar todos los atributos, no incluya un parámetro `AttributesToGet` ni solicitud `AttributesToGet` con un valor de la cadena literal `null`.
- `Filter`: una cadena de filtro con la forma "`AttributeName Filter-Type "AttributeValue"`". Las comillas dentro de la cadena de filtro deben ir precedidas por una barra

oblicua inversa (\). Por ejemplo, `"family_name = \"Reddy\""`. Si la cadena de filtro está vacía, `ListUsers` devuelve todos los usuarios del grupo de usuarios.

- `AttributeName`: el nombre del atributo que debe buscarse. Solo puede buscar los atributos de uno en uno.

  > **Note**
  >
  > Solo puede buscar atributos estándar. No se pueden buscar los atributos personalizados. Esto se debe a que solo se pueden buscar atributos indizados y los atributos personalizados no se pueden indexar.

- `Filter-Type`: para una coincidencia exacta, utilice =; por ejemplo, `given_name = "Jon"`. Para una coincidencia de prefijo ("comienza con"), utilice ^=; por ejemplo, `given_name ^= "Jon"`.
- `AttributeValue`: el valor del atributo que debe asociarse para cada usuario.

- `Limit`: número máximo de usuarios que debe devolverse.
- `PaginationToken`: un token para obtener más resultados de una búsqueda anterior. Amazon Cognito hace que venza el token de paginación después de una hora.
- `UserPoolId`: el ID del grupo de usuarios en el que debe realizarse la búsqueda.

Todas las búsquedas no distinguen entre mayúsculas y minúsculas. Los resultados de la búsqueda se ordenan según el atributo designado por la cadena `AttributeName`, en orden ascendente.

Ejemplos de uso de la API **`ListUsers`**

En el ejemplo siguiente se devuelven todos los usuarios y se incluyen todos los atributos.

```
{
    "AttributesToGet": null,
    "Filter": "",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

En el ejemplo siguiente se devuelven todos los usuarios cuyos números de teléfono empiezan por "+1312" y se incluyen todos los atributos.

```
{
    "AttributesToGet": null,
    "Filter": "phone_number ^= \"+1312\"",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

En el ejemplo siguiente se devuelven los 10 primeros usuarios cuyo apellido es "Reddy". Para cada usuario, los resultados de la búsqueda incluyen el nombre del usuario, su número de teléfono y su dirección de correo electrónico. Si hay más de 10 usuarios que coincidan con la búsqueda en el grupo de usuarios, la respuesta incluirá un token de paginación.

```
{
    "AttributesToGet": [
        "given_name",
        "phone_number",
        "email"
    ],
    "Filter": "family_name = \"Reddy\"",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

Mientras que en el ejemplo anterior se devuelve un token de paginación, en el ejemplo siguiente se devuelven los 10 usuarios siguientes que coincidan con la misma cadena de filtro.

```
{
    "AttributesToGet": [
        "given_name",
        "phone_number",
        "email"
    ],
    "Filter": "family_name = \"Reddy\"",
    "Limit": 10,
    "PaginationToken": "pagination_token_from_previous_search",
    "UserPoolId": "us-east-1_samplepool"
}
```

Recuperación de cuentas de usuario

El parámetro `AccountRecoverySetting` le permite personalizar el método que un usuario puede utilizar para recuperar su contraseña cuando llama a la API `ForgotPassword`. `ForgotPassword` envía un código de recuperación a un correo electrónico verificado o a un número de teléfono verificado. El código de recuperación es válido durante una hora. Cuando especifica un `AccountRecoverySetting` para su grupo de usuarios, Amazon Cognito elige el destino de entrega de código en función de la prioridad establecida.

Cuando se define `AccountRecoverySetting` y un usuario tiene la MFA con SMS configurada, el SMS no se puede utilizar como mecanismo de recuperación de la cuenta. La prioridad de esta configuración se determina con un 1, que es la prioridad más alta. Cognito envía una verificación solo a uno de los métodos especificados.

Por ejemplo, `admin_only` es un valor que se utiliza cuando el administrador no desea que el usuario recupere la cuenta por sí mismo y, en su lugar, requiere que se ponga en contacto con el administrador para restablecerla. No se puede utilizar `admin_only` con ningún otro mecanismo de recuperación de la cuenta.

Si no se especifica `AccountRecoverySetting`, Amazon Cognito utiliza el mecanismo heredado para determinar el método de recuperación de la contraseña. En este caso, Cognito utiliza primero un teléfono verificado. Si no se encuentra el teléfono verificado del usuario, Cognito retrocede y utiliza el correo electrónico verificado a continuación.

Para obtener más información acerca de `AccountRecoverySetting`, consulte CreateUserPool y UpdateUserPool en la Referencia de la API de Amazon Cognito Identity Provider.

Comportamiento de contraseña olvidada

En una hora determinada, permitimos entre 5 y 20 intentos para que un usuario solicite o introduzca un código de restablecimiento de contraseña como parte de las acciones forgot-password (contraseña olvidada) y confirm-forgot-password (confirmar contraseña olvidada). El valor exacto depende de los parámetros de riesgo asociados con las solicitudes. Tenga en cuenta que este comportamiento está sujeto a cambios.

Importación de usuarios a un grupo de usuarios

Existen dos formas de importar o migrar usuarios del directorio de usuarios o de la base de datos de usuarios a los grupos de usuarios Amazon Cognito. Puede migrar usuarios cuando inician sesión por primera vez mediante Amazon Cognito con un desencadenador de Lambda para la migración de

usuarios. Con este método, los usuarios pueden seguir usando sus contraseñas y no es necesario restablecerlas tras la migración al grupo de usuarios. También puede migrar los usuarios de forma masiva cargando un archivo CSV que contiene los atributos del perfil de usuario de todos los usuarios. En las secciones siguientes se describen estos dos métodos.

Temas

Importación de usuarios a grupos de usuarios con un desencadenador de Lambda para la migración de usuarios

Con este enfoque, puede migrar sin problemas usuarios desde el directorio de usuarios existente a grupos de usuarios cuando un usuario inicia sesión con la aplicación o solicita un restablecimiento de la contraseña por primera vez. Agregue una función Migración del desencadenador de Lambda del usuario a su grupo de usuarios, y este recibe metadatos sobre los usuarios que intentan iniciar sesión y devuelve información del perfil de usuario de un origen de identidad externo. Para obtener detalles y un ejemplo de código sobre este desencadenador de Lambda, incluidos los parámetros de solicitud y respuesta, consulte Parámetros del desencadenador de Lambda para migrar usuarios.

Antes de comenzar el proceso de migración de usuarios, cree una función de Lambda para migrar usuarios en su Cuenta de AWS y defina la función de Lambda como el desencadenador de migración del usuario en el grupo de usuarios. Agregue una política de autorización a su función de Lambda que permita acceder únicamente a la entidad principal de la cuenta del servicio de Amazon Cognito, `cognito-idp.amazonaws.com`, para invocar a la función de Lambda y solo en el contexto de su propio grupo de usuarios. Para obtener más información, consulte Uso de políticas basadas en recursos para Lambda de AWS Lambda (políticas de funciones de Lambda).

Proceso de inicio de sesión

1. El usuario abre la aplicación e inicia sesión con la API de grupos de usuarios de Amazon Cognito o a través de la IU alojada de Amazon Cognito. Para obtener más información sobre cómo facilitar el inicio de sesión con las API de Amazon Cognito, consulte Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles.
2. La aplicación envía el nombre de usuario y la contraseña a Amazon Cognito. Si la aplicación tiene una interfaz de usuario de inicio de sesión personalizada que ha creado con un SDK de AWS,

su aplicación debe usar InitiateAuth o AdminInitiateAuth con el flujo `USER_PASSWORD_AUTH` o `ADMIN_USER_PASSWORD_AUTH`. Cuando la aplicación utiliza uno de estos flujos, el SDK envía la contraseña al servidor.

> **Note**
>
> Antes de agregar un desencadenador de migración de usuarios, active el flujo `USER_PASSWORD_AUTH` o `ADMIN_USER_PASSWORD_AUTH` en la configuración del cliente de la aplicación. Debe utilizar estos flujos en lugar del flujo predeterminado `USER_SRP_AUTH`. Amazon Cognito debe enviar una contraseña a la función de Lambda para que pueda verificar la autenticación de su usuario en el otro directorio. Un SRP oculta la contraseña de usuario de la función de Lambda.

3. Amazon Cognito comprueba si el nombre de usuario enviado coincide con un nombre de usuario o un alias del grupo de usuarios. Puede configurar la dirección de correo electrónico, el número de teléfono o el nombre de usuario preferido como alias en el grupo de usuarios. Si el usuario no existe, Amazon Cognito envía parámetros, incluidos el nombre de usuario y la contraseña, a la función Migración del desencadenador de Lambda del usuario.
4. La función Migración del desencadenador de Lambda del usuario comprueba o autentica al usuario con el directorio o la base de datos de usuarios existente. La función devuelve los atributos de usuario que Amazon Cognito almacena en el perfil del usuario en el grupo de usuarios. Puede devolver un parámetro `username` solo si el nombre de usuario enviado coincide con un atributo de alias. Si desea que los usuarios puedan seguir usando las contraseñas existentes, la función establece el atributo `finalUserStatus` en `CONFIRMED` en la respuesta de Lambda. Su aplicación debe devolver todos los parámetros `"response"` mostrados en Parámetros del desencadenador de Lambda para migrar usuarios.

   > **Important**
   >
   > No registre todo el objeto de evento de solicitud en el código de Lambda de migración de usuarios. Este objeto de evento de solicitud incluye la contraseña del usuario. Si no sanea los registros, las contraseñas aparecen en CloudWatch Logs.

5. Amazon Cognito crea el perfil de usuario en el grupo de usuarios y devuelve los tokens al cliente de aplicación.
6. La aplicación admite los tokens, acepta la autenticación de usuarios y procede con el contenido solicitado.

Después de migrar a los usuarios, utilice `USER_SRP_AUTH` para iniciar sesión. El protocolo Secure Remote Password (SRP) no envía la contraseña a través de la red y ofrece beneficios de seguridad con respecto al flujo `USER_PASSWORD_AUTH` utilizado durante la migración.

Si se producen errores durante la migración, incluidos problemas con el dispositivo del cliente o con la red, la aplicación recibe respuestas de error de la API de grupos de usuarios de Amazon Cognito. Cuando esto ocurre, es posible que Amazon Cognito cree o no la cuenta de usuario en el grupo de usuarios. El usuario debería intentar iniciar sesión de nuevo. Si el inicio de sesión falla repetidamente, intente restablecer la contraseña del usuario con el flujo de recuperación de contraseñas olvidadas de la aplicación.

El flujo de recuperación de contraseñas olvidadas también invoca a la función Migración del desencadenador de Lambda del usuario con un origen de eventos `UserMigration_ForgotPassword`. Dado que el usuario no envía una contraseña cuando solicita un restablecimiento de contraseña, Amazon Cognito no incluye ninguna contraseña en caso de que se envíe a la función de Lambda. La función solo puede buscar al usuario en el directorio de usuarios existente y devolver atributos para agregarlos al perfil de usuarios en el grupo de usuarios. Cuando la función completa la invocación y devuelve su respuesta a Amazon Cognito, el grupo de usuarios envía un código de restablecimiento de contraseña por correo electrónico o SMS. En la aplicación, solicite al usuario el código de confirmación y una nueva contraseña y, a continuación, envíe esa información a Amazon Cognito en una solicitud de la API ConfirmForgotPassword. Puede también utilizar páginas integradas para el flujo de contraseña olvidada en la interfaz de usuario alojada de Amazon Cognito.

Importación de usuarios en grupos de usuarios desde un archivo CSV

Puede importar usuarios a un grupo de usuarios de Amazon Cognito. La información de usuario se importa desde un archivo .csv de formato especial. El proceso de importación establece valores para todos los atributos de usuario excepto password (contraseña). No se admite la importación de contraseñas, ya que las prácticas recomendadas de seguridad requieren que las contraseñas no estén disponibles como texto sin formato, y no admitimos la importación de hash. Esto significa que sus usuarios deben cambiar de contraseña la primera vez que inicien sesión. Por lo tanto, los usuarios se encontrarán en el estado RESET_REQUIRED cuando se importen con este método.

Puede establecer las contraseñas de sus usuarios con una solicitud a la API AdminSetUserPassword que establezca el parámetro `Permanent` a `true`.

> **Note**
>
> La fecha de creación de cada usuario es la hora en la que se importó a dicho usuario al grupo de usuarios. La fecha de creación no es uno de los atributos importados.

A continuación, indicamos los pasos básicos:

1. Cree un rol de Registros de Amazon CloudWatch en la consola (IAM) AWS Identity and Access Management.
2. Cree el archivo .csv de importación de usuarios.
3. Cree y ejecute el trabajo de importación de usuarios.
4. Cargue el archivo .csv de importación de usuarios.
5. Inicie y ejecute el trabajo de importación de usuarios.
6. Utilice CloudWatch para comprobar el registro de eventos.
7. Pida a los usuarios importados que restablezcan sus contraseñas.

Temas

Creación del rol de IAM de CloudWatch Logs

Si utiliza la CLI o la API de Amazon Cognito, tiene que crear un rol de IAM para CloudWatch. En el procedimiento siguiente se describe cómo crear un rol de IAM que Amazon Cognito pueda utilizar para escribir los resultados de su trabajo de importación en CloudWatch Logs.

> **Note**
>
> Al crear un trabajo de importación en la consola de Amazon Cognito, puede crear el rol de IAM al mismo tiempo. Cuando elige Create a new IAM role (Crear un nuevo rol de

IAM), Amazon Cognito aplica automáticamente la política de confianza y la política de IAM adecuadas al rol.

Para crear el rol de IAM de CloudWatch Logs para la importación de grupos de usuarios (AWS CLI, API)

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Cree un nuevo rol de IAM para un Servicio de AWS. Para obtener instrucciones detalladas, consulte Creación de un rol para un Servicio de AWS en la Guía del usuario de AWS Identity and Access Management.

   a. Al seleccionar Use case (Caso de uso) para Trusted entity type (Tipo de entidad de confianza), elija cualquier servicio. Amazon Cognito no aparece actualmente en la lista de casos de uso del servicio.

   b. En la pantalla Add permissions (Agregar permisos), elija Create policy (Crear política) e inserte la siguiente declaración de política. Reemplace *REGION* por la Región de AWS de su grupo de usuarios, por ejemplo, `us-east-1`. Reemplace *ACCOUNT* por su ID de Cuenta de AWS, por ejemplo, `111122223333`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:REGION:ACCOUNT:log-group:/aws/cognito/*"
            ]
        }
    ]
}
```

3. Como no ha elegido Amazon Cognito como entidad de confianza al crear el rol, ahora debe editar manualmente la relación de confianza del rol. Elija Roles en el panel de navegación de la consola de IAM y, a continuación, elija el nuevo rol que ha creado.
4. Seleccione la pestaña Trust Relationships (Relaciones de confianza).
5. Elija Edit trust policy (Editar la política de confianza).
6. Pegue la siguiente declaración de política en Edit trust policy (Editar política de confianza) y reemplace cualquier texto existente:

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "cognito-idp.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    }
```

7. Elija Update policy.
8. Apunte el ARN del rol. Proporcionará el ARN cuando cree su trabajo de importación.

Creación del archivo CSV de importación de usuarios

Para poder importar los usuarios existentes a su grupo de usuarios, debe crear un archivo de valores separados por comas (CSV) que contenga los usuarios que desea importar y sus atributos. A partir de su grupo de usuarios, puede recuperar un archivo de importación de usuarios con encabezados que reflejen el esquema de atributos de su grupo de usuarios. A continuación, puede insertar la información de usuario que coincida con los requisitos de formato de Formato del archivo CSV.

Descarga del encabezado del archivo CSV (consola)

Siga este procedimiento para descargar el archivo de encabezado de CSV.

Para descargar el encabezado de archivo CSV

1. Diríjase a la consola de Amazon Cognito. Es posible que se le soliciten sus credenciales de AWS.
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija la pestaña Users.
5. En la sección Import users (Importar usuarios), elija Create an import job (Crear un trabajo de importación).
6. En Upload CSV (Cargar CSV), seleccione el enlace template.csv y descargue el archivo CSV.

Descarga del encabezado del archivo CSV (AWS CLI)

Para obtener una lista de los encabezados correctos, ejecute este comando de la CLI, donde *USER_POOL_ID* es el identificador del grupo de usuarios al que importará los usuarios:

```
aws cognito-idp get-csv-header --user-pool-id "USER_POOL_ID"
```

Respuesta de ejemplo:

```
{
    "CSVHeader": [
        "name",
        "given_name",
        "family_name",
        "middle_name",
        "nickname",
        "preferred_username",
        "profile",
        "picture",
        "website",
        "email",
        "email_verified",
        "gender",
        "birthdate",
        "zoneinfo",
        "locale",
        "phone_number",
        "phone_number_verified",
```

```
        "address",
        "updated_at",
        "cognito:mfa_enabled",
        "cognito:username"
    ],
    "UserPoolId": "USER_POOL_ID"
}
```

Formato del archivo CSV

El archivo de encabezado CSV de importación de usuarios descargado es parecido a la siguiente cadena. También incluye cualquier atributo personalizado que haya agregado a su grupo de usuarios.

```
cognito:username,name,given_name,family_name,middle_name,nickname,preferred_username,profile,pi
```

Edite el archivo CSV para que incluya este encabezado y los valores de atributo de sus usuarios y que tenga un formato que siga estas reglas:

> **Note**
>
> Para obtener más información acerca de los valores de atributos, como el formato adecuado para números de teléfono, consulte Custom pool attributes (.

- La primera línea del archivo es la fila de encabezado descargada, que contiene los nombres de los atributos de usuario.
- El orden de las columnas del archivo CSV no importa.
- Cada línea tras la primera línea contiene los valores de atributo de un usuario.
- Todas las columnas del encabezado tienen que estar presentes, pero no es necesario proporcionar valores para cada columna.
- Los atributos siguientes son obligatorios:
  - cognito:username
  - cognito:mfa_enabled
  - email_verified o phone_number_verified
    - Al menos uno de los atributos verificados automáticamente debe ser `true` para cada usuario. Un atributo verificado automáticamente es una dirección de correo electrónico o un número de

teléfono al que Amazon Cognito envía automáticamente un código cuando un nuevo usuario se une a su grupo de usuarios.

- El grupo de usuarios debe tener al menos un atributo verificado automáticamente, ya sea email_verified o phone_number_verified. Si el grupo de usuarios no tiene atributos verificados automáticamente, el trabajo de importación no empezará.
- Si el grupo de usuarios solo tiene un atributo verificado automáticamente, dicho atributo tiene que verificarse para cada usuario. Por ejemplo, si el grupo de usuarios solo tiene phone_number como un atributo verificado automáticamente, el valor phone_number_verified debe ser `true` para cada usuario.

> **Note**
>
> Para que los usuarios restablezcan sus contraseñas, deben tener un correo electrónico o un número de teléfono verificado. Amazon Cognito envía un mensaje con el código de restablecimiento de contraseña al correo electrónico o al número de teléfono especificado en el archivo CSV. Si el mensaje se envía al número de teléfono, se envía mediante SMS. Para obtener más información, consulte Verificación de la información de contacto durante el registro.

  - email (si email_verified es `true`)
  - phone_number (si phone_number_verified es `true`)
  - Todos los atributos que ha marcado como obligatorios al crear el grupo de usuarios
- Los valores de atributo que son cadenas no deben estar entre comillas.
- Si un valor de atributo contiene una coma, debe poner delante de la coma una barra oblicua inversa (\). Esto se debe a que los campos de un archivo CSV están separados por comas.
- El contenido del archivo CSV debe estar en formato UTF-8 sin marca de orden de bytes.
- El campo cognito:username es obligatorio y debe ser único dentro del grupo de usuarios. Puede ser cualquier cadena Unicode. Sin embargo, no puede contener espacios ni pestañas.
- Los valores birthdate (Fecha de nacimiento), si los hay, deben tener el formato *`mm/dd/aaaa`*. Esto significa, por ejemplo, que la fecha de nacimiento 1 de febrero de 1985 debe codificarse como **`02/01/1985`**.
- El campo cognito:mfa_enabled es obligatorio. Si ha establecido que la autenticación multifactor (MFA) es obligatoria en su grupo de usuarios, este campo debe ser `true` para todos los usuarios. Si ha desactivado la autenticación MFA, este campo debe ser `false` para todos los usuarios. Si

ha definido la autenticación MFA como opcional, este campo puede ser `true` o `false`, pero no puede estar vacío.

- La longitud máxima de la fila es de 16 000 caracteres.
- El tamaño de archivo CSV máximo es de 100 MB.
- El número máximo de filas (usuarios) del archivo es de 500 000. Este máximo no incluye la fila de encabezado.
- Se espera que el valor del campo updated_at (Actualizado a) esté en formato de tiempo Unix en segundos, por ejemplo: **1471453471**.
- Los espacios en blanco del principio y del final de un valor de atributo se eliminan.

La siguiente lista es un ejemplo de archivo de importación CSV para un grupo de usuarios sin atributos personalizados. Su esquema de grupo de usuarios puede diferir de este ejemplo. En ese caso, deberá proporcionar valores de prueba en la plantilla CSV que descargue de su grupo de usuarios.

```
cognito:username,name,given_name,family_name,middle_name,nickname,preferred_username,profile,pi
John,,John,Doe,,,,,,,johndoe@example.com,TRUE,,02/01/1985,,,+12345550100,TRUE,123 Any
 Street,,FALSE
Jane,,Jane,Roe,,,,,,,janeroe@example.com,TRUE,,01/01/1985,,,+12345550199,TRUE,100 Main
 Street,,FALSE
```

Creación y ejecución del trabajo de importación del grupo de usuarios de Amazon Cognito

En esta sección, se describe cómo crear y ejecutar el trabajo de importación del grupo de usuarios mediante la consola de Amazon Cognito y la AWS Command Line Interface (AWS CLI).

Temas

- Importación de usuarios desde un archivo CSV (consola)
- Importación de usuarios (AWS CLI)

Importación de usuarios desde un archivo CSV (consola)

En el procedimiento siguiente se describe cómo importar a los usuarios desde el archivo CSV.

Para importar usuarios desde el archivo CSV (consola)

1. Diríjase a la consola de Amazon Cognito. Es posible que se le soliciten sus credenciales de AWS.
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija la pestaña Users.
5. En la sección Import users (Importar usuarios), elija Create an import job (Crear un trabajo de importación).
6. En la página Create import job (Crear trabajo de importación), ingrese un valor en Job name (Nombre de trabajo).
7. Elija Create a new IAM role (Crear un nuevo rol de IAM) o Use an existing IAM role (Usar un rol de IAM existente).

   a. Si eligió Create a new IAM role (Crear un nuevo rol de IAM), ingrese un nombre para su nuevo rol. Amazon Cognito creará automáticamente un rol con los permisos y la relación de confianza correctos. La entidad principal de IAM que crea el trabajo de importación debe tener permisos para crear roles de IAM.

   b. Si eligió Use an existing IAM role (Utilizar un rol de IAM existente), elija un rol de la lista debajo de IAM role selection (Selección de rol de IAM). Este rol debe tener los permisos y la política de confianza que se describen en Creación del rol de IAM de CloudWatch Logs.
8. Elija Create job (Crear trabajo) para enviar su trabajo, pero inícielo más tarde. Elija Create and start job (Crear e iniciar trabajo) para enviar su trabajo e iniciarlo inmediatamente.
9. Si ha creado el trabajo pero no lo ha iniciado, puede iniciarlo más adelante. En la pestaña Users (Usuarios), en Import users (Importar usuarios), elija su trabajo de importación y, a continuación, seleccione Start (Iniciar). También puede enviar una solicitud de API StartUserImportJob desde un SDK de AWS.
10. Supervise el progreso de su trabajo de importación de usuarios en la pestaña Users (Usuarios), en Import users (Importar usuarios). Si su trabajo no se realiza correctamente, puede seleccionar el valor Status (Estado). Para obtener más detalles, seleccione View the CloudWatch logs for more details (Ver los registros de CloudWatch para obtener más detalles) y consulte cualquier problema en la consola de CloudWatch Logs.

Importación de usuarios (AWS CLI)

Dispone de los comandos de la CLI siguientes para importar usuarios a un grupo de usuarios:

- `create-user-import-job`
- `get-csv-header`
- `describe-user-import-job`
- `list-user-import-jobs`
- `start-user-import-job`
- `stop-user-import-job`

Para obtener la lista de opciones de línea de comandos de estos comandos, utilice la opción de línea de comandos `help`. Por ejemplo:

```
aws cognito-idp get-csv-header help
```

Creación de un trabajo de importación de usuarios

Después de crear el archivo CSV, cree un trabajo de importación de usuarios mediante la ejecución del siguiente comando de la CLI, donde *JOB_NAME* es el nombre elegido para el trabajo, *USER_POOL_ID* es el ID del grupo de usuarios al que se agregarán usuarios y *ROLE_ARN* es el ARN de rol recibido en Creación del rol de IAM de CloudWatch Logs:

```
aws cognito-idp create-user-import-job --job-name "JOB_NAME" --user-pool-id
 "USER_POOL_ID" --cloud-watch-logs-role-arn "ROLE_ARN"
```

El valor de *PRE_SIGNED_URL* devuelto en la respuesta es válido durante 15 minutos. Transcurrido ese tiempo, la URL caducará y será preciso crear otra tarea de importación de usuarios para obtener una URL nueva.

Example Respuesta de ejemplo:

```
{
    "UserImportJob": {
        "Status": "Created",
        "SkippedUsers": 0,
        "UserPoolId": "USER_POOL_ID",
```

```
        "ImportedUsers": 0,
        "JobName": "JOB_NAME",
        "JobId": "JOB_ID",
        "PreSignedUrl": "PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
}
```

Valores de estado para un trabajo de importación de usuarios

En las respuestas a los comandos de importación de usuarios, verá uno de los valores `Status` siguientes:

- `Created`: Se ha creado el trabajo, pero no se ha iniciado.
- `Pending`: Un estado de transición. El trabajo se ha iniciado, pero todavía no se ha empezado a importar los usuarios.
- `InProgress`: El trabajo se ha iniciado y se están importando usuarios.
- `Stopping`: Ha detenido el trabajo, pero el trabajo aún no ha dejado de importar usuarios.
- `Stopped`: Ha detenido el trabajo y este ha dejado de importar usuarios.
- `Succeeded`: El trabajo se ha completado correctamente.
- `Failed`: El trabajo se ha detenido debido a un error.
- `Expired`: Ha creado un trabajo, pero no la ha iniciado en un plazo de 24-48 horas. Todos los datos asociados al trabajo se han eliminado y el trabajo no puede iniciarse.

Carga del archivo CSV

Utilice el comando `curl` siguiente para cargar el archivo CSV que contiene los datos de usuario en la URL prefirmada que ha obtenido de la respuesta del comando `create-user-import-job`.

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
 "PRE_SIGNED_URL"
```

En la salida de este comando, busque la frase "`We are completely uploaded and fine`". Esta frase indica que el archivo se ha cargado correctamente.

Descripción de un trabajo de importación de usuarios

Para obtener una descripción de su tarea de importación de usuarios, utilice el siguiente comando, donde *USER_POOL_ID* es el ID del grupo de usuarios y *JOB_ID* es el ID de trabajo que se ha devuelto al crear el trabajo de importación de usuarios.

```
aws cognito-idp describe-user-import-job --user-pool-id "USER_POOL_ID" --job-id
 "JOB_ID"
```

Example Respuesta de ejemplo:

```
{
    "UserImportJob": {
        "Status": "Created",
        "SkippedUsers": 0,
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "JobName": "JOB_NAME",
        "JobId": "JOB_ID",
        "PreSignedUrl": "PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn":"ROLE_ARN",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
}
```

En el resultado del ejemplo anterior, *PRE_SIGNED_URL* es la URL en la que ha cargado el archivo CSV. *ROLE_ARN* es el ARN del rol para CloudWatch Logs que ha recibido cuando creó el rol.

Visualización de la lista de trabajos de importación de usuarios

Para visualizar una lista de las tareas de importación de usuarios, ejecute el comando siguiente:

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 2
```

Example Respuesta de ejemplo:

```
{
    "UserImportJobs": [
        {
            "Status": "Created",
            "SkippedUsers": 0,
```

```
            "UserPoolId": "USER_POOL_ID",
            "ImportedUsers": 0,
            "JobName": "JOB_NAME",
            "JobId": "JOB_ID",
            "PreSignedUrl":"PRE_SIGNED_URL",
            "CloudWatchLogsRoleArn":"ROLE_ARN",
            "FailedUsers": 0,
            "CreationDate": 1470957431.965
        },
        {
            "CompletionDate": 1470954227.701,
            "StartDate": 1470954226.086,
            "Status": "Failed",
            "UserPoolId": "USER_POOL_ID",
            "ImportedUsers": 0,
            "SkippedUsers": 0,
            "JobName": "JOB_NAME",
            "CompletionMessage": "Too many users have failed or been skipped during the
 import.",
            "JobId": "JOB_ID",
            "PreSignedUrl":"PRE_SIGNED_URL",
            "CloudWatchLogsRoleArn":"ROLE_ARN",
            "FailedUsers": 5,
            "CreationDate": 1470953929.313
        }
    ],
    "PaginationToken": "PAGINATION_TOKEN"
}
```

Las tareas se enumeran en orden cronológico desde la última tarea creada hasta la primera. La cadena *PAGINATION_TOKEN* que sigue al segundo trabajo indica que este comando de lista tiene resultados adicionales. Para publicar la lista de resultados adicionales, utilice la opción `--pagination-token` de la siguiente manera:

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 10 --
pagination-token "PAGINATION_TOKEN"
```

Inicio de un trabajo de importación de usuarios

Para iniciar una tarea de importación de usuarios, ejecute el comando siguiente:

```
aws cognito-idp start-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Solo puede haber un trabajo de importación activo a la vez por cuenta.

Example Respuesta de ejemplo:

```
{
    "UserImportJob": {
        "Status": "Pending",
        "StartDate": 1470957851.483,
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "SkippedUsers": 0,
        "JobName": "JOB_NAME",
        "JobId": "JOB_ID",
        "PreSignedUrl":"PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
        "FailedUsers": 0,
        "CreationDate": 1470957431.965
    }
}
```

Detención de un trabajo de importación de usuarios

Para detener una tarea de importación de usuarios mientras está en curso, ejecute el comando siguiente. Después de detener el trabajo, esta no se puede reiniciar.

```
aws cognito-idp stop-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Example Respuesta de ejemplo:

```
{
    "UserImportJob": {
        "CompletionDate": 1470958050.571,
        "StartDate": 1470958047.797,
        "Status": "Stopped",
        "UserPoolId": "USER_POOL_ID",
        "ImportedUsers": 0,
        "SkippedUsers": 0,
        "JobName": "JOB_NAME",
        "CompletionMessage": "The Import Job was stopped by the developer.",
        "JobId": "JOB_ID",
        "PreSignedUrl":"PRE_SIGNED_URL",
        "CloudWatchLogsRoleArn": "ROLE_ARN",
```

```
        "FailedUsers": 0,
        "CreationDate": 1470957972.387
    }
}
```

Visualización de los resultados de importación del grupo de usuarios en la consola de CloudWatch

Puede ver los resultados de su trabajo de importación en la consola de Amazon CloudWatch.

Temas

- Visualización de los resultados
- Interpretación de los resultados

Visualización de los resultados

En los pasos siguientes se describe cómo ver los resultados de la importación del grupo de usuarios.

Para ver los resultados de la importación del grupo de usuarios

1. Inicie sesión en la AWS Management Console y abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. Seleccione Logs (Registros).
3. Elija el grupo de log de las tareas de importación del grupo de usuarios. El nombre del grupo de log tiene el formato `/aws/cognito/userpools/`*`USER_POOL_ID`*`/`*`USER_POOL_NAME`*.
4. Elija el log de el trabajo de importación de usuarios que acaba de ejecutar. El nombre de log tiene el formato *`JOB_ID`*/*`JOB_NAME`*. Los resultados del log remiten a los usuarios por número de línea. No se escriben datos de usuarios en el log. Por cada usuario, se muestra una línea similar a la siguiente:

    - `[SUCCEEDED] Line Number 5956 - The import succeeded.`
    - `[SKIPPED] Line Number 5956 - The user already exists.`
    - `[FAILED] Line Number 5956 - The User Record does not set any of the auto verified attributes to true. (Example: email_verified to true).`

Interpretación de los resultados

Los usuarios que se han importado correctamente tiene el estado establecido en "PasswordReset".

En los casos siguientes, el usuario no se importa, pero el trabajo de importación continuará:

- Ningún atributo verificado automáticamente se establece en `true`.
- Los datos de usuario no coinciden con el esquema.
- El usuario no se ha podido importar debido a un error interno.

En los casos siguientes, el trabajo de importación fallará:

- No se puede asumir el rol de Amazon CloudWatch Logs, no tiene la política de acceso correcta o se ha eliminado.
- El grupo de usuarios se ha eliminado.
- Amazon Cognito no puede analizar el archivo .csv.

Obligación de que los usuarios importados restablezcan sus contraseñas

La primera vez que cada usuario importado inicia sesión e ingresa la contraseña, se le pide que ingrese una nueva contraseña. En el procedimiento siguiente se describe la experiencia del usuario en una aplicación personalizada con usuarios locales después de importar un archivo CSV. Si sus usuarios inician sesión con la interfaz de usuario alojada, Amazon Cognito les pedirá que establezcan una nueva contraseña cuando inicien sesión por primera vez.

Obligación de que los usuarios importados restablezcan sus contraseñas

1. En la aplicación, intente iniciar sesión de forma silenciosa para el usuario actual con `InitiateAuth` mediante una contraseña aleatoria.
2. Amazon Cognito devuelve `NotAuthorizedException` cuando está habilitado `PreventUserExistenceErrors`. De lo contrario, devuelve `PasswordResetRequiredException`.
3. Su aplicación realiza una solicitud de API `ForgotPassword` y restablece la contraseña del usuario.

    a. La aplicación envía el nombre de usuario en una solicitud de API `ForgotPassword`.

    b. Amazon Cognito envía un código al correo electrónico o número de teléfono verificados. El destino depende de los valores que haya proporcionado para `email_verified` y `phone_number_verified` en su archivo CSV. La respuesta a la solicitud `ForgotPassword` indica el destino del código.

> **Note**
>
> Su grupo de usuarios debe estar configurado para verificar correos electrónicos o números de teléfono. Para obtener más información, consulte Inscripción y confirmación de cuentas de usuario.

c. Su aplicación muestra un mensaje a su usuario para que compruebe la ubicación a la que se envió el código y le pide que ingrese el código y una nueva contraseña.

d. El usuario introduce el código y una nueva contraseña en la aplicación.

e. La aplicación envía el código y la nueva contraseña en una solicitud de API `ConfirmForgotPassword`.

f. La aplicación redirige al usuario para que inicie sesión.

Custom pool attributes (

Los atributos son fragmentos de información de usuarios individuales, como su nombre, la dirección de correo electrónico o su número de teléfono, que ayudan a identificarlos. Los grupos de usuarios nuevos tienen un conjunto de atributos estándar predeterminados. También puede añadir atributos personalizados a la definición de su grupo de usuarios en AWS Management Console. En este tema se describen estos atributos en detalle y se le ofrecen consejos sobre cómo configurar el grupo de usuarios.

No almacene toda la información de los usuarios en atributos. Por ejemplo, guarda los datos de los usuarios que cambien con frecuencia, como las puntuaciones en juegos o las estadísticas de uso, en un almacén de datos independiente, como Amazon Cognito Sync o Amazon DynamoDB.

> **Note**
>
> Algunos documentos y estándares hacen referencia a los atributos como miembros.

Temas

- Atributos estándar
- Nombres de usuario y nombres de usuario preferidos
- Personalización de los atributos de inicio de sesión

- Custom attributes (Atributos personalizados)
- Permisos y ámbitos de los atributos

Atributos estándar

Amazon Cognito asigna a todos los usuarios un conjunto de atributos estándar en función de la OpenID Connect specification. De forma predeterminada, los valores de atributo estándar y personalizados pueden tener un máximo de 2048 caracteres, aunque algunos valores de atributo tienen restricciones de formato.

Los atributos estándar son:

- `address`
- `birthdate`
- `email`
- `family_name`
- `gender`
- `given_name`
- `locale`
- `middle_name`
- `name`
- `nickname`
- `phone_number`
- `picture`
- `preferred_username`
- `profile`
- `sub`
- `updated_at`
- `website`
- `zoneinfo`

A excepción de `sub`, los atributos estándar son opcionales de forma predeterminada para todos los usuarios. Para que un atributo sea obligatorio, durante el proceso de creación del grupo de usuarios,

seleccione laObligatorioLa casilla de verificación situada junto al atributo. Amazon Cognito asigna un valor de identificador de usuario único al atributo `sub` de cada usuario. Solo se pueden verificar los atributos email y phone_number.

> **Note**
>
> Cuando un atributo estándar se marca como Required (Obligatorio), el usuario no puede registrarse, salvo que indique un valor para el atributo. Para crear usuarios y no proporcionar valores para los atributos obligatorios, los administradores pueden usar la AdminCreateUserAPI. Después de crear un grupo de usuarios, no puede cambiar un atributo de obligatorio a no obligatorio y viceversa.

Detalles de atributos estándar y restricciones de formato

birthdate

El valor debe ser una fecha válida de 10 caracteres con el formato AAAA-MM-DD.

email

Los usuarios y los administradores pueden verificar los valores de las direcciones de correo electrónico.

Un administrador con Cuenta de AWS los permisos adecuados puede cambiar la dirección de correo electrónico del usuario y también marcarla como verificada. Marca una dirección de correo electrónico como verificada con la AdminUpdateUserAttributesAPI o el comando admin-update-user-attributes AWS Command Line Interface (AWS CLI). Este comando permite al administrador cambiar el atributo `email_verified` a `true`. También puedes editar un usuario en la pestaña Usuarios de AWS Management Console para marcar una dirección de correo electrónico como verificada.

El valor debe ser una cadena de dirección de correo electrónico válida que siga el formato de correo electrónico estándar con el símbolo @ y el dominio, con una longitud máxima de 2048 caracteres.

phone_number

Si la autenticación multifactor (MFA) por SMS está activa, el usuario debe proporcionar un número de teléfono. Para obtener más información, consulte Adición de MFA a un grupo de usuarios..

Los usuarios y los administradores pueden verificar los valores de números de teléfono.

Un administrador con Cuenta de AWS los permisos adecuados puede cambiar el número de teléfono del usuario y también marcarlo como verificado. Marca un número de teléfono como verificado con la AdminUpdateUserAttributesAPI o el admin-update-user-attributes AWS CLI comando. Este comando permite al administrador cambiar el atributo `phone_number_verified` a `true`. También puedes editar un usuario en la pestaña Usuarios AWS Management Console para marcar un número de teléfono como verificado.

> **Important**
>
> Los números de teléfono deben cumplir con las reglas de formato siguientes: deben comenzar por un signo más (**+**) seguido inmediatamente por el código de país. Un número de teléfono solo puede contener el signo **+** y dígitos. Elimine cualquier otro carácter dentro del número de teléfono como, por ejemplo, paréntesis, espacios o guiones (**-**) antes de enviar el valor al servicio. Por ejemplo, un número de teléfono de Estados Unidos debe tener este formato: **`+14325551212`**.

preferred_username

Puede seleccionar `preferred_username` según sea necesario o como alias, pero no ambas opciones. Si `preferred_username` se trata de un alias, puede realizar una solicitud a la operación de la UpdateUserAttributesAPI y añadir el valor del atributo después de confirmar el usuario.

sub

Indexe y busque los usuarios en función del atributo `sub`. El atributo `sub` es un identificador de usuario único dentro de cada grupo de usuarios. Los usuarios pueden cambiar atributos como `phone_number` y `email`. El atributo `sub` tiene un valor fijo. Para obtener más información sobre cómo encontrar a los usuarios, consulte Gestión y búsqueda de cuentas de usuario.

Ver atributos obligatorios

Utilice el siguiente procedimiento para ver los atributos obligatorios de un grupo de usuarios determinado.

> **Note**
>
> No puede cambiar los atributos obligatorios una vez que se haya creado el grupo de usuarios.

Para ver los atributos obligatorios

1. Vaya a Amazon Cognito en. AWS Management Console Si la consola se lo pide, introduzca sus credenciales. AWS
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija la pestaña Sign-up experience (Experiencia de inscripción).
5. Consulte en la sección Atributos obligatorios qué atributos son obligatorios en el grupo de usuarios.

Nombres de usuario y nombres de usuario preferidos

El valor `username` es un atributo independiente y no es el mismo que el del atributo `name`. Cada usuario tiene un atributo `username`. Amazon Cognito genera automáticamente un nombre de usuario para los usuarios federados. Debe proporcionar un atributo `username` para crear un usuario local en el directorio de Amazon Cognito. Después de crear un usuario, no puede cambiar el valor del atributo `username`.

Los desarrolladores pueden utilizar el atributo `preferred_username` para dar a los usuarios un nombre de usuario que estos puedan cambiar. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

Si la aplicación no exige un nombre de usuario, no tiene que pedir al usuario que proporcione uno. La aplicación puede crear un nombre de usuario único para los usuarios en segundo plano. Esto es útil si, por ejemplo, quiere que los usuarios se registren e inicien sesión con una dirección de correo electrónico y una contraseña. Para obtener más información, consulte Personalización de los atributos de inicio de sesión.

El `username` debe ser único en el grupo de usuarios. Si bien los valores `username` pueden volver a utilizarse, solo es posible hacerlo después de haberse eliminado y ya no se estén usando. Para

obtener información sobre las restricciones de cadena de los `username` atributos, consulta la propiedad username de una solicitud de SignUpAPI.

Personalización de los atributos de inicio de sesión

Al crear un grupo de usuarios, puede configurar los atributos de nombre de usuario si desea que los usuarios puedan registrarse e iniciar sesión con una dirección de correo electrónico o un número de teléfono como nombre de usuario. También puede establecer atributos de alias para dar a los usuarios la opción de incluir varios atributos cuando se registren y, a continuación, iniciar sesión con un nombre de usuario, un nombre de usuario preferido, una dirección de correo electrónico o un número de teléfono.

> **Important**
>
> Una vez que se haya creado el grupo de usuarios, no se podrá cambiar esta opción.

Cómo elegir entre atributos de alias y atributos de nombre de usuario

Su requisito	Atributos de alias	Atributos de nombre de usuario
Los usuarios tienen varios atributos de inicio de sesión	Sí[1]	No[2]
Los usuarios deben verificar la dirección de correo electrónico o el número de teléfono antes de poder iniciar sesión con ellos	Sí	No
Registra a los usuarios con direcciones de correo electrónico o números de teléfono duplicados y evita `UsernameExistsException` errores[3]	Sí	No

Su requisito	Atributos de alias	Atributos de nombre de usuario
Puede asignar el mismo valor de atributo de dirección de correo electrónico o número de teléfono a más de un usuario	Sí[4]	No

[1] Los atributos de inicio de sesión disponibles son: nombre de usuario, dirección de correo electrónico, número de teléfono y nombre de usuario preferido.

[2] Pueden iniciar sesión con la dirección de correo electrónico o con el número de teléfono.

[3] El grupo de usuarios no genera errores `UsernameExistsException` cuando los usuarios se registran con direcciones de correo electrónico o números de teléfono potencialmente duplicados, pero sin nombre de usuario. Este comportamiento es independiente de Evite errores de existencia del nombre de usuario, que se aplica a las operaciones de inicio de sesión, pero no a las de registro.

[4] Solo el último usuario que haya verificado el atributo podrá iniciar sesión con él.

Opción 1: múltiples atributos de inicio de sesión (atributos de alias)

Puede activar los alias si quiere que los usuarios tengan la opción de elegir ingresar el nombre de usuario u otros valores de atributos al iniciar sesión. De forma predeterminada, los usuarios inician sesión con su nombre de usuario y una contraseña. El nombre de usuario es un valor fijo que los usuarios no pueden cambiar. Si marca un atributo como alias, los usuarios pueden iniciar sesión con dicho atributo en vez de usar el nombre de usuario. Los atributos de dirección de correo electrónico, número de teléfono y nombre de usuario preferido pueden marcarse como alias. Por ejemplo, si el correo electrónico y el teléfono se seleccionan como alias de un grupo de usuarios, los usuarios de dicho grupo de usuarios pueden iniciar sesión utilizando el nombre de usuario, la dirección de correo electrónico o el número de teléfono, junto con la contraseña.

Para elegir los atributos de alias, seleccione User name (Nombre de usuario) y al menos una opción de inicio de sesión adicional al crear su grupo de usuarios.

> **Note**
>
> Cuando configura el grupo de usuarios para que no tenga en cuenta el uso de mayúsculas o minúsculas, un usuario puede usar minúsculas o mayúsculas al registrarse o iniciar sesión con su alias. Para obtener más información, consulte la CreateUserPoolreferencia de la API de grupos de usuarios de Amazon Cognito.

Si selecciona la dirección de correo electrónico como alias, Amazon Cognito no aceptará un nombre de usuario que coincida con un formato de dirección de correo electrónico válido. Del mismo modo, si selecciona el número de teléfono como alias, Amazon Cognito no aceptará un nombre de usuario para ese grupo de usuarios que coincida con un formato de número de teléfono válido.

> **Note**
>
> Los valores de alias tienen que ser únicos en un grupo de usuarios. Si se configura un alias para una dirección de correo electrónico o un número de teléfono, el valor proporcionado puede estar en estado verificado solo en una cuenta. Durante el registro, si el usuario proporciona una dirección de correo electrónico o un número de teléfono como valor de alias y otro usuario ya ha utilizado ese valor de alias, el registro se realiza correctamente. No obstante, cuando el usuario intente confirmar la cuenta con ese correo electrónico (o ese número de teléfono) y especifique el código válido, devolverá un error `AliasExistsException`. El error indica al usuario que ya existe una cuenta con ese correo electrónico (o ese número de teléfono). En este punto, el usuario puede desistir de crear una cuenta nueva e intentar restablecer la contraseña de la cuenta antigua. Si el usuario sigue creando la cuenta nueva, la aplicación debe llamar a la API de `ConfirmSignUp` con la opción `forceAliasCreation`. `ConfirmSignUp` con `forceAliasCreation` pasa el alias de la cuenta anterior a la cuenta recién creada y marca el atributo como no verificado en la cuenta anterior.

Los números de teléfono y las direcciones de correo electrónico pasan a ser alias activos de los usuarios únicamente cuando estos verifican los números de teléfono y las direcciones de correo electrónico. Recomendamos que elija la verificación automática de las direcciones de correo electrónico y los números de teléfono si los usa como alias.

Elija atributos de alias para evitar errores `UsernameExistsException` en los atributos de dirección de correo electrónico y número de teléfono cuando sus usuarios se registren.

Active el atributo `preferred_username` para que el usuario pueda cambiar el nombre de usuario que utiliza para iniciar sesión mientras su valor de atributo `username` no cambie. Si desea habilitar esta experiencia de usuario, envíe el nuevo valor de `username` como `preferred_username` y elija `preferred_username` como alias. Esto permitirá a los usuarios iniciar sesión con el valor nuevo que han especificado. Si se ha seleccionado `preferred_username` como alias, el usuario puede proporcionar el valor solo cuando confirma la cuenta. Este valor no se puede proporcionar en el momento de registro.

Cuando el usuario se registra con un nombre de usuario, usted puede elegir si puede iniciar sesión con uno o más de los alias siguientes.

- Dirección de correo electrónico verificada
- Número de teléfono verificado
- Nombre de usuario preferido

Los usuarios pueden cambiar estos alias después de registrarse.

> **Important**
>
> Si el grupo de usuarios admite el inicio de sesión con alias y desea autorizar o buscar a un usuario, no lo identifique por ninguno de sus atributos de inicio de sesión. El identificador de usuario de valor fijo `sub` es el único indicador coherente de la identidad del usuario.

Incluya los siguientes pasos al crear el grupo de usuarios para que los usuarios puedan iniciar sesión con un alias.

Para configurar un grupo de usuarios para iniciar de sesión con un nombre de usuario preferido

1. Diríjase a Amazon Cognito en la AWS Management Console. Si la consola se lo pide, introduzca sus credenciales. AWS
2. ElegirGrupos de usuarios de.
3. En la esquina superior derecha de la página, elija Create a User Pool (Crear un grupo de usuarios).
4. EnConfigurar la experiencia de inicio de sesión, elige la identidadTipos de proveedoresque desea asociar a su grupo de usuarios.

5. En Cognito user pool sign-in options (Opciones de inicio de sesión del grupo de usuarios de Cognito), elija cualquier combinación de User name (Nombre de usuario), Email (Correo electrónico) y Phone number (Número de teléfono).
6. En Requisitos para nombre de usuario, elija Permitir a los usuarios iniciar sesión con un nombre de usuario preferido para que los usuarios puedan establecer un nombre de usuario alternativo para iniciar sesión.
7. Elija Next (Siguiente) y, a continuación, complete todos los pasos del asistente.

Opción 2: dirección de correo electrónico o número de teléfono como atributo de inicio de sesión (atributos de nombre de usuario)

Puede elegir si el usuario solo puede registrarse con una dirección de correo electrónico, solo con un número de teléfono o con cualquiera de estas dos opciones cuando este se registra con una dirección de correo electrónico o un número de teléfono como nombre de usuario.

Para elegir los atributos de nombre de usuario, no seleccione Nombre de usuario como opción de inicio de sesión cuando cree el grupo de usuarios.

El correo electrónico o el número de teléfono deben ser únicos y no pueden estar siendo utilizados por otro usuario. No se tiene que verificar. Después de que el usuario se haya registrado con un correo electrónico o un número de teléfono, no podrá crear una cuenta con el mismo correo electrónico o con el mismo número de teléfono, solo podrá reutilizar la cuenta existente (y restablecer la contraseña si es necesario). El usuario solo puede reutilizar la cuenta existente y restablecer la contraseña de la cuenta, si esto fuera necesario. No obstante, el usuario puede cambiar la dirección de correo electrónico o el número de teléfono por otro nuevo. Si la dirección de correo electrónico o el número de teléfono no se están usando, pasará a ser el nuevo nombre de usuario.

> **Note**
>
> Si un usuario se registra con una dirección de correo electrónico como nombre de usuario, puede cambiarlo por otra dirección de correo electrónico, pero no por un número de teléfono. Si se registra con un número de teléfono, puede cambiar el nombre de usuario por otro número de teléfono, pero no por una dirección de correo electrónico.

Siga estos pasos a la hora de crear el grupo de usuarios para configurar el registro y el inicio de sesión con una dirección de correo electrónico o con un número de teléfono.

Para configurar un grupo de usuarios para registrarse e iniciar sesión con un correo electrónico o un número de teléfono

1. Diríjase a Amazon Cognito en la AWS Management Console. Si la consola se lo pide, introduzca sus credenciales. AWS
2. ElegirGrupos de usuarios de.
3. En la esquina superior derecha de la página, elija Create a User Pool (Crear un grupo de usuarios).
4. En Cognito user pool sign-in options (Opciones de inicio de sesión del grupo de usuarios de Cognito), elija cualquier combinación de Email (Correo electrónico) y Phone number (Número de teléfono) que represente los atributos de alias que el usuario puede usar para iniciar sesión.
5. Elija Next (Siguiente) y, a continuación, complete los pasos restantes del asistente.

> **Note**
>
> No tiene que marcar la dirección de correo electrónico o el número de teléfono como atributos obligatorios para el grupo de usuarios.

Para implementar la opción 2 en la aplicación

1. Llame a la API `CreateUserPool` para crear el grupo de usuarios. Establezca el parámetro `UserNameAttributes` en `phone_number`, `email` o `phone_number | email`.
2. Llame a la API `SignUp` y pase una dirección de correo electrónico o un número de teléfono en el parámetro `username` de la API. Esta API admite lo siguiente:
   - Si la cadena `username` tiene un formato de correo electrónico válido, el grupo de usuarios rellena automáticamente el atributo `email` del usuario con el valor `username`.
   - Si la cadena `username` tiene un formato de número de teléfono válido, el grupo de usuarios rellena automáticamente el atributo `phone_number` del usuario con el valor `username`.
   - Si el formato de cadena `username` no es un formato de dirección de correo electrónico o de número de teléfono, la API de `SignUp` genera una excepción.
   - La API de `SignUp` genera un UUID persistente para el usuario y lo utiliza internamente como el atributo de nombre de usuario inmutable. Este UUID tiene el mismo valor que la notificación `sub` en el token de identidad de usuario.

- Si la cadena `username` contiene una dirección de correo electrónico o un número de teléfono que ya se está usando, la API de `SignUp` genera una excepción.

Puede utilizar una dirección de correo electrónico o un número de teléfono como alias en lugar del nombre de usuario en todas las API, excepto la API `ListUsers`. Cuando llama a `ListUsers`, puede buscar por el atributo `email` o el atributo `phone_number`. Si busca por `username`, debe proporcionar el nombre de usuario real, no un alias.

Custom attributes (Atributos personalizados)

Puede añadir hasta 50 atributos personalizados a un grupo de usuarios. Puede especificar la longitud mínima o máxima de los atributos personalizados. Sin embargo, la longitud máxima de ningún atributo personalizado puede superar los 2048 caracteres.

Cada atributo personalizado incluye las siguientes características:

- Puede definirlo como cadena o como número. Amazon Cognito escribe valores de atributos personalizados en el token de ID solo como cadenas.
- No puede exigir que los usuarios proporcionen un valor para el atributo.
- No puede eliminarlo ni cambiarlo después de agregarlo al grupo de usuarios.
- La longitud de caracteres del nombre de atributo se encuentra dentro del límite aceptable por parte de Amazon Cognito. Para obtener más información, consulte Cuotas en Amazon Cognito.
- Puede ser mutable o inmutable. Solo se puede escribir un valor en un atributo inmutable la primera vez que se crea un usuario. Puede cambiar el valor de un atributo mutable si el cliente de la aplicación tiene permiso de escritura para el atributo. Para obtener más información, consulte Permisos y ámbitos de los atributos.

> **Note**
>
> En el código y en la configuración de reglas para Uso del control de acceso basado en roles, los atributos personalizados han de llevar el prefijo `custom:` para diferenciarse de los atributos estándar.

También puede añadir atributos de desarrollador al crear grupos de usuarios, en la `SchemaAttributes` propiedad de CreateUserPool. Los atributos del desarrollador tienen

un prefijo `dev:`. Solo puede modificar los atributos de desarrollador de un usuario con AWS credenciales. Los atributos de desarrollador son una característica antigua que Amazon Cognito sustituyó por permisos de lectura-escritura del cliente de la aplicación.

Utilice el siguiente procedimiento para crear una en un almacén de claves personalizado.

Para añadir un atributo personalizado con la consola

1. Vaya a Amazon Cognito en. AWS Management Console Si la consola se lo pide, introduzca sus credenciales. AWS
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija la pestaña Sign-up experience (Experiencia de inscripción) y en la pestaña Custom attributes (Atributos personalizados), elija Add custom attributes (Agregar atributos personalizados).
5. En la páginaAgregar atributos personalizados, proporcione los siguientes detalles sobre el nuevo atributo:
   - Escriba un Name (nombre).
   - Seleccione Type (tipo), ya sea String (cadena) o Number (número).
   - Escriba una longitud de cadena o un valor numérico Min (mínima).
   - Escriba una longitud de cadena o un valor numérico Max (máximo).
   - Seleccione Mutable (Mutable) si desea dar permiso a los usuarios para cambiar el valor de un atributo personalizado después de establecer el valor inicial.
6. Elija Guardar cambios.

Permisos y ámbitos de los atributos

Puede establecer permisos de lectura y escritura para cada atributo de usuario para cada una de sus aplicaciones de cliente. Esto permite controlar el acceso del que dispone cualquier aplicación para leer y modificar cada atributo que se almacene para los usuarios. Por ejemplo, puede tener un atributo personalizado que indique si el usuario es cliente de pago o no. Es posible que sus aplicaciones puedan ver este atributo, pero no cambiarlo directamente. Por lo tanto, puede actualizar el atributo mediante una herramienta administrativa o un proceso de fondo. Los permisos para atributos de usuario se pueden configurar desde la consola de Amazon Cognito, la API de Amazon Cognito o la AWS CLI. De forma predeterminada, los nuevos atributos personalizados no están

disponibles hasta que defina permisos de lectura y escritura para ellos. De forma predeterminada, cuando creas un nuevo cliente de aplicación, le concedes permisos de lectura y escritura para todos los atributos estándar y personalizados. Para limitar la aplicación a solo la cantidad de información que necesita, asigne permisos específicos a los atributos de la configuración del cliente de la aplicación.

Como práctica recomendada, especifica los permisos de lectura y escritura de los atributos al crear un cliente de aplicación. Conceda al cliente de la aplicación acceso al conjunto mínimo de atributos de usuario que necesita para el funcionamiento de la aplicación.

> **Note**
>
> DescribeUserPoolClientsolo devuelve valores para `ReadAttributes` y `WriteAttributes` cuando configuras permisos de cliente de aplicaciones distintos de los predeterminados.

Para actualizar los permisos de los atributos (AWS Management Console)

1. Vaya a Amazon Cognito en. AWS Management Console Si la consola se lo pide, introduzca sus credenciales. AWS
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija la pestaña App integration (Integración de aplicaciones) y en la sección App clients (Clientes de aplicaciones), elija un cliente de aplicación de la lista.
5. En la sección Attribute read and write permissions (Permisos de lectura y escritura de atributos), elija Edit (Editar).
6. En la página Edit attribute read and write permissions (Editar permisos de lectura y escritura de atributos), configure los permisos de lectura y escritura y, a continuación, elija Save changes (Guardar cambios).

Repita estos pasos para cada cliente de aplicación que utilice el atributo personalizado.

Por cada aplicación, puede marcar los atributos como de lectura o escritura. Esto es cierto para los atributos estándar y los atributos personalizados. La aplicación puede recuperar el valor de los atributos que marque como legibles y puede establecer o modificar el valor de los atributos que marque como que admiten la escritura. Si la aplicación intenta establecer un valor para un

atributo que no está autorizada a escribir, Amazon Cognito devuelve `NotAuthorizedException`. GetUserlas solicitudes incluyen un token de acceso con una reclamación del cliente de la aplicación; Amazon Cognito solo devuelve valores de los atributos que el cliente de la aplicación puede leer. El token de ID de usuario de una aplicación solo contiene afirmaciones que corresponden a los atributos legibles. Todos los clientes de la aplicación pueden escribir los atributos necesarios para el grupo de usuarios. Solo puede establecer el valor de un atributo en una solicitud de la API de grupos de usuarios de Amazon Cognito si también proporciona un valor para los atributos obligatorios que aún no tienen un valor.

Los atributos personalizados tienen características distintas para permisos de lectura y escritura. Puede crearlos como mutables o inmutables para el grupo de usuarios y puede configurarlos como atributos de lectura o escritura para cualquier cliente de la aplicación.

Un atributo personalizado inmutable se puede actualizar una vez, durante la creación del usuario. Puede rellenar un atributo inmutable con los siguientes métodos.

- `SignUp`: un usuario se registra en un cliente de la aplicación que tiene acceso de escritura a un atributo personalizado inmutable. Proporcionan un valor para ese atributo.
- Inicio de sesión con un IdP externo: un usuario inicia sesión en un cliente de la aplicación que tiene acceso de escritura a un atributo personalizado inmutable. La configuración del grupo de usuarios para su IdP tiene una regla para asignar una notificación proporcionada a un atributo inmutable.
- `AdminCreateUser`: usted proporciona un valor para un atributo inmutable.

Para obtener información sobre los ámbitos que puede asignar a los clientes de la aplicación, consulte Autorización de alcances, M2M y API con servidores de recursos.

Puede cambiar los permisos y los alcances de los atributos después de crear el grupo de usuarios.

Adición de requisitos de contraseña para los grupos de usuarios

Las contraseñas complejas y seguras son una práctica recomendada de seguridad para su grupo de usuarios. Especialmente en las aplicaciones que están abiertas a Internet, las contraseñas poco seguras pueden exponer las credenciales de los usuarios a sistemas que las adivinen e intenten acceder a sus datos. Cuanto más compleja sea una contraseña, más difícil será adivinarla. Amazon Cognito cuenta con herramientas adicionales para los administradores preocupados por la seguridad, como funciones de seguridad avanzadas y ACL AWS WAF web, pero su política de contraseñas es un elemento central de la seguridad de su directorio de usuarios.

Las contraseñas de los usuarios locales de los grupos de usuarios de Amazon Cognito no caducan automáticamente. Como práctica recomendada, registre la hora, la fecha y los metadatos del restablecimiento de las contraseñas de los usuarios en un sistema externo. Con un registro externo de la antigüedad de la contraseña, su aplicación o un activador de Lambda pueden buscar la antigüedad de la contraseña de un usuario y requerir que se restablezca después de un período determinado.

Puede configurar su grupo de usuarios para que requiera una complejidad de contraseña mínima que se ajuste a sus estándares de seguridad. Las contraseñas complejas tienen una longitud mínima de ocho caracteres. También incluyen una combinación de caracteres mayúsculas, numéricos y especiales.

Para restablecer una política de contraseñas de grupo de usuarios

1. Cree un grupo de usuarios y vaya al paso Configurar los requisitos de seguridad, o acceda a un grupo de usuarios existente y vaya a la pestaña Experiencia de inicio de sesión.
2. Vaya a Política de contraseñas.
3. Seleccione Modo de política de contraseñas. Valores predeterminados de Cognito configura su grupo de usuarios con la configuración mínima recomendada. También puede elegir una política de contraseñas personalizada.
4. Establezca una Longitud mínima de la contraseña. Todos los usuarios deben registrarse o crearse con una contraseña cuya longitud sea mayor o igual a este valor. Puede establecer este valor mínimo en 99, pero sus usuarios pueden establecer contraseñas de hasta 256 caracteres.
5. Configure las reglas de complejidad de las contraseñas en Requisitos de contraseña. Elija los tipos de caracteres (números, caracteres especiales, letras mayúsculas y minúsculas) que desee incluir, uno como mínimo, en la contraseña de cada usuario.

   Puede requerir al menos uno de los siguientes caracteres en las contraseñas. Una vez que Amazon Cognito compruebe que las contraseñas contienen el mínimo de caracteres necesario, las contraseñas de los usuarios pueden contener caracteres adicionales de cualquier tipo hasta alcanzar la longitud máxima de la contraseña.

   - Letras del alfabeto latino básico en mayúsculas y minúsculas
   - Números
   - Los siguientes caracteres especiales.

     ```
     ^ $ * . [ ] { } ( ) ? " ! @ # % & / \ , > < ' : ; | _ ~ ` = + -
     ```

- Caracteres sin espacios al principio ni al final.

6. Establezca un valor para Contraseñas temporales establecidas por los administradores que caducan en. Transcurrido este periodo, un nuevo usuario que haya creado en la consola de Amazon Cognito o con `AdminCreateUser` no podrá iniciar sesión ni establecer una contraseña nueva. Después de iniciar sesión con su contraseña temporal, sus cuentas de usuario nunca caducan. Para actualizar la duración de la contraseña en la API de grupos de usuarios de Amazon Cognito, defina un valor para su TemporaryPasswordValidityDays solicitud CreateUserPoolo la UpdateUserPoolAPI.

   - Para restablecer el acceso de una cuenta de usuario caducada, realice una de las siguientes acciones.

     - Elimine el perfil de usuario y cree uno nuevo.
     - Establezca una nueva contraseña permanente en una solicitud de AdminSetUserPasswordAPI.
     - Genera un nuevo código de confirmación en una solicitud de AdminResetUserPasswordAPI.

Configuración de correo electrónico para grupos de usuarios de Amazon Cognito

Algunos eventos de la aplicación cliente de su grupo de usuarios pueden provocar que Amazon Cognito envíe un email a sus usuarios. Por ejemplo, si configura su grupo de usuarios para que se exija la verificación de correo electrónico, Amazon Cognito envía un correo electrónico cuando un usuario se registra con una cuenta nueva en su aplicación o cuando restablece su contraseña. En función de la acción que inicie el correo electrónico, el correo electrónico contendrá un código de verificación o una contraseña temporal.

Para administrar la entrega de correo electrónico, puede utilizar cualquiera de las siguientes opciones:

- La configuración de correo electrónico predeterminada que está integrada en el servicio Amazon Cognito.
- Su configuración de Amazon Simple Email Service (Amazon SES).

Puede cambiar la opción de entrega después de crear el grupo de usuarios.

Amazon Cognito envía mensajes de correo electrónico a los usuarios con un código que pueden ingresar o un enlace URL que pueden seleccionar. En la siguiente tabla se muestran los eventos que pueden generar un mensaje de correo electrónico.

Opciones de mensajes

Actividad	Operación de la API	Opciones de entrega	Opciones de formato	Personalizable	Plantilla de mensaje
¿Ha olvidado la contraseña	ForgotPassword	Correo electrónico, SMS	Código	No	N/A
Invitación	AdminCreateUser	Correo electrónico, SMS	Código	Sí	Mensaje de invitación
Autorregistro	SignUp	Correo electrónico, SMS	código, enlace	Sí	Mensaje de verificación
Verificación de dirección de correo electrónico o número de teléfono	UpdateUserAttributes	Correo electrónico, SMS	Código	Sí	Mensaje de verificación
Autenticación multifactor (MFA)	AdminInitiateAuth, InitiateAuth	SMS	Código	Sí[1]	Mensaje de MFA

[1] Para mensajes SMS.

Amazon SES cobra por los mensajes de correo electrónico. Para obtener más información, consulte Precios de Amazon SES.

Configuración de correo electrónico predeterminada

Amazon Cognito puede usar su configuración de correo electrónico predeterminada para gestionar las entregas de correo electrónico por usted. Si utiliza la opción predeterminada, Amazon Cognito solo permite una cantidad limitada de correos electrónicos al día para su grupo de usuarios. Para obtener más información sobre Service Limits, consulte Cuotas en Amazon Cognito. En el caso de entornos de producción típicos, el límite de correo electrónico predeterminado está por debajo del volumen de entrega requerido. Para habilitar un mayor volumen de envíos, debe utilizar la configuración de email de Amazon SES.

Cuando utiliza la configuración predeterminada, utiliza los recursos de Amazon SES que administra AWS para enviar mensajes de correo electrónico. Amazon SES agrega direcciones de correo electrónico que devuelven un rechazo permanente a una lista de supresión de nivel de cuenta o una lista de supresión global. Si una dirección de correo electrónico que no se puede entregar pasa a ser entregable más adelante, no podrá controlar su eliminación de la lista de supresión mientras su grupo de usuarios esté configurado para usar la configuración predeterminada. Una dirección de correo electrónico puede permanecer indefinidamente en la lista de AWS supresión gestionada. Para administrar las direcciones de correo electrónico que no se pueden entregar, utilice la configuración de correo electrónico de Amazon SES con una lista de supresión en el nivel de cuenta, tal y como se describe en la siguiente sección.

Con la opción predeterminada, puede utilizar cualquiera de las siguientes direcciones de correo electrónico como dirección del remitente:

- La dirección de correo electrónico predeterminada, no-reply@verificationemail.com.
- Una dirección de correo electrónico personalizada. Para poder utilizar su propia dirección de correo electrónico, debe verificarla con Amazon SES y conceder permiso a Amazon Cognito para utilizarla.

Configuración de email de Amazon SES

Es posible que su aplicación requiera un volumen de entregas superior al disponible con la opción predeterminada. Para aumentar el posible volumen de envíos, use los recursos de Amazon SES con su grupo de usuarios para enviar un correo electrónico a los usuarios. También puede supervisar la actividad de envío de correo electrónicocuando envía mensajes de correo electrónico con su propia configuración de Amazon SES.

Para poder usar la configuración de Amazon SES, debe verificar una o más direcciones de email con Amazon SES. Use una dirección de correo electrónico verificada, o una dirección de un dominio verificado, como la dirección de correo electrónico del remitente que asigne a su grupo de usuarios. Cuando Amazon Cognito envía un mensaje de correo electrónico, llama a Amazon SES por usted y utiliza su dirección de correo electrónico.

Cuando utilice la configuración de Amazon SES, se aplicarán las siguientes condiciones:

- Los límites de envío de correo electrónico para su grupo de usuarios son los mismos que se aplican a su dirección de correo electrónico verificada de Amazon SES en su cuenta de Cuenta de AWS.
- Puede administrar sus mensajes a direcciones de correo electrónico que no se pueden entregar con una lista de supresión en el nivel de cuenta en Amazon SES, la cual anula la lista de supresión global. Cuando utilizas una lista de supresión en el nivel de cuenta, los rechazos de mensajes de correo electrónico afectan a la reputación de su cuenta como remitente. Para obtener más información, consulte Uso de la lista de supresión de nivel de cuenta de Amazon SES en la guía para desarrolladores de Amazon Simple Email Service.

Regiones de configuración de correo electrónico de Amazon SES

El Región de AWS lugar donde cree un grupo de usuarios tendrá uno de los tres requisitos para la configuración de los mensajes de correo electrónico con Amazon SES. Puede enviar mensajes de correo electrónico desde Amazon SES desde la misma región que su grupo de usuarios, desde varias regiones, incluida la misma región, o desde una o más regiones remotas. Para obtener el mejor rendimiento, envíe mensajes de correo electrónico con una identidad verificada de Amazon SES en la misma región que su grupo de usuarios cuando tenga la opción.

Categorías de requisitos regionales para las identidades verificadas de Amazon SES

Solo en la región

Sus grupos de usuarios pueden enviar mensajes de correo electrónico con identidades verificadas al Región de AWS igual que el grupo de usuarios. En la configuración de correo electrónico predeterminada sin una dirección de `FROM` correo electrónico personalizada, Amazon Cognito utiliza una identidad `no-reply@verificationemail.com` verificada en la misma región.

Compatible con versiones anteriores

Sus grupos de usuarios pueden enviar mensajes de correo electrónico con identidades verificadas en la misma región Región de AWS o en una de las siguientes regiones alternativas:

- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Oregón)
- Europa (Irlanda)

Esta función permite la continuidad de los recursos del grupo de usuarios que podría haber creado para cumplir con los requisitos de Amazon Cognito cuando se lanzó el servicio. Los grupos de usuarios de ese período solo podían enviar mensajes de correo electrónico con identidades verificadas en un número limitado de Regiones de AWS. En la configuración de correo electrónico predeterminada sin una dirección de `FROM` correo electrónico personalizada, Amazon Cognito utiliza una identidad `no-reply@verificationemail.com` verificada en la misma región.

Región alternativa

Sus grupos de usuarios pueden enviar mensajes de correo electrónico con identidades verificadas en una alternativa Región de AWS que se encuentre fuera de la región del grupo de usuarios. Esta configuración se produce cuando Amazon SES no está disponible en una región en la que Amazon Cognito esté disponible.

La política de autorización de envío de Amazon SES para su identidad verificada en la región alternativa debe confiar en el director de servicio de Amazon Cognito de la región de origen. Para obtener más información, consulte Para conceder permisos para usar la configuración de correo electrónico predeterminada.

En algunas de estas regiones, Amazon Cognito divide los mensajes de correo electrónico entre dos regiones alternativas para la configuración de correo electrónico predeterminada de. `COGNITO_DEFAULT` En estos casos, para utilizar una dirección de `FROM` correo electrónico personalizada, la política de autorización de envío de Amazon SES para su identidad verificada en cada región alternativa debe confiar en el director de servicio de Amazon Cognito de la región de origen. Para obtener más información, consulte Para conceder permisos para usar la configuración de correo electrónico predeterminada. Con la configuración de correo electrónico de Amazon SES `DEVELOPER` en estas regiones, debe usar una identidad verificada en la primera región de la lista y configurarla para que confíe en el director del servicio de Amazon Cognito en la región del grupo de usuarios. Por ejemplo, en un grupo de usuarios de Oriente Medio

(Emiratos Árabes Unidos), configure una identidad verificada en Europa (Fráncfort) para que sea de confianza`cognito-idp.me-central-1.amazonaws.com`. En la configuración de correo electrónico predeterminada sin una dirección de `FROM` correo electrónico personalizada, Amazon Cognito utiliza una identidad `no-reply@verificationemail.com` verificada en cada región.

> **Note**
>
> En la siguiente combinación de condiciones, debe especificar el `SourceArn` parámetro EmailConfigurationcon un comodín en el elemento Región, en el formato. `arn:${Partition}:ses:*:${Account}:identity/${IdentityName}` Esto permite a su grupo de usuarios enviar mensajes de correo electrónico con identidades verificadas idénticas a las suyas Cuenta de AWS en ambos Regiones de AWS casos.
>
> - EmailSendingAccount El tuyo es`COGNITO_DEFAULT`.
> - Quieres usar una `FROM` dirección personalizada.
> - Su grupo de usuarios envía correos electrónicos a una región alternativa.
> - Su grupo de usuarios tiene una segunda región [1]alternativa especificada en la siguiente tabla de regiones compatibles con Amazon SES.

Si crea un grupo de usuarios mediante programación (con un AWS SDK, la API o CLI de Amazon Cognito, o AWS CloudFormation), su grupo de usuarios AWS CDK envía mensajes de correo electrónico con la identidad de Amazon SES que el `SourceArn` parámetro especifica para su grupo de usuarios. EmailConfiguration La identidad de Amazon SES debe ocupar un espacio compatible Región de AWS. Si su `EmailSendingAccount` es `COGNITO_DEFAULT` y no especifica un parámetro `SourceArn`, Amazon Cognito envía mensajes de correo electrónico desde `no-reply@verificationemail.com` utilizando recursos de la región donde creó el grupo de usuarios.

En la siguiente tabla se muestra Regiones de AWS dónde puede utilizar las identidades de Amazon SES con Amazon Cognito.

Región del grupo de usuarios	Opción de región	Regiones compatibles con Amazon SES
Este de EE. UU. (Norte de Virginia)	Compatible con versiones anteriores	Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Este de EE. UU. (Ohio)	Compatible con versiones anteriores	Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia) , Oeste de EE. UU. (Oregón), Europa (Irlanda)
Oeste de EE. UU. (Norte de California)	Solo en la región	Oeste de EE. UU. (Norte de California)
Oeste de EE. UU. (Oregón)	Compatible con versiones anteriores	Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Canadá (centro)	Compatible con versiones anteriores	Canadá (centro), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Asia-Pacífico (Tokio)	Compatible con versiones anteriores	Asia-Pacífico (Tokio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Asia-Pacífico (Seúl)	Compatible con versiones anteriores	Asia-Pacífico (Seúl), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Asia-Pacífico (Bombay)	Compatible con versiones anteriores	Asia-Pacífico (Bombay), Este de EE. UU. (Norte de Virginia) , Oeste de EE. UU. (Oregón), Europa (Irlanda)

Región del grupo de usuarios	Opción de región	Regiones compatibles con Amazon SES
Asia-Pacífico (Hyderabad)	Región alternativa	Asia Pacífico (Bombay), Asia Pacífico (Singapur) [1]
Asia-Pacífico (Singapur)	Compatible con versiones anteriores	Asia-Pacífico (Singapur), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Asia-Pacífico (Sídney)	Compatible con versiones anteriores	Asia-Pacífico (Sídney), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Asia-Pacífico (Osaka)	Solo en la región	Asia-Pacífico (Osaka)
Asia-Pacífico (Yakarta)	Solo en la región	Asia-Pacífico (Yakarta)
Asia-Pacífico (Melbourne)	Región alternativa	Asia Pacífico (Sídney), Asia Pacífico (Singapur) [1]
Europa (Irlanda)	Compatible con versiones anteriores	Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Europa (Londres)	Compatible con versiones anteriores	Europa (Londres), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)
Europa (París)	Solo en la región	Europa (París)
Europa (Fráncfort)	Compatible con versiones anteriores	Europa (Londres), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Europa (Irlanda)

Región del grupo de usuarios	Opción de región	Regiones compatibles con Amazon SES
Europa (Zúrich)	Región alternativa	Europa (Fráncfort), Europa (Londres) [1]
Europa (Estocolmo)	Solo en la región	Europa (Estocolmo)
Europa (Milán)	Solo en la región	Europa (Milán)
Europa (España)	Región alternativa	Europa (París), Europa (Estocolmo) [1]
Medio Oriente (Baréin)	Solo en la región	Medio Oriente (Baréin)
Medio Oriente (EAU)	Región alternativa	Europa (Fráncfort), Europa (Londres) [1]
América del Sur (São Paulo)	Solo en la región	América del Sur (São Paulo)
Israel (Tel Aviv)	Solo en la región	Israel (Tel Aviv)
África (Ciudad del Cabo)	Solo en la región	África (Ciudad del Cabo)

[1] Se utiliza en grupos de usuarios con la configuración de correo electrónico predeterminada. Amazon Cognito distribuye los mensajes de correo electrónico entre identidades verificadas con la misma dirección de correo electrónico en cada región. Para usar una `FROM` dirección personalizada, configúrela `EmailConfiguration` con un `SourceArn` parámetro del formato. `arn:${Partition}:ses:*:${Account}:identity/${IdentityName}`

Configuración de correo electrónico para el grupo de usuarios

Siga los pasos que se indican a continuación para configurar las opciones de correo electrónico para el grupo de usuarios. Según la configuración que desee utilizar, es posible que deba completar los pasos con Amazon SES, AWS Identity and Access Management (IAM) y Amazon Cognito.

> **Note**
>
> Los recursos que cree en estos pasos no se pueden compartir entre Cuentas de AWS. Por ejemplo, no se puede configurar un grupo de usuarios en una cuenta con una dirección de email de Amazon SES que esté en otra cuenta. Por lo tanto, si utiliza Amazon Cognito en varias cuentas, recuerde repetir estos pasos en cada una de ellas.

Paso 1: Verificar su dirección de correo electrónico con Amazon SES

Antes de configurar su grupo de usuarios, debe verificar una o más direcciones de correo electrónico con Amazon SES si desea realizar alguna de las siguientes acciones:

- Usar su propia dirección de correo electrónico como dirección de remitente
- Uso de la configuración de Amazon SES para controlar el envío de correos electrónicos

Al verificar su dirección de correo electrónico, confirma que es la suya, lo que ayuda a evitar el uso no autorizado.

Para obtener más información sobre la verificación de email de Amazon SES, consulte Verificación de direcciones de email en la Guía para desarrolladores de Amazon Simple Email Service. Para obtener más información sobre cómo verificar un dominio con Amazon SES, consulte la sección Verificación de un dominio.

Paso 2: Quitar la cuenta del entorno de pruebas de Amazon SES

Omita este paso si utiliza la configuración de correo electrónico predeterminada de Amazon Cognito.

La primera vez que utilice Amazon SES en una región Región de AWS, estará Cuenta de AWS en el entorno limitado de Amazon SES de esa región. Amazon SES utiliza el entorno aislado para evitar el fraude y el abuso. Si utiliza su configuración de Amazon SES para administrar el envío de correos electrónicos, debe quitar su Cuenta de AWS del entorno aislado para que Amazon Cognito pueda enviar un correo electrónico a sus usuarios.

En el entorno de pruebas, Amazon SES impone restricciones sobre cuántos correos electrónicos puede enviar y a dónde puede enviarlos. Puede enviar correos electrónicos solo a direcciones y dominios que haya verificado con Amazon SES o puede enviarlos a direcciones del simulador de buzón de correo de Amazon SES. Mientras Cuenta de AWS permanezca en la zona de pruebas, no utilice su configuración de Amazon SES para aplicaciones que estén en producción. En esta

situación, Amazon Cognito no puede enviar mensajes a las direcciones de correo electrónico de sus usuarios.

Para sacarte Cuenta de AWS del entorno limitado, consulta Cómo salir del entorno limitado de Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service.

Paso 3: Conceder permisos de correo electrónico a Amazon Cognito

Es posible que tenga que conceder permisos específicos a Amazon Cognito para que pueda enviar correos electrónicos a sus usuarios. Los permisos que conceda y el proceso que utilice para concederlos dependen de si utiliza la configuración de correo electrónico predeterminada o la configuración de Amazon SES.

Para conceder permisos para usar la configuración de correo electrónico predeterminada

Complete este paso solo si configura su grupo de usuarios para que envíe correo electrónico con Cognito o esté configurado `EmailSendingAccount` en. `COGNITO_DEFAULT`

Con la configuración de correo electrónico predeterminada, su grupo de usuarios puede enviar mensajes de correo electrónico con cualquiera de las siguientes direcciones.

- La dirección predeterminadano-reply@verificationemail.com.
- Una dirección FROM personalizada de sus direcciones de correo electrónico o dominios verificados en Amazon SES.

Si utiliza una dirección personalizada, Amazon Cognito necesita otros permisos para poder usar esta dirección con el fin de enviar los email a sus usuarios. Estos permisos se otorgan mediante una política de autorización de envío para la dirección o el dominio de Amazon SES. Si utiliza la consola de Amazon Cognito para agregar una dirección personalizada al grupo de usuarios, la política se asoica automáticamente a la dirección de correo electrónico verificada de Amazon SES. Sin embargo, si configura su grupo de usuarios fuera de la consola, por ejemplo, mediante la API AWS CLI o la API de Amazon Cognito, debe adjuntar la política mediante la consola o la PutIdentityPolicyAPI de Amazon SES.

> **Note**
>
> Solo puede configurar una dirección FROM en un dominio verificado mediante la AWS CLI o la API de Amazon Cognito.

Una política de autorización de envío permite o deniega el acceso en función de los recursos de la cuenta que utilizan Amazon Cognito para invocar Amazon SES. Para obtener más información sobre las políticas basadas en recursos, consulte la Guía del usuario de IAM. También puede ver ejemplos de políticas basadas en recursos en la Guía para desarrolladores de Amazon SES.

Example Política de autorización de envío

En el siguiente ejemplo, la política de envío de autorización otorga a Amazon Cognito la capacidad limitada de utilizar una identidad verificada de Amazon SES. Amazon Cognito solo puede enviar mensajes de correo electrónico cuando lo hace en nombre del grupo de usuarios en la condición `aws:SourceArn` y la cuenta en la condición `aws:SourceAccount`.

Regions with Amazon SES

Su política de autorización de envío en la región del grupo de usuarios o en una región alternativa debe permitir al director del servicio de Amazon Cognito enviar mensajes de correo electrónico. Consulte la tabla de regiones para obtener más información. Si la región de su grupo de usuarios coincide con al menos un valor de la región de Amazon SES, configure su política de autorización de envío con el principal de servicio global en el siguiente ejemplo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "stmnt1234567891234",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "email.cognito-idp.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "<your SES identity ARN>",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<your account number>"
                },
                "ArnLike": {
                    "aws:SourceArn": "<your user pool ARN>"
```

```
                }
            }
        }
    ]
}
```

Opt-in Regions without Amazon SES

Amazon SES no está disponible en todas las suscripciones en las Regiones de AWS que Amazon Cognito sí lo está. Oriente Medio (Emiratos Árabes Unidos) es un ejemplo y solo puede enviar correos electrónicos con identidades verificadas en Europa (Fráncfort) (`eu-central-1`). En los grupos de usuarios con la configuración de correo electrónico predeterminada, Amazon Cognito también envía mensajes de correo electrónico con una identidad verificada en cada una de las dos regiones. En el caso de Oriente Medio (Emiratos Árabes Unidos), la región adicional es Europa (Londres). Debes actualizar la política de autorización de envío en ambas regiones.

La política de autorización de envío en cada una de las regiones alternativas debe permitir al director de servicio de Amazon Cognito de la región de suscripción del grupo de usuarios enviar mensajes de correo electrónico. Consulte la tabla de regiones para obtener más información. Si su región está marcada como región alternativa, configure las políticas de autorización de envío con el director del servicio regional, como se muestra en el siguiente ejemplo. Sustituya el identificador de región de ejemplo *`me-central-1`* por el ID de región requerido, según sea necesario.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "cognito-idp.me-central-1.amazonaws.com"
                ]
            },
            "Action": [
                "SES:SendEmail",
                "SES:SendRawEmail"
            ],
            "Resource": "<your SES identity ARN>",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:SourceAccount": "<your account number>"
                },
                "ArnLike": {
                    "aws:SourceArn": "<your user pool ARN>"
                }
            }
        }
    ]
}
```

Para obtener más información sobre la sintaxis de la política, consulte Políticas autorización de envío con Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service.

Para ver más ejemplos, consulte Ejemplos de la política de autorización de envío con Amazon SES en la Guía para desarrolladores de Amazon Simple Email Service.

A fin de conceder permisos para usar la configuración de Amazon SES, siga estos pasos:

Si configura su grupo de usuarios para utilizar su configuración de Amazon SES, Amazon Cognito necesita otros permisos para llamar a Amazon SES en su nombre cuando envía un correo electrónico a sus usuarios. Esta autorización se concede con el servicio de IAM.

Al configurar su grupo de usuarios con esta opción, Amazon Cognito crea un rol vinculado al servicio, que es un tipo de rol de IAM, en su Cuenta de AWS. En este rol se incluyen los permisos para que Amazon Cognito acceda a Amazon SES y envíe correos electrónicos con su dirección.

Amazon Cognito crea su función vinculada al servicio con AWS las credenciales de la sesión de usuario que establece la configuración. Los permisos de IAM de esta sesión deben incluir la acción `iam:CreateServiceLinkedRole`. Para obtener más información sobre los permisos en IAM, consulte la administración del acceso a AWS los recursos en la Guía del usuario de IAM.

Para obtener más información acerca del rol vinculado al servicio que crea Amazon Cognito, consulte Uso de roles vinculados a servicios para Amazon Cognito.

Paso 4: Configurar el nodo grupo de usuarios

Realice los siguientes pasos si desea configurar el grupo de usuarios con cualquiera de los siguientes elementos:

- Una dirección de remitente personalizada que aparece como el remitente del correo electrónico

- Una dirección de destinatario personalizada que recibe los mensajes que sus usuarios envían a su dirección de remitente.
- Su configuración de Amazon SES

> **Note**
>
> Si su identidad verificada es una dirección de correo electrónico, Amazon Cognito establece esa dirección de correo electrónico como la dirección de correo electrónico DE y RESPUESTA forma predeterminada. Sin embargo, si su identidad verificada es un dominio, debe proporcionar un valor para las direcciones de correo electrónico DE y RESPUESTA. Por ejemplo, si su dominio verificado es example.com, puede establecer no-reply@example.com como dirección de correo electrónico DE y RESPUESTA.

Omita este procedimiento si desea utilizar la configuración y la dirección de correo electrónico predeterminadas de Amazon Cognito.

Para configurar el grupo de usuarios de modo que use una dirección de email personalizada

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus credenciales. AWS
2. Elija User Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija el iconoMensajeríatab, busqueConfiguración de correo electrónico, eligeEditar.
5. En la páginaEditar la configuración de correopágina, seleccioneEnviar correo electrónico desde Amazon SESoEnviar correo electrónico con Amazon Cognito. Puede personalizar laRegión SES,Conjunto de configuración, yFROM remitente namesolo cuando elijasEnviar correo electrónico desde Amazon SES.
6. Para utilizar una dirección FROM personalizada, siga los pasos que se describen a continuación:

    a. En SES region (Región de SES), elija la Región que contiene la dirección de email verificada.

    b. En FROM email address (Dirección de correo electrónico del remitente) elija su dirección de correo electrónico. Use una dirección de correo electrónico verificada con Amazon SES.

    c. (Opcional) En Configuration set (Conjunto de configuración), elija un conjunto de configuración para utilizarlo con Amazon SES. Si realiza y guarda este cambio, se crea un rol vinculado al servicio.

d. (Opcional) EnFROM remitente address, introduzca una dirección de correo electrónico. Puede proporcionar solo la dirección de email o la dirección de email junto con un nombre en el formato `Jane Doe <janedoe@example.com>`.

e. (Opcional) En REPLY-TO email address (RESPONER-A dirección de email), ingrese la dirección de email en la que desea recibir los mensajes que sus usuarios envían a la dirección de remitente.

7. Elija Guardar cambios.

Temas relacionados

- Personalización de los mensajes de verificación de correo electrónico
- Personalización de los mensajes de invitación a usuarios

Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito

Algunos eventos de Amazon Cognito para su grupo de usuarios pueden hacer que Amazon Cognito envíe mensajes de texto SMS a sus usuarios. Por ejemplo, si configura su grupo de usuarios para que requiera la verificación por teléfono, Amazon Cognito envía un mensaje de texto SMS cuando un usuario se registra con una cuenta nueva en su aplicación o cuando restablece su contraseña. En función de la acción que inicie el mensaje de texto SMS, en este se incluirá un código de verificación, una contraseña temporal o un mensaje de bienvenida.

En Amazon Cognito, se utiliza Amazon Simple Notification Service (Amazon SNS) para el envío de mensajes de texto SMS. Si es la primera vez que envía un mensaje de texto a través de Amazon Cognito o Amazon SNS, Amazon SNS lo colocará en un entorno aislado. En el entorno aislado, puede probar los mensajes de texto SMS de sus aplicaciones. En el entorno de pruebas, los mensajes solo se pueden enviar a números de teléfono verificados.

Amazon SNS cobra por los mensajes de texto SMS. Para obtener más información, consulte Precios de Amazon SNS.

> **Note**
>
> Debido al volumen de tráfico de SMS no solicitado en todo el mundo, algunos gobiernos imponen barreras entre los remitentes y los destinatarios de los mensajes SMS. Cuando

utilice mensajes SMS de la autenticación multifactor y las actualizaciones de usuario, debe tomar medidas adicionales para garantizar que los mensajes se entreguen. También debe supervisar las normas relacionadas con los mensajes SMS en los países en los que puedan vivir los usuarios y mantener actualizada la configuración de los mensajes SMS. Para obtener más información, consulte Mensajería de texto móvil (SMS) en la Guía para desarrolladores de Amazon Simple Notification Service.
El uso de mensajes SMS para autenticar y verificar a los usuarios no es una práctica recomendada de seguridad. Los números de teléfono pueden cambiar de propietario y es posible que no representen de manera fiable algo que tenga del factor de MFA para los usuarios. En su lugar, implemente TOTP MFA en tu aplicación o con el IdP de terceros. Además, puede crear factores adicionales de autenticación personalizados con Desencadenadores de Lambda de desafío de autenticación personalizado.

Amazon Cognito envía mensajes SMS a los usuarios con un código que pueden ingresar. En la siguiente tabla se muestran los eventos que pueden generar un mensaje SMS.

Opciones de mensajes

Actividad	Operación de la API	Opciones de entrega	Opciones de formato	Personalizable	Plantilla de mensaje
Contraseña olvidada	ForgotPassword	Correo electrónico, SMS	Código	No	N/A
Invitación	AdminCreateUser	Correo electrónico, SMS	Código	Sí	Mensaje de invitación
Autorregistro	SignUp	Correo electrónico, SMS	código, enlace	Sí	Mensaje de verificación
Verificación de dirección de correo electrónico	UpdateUserAttributes	Correo electrónico, SMS	Código	Sí	Mensaje de verificación

Actividad	Operación de la API	Opciones de entrega	Opciones de formato	Personalizable	Plantilla de mensaje
o número de teléfono					
Autenticación multifactor (MFA)	AdminInitiateAuth, InitiateAuth	SMS, aplicación de autenticación	Código	Sí[1]	Mensaje de MFA

[1] Para mensajes SMS.

Configuración de mensajes SMS por primera vez en grupos de usuarios de Amazon Cognito

En Amazon Cognito, se utiliza Amazon SNS para enviar mensajes SMS a los grupos de usuarios. También puede utilizar un Desencadenador de Lambda para remitentes personalizados de SMS para usar sus propios recursos para enviar mensajes SMS. La primera vez que configura Amazon SNS para enviar mensajes de texto SMS en una determinada región Región de AWS, Amazon SNS lo coloca Cuenta de AWS en el entorno limitado de SMS de esa región. Amazon SNS utiliza el entorno limitado para evitar el fraude y el abuso y para cumplir los requisitos de conformidad. Cuando estás en Cuenta de AWS un entorno de pruebas, Amazon SNS impone algunas restricciones. Por ejemplo, puede enviar mensajes de texto a un máximo de 10 números de teléfono que haya verificado con Amazon SNS. Mientras Cuenta de AWS permanezca en el entorno limitado, no utilice la configuración de Amazon SNS para aplicaciones que estén en producción. Cuando se encuentra en el entorno de pruebas, Amazon Cognito no puede enviar mensajes a los números de teléfono de sus usuarios.

Para enviar mensajes de texto SMS a los usuarios del grupo de usuarios

1. Prepare un rol de IAM que Amazon Cognito pueda usar para enviar mensajes SMS con Amazon SNS
2. Elija el Región de AWS para los mensajes SMS de Amazon SNS
3. Obtener una identidad de origen para enviar mensajes SMS a números de teléfono de EE. UU.
4. Confirmar que se encuentra en el entorno de pruebas de SMS
5. Quitar la cuenta del entorno de pruebas de Amazon SNS

6. Verificar los números de teléfono de Amazon Cognito en Amazon SNS
7. Completar la configuración del grupo de usuarios en Amazon Cognito

Prepare un rol de IAM que Amazon Cognito pueda usar para enviar mensajes SMS con Amazon SNS

Cuando envía un mensaje SMS desde su grupo de usuarios, Amazon Cognito asume un rol de IAM en su cuenta. En Amazon Cognito, se utiliza el permiso `sns:Publish` asignado a ese rol para enviar mensajes SMS a los usuarios. En la consola de Amazon Cognito, puede configurar una IAM role selection (Selección de roles de IAM) desde la pestaña Messaging (Mensajes) de su grupo de usuarios, en SMS o haga esta selección con el asistente de creación de grupos de usuarios.

En el ejemplo siguiente de política de confianza de rol de IAM se concede a grupos de usuarios de Amazon Cognito una capacidad limitada para que adopte un rol de IAM. Amazon Cognito solo puede adoptar el rol cuando lo hace en nombre de los grupos de usuarios en la condición `aws:SourceArn` y en la Cuenta de AWS en la condición `aws:SourceAccount`.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cognito-idp.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "<your account number>"
            },
            "ArnLike": {
                "aws:SourceArn": "<your user pool ARN>"
            }
        }
    }]
}
```

Puede especificar un ARN del grupo de usuarios o un ARN comodín en el valor de la condición `aws:SourceArn`. Busque los ARN de sus grupos de usuarios en el AWS Management Console quirófano mediante una DescribeUserPoolsolicitud de API.

Para obtener más información sobre los roles de IAM y las políticas de confianza, consulte Términos y conceptos de roles en la Guía del usuario de AWS Identity and Access Management .

Elija el Región de AWS para los mensajes SMS de Amazon SNS

En algunas Regiones de AWS, puede elegir la región que contiene los recursos de Amazon SNS que quiere usar para los mensajes SMS de Amazon Cognito. En cualquier Región de AWS lugar donde Amazon Cognito esté disponible, excepto en Asia Pacífico (Seúl), puede utilizar los recursos de Amazon SNS en Región de AWS el lugar donde creó su grupo de usuarios. Para que la mensajería SMS sea más rápida y fiable cuando pueda elegir entre regiones, utilice los recursos de Amazon SNS en la misma región que el grupo de usuarios.

> **Note**
>
> En AWS Management Console, solo puede cambiar la región de los recursos de SMS después de cambiarse a la nueva experiencia de consola de Amazon Cognito.

Elija una región para los recursos SMS en el paso Configurar entrega de mensajes del nuevo asistente del grupo de usuarios. También puede elegir Edit (Editar) bajo SMS en la pestaña Messaging (Mensajería) de un grupo de usuarios existente.

En el momento del lanzamiento Regiones de AWS, Amazon Cognito envió mensajes SMS con recursos de Amazon SNS en una región alternativa. Para establecer su región preferida, utilice el `SnsRegion` parámetro del SmsConfigurationTypeobjeto para su grupo de usuarios. Si crea un recurso de grupos de usuarios de Amazon Cognito mediante programación en una región de Amazon Cognito de la siguiente tabla y no proporciona un parámetro `SnsRegion`, el grupo de usuarios puede enviar mensajes SMS con recursos de Amazon SNS en una región de Amazon SNS heredada.

Los grupos de usuarios de Amazon Cognito en Asia Pacífico (Seúl) Región de AWS deben usar su configuración de Amazon SNS en la región Asia Pacífico (Tokio).

Amazon SNS establece la cuota de gasto para todas las cuentas nuevas en 1,00 USD al mes. Es posible que haya aumentado el límite de gasto en uno Región de AWS que utiliza con Amazon Cognito. Antes de cambiar Región de AWS los mensajes SMS de Amazon SNS, abre un caso de aumento de cuota en el AWS Support Center para aumentar tu límite en la nueva región. Para obtener más información, consulte Requesting increases to your monthly SMS spending quota for

Amazon SNS (Solicitud de aumento de la cuota de gasto mensual de SMS para Amazon SNS) en Amazon Simple Notification Service Developer Guide (Guía para desarrolladores de Amazon Simple Notification Service).

Puede enviar mensajes SMS a cualquier región de Amazon Cognito de la siguiente tabla con los recursos de Amazon SNS en la correspondiente región de Amazon SNS.

Región de Amazon Cognito	Región de Amazon SNS
Este de EE. UU. (Ohio)	EE. UU. Este (Ohio), EE. UU. Este (Norte de Virginia)
Canadá (centro)	Canadá (centro), este de EE. UU. (Norte de Virginia)
Europa (Fráncfort)	Europa (Fráncfort), Europa (Irlanda)
Europa (Londres)	Europa (Londres), Europa (Irlanda)
Asia-Pacífico (Seúl)	Asia-Pacífico (Tokio)
Este de EE. UU. (Norte de Virginia)	Este de EE. UU. (Norte de Virginia)
Oeste de EE. UU. (Norte de California)	Oeste de EE. UU. (Norte de California)
Oeste de EE. UU. (Oregón)	Oeste de EE. UU. (Oregón)
Asia-Pacífico (Bombay)	Asia Pacífico (Mumbai), Asia Pacífico (Singapur)
Asia-Pacífico (Hyderabad)	Asia-Pacífico (Hyderabad)
Asia-Pacífico (Singapur)	Asia-Pacífico (Singapur)
Asia-Pacífico (Sídney)	Asia-Pacífico (Sídney)
Asia-Pacífico (Tokio)	Asia-Pacífico (Tokio)
Asia-Pacífico (Yakarta)	Asia-Pacífico (Yakarta)
Asia-Pacífico (Osaka)	Asia-Pacífico (Osaka)

Región de Amazon Cognito	Región de Amazon SNS
Asia-Pacífico (Melbourne)	Asia-Pacífico (Melbourne)
Europa (Irlanda)	Europa (Irlanda)
Europa (París)	Europa (París)
Europa (Estocolmo)	Europa (Estocolmo)
Europa (Milán)	Europa (Milán)
Europa (España)	Europa (España)
Medio Oriente (Baréin)	Medio Oriente (Baréin)
América del Sur (São Paulo)	América del Sur (São Paulo)
Israel (Tel Aviv)	Israel (Tel Aviv)
África (Ciudad del Cabo)	África (Ciudad del Cabo)
Medio Oriente (EAU)	Medio Oriente (EAU)
Europa (Zúrich)	Europa (Zúrich)

Obtener una identidad de origen para enviar mensajes SMS a números de teléfono de EE. UU.

Si tiene previsto enviar mensajes de texto SMS a números de teléfono de EE. UU., debe obtener una identidad de origen, independientemente de si crea un entorno de pruebas aislado de SMS o un entorno de producción.

A partir del 1 de junio de 2021, los operadores estadounidenses exigen una identidad de origen para enviar mensajes a números de teléfono de EE. UU. Si no dispone de una identidad de origen, debe obtener una. Para saber cómo obtener una identidad de origen, consulte Solicitud de un número en la Guía del usuario de Amazon Pinpoint.

Si opera de la siguiente manera Regiones de AWS, debe abrir un AWS Support ticket para obtener una identidad de origen. Para obtener instrucciones, consulte Solicitud de soporte para mensajería SMS en la Guía para desarrolladores de Amazon Simple Notification Service.

- Este de EE. UU. (Ohio)
- Europa (Estocolmo)
- Europa (París)
- Europa (Milán)
- Middle East (Bahrain)
- América del Sur (São Paulo)
- Oeste de EE. UU. (Norte de California)

Si tiene más de una identidad de origen en la misma Región de AWS, Amazon SNS elige un tipo de identidad de origen en el siguiente orden de prioridad: código corto, 10 DLC, número gratuito. No puede cambiar este valor. Para obtener más información, consulte las preguntas frecuentes de Amazon SNS.

Confirmar que se encuentra en el entorno de pruebas de SMS

Utilice el procedimiento siguiente para confirmar que está en el entorno aislado de SMS. Repita el procedimiento para cada uno de los Región de AWS lugares en los que tenga grupos de usuarios de Amazon Cognito de producción.

Revisión del estado del entorno aislado de SMS en la consola de Amazon Cognito

Confirmar que se encuentra en el entorno de pruebas de SMS

1. Vaya a la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS .
2. ElegirUser Pools (Grupos de usuarios).
3. Elija en la lista un usuario existente.
4. Elija el iconoMensajeríapestaña.
5. En el navegadorConfiguración de SMSsección, expandirMover al entorno de producción de Amazon SNS. Si su cuenta se encuentra en el entorno de pruebas de SMS, verá el siguiente mensaje en Amazon Cognito.

   ```
   You are currently in the SMS Sandbox and cannot send SMS messages to
   unverified numbers.
   ```

Si no ve este mensaje, signfica que alguien ya ha realizado los pasos necesarios para configurar los mensajes SMS en su cuenta. Vaya a Completar la configuración del grupo de usuarios en Amazon Cognito.

6. Elija el enlace de Amazon SNS en el mensaje. Esto abre la consola de Amazon SNS en una pestaña nueva.
7. Compruebe que se encuentre en el entorno de pruebas. El mensaje de la consola indica el estado de la zona de pruebas y Región de AWS, de la siguiente manera:

   ```
   This account is in the SMS sandbox in US East (N. Virginia).
   ```

Quitar la cuenta del entorno de pruebas de Amazon SNS

Si está probando la aplicación y solo necesita enviar mensajes SMS a números de teléfono que los administradores puedan verificar, omita este paso.

Para utilizar su aplicación en producción, quite la cuenta del entorno aislado de SMS y entre en producción. Tras configurar una identidad de origen Región de AWS que contenga los recursos de Amazon SNS que desea que utilice Amazon Cognito, podrá verificar los números de teléfono de EE. UU. mientras permanece en el entorno limitado de Cuenta de AWS SMS. Cuando su entorno de Amazon SNS esté en producción, no tendrá que verificar los números de teléfono de los usuarios en Amazon SNS para enviar mensajes SMS a sus usuarios.

Para obtener instrucciones detalladas, consulte la sección de salida del entorno aislado de Amazon SNS en la Guía para desarrolladores de Amazon Simple Notification Service.

Verificar los números de teléfono de Amazon Cognito en Amazon SNS

Si ha quitado la cuenta del entorno aislado de SMS, omita este paso.

Cuando esté en el entorno aislado de SMS, podrá enviar mensajes a cualquier número de teléfono que haya verificado con Amazon SNS.

Para verificar un número de teléfono, haga lo siguiente:

1. Añada un Sandbox destination phone number (Número de teléfono de destino de entorno aislado) en la sección de mensajería de texto (SMS) de la consola de Amazon SNS.
2. Reciba un mensaje SMS con un código en el número de teléfono que ha proporcionado.
3. Escriba el Código de verificación del mensaje SMS en la consola de Amazon SNS.

Para obtener instrucciones detalladas, consulte Agregar y verificar números de teléfono en el entorno de pruebas de SMS en la Guía para desarrolladores de Amazon Simple Notification Service.

> **Note**
>
> Amazon SNS limita la cantidad de números de teléfono de destino que puede verificar mientras se encuentra en el entorno aislado de SMS. Consulte el sección sobre el entorno aislado de SMS en la Guía para desarrolladores de Amazon Simple Notification Service.

Completar la configuración del grupo de usuarios en Amazon Cognito

Vuelva a la pestaña del navegador donde estaba creando o editando su grupo de usuarios. Complete el procedimiento . Cuando haya añadido correctamente la configuración de SMS a su grupo de usuarios, Amazon Cognito envía un mensaje de prueba a un número de teléfono interno para comprobar que la configuración funciona. Amazon SNS cobra por cada mensaje SMS de prueba.

Uso de tokens con grupos de usuarios

Autentique usuarios y conceda acceso a los recursos con tokens. Las reclamaciones que aparecen en los tokens son información sobre su usuario. El token de ID contiene reclamaciones sobre la identidad, como el nombre de usuario, apellido y dirección de correo electrónico. El token de acceso contiene reclamaciones como el `scope` que el usuario autenticado puede utilizar para acceder a las API de terceros, las operaciones de la API de autoservicio del usuario de Amazon Cognito y Punto de conexión de UserInfo. Los tokens de ID y acceso como un reclamación `cognito:groups` que contiene la pertenencia del grupo del usuario en el grupo de usuarios. Para obtener más información acerca de los conjuntos de grupos de usuarios, consulte Agregar grupos a un grupo de usuarios.

Amazon Cognito también tiene tokens de actualización que puede utilizar para obtener nuevos tokens o revocar los existentes. Actualice un token para recuperar un ID nuevo y tokens de acceso. Revoque un token para denegar el acceso del usuario admitido por los tokens de actualización.

Amazon Cognito emite tokens como cadenas codificadas en Base64. Puede descodificar cualquier ID de Amazon Cognito o token de acceso de base64 a JSON de texto sin formato. Los tokens de actualización de Amazon Cognito están cifrados, son opacos para los usuarios y administradores de grupos de usuarios y solo los puede leer el grupo de usuarios.

Autenticación con tokens

Cuando un usuario inicia sesión en su aplicación, Amazon Cognito verifica la información de inicio de sesión. Si el inicio de sesión es correcto, Amazon Cognito crea una sesión y devuelve un token de ID, un token de acceso y un token de actualización para el usuario autenticado. Puede utilizar los tokens para conceder a los usuarios acceso a las API y los recursos descendentes, como Amazon API Gateway. O bien, puede intercambiarlos por credenciales temporales de AWS para acceder a otros Servicios de AWS.

Almacenamiento de tokens

La aplicación debe poder almacenar tokens de distintos tamaños. El tamaño del token puede cambiar por diferentes motivos, entre los que se incluyen notificaciones adicionales, cambios en los algoritmos de codificación y cambios en los algoritmos de cifrado. Cuando habilita la revocación de tokens en el grupo de usuarios, Amazon Cognito agrega reclamaciones adicionales a los JSON Web Tokens, lo que aumenta su tamaño. Las nuevas notificaciones `origin_jti` y `jti` se agregan a los tokens de acceso e ID. Para obtener más información acerca de la revocación de tokens, consulte Revocación de tokens.

> **Important**
>
> Como práctica recomendada, asegure todos los tokens en tránsito y el almacenamiento en el contexto de la aplicación. Los tokens pueden contener información de identificación personal acerca de los usuarios e información sobre el modelo de seguridad que utiliza para el grupo de usuarios.

Personalización de tokens

Puede personalizar los tokens de acceso e ID que Amazon Cognito transfiere a la aplicación. En Desencadenador de Lambda anterior a la generación del token, puede agregar, modificar y suprimir las reclamaciones de tokens. El desencadenador previo a la generación del token es una función de Lambda a la que Amazon Cognito envía un conjunto predeterminado de reclamaciones. Entre las reclamaciones se incluyen los ámbitos de OAuth 2.0, la pertenencia a grupos de usuarios, los atributos de los usuarios, etc. Luego, la función puede aprovechar la oportunidad para realizar

cambios en tiempo de ejecución y devolver las reclamaciones de token actualizadas a Amazon Cognito.

El acceso a la personalización del token con los eventos de la versión 2 conlleva costos adicionales. Para obtener más información, consulte Precios de Amazon Cognito.

Temas

Uso del token de ID

El token de ID es un JSON Web Token (JWT) que contiene notificaciones acerca de la identidad del usuario autenticado, como por ejemplo, `name`, `email` y `phone_number`. Puede utilizar esta información de identidad dentro de la aplicación. El token de ID también puede utilizarse para autenticar a los usuarios en sus servidores de recursos o aplicaciones de servidor. También puede usar un token de ID fuera de la aplicación con sus operaciones de API web. En esos casos, debe verificar la firma del token de ID antes de poder confiar en las notificaciones que contiene. Consulte Verificación de un JSON Web Token.

Puede usar cualquier valor de entre 5 minutos y 1 día para establecer el vencimiento del token de ID. Puede configurar este valor por cliente de aplicación.

> **Important**
>
> Cuando el usuario inicia sesión con la IU alojada o con un proveedor de identidad federado (IdP), Amazon Cognito establece cookies de sesión válidas durante 1 hora. Si utiliza la IU alojada o la federación y especifica una duración mínima de menos de 1 hora para sus tokens de acceso e ID, los usuarios seguirán teniendo una sesión válida hasta que caduque la cookie. Si el usuario tiene tokens que caducan durante la sesión de una hora, podrá actualizar sus tokens sin necesidad de volver a autenticarse.

Encabezado del token de ID

El encabezado contiene dos bloques de información: el ID de clave (`kid`) y el algoritmo (`alg`).

```
{
"kid" : "1234example=",
"alg" : "RS256"
}
```

`kid`

ID de la clave. Este valor indica la clave que se ha utilizado para proteger la firma web JSON (JWS) del token. Puede ver los ID de las claves de firma de su grupo de usuarios en el punto de conexión de `jwks_uri`.

Para obtener más información sobre el parámetro`kid`, consulte Parámetro de encabezado de identificador de clave (kid).

`alg`

El algoritmo criptográfico que Amazon Cognito utilizó para proteger el token de acceso. Los grupos de usuarios usan un algoritmo criptográfico RS256, que es una firma RSA con SHA-256.

Para obtener más información sobre el `alg`parámetro, consulte Parámetro de encabezado de algoritmos (alg).

Carga útil predeterminada del token de ID

Este es un ejemplo de carga útil de un token de ID. Contiene notificaciones sobre el usuario autenticado. Para obtener más información sobre las afirmaciones estándar de OpenID Connect (OIDC), consulte la lista de afirmaciones estándar de OIDC. Puede añadir afirmaciones de su propio diseño con un. Desencadenador de Lambda anterior a la generación del token

```
<header>.{
    "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "cognito:groups": [
        "test-group-a",
        "test-group-b",
        "test-group-c"
    ],
```

```
    "email_verified": true,
    "cognito:preferred_role": "arn:aws:iam::111122223333:role/my-test-role",
    "iss": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
    "cognito:username": "my-test-user",
    "middle_name": "Jane",
    "nonce": "abcdefg",
    "origin_jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "cognito:roles": [
        "arn:aws:iam::111122223333:role/my-test-role"
    ],
    "aud": "xxxxxxxxxxxxexample",
    "identities": [
        {
            "userId": "amzn1.account.EXAMPLE",
            "providerName": "LoginWithAmazon",
            "providerType": "LoginWithAmazon",
            "issuer": null,
            "primary": "true",
            "dateCreated": "1642699117273"
        }
    ],
    "event_id": "64f513be-32db-42b0-b78e-b02127b4f463",
    "token_use": "id",
    "auth_time": 1676312777,
    "exp": 1676316377,
    "iat": 1676312777,
    "jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "email": "my-test-user@example.com"
}
.<token signature>
```

sub

Un identificador único (UUID), o asunto, para el usuario autenticado. Es posible que el nombre de usuario no sea único en el grupo de usuarios. La reclamación `sub` es la mejor forma de identificar a un usuario determinado.

cognito:groups

Una matriz con los nombres de los grupos de usuarios que tienen a su usuario como miembro. Los grupos pueden ser un identificador que se presenta en la aplicación o pueden generar una solicitud para un rol de IAM preferido desde un grupo de identidades.

`cognito:preferred_role`

El ARN del rol de IAM que asoció al grupo de usuarios de mayor prioridad de su usuario. Para obtener más información sobre cómo el grupo de usuarios selecciona esta reclamación de rol, consulte Asignación de valores de prioridad a los grupos.

`iss`

El proveedor de identidad que emitió el token. La reclamación tiene el formato siguiente:

`https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>`

`cognito:username`

El nombre de usuario del usuario en el grupo de usuarios.

`nonce`

La notificación `nonce` proviene de un parámetro del mismo nombre que puede agregar a las solicitudes al punto de conexión `authorize` de OAuth 2.0. Cuando agrega el parámetro, la notificación `nonce` se incluye en el token de ID que emite Amazon Cognito y puede utilizarla para protegerse de los ataques de repetición. Si no proporciona un valor `nonce` en la solicitud, Amazon Cognito genera y valida de forma automática un nonce cuando se autentica a través de un proveedor de identidad de terceros y, a continuación, lo agrega como la notificación `nonce` al token de ID. La implementación de la notificación `nonce` en Amazon Cognito se basa en estándares OIDC.

`origin_jti`

Un identificador de revocación de tokens asociado al token de actualización del usuario. Amazon Cognito hace referencia a la `origin_jti` reclamación cuando comprueba si revocó el token de su usuario con la operación Revocación de puntos de conexión o la RevokeTokenAPI. Al revocar un token, Amazon Cognito invalida todos los tokens de acceso e ID con el mismo valor `origin_jti`.

`cognito:roles`

Una matriz con los nombres de los roles de IAM asociados a los grupos de su usuario. Cada grupo de usuarios puede tener un rol de IAM asociado. Esta matriz representa todos los roles de IAM de los grupos de usuarios, independientemente de su prioridad. Para obtener más información, consulte Agregar grupos a un grupo de usuarios.

`aud`

El cliente de la aplicación del grupo de usuarios que ha autenticado a su usuario. Amazon Cognito representa el mismo valor en la reclamación `client_id` del token de acceso.

`identities`

El contenido del atributo `identities` del usuario. El atributo contiene información sobre cada perfil de proveedor de identidad externo que haya vinculado a un usuario, ya sea mediante un inicio de sesión federado o mediante la vinculación de un usuario federado a un perfil local. Esta información contiene el nombre del proveedor, el identificador único del proveedor y otros metadatos.

`token_use`

El objetivo para el que se creó el token. En un token de identificación, su valor es `id`.

`auth_time`

La hora de autenticación, en formato de hora de Unix, a la que el usuario completó la autenticación.

`exp`

La hora de caducidad, en formato de hora de Unix, en la que vence el token de su usuario.

`iat`

La hora de emisión, en formato de hora de Unix, a la que Amazon Cognito emitió el token de su usuario.

`jti`

El identificador único del JWT.

El token de ID puede contener notificaciones estándar de OIDC definidas en las notificaciones estándar de OIDC. También puede contener atributos personalizados que se definen en el grupo de usuarios. Amazon Cognito escribe valores de atributos personalizados en el token de ID como cadenas, independientemente del tipo de atributo.

> **Note**
>
> Los atributos personalizados del grupo de usuarios siempre llevan el prefijo. `custom:`

Firma del token de ID

La firma del token de ID se calcula en función del encabezado y la carga del token JWT. Antes de aceptar las reclamaciones en cualquier token de ID que reciba su aplicación, verifique la firma del token. Para obtener más información, consulte Verificación de un JSON Web Token. Verificación de un JSON Web Token

Uso del token de acceso

El token de acceso de grupo de usuarios contiene notificaciones acerca del usuario autenticado, una lista de los grupos de usuarios y una lista de ámbitos. La finalidad del token de acceso es autorizar operaciones de la API. Su grupo de usuarios acepta tokens de acceso para autorizar las operaciones de autoservicio de los usuarios. Por ejemplo, puede utilizar el token de acceso para conceder acceso a sus usuarios a fin de agregar, cambiar o eliminar atributos de usuarios.

Con los ámbitos de OAuth 2.0 en un token de acceso, que proceden de los ámbitos personalizados que añade a su grupo de usuarios, puede autorizar a su usuario a recuperar información de una API. Por ejemplo, Amazon API Gateway admite la autorización con los tokens de acceso de Amazon Cognito. Puede rellenar un autorizador de la API de REST con información del grupo de usuarios o utilizar Amazon Cognito como autorizador de JSON Web Token (JWT) para una API de HTTP. Para generar un token de acceso con ámbitos personalizados, debe solicitarlo a través de los puntos de conexión públicos de su grupo de usuarios.

El token de acceso de su usuario es un permiso para solicitar más información sobre los atributos de su usuario a Punto de conexión de UserInfo. El token de acceso de su usuario también es un permiso para leer y escribir atributos de usuario. El nivel de acceso a los atributos que otorga su token de acceso depende de los permisos que asigne al cliente de su aplicación y de los ámbitos que conceda en el token.

El token de acceso es un token web JSON (JWT). El encabezado del token de acceso tiene la misma estructura que el token de ID. Amazon Cognito firma los tokens de acceso con una clave diferente a la clave que firma los tokens de ID. El valor de una reclamación de ID de clave de acceso (`kid`) no coincide con el valor de la reclamación `kid` de un token de ID de la misma sesión de usuario. En el código de su aplicación, verifique los tokens de ID y los tokens de acceso de forma independiente. No confíe en las reclamaciones de un token de acceso hasta que verifique la firma. Para obtener más información, consulte Verificación de un JSON Web Token. Puede utilizar cualquier valor de entre 5 minutos y 1 día para configurar el vencimiento del token de acceso. Puede configurar este valor por cliente de aplicación.

> **Important**
>
> Para los tokens de acceso e ID, no especifique un valor mínimo inferior a una hora si utiliza la IU alojada. Amazon Cognito HostedUI utiliza cookies que son válidas durante una hora. Si ingresa un mínimo inferior a una hora, no obtendrá un tiempo de caducidad inferior.

Encabezado del token de acceso

El encabezado contiene dos bloques de información: el ID de clave (`kid`) y el algoritmo (`alg`).

```
{
"kid" : "1234example="
"alg" : "RS256",
}
```

`kid`

ID de la clave. Este valor indica la clave que se ha utilizado para proteger la firma web JSON (JWS) del token. Puede ver los ID de las claves de firma de su grupo de usuarios en el punto de conexión de `jwks_uri`.

Para obtener más información sobre el parámetro`kid`, consulte Parámetro de encabezado de identificador de clave (kid).

`alg`

El algoritmo criptográfico que Amazon Cognito utilizó para proteger el token de acceso. Los grupos de usuarios usan un algoritmo criptográfico RS256, que es una firma RSA con SHA-256.

Para obtener más información sobre el `alg`parámetro, consulte Parámetro de encabezado de algoritmos (alg).

Carga útil predeterminada del token de acceso

Esta es una carga de muestra de un token de acceso. Para obtener más información, consulte las notificaciones JWT. Puede añadir imágenes de su propio diseño con unDesencadenador de Lambda anterior a la generación del token.

```
<header>.
```

```
{
   "sub":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "device_key": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "cognito:groups":[
      "testgroup"
   ],
   "iss":"https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
   "version":2,
   "client_id":"xxxxxxxxxxxxexample",
   "origin_jti":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "event_id":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "token_use":"access",
   "scope":"phone openid profile resourceserver.1/appclient2 email",
   "auth_time":1676313851,
   "exp":1676317451,
   "iat":1676313851,
   "jti":"aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
   "username":"my-test-user"
}
.<token signature>
```

`sub`

Un identificador único (UUID), o asunto, para el usuario autenticado. Es posible que el nombre de usuario no sea único en el grupo de usuarios. La reclamación `sub` es la mejor forma de identificar a un usuario determinado.

`cognito:groups`

Una matriz con los nombres de los grupos de usuarios que tienen a su usuario como miembro.

`iss`

El proveedor de identidad que emitió el token. La reclamación tiene el formato siguiente:

`https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>`

`client_id`

El cliente de la aplicación del grupo de usuarios que ha autenticado a su usuario. Amazon Cognito representa el mismo valor en la reclamación `aud` del token de ID.

`origin_jti`

Un identificador de revocación de tokens asociado al token de actualización del usuario. Amazon Cognito hace referencia a la `origin_jti` reclamación cuando comprueba si revocó el token

de su usuario con la operación Revocación de puntos de conexión o la RevokeTokenAPI. Al revocar un token, Amazon Cognito invalida todos los tokens de acceso e ID con el mismo valor `origin_jti`.

`token_use`

El objetivo para el que se creó el token. En un token de acceso, su valor es `access`.

`scope`

Una lista de ámbitos de OAuth 2.0 que definen el acceso que proporciona el token. Un token de Punto de conexión de token puede contener cualquier ámbito que admita el cliente de la aplicación. Un token del inicio de sesión de la API de Amazon Cognito solo contiene el ámbito `aws.cognito.signin.user.admin`.

`auth_time`

La hora de autenticación, en formato de hora de Unix, a la que el usuario completó la autenticación.

`exp`

La hora de caducidad, en formato de hora de Unix, en la que vence el token de su usuario.

`iat`

La hora de emisión, en formato de hora de Unix, a la que Amazon Cognito emitió el token de su usuario.

`jti`

El identificador único del JWT.

`username`

El nombre de usuario del usuario en el grupo de usuarios.

Firma del token de acceso

La firma del token de acceso se calcula en función del encabezado y la carga del token JWT. Cuando se utiliza fuera de una aplicación en las API web, se debe verificar siempre esta firma para poder aceptar el token. Para obtener más información, consulte Verificación de un JSON Web Token.

Uso del token de actualización

Puede utilizar el token de actualización para recuperar tokens de ID y de acceso nuevos. De forma predeterminada, el token de actualización vence 30 días después de que el usuario de la aplicación inicie sesión en el grupo de usuarios. Al crear una aplicación para el grupo de usuarios, puede utilizar cualquier valor comprendido entre 60 minutos y 10 años a fin de configurar el vencimiento del token de actualización de la aplicación.

SDK para móviles para iOS, SDK para móviles para Android, Amplify para iOS, Android y Flutter actualizan de forma automática los tokens de ID y acceso si hay un token de actualización válido (sin vencer). Los tokens de ID y de acceso tienen una validez mínima restante de 2 minutos. Si el token de actualización se ha vencido, el usuario de la aplicación debe volver a autenticarse al iniciar sesión de nuevo en el grupo de usuarios. Si el valor mínimo para el token de acceso y el token de ID se establece en 5 minutos y está utilizando el SDK, el token de actualización se utilizará de forma continua para recuperar los nuevos tokens de ID y de acceso. Observará el funcionamiento esperado con un mínimo de 7 minutos, en lugar de 5 minutos.

La cuenta de usuario en sí no vence, siempre y cuando el usuario haya iniciado sesión, como mínimo, una vez antes del plazo `UnusedAccountValidityDays` indicado para las cuentas nuevas.

Obtener nuevos tokens de acceso e identidad con un token de actualización

Utilice la API o la IU alojada para iniciar la autenticación de los tokens de actualización.

Para usar el token de actualización para obtener un nuevo ID y acceder a los tokens con la API de grupos de usuarios, utilice las operaciones de la InitiateAuthAPI AdminInitiateAutho. Pasar `REFRESH_TOKEN_AUTH` para el parámetro `AuthFlow`. En la propiedad `AuthParameters` de `AuthFlow`, pase el token de actualización del usuario como el valor de `"REFRESH_TOKEN"`. Amazon Cognito devuelve nuevos tokens de ID y acceso después de que la solicitud de API supera todos los desafíos.

> **Note**
>
> Para utilizar la API de grupos de usuarios de Amazon Cognito para actualizar los tokens de un usuario de interfaz de usuario alojado, genere una solicitud de `InitiateAuth`.

También puede enviar los tokens de actualización a Punto de conexión de token en un grupo de usuarios en el que haya configurado un dominio. En el cuerpo de la solicitud, incluya un

valor `grant_type` de `refresh_token` y un valor `refresh_token` del token de actualización del usuario.

Revocación de los tokens de actualización

Puede revocar los tokens de actualización que pertenecen a un usuario. Para obtener más información acerca de la revocación de tokens, consulte Revocación de tokens.

> **Note**
>
> Al revocar el token de actualización, se revocarán todos los ID y tokens de acceso que Amazon Cognito emitió a partir de las solicitudes de actualización con ese token.

Los usuarios pueden cerrar sesión en todos los dispositivos en los que tengan la sesión iniciada en ese momento si se revocan todos los tokens de usuario mediante las operaciones de la API `GlobalSignOut` y `AdminUserGlobalSignOut`. Cuando el usuario cierra sesión, se producen los siguientes efectos.

- El token de actualización del usuario no puede obtener nuevos tokens para el usuario.
- El token de acceso del usuario no puede realizar solicitudes de la API autorizadas por un token.
- El usuario deberá volver a autenticarse para obtener tokens nuevos. Como las cookies de sesión de la interfaz de usuario alojada no caducan automáticamente, el usuario puede volver a autenticarse con una cookie de sesión, sin necesidad de solicitar credenciales adicionales. Después de cerrar la sesión de los usuarios de la interfaz de usuario alojada, rediríjalos a Punto de conexión Logout, donde Amazon Cognito borrará la cookie de sesión.

Con los tokens de actualización, puede mantener las sesiones de los usuarios en la aplicación durante mucho tiempo. Con el tiempo, es posible que los usuarios deseen desautorizar algunos dispositivos en los que han iniciado sesión y actualizar la sesión continuamente. Para cerrar la sesión del usuario de un único dispositivo, revoque el token de actualización. Cuando tu usuario quiera cerrar sesión en todas las sesiones autenticadas, genera una solicitud de GlobalSignOutAPI. La aplicación puede ofrecer al usuario una opción como Cerrar sesión en todos los dispositivos. `GlobalSignOut` acepta un token de acceso válido inalterado, no caducado y no revocado de un usuario. Como esta API está autorizada por un token, un usuario no puede usarla para iniciar el cierre de sesión de otro usuario.

Sin embargo, puedes generar una solicitud de AdminUserGlobalSignOutAPI que autorices con tus AWS credenciales para cerrar la sesión de cualquier usuario en todos sus dispositivos. La aplicación de administrador debe llamar a esta operación de API con las credenciales de AWS desarrollador y pasar como parámetros el ID del grupo de usuarios y el nombre de usuario del usuario. La API `AdminUserGlobalSignOut` puede cerrar la sesión de cualquier usuario del grupo de usuarios.

Para obtener más información sobre las solicitudes que puede autorizar con AWS credenciales o con un token de acceso de usuario, consulteOperaciones de API autenticadas y no autenticadas de los grupos de usuarios de Amazon Cognito.

Revocación de tokens

Puedes revocar un token de actualización para un usuario mediante la AWS API. Cuando se revoca un token de actualización, todos los tokens de acceso que este token de actualización haya emitido con anterioridad pierden su validez. Los otros tokens de actualización emitidos al usuario no se ven afectados.

> **Note**
>
> Los tokens de JWT son autónomos y cuentan con una firma y una fecha de vencimiento que se asignó cuando se creó el token. Los tokens revocados no se pueden utilizar con ninguna llamada a la API de Amazon Cognito que requiera un token. Sin embargo, los tokens revocados seguirán siendo válidos si se verifican con cualquier biblioteca JWT que verifique la firma y el vencimiento del token.

Puede revocar un token de actualización para un cliente de un grupo de usuarios con la revocación de tokens habilitada. Cuando se crea un nuevo cliente de grupos de usuarios, la revocación de tokens se habilita de forma predeterminada.

Habilitar la revocación de tokens

Antes de poder revocar un token para un cliente actual de grupos de usuarios, debe habilitar la revocación de tokens. Puedes habilitar la revocación del token para los clientes del grupo de usuarios existentes mediante la API AWS CLI o la AWS API. Para ello, llame al comando de CLI `aws cognito-idp describe-user-pool-client` o a la operación de la API `DescribeUserPoolClient` para recuperar la configuración actual del cliente de la aplicación. Luego, llame al comando de CLI `aws cognito-idp update-user-pool-client` o a la

operación de la API `UpdateUserPoolClient`. Incluye la configuración actual del cliente de la aplicación y establece el parámetro `EnableTokenRevocation` en `true`.

Al crear un nuevo cliente de grupo de usuarios mediante la AWS Management Console, la API o la AWS API AWS CLI, la revocación de tokens se habilita de forma predeterminada.

Después de habilitar la revocación de tokens, se agregan nuevas reclamaciones en los JSON Web Tokens de Amazon Cognito. Las notificaciones `origin_jti` y `jti` se agregan a los tokens de acceso e ID. Estas notificaciones aumentan la dimensión de los tokens de acceso e ID del cliente de la aplicación.

Para crear o modificar un cliente de aplicaciones con la revocación de token habilitada, incluye el siguiente parámetro en tu solicitud CreateUserPoolClient o en la de la UpdateUserPoolClientAPI.

```
"EnableTokenRevocation": true
```

Revocación de un token

Puede revocar un token de actualización mediante una solicitud de RevokeTokenAPI, por ejemplo, con el comando `aws cognito-idp revoke-token` CLI. También puede revocar los tokens mediante Revocación de puntos de conexión. Este punto de enlace se encuentra disponible después de agregar un dominio a su grupo de usuarios. Puede utilizar el punto de conexión de revocación en un dominio alojado en Amazon Cognito o en su propio dominio personalizado.

> **Note**
>
> La solicitud para revocar un token de actualización debe incluir el ID del cliente que se utilizó para obtener el token.

A continuación, se muestra el cuerpo de una solicitud de la API de `RevokeToken` de ejemplo.

```
{
   "ClientId": "1example23456789",
   "ClientSecret": "abcdef123456789ghijklexample",
   "Token": "eyJjdHkiOiJKV1QiEXAMPLE"
}
```

A continuación, se muestra un ejemplo de solicitud cURL al punto de conexión `/oauth2/revoke` de un grupo de usuarios con un dominio personalizado.

```
curl --location 'auth.mydomain.com/oauth2/revoke' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic Base64Encode(client_id:client_secret)' \
--data-urlencode 'token=abcdef123456789ghijklexample' \
--data-urlencode 'client_id=1example23456789'
```

La operación `RevokeToken` y el punto de conexión `/oauth2/revoke` no requieren ninguna autorización adicional a menos que el cliente de la aplicación tenga un secreto de cliente.

Verificación de un JSON Web Token

En estos pasos, se describe cómo verificar un JSON Web Token (JWT) de grupo de usuarios.

Temas

- Requisitos previos
- Valide los tokens con aws-jwt-verify
- Descripción e inspección de tokens

Requisitos previos

Es posible que la biblioteca, el SDK o el marco de software ya gestionen las tareas de esta sección. AWS Los SDK proporcionan herramientas para gestionar y gestionar los tokens del grupo de usuarios de Amazon Cognito en su aplicación. AWS Amplify incluye funciones para recuperar y actualizar los tokens de Amazon Cognito.

Para obtener más información, consulte las páginas siguientes.

- Integración de la autenticación y la autorización de Amazon Cognito con aplicaciones web y móviles
- Ejemplos de código para Amazon Cognito Identity Provider mediante SDK AWS
- Flujos de trabajo avanzados en Amplify Dev Center

Hay muchas bibliotecas disponibles para decodificar y verificar un JSON Web Token (JWT). Estas bibliotecas pueden resultarle de ayuda si desea procesar de forma manual los tokens para el procesamiento de la API del lado del servidor o si utiliza otros lenguajes de programación. Consulte la lista de bibliotecas de OpenID Foundation para trabajar con tokens JWT.

Valide los tokens con aws-jwt-verify

En una aplicación de Node.js, AWS recomienda que la aws-jwt-verifybiblioteca valide los parámetros del token que el usuario pasa a la aplicación. Con `aws-jwt-verify`, puede rellenar `CognitoJwtVerifier` con los valores de las reclamaciones que desea verificar para uno o varios grupos de usuarios. Estos son algunos de los valores que puede comprobar:

- Que esos tokens de acceso o ID no tengan un formato incorrecto ni hayan caducado y tengan una firma válida.
- Que esos tokens de acceso procedan de los grupos de usuarios y clientes de aplicaciones correctos.
- Que las reclamaciones del token de acceso contengan los ámbitos de OAuth 2.0 correctos.
- Que las claves que firmaron sus tokens de acceso e ID coincidan con una clave `kid` de firma del URI de JWKS de sus grupos de usuarios.

  El URI de JWKS contiene información pública sobre la clave privada que firmó el token de su usuario. Puede encontrar el URI de JWKS para su grupo de usuarios en `https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/.well-known/jwks.json`.

Para obtener más información y un código de ejemplo que puedes usar en una aplicación de Node.js o en un AWS Lambda autorizador, consulta aws-jwt-verifyen GitHub.

Descripción e inspección de tokens

Antes de integrar la inspección de tokens en su aplicación, analice la forma en que Amazon Cognito ensambla los JWT. Obtenga tokens de ejemplo de su grupo de usuarios. Decodifíquelos y examínelos bien para conocer sus características y determinar qué desea verificar y cuándo. Por ejemplo, es posible que desee examinar la pertenencia a un grupo en un escenario y los ámbitos en otro.

En las siguientes secciones, se describe un proceso para inspeccionar manualmente los JWT de Amazon Cognito mientras prepara la aplicación.

Confirmar la estructura del JWT

Un token web JSON (JWT) incluye tres secciones con un delimitador `.` (punto) entre ellas.

Encabezado

El ID de clave, `kid`, y el algoritmo RS, `alg`, que Amazon Cognito utilizó para firmar el token. Amazon Cognito firma los tokens con un `alg` de `RS256`.

Carga

Reclamaciones de tokens. En un token de ID, las reclamaciones incluyen atributos de usuario e información sobre el grupo de usuarios, `iss`, y el cliente de la aplicación, `aud`. En un token de acceso, la carga incluye los ámbitos, la pertenencia a grupos, el nombre de su grupo de usuarios como `iss` y el de su cliente de aplicación como `client_id`.

Signature

La firma no se puede descodificar en base64 como el encabezado y la carga. Es un identificador RSA256 que proviene de una clave de firma y unos parámetros que puede observar en su URI de JWKS.

El encabezado y la carga son JSON codificados en base64. Puede identificarlos por los caracteres de apertura `eyJ` que se descodifican para formar el carácter inicial `{`. Si su usuario presenta un JWT codificado en base64 a su aplicación y no está en el formato `[JSON Header].[JSON Payload].[Signature]`, eso significa que no es un token de Amazon Cognito válido, por lo que puede descartarlo.

Validación del JWT

La firma JWT es una combinación con hash del encabezado y la carga. Amazon Cognito genera dos pares de claves criptográficas RSA para cada grupo de usuarios. Una clave privada firma los tokens de acceso y la otra firma los tokens de ID.

Para verificar la firma de un token JWT

1. Descodifique el token de ID.

   OpenID Foundation también mantiene una lista de bibliotecas para trabajar con tokens JWT.

   También se puede utilizar AWS Lambda para decodificar los JWT de grupos de usuarios. Para obtener más información, consulte Decodificar y verificar los tokens JWT de Amazon Cognito mediante. AWS Lambda
2. Compare el ID de clave local (`kid`) con el `kid` público.

a. Descargue y almacene la JSON Web Key (JWK) pública correspondiente del grupo de usuarios. Está disponible como parte de un JSON Web Key Set (JWKS). Para localizarla, construya la siguiente URI `jwks_uri` para su entorno:

```
https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/.well-known/jwks.json
```

Para obtener más información sobre JWK y los conjuntos JWK, consulte JSON Web Key (JWK).

> **Note**
>
> Es posible que Amazon Cognito rote las claves de firma en su grupo de usuarios. Como práctica recomendada, almacene en caché las claves públicas en su aplicación utilizando el `kid` como clave de caché y actualice la caché periódicamente. Compare el `kid` de los tokens que recibe su aplicación con su caché.
>
> Si recibe un token con el emisor correcto pero con un `kid` diferente, es posible que Amazon Cognito haya rotado la clave de firma. Actualice la memoria caché desde el punto de conexión `jwks_uri` de su grupo de usuarios.

Este es un archivo `jwks.json` de muestra:

```
{
 "keys": [{
  "kid": "1234example=",
  "alg": "RS256",
  "kty": "RSA",
  "e": "AQAB",
  "n": "1234567890",
  "use": "sig"
 }, {
  "kid": "5678example=",
  "alg": "RS256",
  "kty": "RSA",
  "e": "AQAB",
  "n": "987654321",
  "use": "sig"
 }]
```

```
}
```

ID de clave (**`kid`**)

El parámetro `kid` es una sugerencia que indica la clave que se ha utilizado para proteger la firma web JSON (JWS) del token.

Algoritmo (**`alg`**)

El parámetro de encabezado `alg` representa el algoritmo criptográfico que se utiliza para proteger el token de ID. Los grupos de usuarios usan un algoritmo criptográfico RS256, que es una firma RSA con SHA-256. Para obtener más información sobre RSA, consulte Criptografía de RSA.

Tipo de clave (**`kty`**)

El parámetro `kty` identifica la familia de algoritmos criptográficos que se utilizan con la clave, como “RSA” en este ejemplo.

Exponente RSA (**`e`**)

El parámetro `e` contiene el valor del exponente de la clave pública RSA. Se representa como un valor codificado en Base64urlUInt.

Módulo RSA (**`n`**)

El parámetro `n` contiene el valor del módulo de la clave pública RSA. Se representa como un valor codificado en Base64urlUInt.

Uso (**`use`**)

El parámetro `use` describe el uso previsto de la clave pública. En este ejemplo, el `use` valor `sig` representa la firma.

b. Busque la clave JSON web pública para un `kid` que coincida con el `kid` del JWT.

3. Utilice una biblioteca JWT para comparar la firma del emisor con la firma en el token. La firma del emisor se deriva de la clave pública (el módulo RSA`"n"`) del`kid` in jwks.json que coincide con el token`kid`. Es posible que tenga que convertir primero la JWK al formato PEM. En este ejemplo, se emplea el JWT y la JWK, y se utiliza la biblioteca de Node.js, jsonwebtoken, para verificar la firma JWT:

Node.js

```
var jwt = require('jsonwebtoken');
var jwkToPem = require('jwk-to-pem');
var pem = jwkToPem(jwk);
jwt.verify(token, pem, { algorithms: ['RS256'] }, function(err, decodedToken) {
});
```

Comprobar las notificaciones

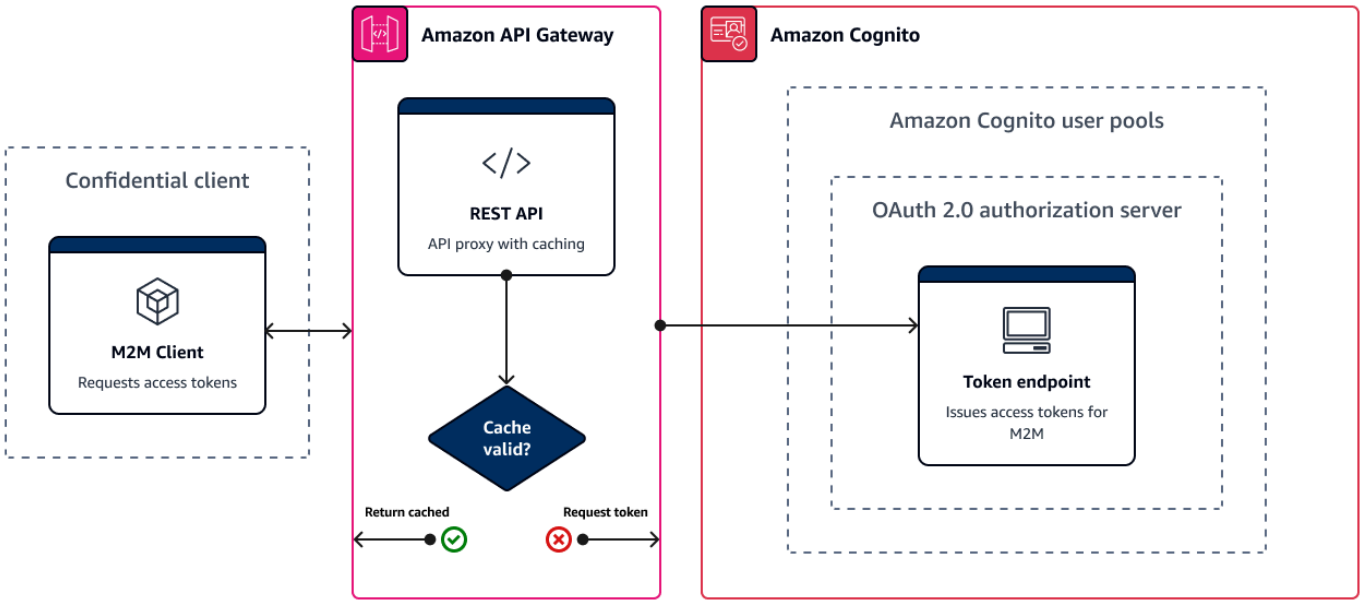
Para comprobar las notificaciones JWT

1. Mediante uno de los siguientes métodos, compruebe que el token no haya caducado.
   a. Descodifique el token y compare la reclamación `exp` con la hora actual.
   b. Si su token de acceso incluye una `aws.cognito.signin.user.admin` reclamación, envíe una solicitud a una API como. GetUser Las solicitudes de API que autorice con un token de acceso devuelven un error si el token ha caducado.
   c. Presente el token de acceso en una solicitud a Punto de conexión de UserInfo. La solicitud devuelve un error si el token ha caducado.
2. La afirmación `aud` en un token de ID y la afirmación `client_id` de un token de acceso deberían coincidir con el ID de cliente de la aplicación creado en el grupo de usuarios de Amazon Cognito.
3. La notificación de emisor (`iss`) debería coincidir con el grupo de usuarios. Por ejemplo, un grupo de usuarios creado en la región `us-east-1` tendrá el siguiente valor `iss`:

   `https://cognito-idp.us-east-1.amazonaws.com/<userpoolID>`.
4. Compruebe la notificación `token_use`.
   - Si solo acepta el token de acceso en las operaciones de la API web, su valor debe ser `access`.
   - Si solo usa el token de ID, su valor debe ser `id`.
   - Si utiliza tokens de ID y de acceso, la notificación `token_use` debe ser `id` o `access`.

Ahora puede confiar en las notificaciones del token.

Almacenamiento en caché de tokens



La aplicación debe completar correctamente una de las siguientes solicitudes cada vez que desee obtener un nuevo JSON Web Token (JWT).

- Solicite las credenciales o la concesión del código de autorización desde el Punto de conexión de token.
- Solicite una concesión implícita desde la IU alojada.
- Autentique a un usuario local en una solicitud de API de Amazon Cognito como. InitiateAuth

Puede configurar el grupo de usuarios para que los tokens caduquen en minutos, horas o días. Para garantizar el rendimiento y la disponibilidad de su aplicación, utilice los tokens de Amazon Cognito hasta que caduquen y solo entonces recupere los nuevos. Una solución de caché que cree para su aplicación mantiene los tokens disponibles y evita que Amazon Cognito rechace las solicitudes cuando el porcentaje de solicitudes sea demasiado alto. Una aplicación del lado del cliente debe almacenar los tokens en una memoria caché. Una aplicación del lado del servidor puede añadir un mecanismo de caché cifrado para almacenar los tokens.

Cuando su grupo de usuarios genera un gran volumen de usuarios o machine-to-machine actividad, es posible que se encuentre con los límites que Amazon Cognito establece en cuanto al número de solicitudes de tokens que puede realizar. Para reducir el número de solicitudes que realiza a los

puntos de conexión de Amazon Cognito, puede almacenar y reutilizar los datos de autenticación de forma segura o implementar retrocesos y reintentos exponenciales.

Los datos de autenticación provienen de dos clases de puntos de conexión. Los puntos de conexión de OAuth 2.0 de Amazon Cognito incluyen el punto de conexión del token, que atiende las credenciales del cliente y las solicitudes de código de autorización de la interfaz Los puntos de conexión de servicio responden a solicitudes de API de grupos de usuarios como `InitiateAuth` y `RespondToAuthChallenge`. Cada tipo de solicitud tiene su propio límite. Para obtener más información acerca de los límites, consulte Cuotas en Amazon Cognito.

Almacenamiento en caché de los tokens de machine-to-machine acceso con Amazon API Gateway

Con el almacenamiento en caché de tokens de API Gateway, su aplicación puede reducir horizontalmente en respuesta a eventos que superen la cuota de solicitudes predeterminada de los puntos de conexión de Amazon Cognito OAuth.

Puedes almacenar en caché los tokens de acceso para que su aplicación solo solicite un nuevo token de acceso si un token en caché ha caducado. De lo contrario, el punto de conexión de almacenamiento en caché devuelve un token de la caché. Esto evita una llamada adicional a un punto de conexión de la API de Amazon Cognito. Cuando utilice Amazon API Gateway como proxy para Punto de conexión de token, su API responde a la mayoría de las solicitudes que, de otro modo, contribuirían a su cuota de solicitudes, lo que evita las solicitudes fallidas como resultado de la limitación de la tarifa.

La siguiente solución basada en API Gateway ofrece una implementación del almacenamiento en caché de tokens de baja latencia, bajo código o sin código. Las API de API Gateway se cifran en tránsito y, opcionalmente, en reposo. Una caché de API Gateway es ideal para la concesión de credenciales de clientes de OAuth 2.0, un tipo de concesión que suele ser de gran volumen y que produce tokens de acceso para autorizar sesiones machine-to-machine y microservicios. En el caso de que se produzca un aumento de tráfico que provoque que tus microservicios escalen horizontalmente, es posible que muchos sistemas utilicen las mismas credenciales de cliente con un volumen que supere el límite de AWS frecuencia de solicitudes de tu grupo de usuarios o cliente de la aplicación. Para preservar la disponibilidad de las aplicaciones y la baja latencia, se recomienda utilizar una solución de almacenamiento en caché en estos casos.

En esta solución, define una caché en su API para almacenar un token de acceso independiente para cada combinación de ámbitos y el cliente de aplicación de OAuth que quiera solicitar en su aplicación. Cuando la aplicación realiza una solicitud que coincide con la clave de caché, la API

responde con un token de acceso que Amazon Cognito emitió a la primera solicitud que coincidió con la clave de caché. Cuando caduca la duración de la clave de caché, la API reenvía la solicitud al punto de conexión del token y almacena en caché un nuevo token de acceso.

> **Note**
>
> La duración de la clave de caché debe ser inferior a la duración del token de acceso de su cliente de aplicación.

La clave de caché es una combinación de los ámbitos de OAuth que solicita en el parámetro de URL `scope` y el encabezado `Authorization` de la solicitud. El encabezado `Authorization` contiene el ID de cliente y el secreto de cliente de la aplicación. No tiene que implementar una lógica adicional en su aplicación para implementar esta solución. Solo debe actualizar la configuración para cambiar la ruta al punto de conexión del token del grupo de usuarios.

También puedes implementar el almacenamiento en caché de los tokens para Redis. ElastiCache Para un control detallado con políticas de AWS Identity and Access Management (IAM), considere una caché de Amazon DynamoDB.

> **Note**
>
> El almacenamiento en caché en API Gateway está sujeto a un coste adicional. Para obtener más información, consulte los precios.

Para configurar un proxy de almacenamiento en caché con API Gateway

1. Abra la consola de API Gateway y cree una API de REST.
2. En Resources (Recursos), cree un método POST.
    a. Elija el Integration type (Tipo de integración) de HTTP.
    b. Seleccione Use HTTP proxy integration (Usar integración de proxy HTTP).
    c. Introduzca una Endpoint URL (URL de punto de conexión) de `https://<your user pool domain>/oauth2/token`.
3. En Resources (Recursos), configure la clave de caché.
    a. Edite la Method request (Solicitud de método) de su método POST.

b. Establezca su parámetro `scope` y el encabezado `Authorization` como clave de almacenamiento en caché.

   i. Agregue una cadena de consulta a los URL query string parameters (Parámetros de la cadena de consulta URL) y elija Caching (Almacenamiento en caché) para la cadena `scope`.

   ii. Agregue un encabezado a los HTTP request headers (Encabezados de solicitud HTTP) y elija Caching (Almacenamiento en caché) para el encabezado `Authorization`.

4. En Stages (Etapas), configure el almacenamiento en caché.

   a. Elija la etapa que desea modificar.

   b. En Settings (Configuración), seleccione Enable API cache (Habilitar caché de API).

   c. Elija una Cache capacity (Capacidad de caché).

   d. Elija una caché time-to-live (TTL) de al menos 3600 segundos.

   e. Desmarque la casilla de verificación Requerir autorización.

5. En Stages (Etapas), anote la Invoke URL (URL de invocación).

6. Actualice su aplicación para solicitar el token POST a la Invoke URL (URL de invocación) de su API en lugar del punto de conexión de `/oauth2/token` de su grupo de usuarios.

Acceso a los recursos después de una autenticación correcta con el grupo de usuarios

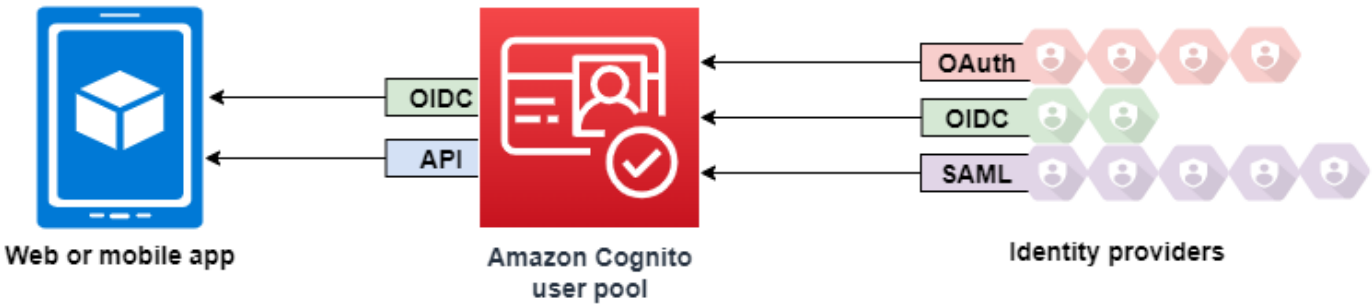
Los usuarios de tu aplicación pueden iniciar sesión directamente a través de un grupo de usuarios o pueden federarse a través de un proveedor de identidad (IdP) externo. El grupo de usuarios gestiona la sobrecarga de gestión de los tokens que se devuelven al iniciar sesión en redes sociales a través de Facebook, Google, Amazon y Apple, y desde OpenID Connect (OIDC) y SAML. IdPs Para obtener más información, consulte Uso de tokens con grupos de usuarios.

Tras una autenticación correcta, la aplicación web o móvil recibirá tokens de grupos de usuarios desde Amazon Cognito. Puedes usar los tokens del grupo de usuarios para:

- Recupere AWS las credenciales que autorizan las solicitudes de recursos de aplicaciones, Servicios de AWS como Amazon DynamoDB y Amazon S3.
- Proporcione una prueba de autenticación temporal y revocable.
- Introduce los datos de identidad en un perfil de usuario de tu aplicación.

- Autoriza los cambios en el perfil del usuario que ha iniciado sesión en el directorio del grupo de usuarios.
- Autoriza las solicitudes de información de los usuarios con un token de acceso.
- Autorice las solicitudes a los datos que se encuentran detrás de las API externas protegidas con acceso mediante tokens de acceso.
- Autorice el acceso a los activos de la aplicación que están almacenados en el cliente o el servidor con los permisos verificados de Amazon.

Para obtener más información, consulte Flujo de autenticación de los grupos de usuarios y Uso de tokens con grupos de usuarios.



Temas

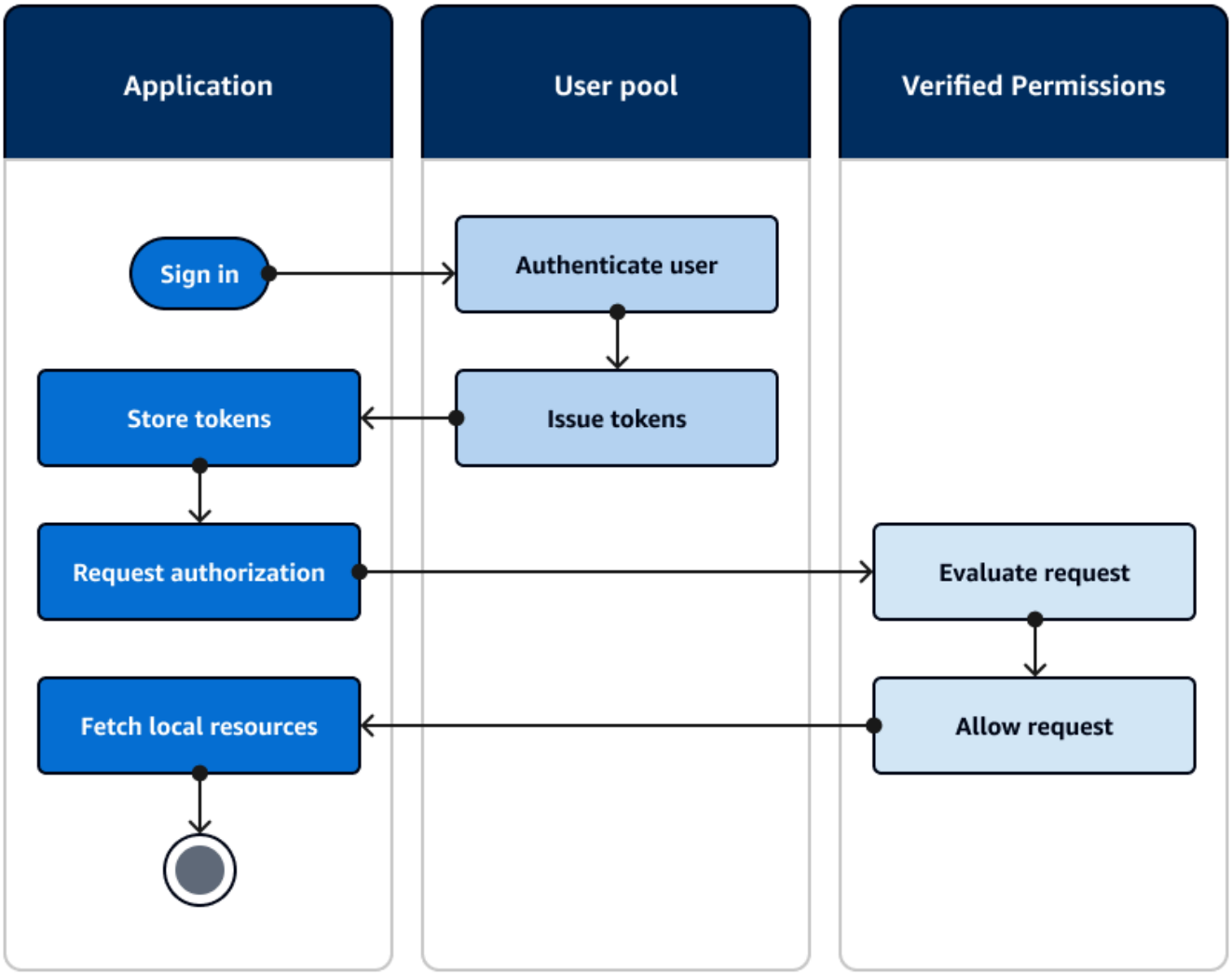
- Autorizar el acceso a los recursos del cliente o del servidor con los permisos verificados de Amazon
- Acceso a los recursos con API Gateway después de iniciar sesión
- Acceder Servicios de AWS mediante un grupo de identidades después de iniciar sesión

Autorizar el acceso a los recursos del cliente o del servidor con los permisos verificados de Amazon

Tu aplicación puede transferir los tokens de un usuario que haya iniciado sesión a Amazon Verified Permissions. Verified Permissions es un servicio de autorización y administración de permisos escalable y detallado para las aplicaciones personalizadas que hayas creado. Un grupo de usuarios de Amazon Cognito puede ser una fuente de identidad para un almacén de políticas de permisos verificados. Verified Permissions toma decisiones de autorización para las acciones y los recursos

solicitados, `GetPhoto` por ejemplo`premium_badge.png`, a partir del elemento principal y sus atributos en los tokens del grupo de usuarios.

En el siguiente diagrama, se muestra cómo la aplicación puede transferir el token de un usuario a Verified Permissions en una solicitud de autorización.



Comience con los permisos verificados de Amazon

Tras integrar su grupo de usuarios con los permisos verificados, obtendrá una fuente central de autorización detallada para todas sus aplicaciones de Amazon Cognito. Esto elimina la necesidad de una lógica de seguridad detallada que, de otro modo, tendría que codificar y replicar entre todas sus aplicaciones. Para obtener más información sobre la autorización con permisos verificados, consulte. [Autorización con Amazon Verified Permissions]

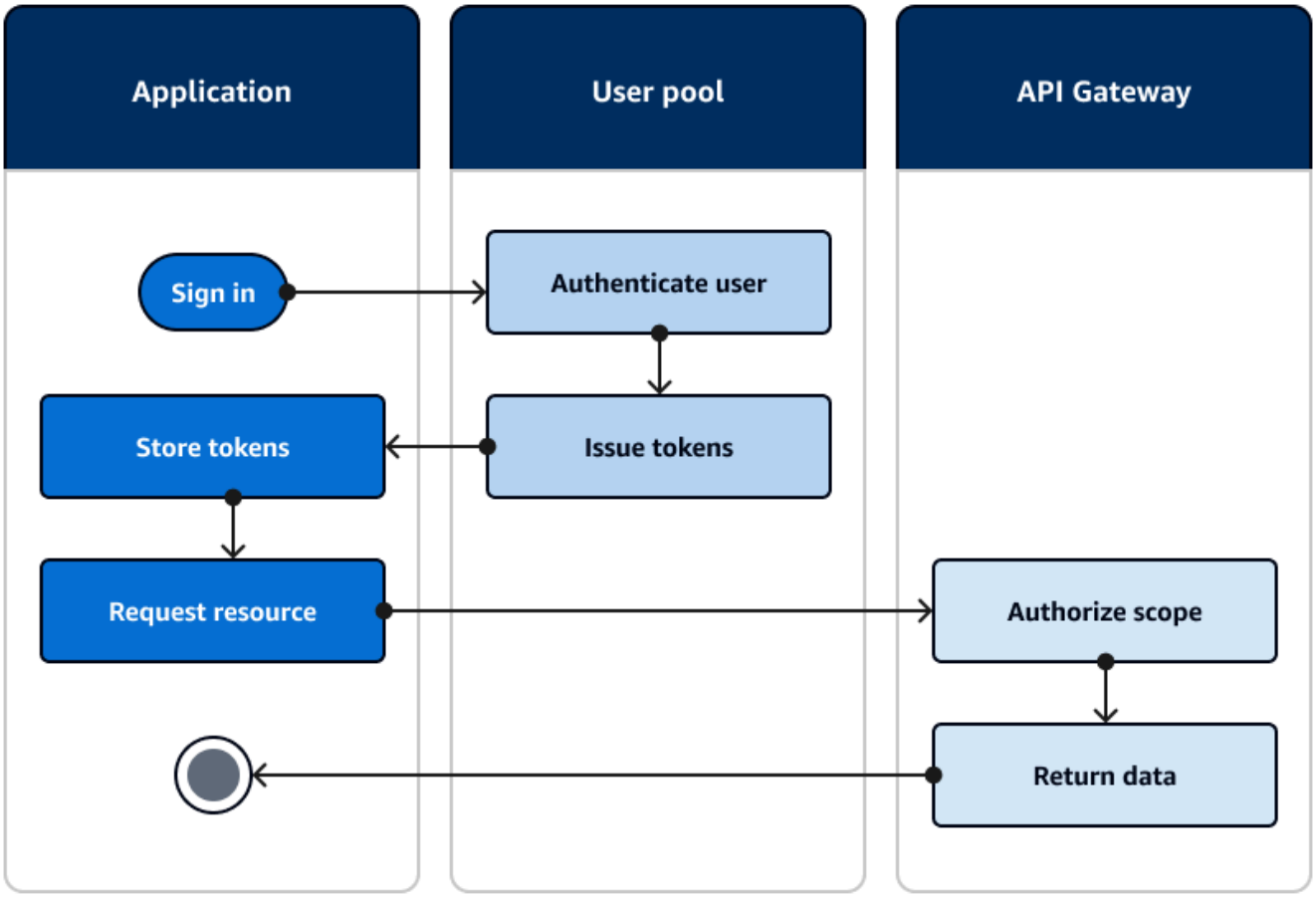
Las solicitudes de autorización de permisos verificados requieren AWS credenciales. Puede implementar algunas de las siguientes técnicas para aplicar las credenciales de forma segura a las solicitudes de autorización.

- Utilice una aplicación web que pueda almacenar secretos en el backend del servidor.
- Adquiera credenciales autenticadas del grupo de identidades.
- Utilice el proxy para las solicitudes de los usuarios a través de una access-token-authorized API y adjunte AWS las credenciales a la solicitud.

Acceso a los recursos con API Gateway después de iniciar sesión

Un uso común de los tokens de grupos de usuarios de Amazon Cognito es autorizar las solicitudes a una API REST de API Gateway. Los ámbitos de OAuth 2.0 de los tokens de acceso pueden autorizar un método y una ruta, por ejemplo, para. `HTTP GET /app_assets` Los identificadores pueden servir como autenticación genérica para una API y pueden transferir los atributos del usuario al servicio de backend. API Gateway tiene opciones de autorización personalizadas adicionales, como los autorizadores JWT para las API HTTP y los autorizadores Lambda que pueden aplicar una lógica más detallada.

El siguiente diagrama ilustra una aplicación que está accediendo a una API REST con los alcances de OAuth 2.0 en un token de acceso.



La aplicación debe recopilar los tokens de las sesiones autenticadas y añadirlos como tokens portadores a un `Authorization` encabezado de la solicitud. Configura el autorizador que configuraste para la API, la ruta y el método para evaluar el contenido de los tokens. API Gateway devuelve datos solo si la solicitud cumple con las condiciones que configuraste para tu autorizador.

Algunas posibles formas en las que la API API Gateway puede aprobar el acceso desde una aplicación son las siguientes:

- El token de acceso contiene el alcance correcto de OAuth 2.0. El autorizador de grupos de usuarios de Amazon Cognito para una API REST es una implementación común con una barrera de entrada baja. También puede evaluar el cuerpo, los parámetros de la cadena de consulta y los encabezados de una solicitud dirigida a este tipo de autorizador.
- El token de identificación es válido y no ha caducado. Al pasar un token de ID a un autorizador de Amazon Cognito, puede realizar una validación adicional del contenido del token de ID en el servidor de aplicaciones.

- Un grupo, afirmación, atributo o rol en un token de acceso o ID cumple con los requisitos que se definen en una función Lambda. Un autorizador de Lambda analiza el token en el encabezado de la solicitud y lo evalúa para tomar una decisión de autorización. Puedes crear una lógica personalizada en tu función o realizar una solicitud de API a Amazon Verified Permissions.

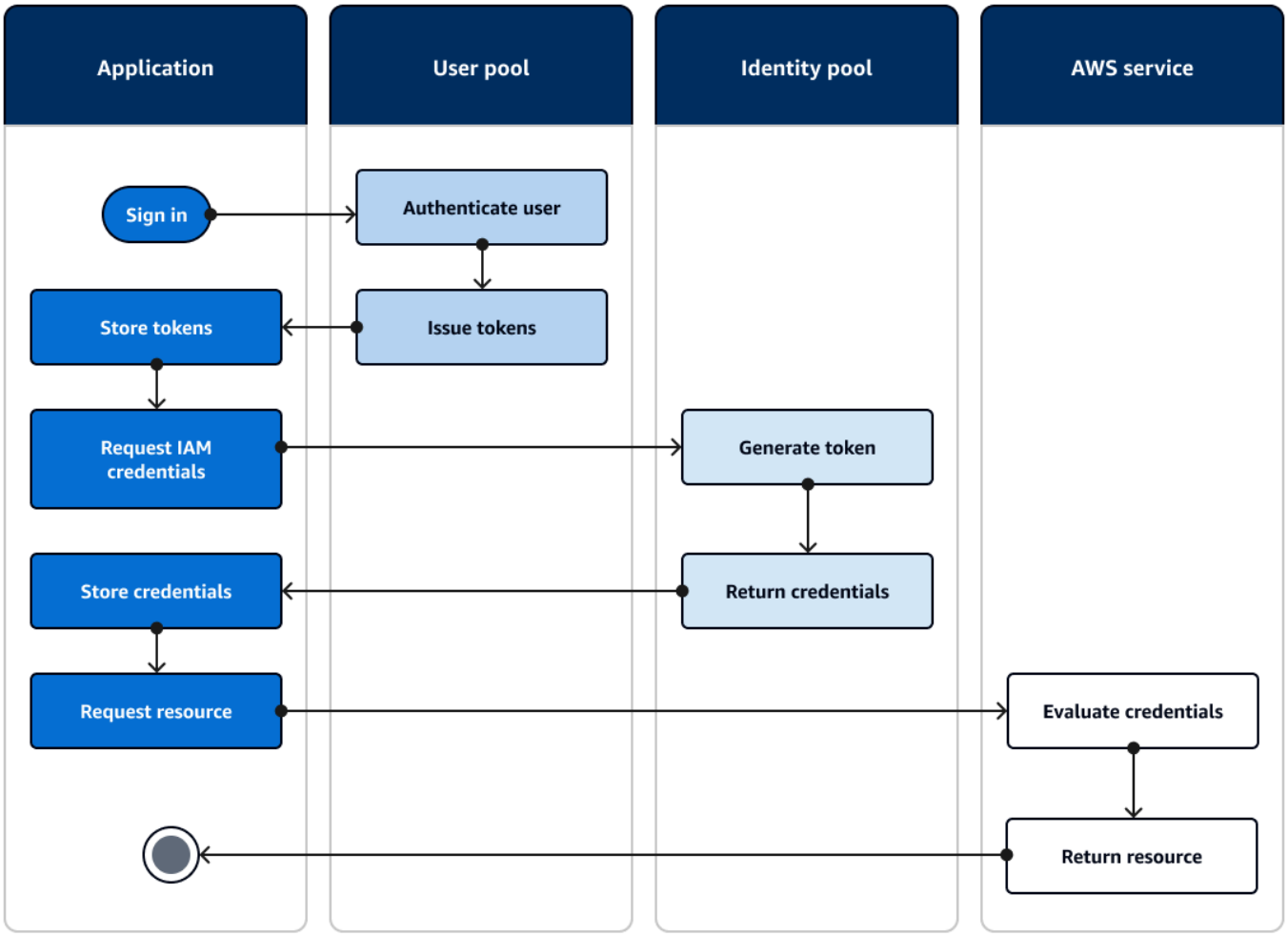
También puedes autorizar las solicitudes a una API de AWS AppSync GraphQL con tokens de un grupo de usuarios.

Acceder Servicios de AWS mediante un grupo de identidades después de iniciar sesión

Una vez que los usuarios inician sesión con un grupo de usuarios, pueden acceder Servicios de AWS con credenciales de API temporales emitidas desde un grupo de identidades.

Su aplicación web o móvil recibe los tokens de un grupo de usuarios. Cuando configura su grupo de usuarios como un proveedor de identidades para su grupo de identidades, el grupo de identidades intercambia los tokens por AWS credenciales temporales. Estas credenciales se pueden ajustar a las funciones de IAM y a sus políticas, que permiten a los usuarios acceder a un conjunto limitado de AWS recursos. Para obtener más información, consulte Flujo de autenticación de grupos de identidades (identidades federadas).

En el siguiente diagrama se muestra cómo una aplicación inicia sesión en un grupo de usuarios, recupera las credenciales del grupo de identidades y solicita un activo a un. Servicio de AWS



Puede usar las credenciales del grupo de identidades para:

- Realiza solicitudes de autorización detalladas a Amazon Verified Permissions con las propias credenciales de tu usuario.
- Conéctese a una API REST de Amazon API Gateway o a una API de AWS AppSync GraphQL que autorice las conexiones con IAM.
- Conéctese a un backend de base de datos, como Amazon DynamoDB o Amazon RDS, que autorice las conexiones con IAM.
- Recupere los activos de la aplicación de un bucket de Amazon S3.
- Inicie una sesión con un escritorio WorkSpaces virtual de Amazon.

Los grupos de identidades no funcionan exclusivamente dentro de una sesión autenticada con un grupo de usuarios. También aceptan la autenticación directamente de proveedores de identidad externos y pueden generar credenciales para los usuarios invitados no autenticados.

Para obtener más información sobre el uso de grupos de identidades junto con grupos de grupos de usuarios para controlar el acceso a sus AWS recursos, consulte Agregar grupos a un grupo de usuarios y. Uso del control de acceso basado en roles Además, para obtener más información sobre los grupos de identidades AWS Identity and Access Management, consulteConceptos de grupos de identidades.

Configuración de un grupo de usuarios con AWS Management Console

Cree un grupo de usuarios de Amazon Cognito y anote el ID del grupo de usuarios y el ID del cliente de la aplicación de cada una de sus aplicaciones cliente. Para obtener más información acerca de la creación de grupos de usuarios, consulte Introducción a los grupos de usuarios.

Configurar un grupo de identidades con AWS Management Console

El siguiente procedimiento describe cómo usarlo AWS Management Console para integrar un grupo de identidades con uno o más grupos de usuarios y aplicaciones cliente.

Para agregar un proveedor de identidades (IdP) de grupos de usuarios de Amazon Cognito

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Elija Grupo de usuarios de Amazon Cognito.
5. Introduzca un ID de grupo de usuarios y un ID de cliente de aplicación.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.

    a. Puede asignar a los usuarios de ese IdP el rol predeterminado que configuró cuando configuró su rol autenticado, o puede elegir un rol con reglas. Con un IdP del grupo de usuarios de Amazon Cognito, también puede Elegir un rol con solicitud preferred_role en los tokens. Para obtener más información acerca de la reclamación de `cognito:preferred_role`, consulte Asignación de valores de prioridad a los grupos.

i. Si ha elegido Elegir un rol con reglas, introduzca la reclamación de origen obtenida de la autenticación de su usuario, el operador que desee utilizar para comparar la afirmación con la regla, el valor que provocará una coincidencia con esta elección de función y la función que desee asignar cuando la asignación de funciones coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

ii. Si seleccionó la afirmación Choose role with preferred_role en los tokens, Amazon Cognito emitirá las credenciales para el rol en la afirmación de su usuario. `cognito:preferred_role` Si no hay ninguna solicitud de rol preferido, Amazon Cognito emite las credenciales basándose en su Resolución de rol.

b. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.

   - Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.
   - Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.
   - Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Integración de un grupo de usuarios con un grupo de identidades

Una vez que el usuario de la aplicación esté autenticado, añada el token de identidad de dicho usuario en la asignación de inicios de sesión en el proveedor de credenciales. El nombre del proveedor dependerá del ID del grupo de usuarios de Amazon Cognito. Tendrá la estructura siguiente:

```
cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>
```

Puede obtener el valor de a partir del ID del grupo de usuarios. <region> Por ejemplo, si el ID del grupo de usuarios es`us-east-1_EXAMPLE1`, entonces <region>es`us-east-1`. Si el ID del grupo de usuarios es`us-west-2_EXAMPLE2`, entonces <region>es`us-west-2`.

JavaScript

```
var cognitoUser = userPool.getCurrentUser();

if (cognitoUser != null) {
 cognitoUser.getSession(function(err, result) {
  if (result) {
   console.log('You are now logged in.');

   // Add the User's Id Token to the Cognito credentials login map.
   AWS.config.credentials = new AWS.CognitoIdentityCredentials({
    IdentityPoolId: 'YOUR_IDENTITY_POOL_ID',
    Logins: {
     'cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>':
 result.getIdToken().getJwtToken()
    }
   });
  }
 });
}
```

Android

```
cognitoUser.getSessionInBackground(new AuthenticationHandler() {
 @Override
 public void onSuccess(CognitoUserSession session) {
  String idToken = session.getIdToken().getJWTToken();

  Map<String, String> logins = new HashMap<String, String>();
  logins.put("cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>",
 session.getIdToken().getJWTToken());
  credentialsProvider.setLogins(logins);
 }

});
```

iOS - objective-C

```
AWSServiceConfiguration *serviceConfiguration = [[AWSServiceConfiguration alloc]
 initWithRegion:AWSRegionUSEast1 credentialsProvider:nil];
AWSCognitoIdentityUserPoolConfiguration *userPoolConfiguration =
 [[AWSCognitoIdentityUserPoolConfiguration alloc] initWithClientId:@"YOUR_CLIENT_ID"
  clientSecret:@"YOUR_CLIENT_SECRET" poolId:@"YOUR_USER_POOL_ID"];
[AWSCognitoIdentityUserPool
 registerCognitoIdentityUserPoolWithConfiguration:serviceConfiguration
 userPoolConfiguration:userPoolConfiguration forKey:@"UserPool"];
AWSCognitoIdentityUserPool *pool = [AWSCognitoIdentityUserPool
 CognitoIdentityUserPoolForKey:@"UserPool"];
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider
 alloc] initWithRegionType:AWSRegionUSEast1 identityPoolId:@"YOUR_IDENTITY_POOL_ID"
 identityProviderManager:pool];
```

iOS - swift

```
let serviceConfiguration = AWSServiceConfiguration(region: .USEast1,
 credentialsProvider: nil)
let userPoolConfiguration = AWSCognitoIdentityUserPoolConfiguration(clientId:
 "YOUR_CLIENT_ID", clientSecret: "YOUR_CLIENT_SECRET", poolId: "YOUR_USER_POOL_ID")
AWSCognitoIdentityUserPool.registerCognitoIdentityUserPoolWithConfiguration(serviceConfigura
 userPoolConfiguration: userPoolConfiguration, forKey: "UserPool")
let pool = AWSCognitoIdentityUserPool(forKey: "UserPool")
let credentialsProvider = AWSCognitoCredentialsProvider(regionType: .USEast1,
 identityPoolId: "YOUR_IDENTITY_POOL_ID", identityProviderManager:pool)
```

Uso de las características de seguridad de los grupos de usuarios de Amazon Cognito

Puede añadir la autenticación multifactor (MFA) a un grupo de usuarios para proteger la identidad de los usuarios. La MFA agrega un segundo método de autenticación de manera que el grupo de usuarios no depende exclusivamente del nombre de usuario y la contraseña. Puede optar por utilizar mensajes de texto SMS o contraseñas temporales de un solo uso (TOTP) como segundo factor para que los usuarios inicien sesión. También puede utilizar la autenticación flexible con su modelo basado en riesgos para predecir cuándo es posible que sea necesario utilizar otro factor de autenticación. Las características de seguridad avanzada del grupo de usuarios incluyen autenticación adaptativa y protecciones contra credenciales atacadas.

Temas

Adición de MFA a un grupo de usuarios.

La autenticación multifactor (MFA) aumenta la seguridad de la aplicación. Agrega un factor de autenticación algo que poseemos al factor algo que sabemos de nombre de usuario y contraseña. Puede utilizar mensajes de texto SMS o contraseñas temporales de un solo uso (TOTP) como segundo factor para el inicio de sesión de los usuarios.

> **Note**
>
> La primera vez que un nuevo usuario inicia sesión en su aplicación, Amazon Cognito emite tokens de OAuth 2.0, incluso si el grupo de usuarios exige MFA. El segundo factor de autenticación cuando el usuario inicia sesión por primera vez es la confirmación del mensaje de verificación que Amazon Cognito le envía. Si su grupo de usuarios exige MFA, Amazon Cognito le pide al usuario que registre un factor de inicio de sesión adicional para utilizarlo cada vez que se intente iniciar sesión después de la primera vez.

Con la autenticación flexible, puede configurar el grupo de usuarios para que requiera la autenticación de segundo factor en respuesta a un aumento del nivel de riesgo. Para añadir la autenticación flexible a un grupo de usuarios, consulte Adición de seguridad avanzada a un grupo de usuarios.

Al configurar la MFA en `required` para un grupo de usuarios, todos los usuarios deben completar la MFA para iniciar sesión. Cada usuario debe configurar como mínimo un factor MFA, como un SMS o TOTP. Al configurar la MFA en `required`, debe incluir la configuración de MFA en la incorporación de usuarios para que el grupo de usuarios les permita iniciar sesión.

Si activa los SMS como factor MFA, puede solicitar a los usuarios que proporcionen números de teléfono y que los verifiquen durante el registro. Si ha establecido la MFA en `required` y solo

admite SMS como factor, los usuarios deben proporcionar números de teléfono. Los usuarios que no tengan un número de teléfono necesitarán su ayuda para agregar un número de teléfono a su perfil antes de poder iniciar sesión. Puede utilizar números de teléfono no verificados para la MFA por SMS. Estos números recibirán el estado verificado después de que la MFA se haya realizado correctamente.

Si ha configurado la MFA para que se requiera y ha activado SMS y TOTP como métodos de verificación compatibles, Amazon Cognito pedirá a los nuevos usuarios sin números de teléfono que configuren la MFA con TOTP. Si ha configurado la MFA para que se requiera y el único método de MFA que ha activado es TOTP, Amazon Cognito pedirá a todos los nuevos usuarios que configuren la MFA con TOTP la segunda vez que inicien sesión. Amazon Cognito plantea el desafío de configurar el MFA TOTP en respuesta InitiateAutha las operaciones de la API. AdminInitiateAuth

La interfaz de usuario alojada solicita a los usuarios que configuren la MFA cuando se establece que la MFA es obligatoria. Al configurar la MFA como opcional en el grupo de usuarios, la interfaz de usuario alojada no se lo pide a los usuarios. Para trabajar con la MFA opcional, debe crear una interfaz en la aplicación que pida a los usuarios que seleccionen si desean configurar la MFA y, a continuación, los guíe por las entradas de la API para comprobar el factor de inicio de sesión adicional.

Tras cinco intentos erróneos de presentar un código MFA, Amazon Cognito inicia el proceso de bloqueo por tiempo de espera exponencial descrito en Flujo de autenticación de los grupos de usuarios.

Temas

- Requisitos previos
- Configuración de la autenticación multifactor
- MFA por mensaje de texto SMS
- MFA con token de software TOTP

Requisitos previos

Antes de configurar la MFA, tenga en cuenta lo siguiente:

- Si activa la MFA en el grupo de usuarios y elige Mensaje de texto SMS como segundo factor, puede enviar mensajes SMS a un atributo de número de teléfono que no haya verificado en Amazon Cognito. Una vez que el usuario complete la MFA por SMS, Amazon Cognito establece su atributo `phone_number_verified` como `true`.

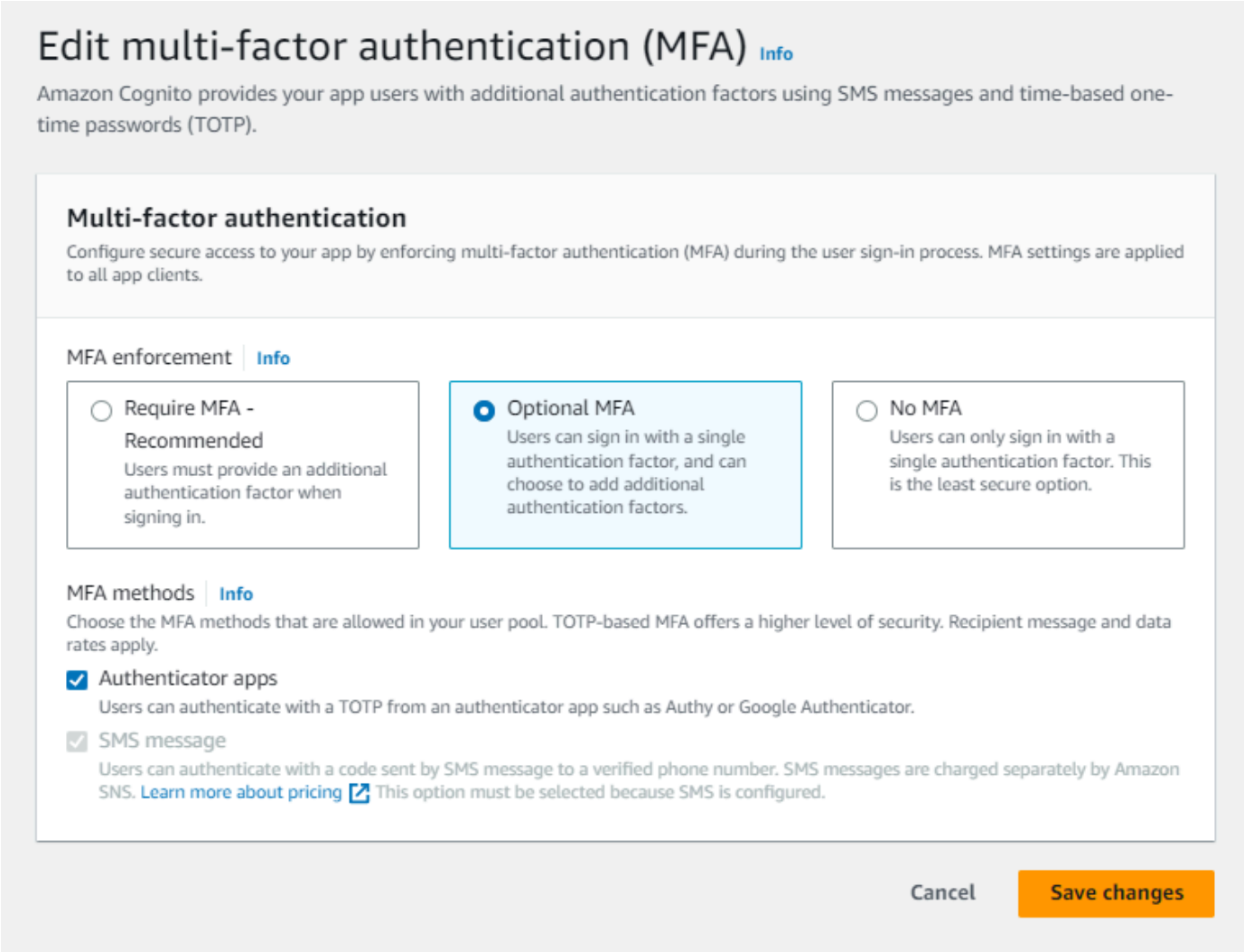
- Si su cuenta se encuentra en el entorno limitado de SMS Región de AWS que contiene los recursos del Amazon Simple Notification Service (Amazon SNS) para su grupo de usuarios, debe verificar los números de teléfono en Amazon SNS antes de poder enviar un mensaje SMS. Para obtener más información, consulte Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito.
- Las características de seguridad avanzadas requieren que active la MFA y que se establezca como opcional en la consola del grupo de usuarios de Amazon Cognito. Para obtener más información, consulte Adición de seguridad avanzada a un grupo de usuarios.

Configuración de la autenticación multifactor

Puede configurar la MFA en la consola de Amazon Cognito.

Para configurar la MFA en la consola de Amazon Cognito, siga estos pasos:

1. Inicie sesión en la consola de Amazon Cognito.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña Sign-in experience (Experiencia de inicio de sesión). Localice Multi-factor authentication (Autenticación multifactor) y elija Edit (Editar)
5. Elija el método MFA enforcement (Aplicación de MFA) que desea utilizar con su grupo de usuarios.



a. Require MFA (Requerir MFA): todos los usuarios de su grupo de usuarios deben iniciar sesión con un factor de código SMS adicional o de contraseña temporal de un solo uso (TOTP).

b. Optional MFA (MFA opcional): puede dar a sus usuarios la opción de registrar un factor de inicio de sesión adicional y seguir permitiendo el inicio de sesión de los usuarios sin tener una MFA configurada. Elija esta opción si utiliza la autenticación adaptativa. Para obtener más información sobre la autenticación flexible, consulte Adición de seguridad avanzada a un grupo de usuarios.

c. No MFA (Sin MFA): los usuarios no pueden registrar un factor de inicio de sesión adicional.

6. Elija los MFA methods (Métodos MFA) compatibles con su aplicación. Puede establecer tanto SMS message (Mensaje SMS) como Authenticator apps (Aplicaciones autenticadoras) para generar TOTP como segundo factor. Le recomendamos que implemente una MFA basada en TOTP para que la recuperación de cuentas pueda usar mensajes SMS.

7. Si utiliza los mensajes de texto SMS como segundo factor y no tiene un rol de IAM configurado para usar con Amazon Simple Notification Service (Amazon SNS) para mensajes SMS, cree uno en la consola. En la pestaña Messaging (Mensajería) para el grupo de usuarios, localice SMS y elija Edit (Editar). También puede utilizar un rol existente que permita a Amazon Cognito enviar mensajes SMS a los usuarios por usted. Para obtener más información, consulte Roles de IAM.
8. Elija Save changes (Guardar cambios).

MFA por mensaje de texto SMS

Cuando los usuarios inician sesión con la MFA habilitada, primero ingresan y envían su nombre de usuario y su contraseña. La aplicación cliente recibe una respuesta `getMFA` que indica dónde se ha enviado el código de autorización. La aplicación cliente debe indicar al usuario dónde buscar el código (por ejemplo, a qué número de teléfono se envió el código). A continuación, se proporciona un formulario para ingresar el código. Por último, la aplicación cliente envía el código para completar el proceso de inicio de sesión. El destino está enmascarado, lo que oculta todos los dígitos del número de teléfono, excepto los últimos cuatro. Si en una aplicación se utiliza la IU alojada de Amazon Cognito, se muestra una página para que el usuario ingrese el código de MFA.

El código de autorización del mensaje de texto SMS es válido para Authentication flow session duration (Duración de sesión del flujo de autenticación) configurado para el cliente de aplicaciones.

Defina la duración de una sesión de flujo de autenticación en la consola de Amazon Cognito en la pestaña App integration (Integración de aplicaciones), cuando modifique el cliente de su aplicación en App clients and analytics (Clientes de aplicaciones y análisis). También puede establecer la duración de la sesión del flujo de autenticación en una solicitud de API de `CreateUserPoolClient` o `UpdateUserPoolClient`. Para obtener más información, consulte Flujo de autenticación de los grupos de usuarios.

Si un usuario ya no tiene acceso al dispositivo al que se envían los códigos de MFA por mensaje de texto SMS, debe solicitar ayuda al servicio de atención al cliente. Un administrador con Cuenta de AWS los permisos necesarios puede cambiar el número de teléfono del usuario, pero solo a través de la API AWS CLI .

Cuando un usuario realiza correctamente el flujo de MFA por mensaje de texto SMS, su número de teléfono también se marca como verificado.

> **Note**
>
> Los mensajes SMS de la autenticación multifactor se facturan por separado. (No se aplica ningún cargo por el envío de códigos de verificación a las direcciones de correo electrónico). Para obtener información sobre los precios de Amazon SNS, consulte Precios de SMS en todo el mundo. Para ver la lista actual de los países en los que los mensajes SMS están disponibles, consulte Regiones y países admitidos.

> **Important**
>
> Para garantizar que se envíen mensajes SMS con el fin de verificar números de teléfono y MFA por mensaje de texto SMS, debe solicitar un aumento del límite de gasto a Amazon SNS.
> En Amazon Cognito, se utiliza Amazon SNS para enviar mensajes SMS a los usuarios. La cantidad de mensajes SMS que Amazon SNS entrega está sujeta a los límites de gasto. Los límites de gasto se pueden especificar para una AWS cuenta y para mensajes individuales, y los límites se aplican únicamente al coste del envío de mensajes SMS.
> El límite de gasto predeterminado por cuenta (si no se especifica) es de 1,00 USD al mes. Si quieres aumentar el límite, presenta un caso de aumento del límite de SNS en el AWS Support Centro. En New limit value (Nuevo valor del límite), especifique el límite de gasto mensual que desee. En el campo Use Case Description (Descripción de caso de uso), explique que solicita un aumento del límite de gasto mensual en SMS.

Para añadir la MFA a un grupo de usuarios, consulte Adición de MFA a un grupo de usuarios.. Para obtener más información sobre los mensajes SMS con Amazon SNS en su grupo de usuarios, consulte. Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito

MFA con token de software TOTP

Al configurar la MFA de token de software TOTP en el grupo de usuarios, el usuario inicia sesión con un nombre de usuario y una contraseña y, a continuación, utiliza una TOTP para completar la autenticación. Después de que el usuario establezca y verifique un nombre de usuario y una contraseña, puede activar un token de software TOTP para la MFA. Si la aplicación utiliza la interfaz de usuario alojada de Amazon Cognito para el inicio de sesión de los usuarios, el usuario envía el nombre de usuario y la contraseña y, a continuación, envía la contraseña TOTP en una página de inicio de sesión adicional.

Puede activar la MFA con TOTP para el grupo de usuarios en la consola de Amazon Cognito o utilizar las operaciones de la API de Amazon Cognito. A nivel del grupo de usuarios, puede llamar SetUserPoolMfaConfigpara configurar el MFA y habilitar el MFA TOTP.

> **Note**
>
> Si no activa la MFA con token de software de TOTP para el grupo de usuarios, Amazon Cognito no podrá usar el token para asociar ni verificar usuarios. En este caso, los usuarios reciben un excepción `SoftwareTokenMFANotFoundException` con la descripción `Software Token MFA has not been enabled by the userPool`. Si posteriormente desactiva la MFA con token de software para el grupo de usuarios, los usuarios que asociaron y verificaron previamente un token de TOTP podrán seguir utilizándolo para la MFA.

La configuración de TOTP para el usuario es un proceso de varios pasos en el que el usuario recibe un código secreto que valida introduciendo una contraseña de un solo uso. A continuación, se puede activar la MFA con TOTP para el usuario o configurar TOTP como método de MFA preferido para el usuario.

Cuando configure su grupo de usuarios para solicitar que MFA con TOTP y sus usuarios se registren en su aplicación en la IU alojada, Amazon Cognito automatiza el proceso del usuario. Amazon Cognito pide al usuario que elija un método de MFA, muestra un código QR para configurar su aplicación de autenticación y verifica su registro de MFA. En los grupos de usuarios en los que ha permitido a los usuarios elegir entre MFA por SMS y TOTP, Amazon Cognito también ofrece al usuario una selección de métodos. Para obtener más información sobre la experiencia de registro en la interfaz de usuario alojada, consulte Cómo registrarse en una nueva cuenta en la IU alojada de Amazon Cognito.

> **Important**
>
> Si tiene una ACL AWS WAF web asociada a un grupo de usuarios y una regla de su ACL web presenta un CAPTCHA, esto puede provocar un error irrecuperable en el registro del TOTP de la interfaz de usuario alojada. Para crear una regla que tenga una acción CAPTCHA y no afecte al TOTP de la IU alojada, consulte Configuración de la ACL AWS WAF web para la interfaz de usuario alojada en el MFA. Para obtener más información sobre

las ACL AWS WAF web y Amazon Cognito, consulte. Asociar una ACL AWS WAF web a un grupo de usuarios

Para implementar MFA con TOTP en una IU personalizada en la que utilice la API de Amazon Cognito, consulte Configuración de MFA para un usuario en la API del grupo de usuarios de Amazon Cognito.

Para añadir la MFA a un grupo de usuarios, consulte Adición de MFA a un grupo de usuarios..

Condiciones y limitaciones de la MFA con TOTP

1. Amazon Cognito admite la MFA con token de software a través de una aplicación de autenticación que genera códigos de TOTP. Amazon Cognito no admite la MFA basada en hardware.
2. Cuando el grupo de usuarios requiere una TOTP para un usuario que no la ha configurado, este recibe un token de acceso de un solo uso que la aplicación puede utilizar para activar la MFA con TOTP para dicho usuario. Los intentos de inicio de sesión posteriores fallarán hasta que el usuario haya registrado un factor de inicio de sesión adicional con TOTP.
   - Un usuario que se registra en el grupo de usuarios con la operación de la API `SignUp` o a través de la IU alojada recibirá tokens de un solo uso al finalizar el registro.
   - Después de crear un usuario y de que este configure su contraseña inicial, Amazon Cognito emite tokens de un solo uso desde la IU alojada para el usuario. Si establece una contraseña permanente para el usuario, Amazon Cognito emite tokens de un solo uso cuando el usuario inicia sesión por primera vez.
   - Amazon Cognito no emite tokens de un solo uso a un usuario creado por el administrador que inicia sesión con las operaciones de la API o la API. InitiateAuthAdminInitiateAuth Después de que el usuario supere el desafío de establecer su contraseña inicial o si usted establece una contraseña permanente para el usuario, Amazon Cognito desafía inmediatamente al usuario para que configure la MFA.
3. Si un usuario de un grupo de usuarios que requiere la MFA ya ha recibido un token de acceso de un solo uso pero no ha configurado la MFA con TOTP, el usuario no podrá iniciar sesión en la IU alojada hasta que haya configurado la MFA. En lugar del token de acceso, puede usar el valor de `session` respuesta de un `MFA_SETUP` desafío a una solicitud InitiateAutho AdminInitiateAuthen una solicitud. AssociateSoftwareToken
4. Si los usuarios han configurado una TOTP, pueden usarla para la MFA, incluso si posteriormente desactiva la TOTP para el grupo de usuarios.

5. Amazon Cognito solo acepta los TOTP de las aplicaciones de autenticación que generan códigos con la función de inserción SHA-1. Los códigos generados con la función de inserción SHA-256 devuelven un error de `Code mismatch`.

Configuración de MFA para un usuario en la API del grupo de usuarios de Amazon Cognito

Cuando un usuario inicia sesión por primera vez, la aplicación utiliza su token de acceso de un solo uso para generar la clave privada TOTP y presentarla al usuario en formato de texto o código QR. El usuario configura su aplicación de autenticación y proporciona un TOTP para los intentos de inicio de sesión posteriores. Su aplicación o la IU alojada presentan el TOTP a Amazon Cognito en las respuestas al desafío de MFA.

Temas

- Asociar el token de software TOTP
- Verificar el token de TOTP
- Describe cómo iniciar sesión utilizando la MFA con TOTP
- Eliminación del token de TOTP

Asociar el token de software TOTP

Para asociar el token de TOTP, debe enviar un código secreto al usuario que este debe validar con una contraseña de un solo uso. Para asociar el token se deben seguir tres pasos.

1. Cuando el usuario elija el MFA del token de software TOTP, AssociateSoftwareTokenllame para obtener un código clave secreto compartido generado único para la cuenta de usuario. Puede autorizar AssociateSoftwareToken con un token de acceso o una cadena de sesión.
2. La aplicación presenta al usuario la clave privada o un código QR que usted genera a partir de la clave privada. El usuario debe ingresar la clave en una aplicación generadora de TOTP, como Google Authenticator. Puede usar libqrencode para generar un código QR.
3. Cuando el usuario introduce la clave o escanea el código QR en una aplicación de autenticación como Google Authenticator, la aplicación comienza a generar códigos.

Verificar el token de TOTP

A continuación, verifique el token de TOTP. Para solicitar los códigos de muestra a su usuario y proporcionárselos al servicio Amazon Cognito para confirmar que el usuario está generando correctamente códigos TOTP, siga estos pasos.

1. La aplicación solicita al usuario un código para demostrar que ha configurado correctamente su aplicación de autenticación.
2. La aplicación de autenticación del usuario muestra una contraseña temporal. La aplicación de autenticación basa la contraseña en la clave secreta que usted le dio al usuario.
3. El usuario ingresa su contraseña temporal. La aplicación pasa la contraseña temporal a Amazon Cognito en una solicitud a la API `VerifySoftwareToken`.
4. Amazon Cognito ha conservado la clave secreta asociada al usuario, genera un TOTP y la compara con la que proporcionó el usuario. Si coinciden, `VerifySoftwareToken` devuelve una respuesta `SUCCESS`.
5. Amazon Cognito asocia el factor TOTP al usuario.
6. Si la operación `VerifySoftwareToken` devuelve una respuesta `ERROR`, asegúrese de que el reloj del usuario sea correcto y de que no haya superado el número máximo de reintentos. Amazon Cognito acepta los tokens TOTP que se encuentran dentro de los 30 segundos anteriores o posteriores al intento, para tener en cuenta el sesgo menor del reloj. Cuando haya resuelto el problema, vuelva a intentar la VerifySoftwareToken operación.

Describe cómo iniciar sesión utilizando la MFA con TOTP

En este punto, el usuario inicia sesión con la contraseña temporal de un solo uso. El proceso es el siguiente.

1. El usuario ingresa el nombre de usuario y la contraseña para iniciar sesión en la aplicación cliente.
2. Se invoca el desafío de la MFA con TOTP y, desde la aplicación, se le pide al usuario que ingrese una contraseña temporal.
3. El usuario obtiene la contraseña temporal de una aplicación generadora de TOTP asociada.
4. El usuario introduce el código de TOTP en la aplicación cliente. La aplicación solicita al servicio de Amazon Cognito que la verifique. RespondToAuthChallengeDebe llamarlo cada vez que inicie sesión para obtener una respuesta al nuevo desafío de autenticación del TOTP.
5. Si Amazon Cognito verifica el token, el inicio de sesión es exitoso y el usuario continúa con el flujo de autenticación.

Eliminación del token de TOTP

Por último, la aplicación debería permitir al usuario desactivar la configuración de TOTP. En la actualidad, no puede eliminar el token del software TOTP de un usuario. Para reemplazar el token de software del usuario, asocie y verifique un nuevo token de software. Para desactivar el MFA TOTP para un usuario, llame a SetUserMFAPreference para modificar su usuario de modo que no utilice ningún MFA o solo MFA por SMS.

1. Cree una interfaz en la aplicación para los usuarios que deseen restablecer la MFA. Pida a un usuario de esta interfaz que ingrese la contraseña.
2. Si Amazon Cognito devuelve un desafío de MFA TOTP, actualice la preferencia de MFA del usuario con MFAPreference. SetUser
3. En la aplicación, comunique al usuario que ha desactivado la MFA y pídale que vuelva a iniciar sesión.

Configuración de la ACL AWS WAF web para la interfaz de usuario alojada en el MFA

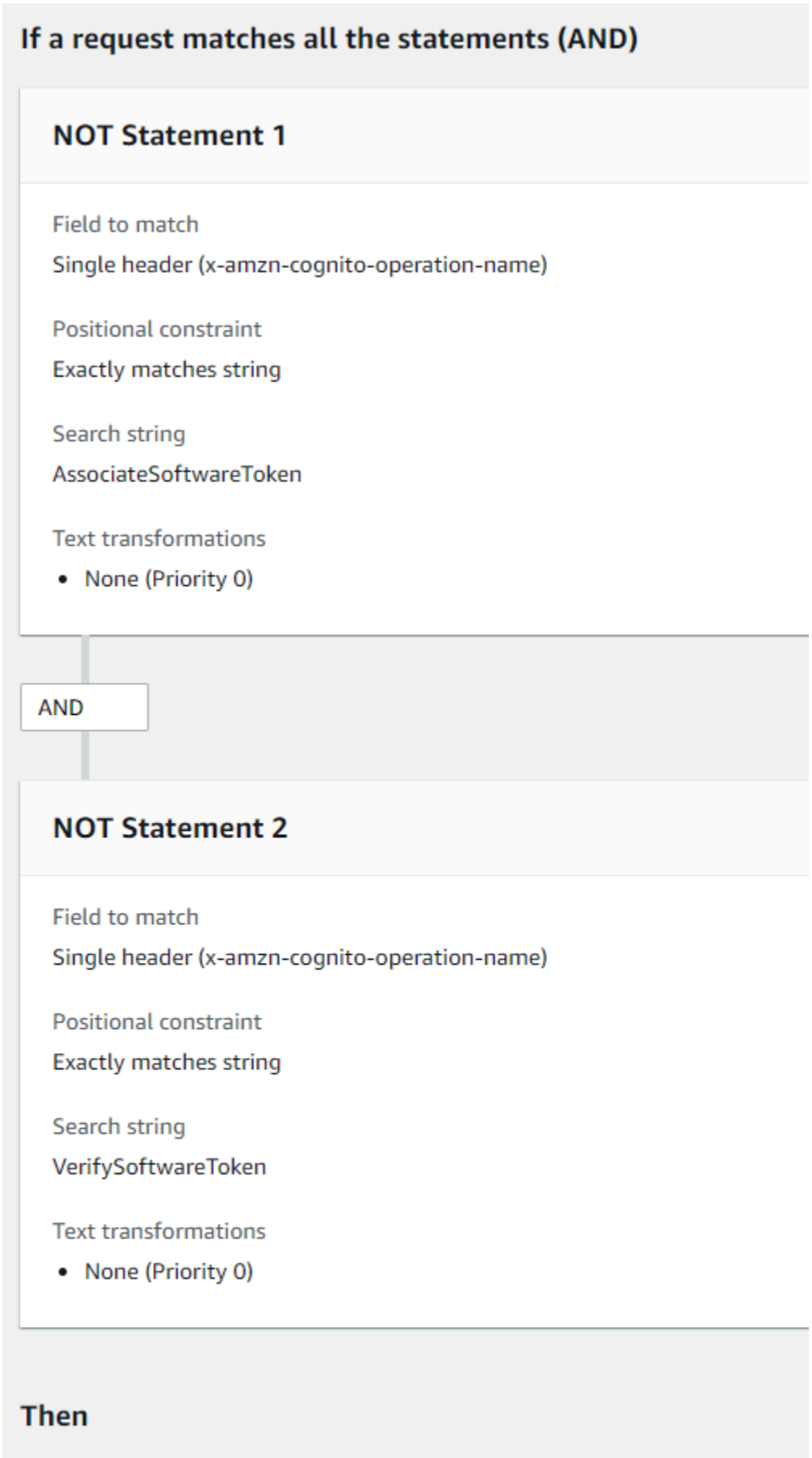
Si tiene una ACL AWS WAF web asociada a un grupo de usuarios y una regla de su ACL web presenta un CAPTCHA, esto puede provocar un error irrecuperable en el registro del TOTP de la interfaz de usuario alojada. AWS WAF Las reglas de CAPTCHA solo afectan al MFA TOTP en la interfaz de usuario alojada de esta manera. La MFA de SMS no se ve afectada.

Amazon Cognito muestra el siguiente error cuando la regla de CAPTCHA no permite que un usuario complete la configuración de MFA con TOTP.

La solicitud no se admite debido al captcha de WAF.

Este error se produce cuando se AWS WAF solicita un CAPTCHA en respuesta a AssociateSoftwareTokenlas solicitudes de VerifySoftwareTokenAPI que el grupo de usuarios realiza en segundo plano. Para crear una regla que tenga una acción CAPTCHA y no afecte al TOTP de IU alojada, excluya los valores del encabezado `x-amzn-cognito-operation-name` de `AssociateSoftwareToken` y `VerifySoftwareToken` de la acción CAPTCHA en tu regla.

En la siguiente captura de pantalla se muestra un ejemplo de AWS WAF regla que aplica una acción de CAPTCHA a todas las solicitudes que no tienen un `x-amzn-cognito-operation-name` valor de encabezado igual o. `AssociateSoftwareToken` `VerifySoftwareToken`


If a request matches all the statements (AND)
NOT Statement 1
Field to match
Single header (x-amzn-cognito-operation-name)
Positional constraint
Exactly matches string
Search string
AssociateSoftwareToken
Text transformations
None (Priority 0)
AND
NOT Statement 2
Field to match
Single header (x-amzn-cognito-operation-name)
Positional constraint
Exactly matches string
Search string
VerifySoftwareToken
Text transformations
None (Priority 0)
Then

Para obtener más información sobre las ACL AWS WAF web y Amazon Cognito, consulte. Asociar una ACL AWS WAF web a un grupo de usuarios

Adición de seguridad avanzada a un grupo de usuarios

Después de crear el grupo de usuarios, recibirá acceso a la opción Advanced security (Seguridad avanzada) de la barra de navegación de la consola de Amazon Cognito. Se pueden activar las características de seguridad avanzadas del grupo de usuarios y personalizar las acciones que se toman en respuesta a los distintos riesgos. También es posible utilizar el modo de auditoría para recopilar métricas sobre los riesgos detectados sin necesidad de aplicar mitigación alguna de seguridad. En el modo auditoría, las funciones de seguridad avanzadas publican las métricas en Amazon CloudWatch. Puede ver las métricas de seguridad avanzadas después de que Amazon Cognito genere el primer evento de seguridad avanzada. Consulte Visualización de las métricas de seguridad avanzadas.

Las características de seguridad avanzada incluyen detección de credenciales comprometidas y autenticación adaptiva.

Credenciales comprometidas

Los usuarios reutilizan las contraseñas de varias cuentas de usuario. La característica de credenciales comprometidas de Amazon Cognito recopila datos de filtraciones públicas de nombres de usuario y contraseñas y compara las credenciales de los usuarios con listas de credenciales filtradas. La detección de credenciales comprometidas también comprueba las contraseñas que se suelen adivinar.

Puede elegir las acciones del usuario que solicitan la comprobación de credenciales comprometidas y la acción que desea que Amazon Cognito realice en respuesta. Para los eventos de inicio de sesión, registro y cambio de contraseña, Amazon Cognito puede Bloquear el inicio de sesión o Permitir el inicio de sesión. En ambos casos, Amazon Cognito genera un registro de actividad del usuario, donde puede encontrar más información sobre el evento.

Autenticación flexible

Amazon Cognito puede revisar la información sobre la ubicación y el dispositivo de las solicitudes de inicio de sesión de los usuarios y aplicar una respuesta automática para proteger las cuentas de usuario del grupo de usuarios contra actividades sospechosas.

Al activar la seguridad avanzada, Amazon Cognito asigna una puntuación de riesgo a la actividad del usuario. Puede asignar una respuesta automática a una actividad sospechosa: puede solicitar la MFA, bloquear el inicio de sesión o simplemente registrar los detalles de la actividad y la

puntuación de riesgo. También puede enviar automáticamente mensajes de correo electrónico para notificar al usuario la actividad sospechosa para que pueda restablecer la contraseña o realizar otras acciones autoguiadas.

Personalización del token de acceso

Al activar las características de seguridad avanzadas, puede configurar el grupo de usuarios para que acepte respuestas a un evento desencadenante de Lambda versión 2. Con la versión 2, puede personalizar los alcances y otras reclamaciones de los tokens de acceso. Esto aumenta la capacidad de crear resultados de autorización flexibles cuando los usuarios se autentican. Para obtener más información, consulte Personalización del token de acceso.

Temas

Consideraciones y limitaciones

- Se aplican otros precios para las características de seguridad avanzadas de Amazon Cognito. Consulte la página de precios de Amazon Cognito.
- Amazon Cognito admite la autenticación adaptativa y la detección de credenciales comprometidas con los siguientes flujos de autenticación estándar:, y. `USER_PASSWORD_AUTH` `ADMIN_USER_PASSWORD_AUTH` `USER_SRP_AUTH` No se puede utilizar la seguridad avanzada con un flujo `CUSTOM_AUTH` y Desencadenadores de Lambda de desafío de autenticación personalizado, o con inicio de sesión federado.
- Con las características de seguridad avanzadas de Amazon Cognito en modo Funcionalidad completa, puede crear una dirección IP con las excepciones Bloquear siempre y Permitir siempre. A una sesión de una dirección IP en la lista de excepciones Always block (Bloquear siempre) no se le asigna un nivel de riesgo mediante la autenticación adaptativa y no puede iniciar sesión en su grupo de usuarios.

- Las solicitudes bloqueadas de direcciones IP en una lista de excepciones Always block (Bloquear siempre) del grupo de usuarios contribuye a las cuotas de las tasas de solicitudes para los grupos de usuarios. Las características de seguridad avanzadas de Amazon Cognito no impiden los ataques de denegación de servicio distribuido (DDoS). Para implementar defensas contra los ataques volumétricos en sus grupos de usuarios, añada ACL web. AWS WAF Para obtener más información, consulte Asociar una ACL AWS WAF web a un grupo de usuarios.
- La concesión de credenciales de cliente está destinada a la autorización machine-to-machine (M2M) sin conexión con las cuentas de usuario. Las características de seguridad avanzadas solo supervisan las cuentas de usuario y contraseñas de su grupo de usuarios. Para implementar funciones de seguridad en su actividad M2M, tenga en cuenta las capacidades de monitoreo de AWS WAF las tasas de solicitudes y el contenido. Para obtener más información, consulte Asociar una ACL AWS WAF web a un grupo de usuarios.

Requisitos previos

Antes de comenzar, necesitará lo siguiente:

- Un grupo de usuarios con un cliente de aplicación. Para obtener más información, consulte Introducción a los grupos de usuarios.
- Establezca la autenticación multifactor (MFA) en Optional (Opcional) en la consola de Amazon Cognito para utilizar la característica de autenticación flexible basada en riesgos. Para obtener más información, consulte Adición de MFA a un grupo de usuarios..
- Si utiliza notificaciones por correo electrónico, diríjase a la consola de Amazon SES para configurar y verificar un dominio o una dirección de correo electrónico con el fin de usar notificaciones por correo electrónico. Para obtener más información sobre Amazon SES, consulte Verificación de identidades en Amazon SES.

Configuración de las características de seguridad avanzadas

Puede configurar las funciones de seguridad avanzadas de Amazon Cognito en el AWS Management Console.

Para configurar la seguridad avanzada para un grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).

3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña App integration (Integración de aplicaciones). Localice Advanced security (Seguridad avanzada) y elija Enable (Habilitar). Si habilitó la seguridad avanzada anteriormente, elija Edit (Editar).
5. Seleccione Full function (Funcionalidad completa) para configurar respuestas de seguridad avanzadas a las credenciales atacadas y la autenticación adaptativa. Seleccione Auditar solo para recopilar información y enviar los datos del grupo de usuarios a CloudWatch. Los precios de seguridad avanzados se aplican en ambos:Audit only (Solo auditoría) y Full function (Función completa). Para obtener más información, consulte Precios de Amazon Cognito.

   Recomendamos mantener las características de seguridad avanzadas en el modo de auditoría durante dos semanas antes de activar las acciones. En ese tiempo, Amazon Cognito puede aprender los patrones de uso de los usuarios de la aplicación.
6. Si seleccionó Audit only (Solo auditoría), elija Save changes (Guardar los cambios). Si seleccionó Full function (Función completa):

   a. Seleccione si va a realizar una acción Custom (Personalizada) o a utilizar los Cognito defaults (Valores predeterminados de Cognito) para responder ante supuestas Compromised credentials (Credenciales atacadas). Los Cognito defaults (Valores predeterminados de Cognito) son:

      i. Detectar credenciales atacadas enSign-in (Inicio de sesión), Sign-up (Registro), y Password change (Cambio de contraseña).

      ii. Responder ante credenciales atacadas con la acción Block sign-in (Bloquear inicio de sesión).

   b. Si ha seleccionado acciones Custom (Personalizadas) para Compromised credentials (Credenciales comprometidas), elija las acciones del grupo de usuarios que Amazon Cognito utilizará para la Event detection (Detección de eventos) y las Compromised credentials responses (Respuestas contra credenciales atacadas) que desearía que Amazon Cognito realizara. PuedeBlock sign-in (Bloquear inicio de sesión) o Allow sign-in (Permitir inicio de sesión) con supuestas credenciales atacadas.

   c. Elija cómo responder a los intentos de inicio de sesión maliciosos en Adaptive authentication (Autenticación flexible). Seleccione si va a realizar una acción Custom (Personalizada) o a utilizar los Cognito defaults (Valores predeterminados de Cognito) para responder ante supuestas Compromised credentials (Credenciales atacadas). Cuando

selecciona Cognito defaults (Valores predeterminados de Cognito), Amazon Cognito bloquea el inicio de sesión en todos los niveles de riesgo y no notifica al usuario.

d. Si selecciona acciones Custom (Personalizadas) para Adaptive Authentication (Autenticación flexible), elija acciones de Automatic risk response (Respuesta automática al riesgo) que Amazon Cognito llevará a cabo en respuesta a los riesgos detectados en función del nivel de gravedad. Cuando asigna una respuesta a un nivel de riesgo, no se puede asignar una respuesta menos restrictiva a un nivel de riesgo más alto. Puede asignar las siguientes respuestas a los niveles de riesgo:

   i. Allow sign-in (Permitir inicio de sesión): No se realizan acciones preventivas.

   ii. Optional MFA (MFA opcional): si el usuario tiene MFA configurada, Amazon Cognito siempre requerirá que el usuario proporcione un SMS adicional o un factor de contraseña temporal de un solo uso (TOTP) cuando inicie sesión. Si el usuario no tiene MFA configurada, puede seguir iniciando sesión normalmente.

   iii. Require MFA (Requerir MFA): si el usuario tiene MFA configurada, Amazon Cognito siempre requerirá que el usuario proporcione un SMS adicional o un factor de contraseña temporal de un solo uso (TOTP) cuando inicie sesión. Si el usuario no tiene MFA configurada, Amazon Cognito le pedirá que configure la MFA. Antes de requerir automáticamente la MFA para los usuarios, configure un mecanismo en la aplicación para capturar números de teléfono para la MFA por SMS o para registrar aplicaciones autenticadoras para la MFA por TOTP.

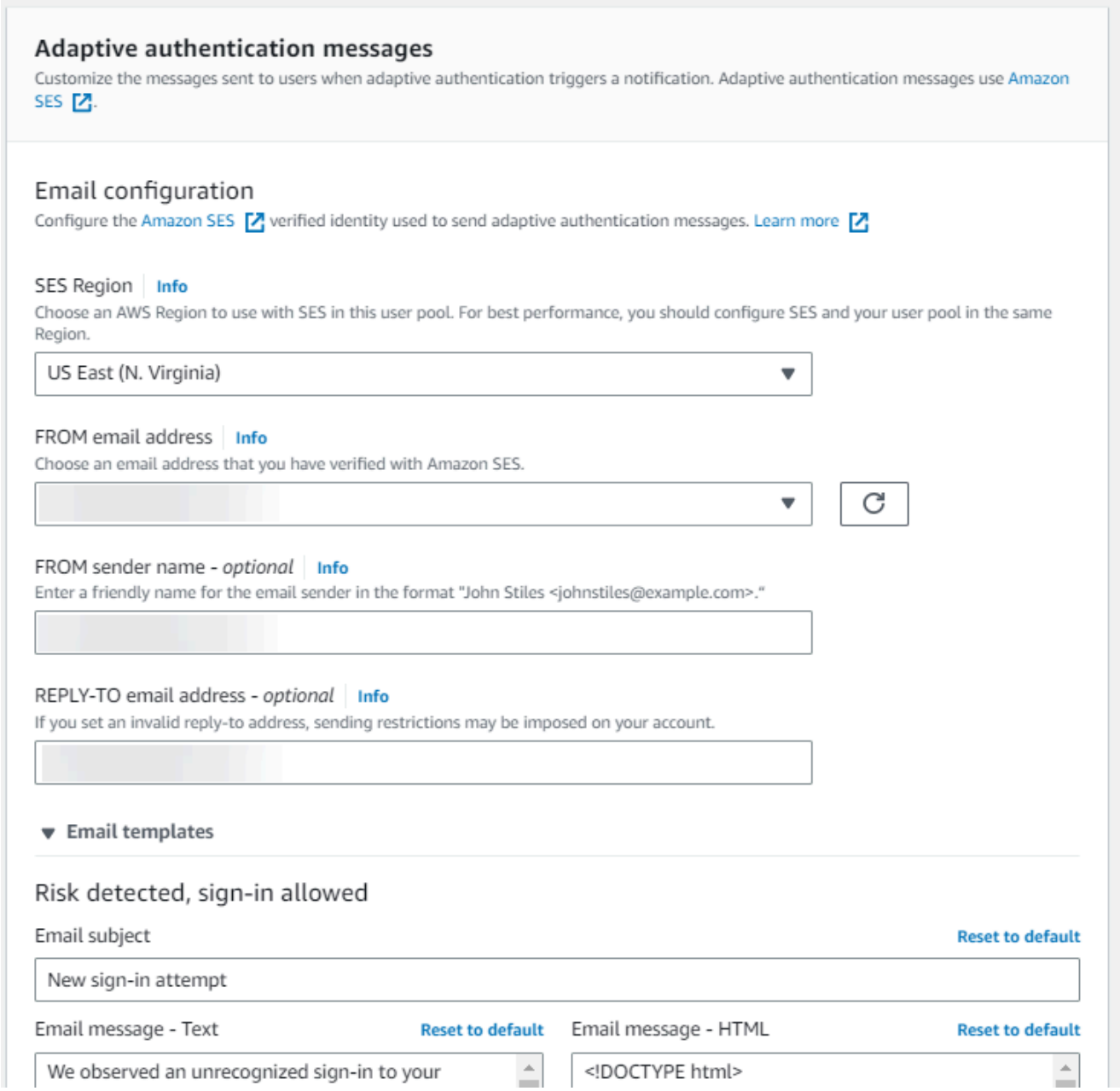
   iv. Block sign-in (Bloquear inicio de sesión): impide que el usuario inicie sesión.

   v. Notify user (Notificar al usuario): envía un mensaje de correo electrónico al usuario con información sobre el riesgo que Amazon Cognito detectó junto con la respuesta que se ha realizado. Puede personalizar plantillas de correo electrónico para los mensajes que envíe.

7. Si eligió Notify user (Notificar al usuario) en el paso anterior, puede personalizar la configuración de entrega de correo electrónico y las plantillas de menajes de correo electrónico para una autenticación flexible.

   a. En Email configuration (Configuración de correo electrónico), elija las opciones SES Region (Región SES), FROM email address (DE dirección de correo electrónico), FROM sender name (DE remitente) y REPLY-TO email address (Dirección de correo electrónico del DESTINATARIO) que desea utilizar con la autenticación flexible. Para obtener más información sobre cómo integrar los mensajes de correo electrónico del grupo de usuarios

con Amazon Simple Email Service, consulte Configuración de correo electrónico para grupos de usuarios de Amazon Cognito.



b. Expanda Email templates (Plantillas de correo electrónico) para personalizar las notificaciones de autenticación flexible con versiones de correo electrónico en HTML y de texto sin formato. Para obtener más información sobre las plantillas de mensajes de correo electrónico, consulte Plantillas de mensaje.

8. Expanda IP address exceptions (Excepciones de dirección IP) para crear una lista de Always allow (Permitir siempre) o deAlways-block (Bloquear siempre) de los intervalos de direcciones IPv4 o IPv6 que siempre se permitirán o bloquearán, independientemente de la evaluación de

riesgos de seguridad avanzada. Especifique los intervalos de direcciones IP en CIDR notation (Notación CIDR) (como por ejemplo: 192.168.100.0/24).

9. Elija Guardar cambios.

Comprobación de las credenciales atacadas

Amazon Cognito puede detectar si el nombre de usuario y la contraseña de un usuario se han visto comprometidos en otro sitio. Esto puede ocurrir cuando los usuarios reutilizan las credenciales en más de un sitio, o cuando utilizan contraseñas poco seguras. Amazon Cognito comprueba los usuarios locales que inician sesión con nombre de usuario y contraseña, en la interfaz de usuario alojada y con la API de Amazon Cognito. Un usuario local existe exclusivamente en el directorio del grupo de usuarios sin federación a través de un IdP externo.

Desde Advanced security (Seguridad avanzada) en la pestaña App integration (Integración de aplicaciones) de la consola de Amazon Cognito, puede configurar Compromised credentials (Credenciales comprometidas). Configure Event detection (Detección de eventos) para elegir los eventos de usuario que desea supervisar en busca de credenciales comprometidas. Configure Compromised credentials responses (Respuestas ante credenciales comprometidas) para elegir si desea permitir o bloquear al usuario si se han detectado credenciales comprometidas. Amazon Cognito puede comprobar si hay credenciales comprometidas durante el inicio de sesión, el registro o los cambios de contraseña.

Si selecciona Permitir inicio de sesión, puede revisar Amazon CloudWatch Logs para supervisar las evaluaciones que Amazon Cognito realiza sobre los eventos de los usuarios. Para obtener más información, consulte Visualización de las métricas de seguridad avanzadas. Cuando elige Block sign-in (Bloquear el inicio de sesión), Amazon Cognito impide el inicio de sesión de los usuarios que utilizan credenciales comprometidas. Cuando Amazon Cognito bloquea el inicio de sesión de un usuario, establece el `UserStatus` de usuario en `RESET_REQUIRED`. Un usuario con un estado `RESET_REQUIRED` debe cambiar su contraseña antes de poder iniciar sesión de nuevo.

> **Note**
>
> En este momento, Amazon Cognito no comprueba si hay credenciales comprometidas para las operaciones de inicio de sesión con el flujo Secure Remote Password (SRP). SRP envía una prueba de contraseña cifrada durante el inicio de sesión. Amazon Cognito no tiene acceso a las contraseñas internamente, por lo que solo puede evaluar una contraseña que el cliente le transmita en texto plano.

Amazon Cognito comprueba los inicios de sesión que utilizan la AdminInitiateAuthAPI con `ADMIN_USER_PASSWORD_AUTH` flow y la InitiateAuthAPI con `USER_PASSWORD_AUTH` flow para detectar credenciales comprometidas.

Para añadir protecciones contra credenciales atacadas a un grupo de usuarios, consulte Adición de seguridad avanzada a un grupo de usuarios.

Uso de la autenticación flexible

Con la autenticación flexible, puede configurar el grupo de usuarios para bloquear los inicios de sesión sospechosos o agregar la autenticación de segundo factor en respuesta a un aumento del nivel de riesgo. Para cada intento de inicio de sesión, Amazon Cognito genera una puntuación de riesgo que indica la probabilidad de que la solicitud de inicio de sesión proceda de un origen comprometido. Esta puntuación de riesgo se basa en factores que incluyen la información del dispositivo y del usuario. La autenticación adaptativa puede activarse o requerir la autenticación multifactor (MFA) para un usuario de su grupo de usuarios cuando Amazon Cognito detecta un riesgo en la sesión de un usuario y este aún no ha elegido un método de MFA. Cuando se activa la MFA para un usuario, siempre se le presenta el desafío de proporcionar o configurar un segundo factor durante la autenticación, independientemente de cómo se haya configurado la autenticación adaptativa. Desde el punto de vista del usuario, la aplicación ofrece ayuda para configurar la MFA y, opcionalmente, Amazon Cognito le impide volver a iniciar sesión hasta que haya configurado un factor adicional.

Amazon Cognito publica los intentos de inicio de sesión, sus niveles de riesgo y las impugnaciones fallidas para Amazon. CloudWatch Para obtener más información, consulte Visualización de las métricas de seguridad avanzadas.

Para añadir la autenticación flexible a un grupo de usuarios, consulte Adición de seguridad avanzada a un grupo de usuarios.

Temas

- Información general sobre la autenticación flexible
- Adición de datos de sesión y dispositivos de usuario a las solicitudes de API
- Visualización del historial de eventos de los usuarios
- Suministro de comentarios sobre los eventos
- Envío de mensajes de notificación

Información general sobre la autenticación flexible

En la página Seguridad avanzada de la pestaña Integración de la aplicación de la consola de Amazon Cognito, puede elegir la configuración de la autenticación adaptativa, incluidas las acciones que se deben realizar para los distintos niveles de riesgo y la personalización de los mensajes de notificación para los usuarios. Puede asignar una configuración de seguridad avanzada global a todos sus clientes de aplicaciones, pero aplicar una configuración de nivel de cliente a los clientes de aplicaciones individuales.

La autenticación adaptativa de Amazon Cognito asigna uno de los siguientes niveles de riesgo a cada sesión de usuario: alto, medio, bajo o sin riesgo.

Estudie bien sus opciones cuando cambie Enforcement method (Método de aplicación) de Audit-only (Solo auditoría) a Full-function (Función completa). Las respuestas automáticas que se aplican a los niveles de riesgo influyen en el nivel de riesgo que Amazon Cognito asigna a las sesiones de usuario posteriores con las mismas características. Por ejemplo, si decide no realizar ninguna acción o marcar Allow (Permitir) en las sesiones de usuario que Amazon Cognito evalúa inicialmente como de alto riesgo, Amazon Cognito considera que las sesiones similares tienen un riesgo menor.

Para cada nivel de riesgo, puede elegir entre las opciones siguientes:

Opción	Acción
Permitir	Los usuarios pueden iniciar sesión sin un factor adicional.
Optional MFA (MFA opcional)	Los usuarios que tengan configurado un segundo factor deberán superar un segundo desafío de segundo factor para iniciar sesión. Los segundos factores disponibles son un número de teléfono para SMS y un token de software TOTP. Los usuarios que no tienen un segundo factor configurado pueden iniciar sesión con un solo conjunto de credenciales.
Require MFA (Requerir MFA)	Los usuarios que tengan configurado un segundo factor deberán superar un desafío de segundo factor para iniciar sesión. Amazon Cognito bloquea el inicio de sesión de los

Opción	Acción
	usuarios que no hayan configurado un segundo factor.
Bloque	Amazon Cognito bloquea todos los intentos de inicio de sesión con el nivel de riesgo designado.

> **Note**
>
> No es necesario verificar los números de teléfono para utilizarlos como segundo factor de autenticación para SMS.

Adición de datos de sesión y dispositivos de usuario a las solicitudes de API

Puede recopilar y transferir información sobre la sesión de su usuario a la seguridad avanzada de Amazon Cognito cuando usa la API para registrarlo, iniciarlo y restablecer su contraseña. Esta información incluye la dirección IP de su usuario y un identificador de dispositivo único.

Es posible que tenga un dispositivo de red intermedio entre sus usuarios y Amazon Cognito, como un servicio proxy o un servidor de aplicaciones. Puede recopilar los datos de contexto de los usuarios y pasarlos a Amazon Cognito para que la autenticación adaptativa calcule el riesgo en función de las características del punto de conexión de usuario, en lugar de su servidor o proxy. Si la aplicación del lado del cliente llama directamente a las operaciones de la API de Amazon Cognito, la autenticación adaptativa registra automáticamente la dirección IP de origen. Sin embargo, no registra otra información del dispositivo, como el `user-agent`, a menos que también recoja una huella digital del dispositivo.

Genere estos datos con la biblioteca de recopilación de datos contextuales de Amazon Cognito y envíelos a Amazon Cognito Advanced Security con ContextDatalos parámetros y. UserContextData La biblioteca de recopilación de datos contextuales se incluye en los AWS SDK. Para obtener más información, consulte Integración de Amazon Cognito en aplicaciones web y móviles. Puede enviar `ContextData` si ha activado características de seguridad avanzadas en su grupo de usuarios. Para obtener más información, consulte Configuración de las características de seguridad avanzadas.

Cuando llame a las siguientes operaciones de API autenticadas de Amazon Cognito desde el servidor de aplicaciones, pase la IP del dispositivo del usuario en el parámetro `ContextData`. Además, debe transferir el nombre del servidor, la ruta del servidor y los datos de la huella dactilar codificada del dispositivo.

- AdminInitiateAuth
- AdminRespondToAuthChallenge

Cuando llama a operaciones de API sin autenticar de Amazon Cognito, puede enviar `UserContextData` a las características de seguridad avanzadas de Amazon Cognito. Estos datos incluyen una huella digital de dispositivo en el parámetro `EncodedData`. También puede enviar un parámetro `IpAddress` en su `UserContextData` si cumple las condiciones siguientes:

- Ha activado funciones de seguridad avanzadas en su grupo de usuarios. Para obtener más información, consulte Configuración de las características de seguridad avanzadas.
- El cliente de aplicación tiene un secreto de cliente. Para obtener más información, consulte Configuración de un cliente de aplicación para un grupo de usuarios.
- Ha activado Accept additional user context data (Aceptar datos de contexto de usuario adicionales) en el cliente de aplicación. Para obtener más información, consulte Aceptación de datos de contexto de usuario adicionales (AWS Management Console).

Su aplicación puede rellenar el parámetro `UserContextData` con datos codificados de huellas digitales del dispositivo y la dirección IP del dispositivo del usuario en las siguientes operaciones de API no autenticadas de Amazon Cognito.

- InitiateAuth
- RespondToAuthChallenge
- SignUp
- ConfirmSignUp
- ForgotPassword
- ConfirmForgotPassword
- ResendConfirmationCode

Aceptación de datos de contexto de usuario adicionales (AWS Management Console)

El grupo de usuarios acepta una dirección IP en un parámetro `UserContextData` después de activar la característica Accept additional user context data (Aceptar datos de contexto de usuario adicionales). No es necesario activar esta característica si:

- Sus usuarios solo inician sesión con operaciones de API autenticadas, por ejemplo AdminInitiateAuth , y usted usa el `ContextData` parámetro.
- Solo desea que las operaciones de API no autenticadas envíen una huella digital del dispositivo, pero no una dirección IP, a las características de seguridad avanzadas de Amazon Cognito.

Actualice el cliente de aplicación como se indica a continuación en la consola de Amazon Cognito para agregar compatibilidad con datos de contexto de usuario adicionales.

1. Inicie sesión en la consola de Amazon Cognito.
2. En el panel de navegación, elija Manage your User Pools (Administrar sus grupos de usuarios) y elija el grupo de usuarios que desea editar.
3. Elija la pestaña App integration (Integración de aplicaciones).
4. En App clients and analytics (Clientes de aplicaciones y análisis), elija o cree un cliente de aplicación. Para obtener más información, consulte Configuración de un cliente de aplicación para grupos de usuarios.
5. Elija Edit (Editar) desde el contenedor de App client information (Información del cliente de aplicación).
6. En Advanced authentication settings (Configuración avanzada de autenticación) del cliente de aplicación, elija Accept additional user context data (Aceptar datos de contexto de usuario adicionales).
7. Elija Guardar cambios.

Para configurar el cliente de la aplicación para que acepte datos de contexto de usuario en la API de Amazon Cognito, `EnablePropagateAdditionalUserContextData` configúrelo `true` en una solicitud CreateUserPoolClientto UpdateUserPoolClient. Para obtener información sobre cómo activar la seguridad avanzada desde la aplicación web o móvil, consulte Activación de la seguridad avanzada del grupo de usuarios desde la aplicación. Recopile los datos contextuales del usuario desde el lado del cliente cuando la aplicación llame a Amazon Cognito desde el servidor. A continuación, se muestra un ejemplo en el que se utiliza el método JavaScript `getData` SDK.

```
var encodedData =
 AmazonCognitoAdvancedSecurityData.getData(username, userPoolId, clientId);
```

Cuando diseñe la aplicación para utilizar la autenticación flexible, le recomendamos que incorpore el último SDK de Amazon Cognito a la aplicación. La última versión del SDK recopila la información de las huellas dactilares del dispositivo como el ID, el modelo y la zona horaria del dispositivo. Para obtener más información sobre los SDK de Amazon Cognito, consulte Instalación de un SDK de grupo de usuarios. La seguridad avanzada de Amazon Cognito solo guarda y asigna una puntuación de riesgo a los eventos que la aplicación envía en el formato correcto. Si Amazon Cognito devuelve una respuesta de error, compruebe que su solicitud incluya un hash de secreto válido y que el parámetro `IPaddress` sea una dirección IPv4 o IPv6 válida.

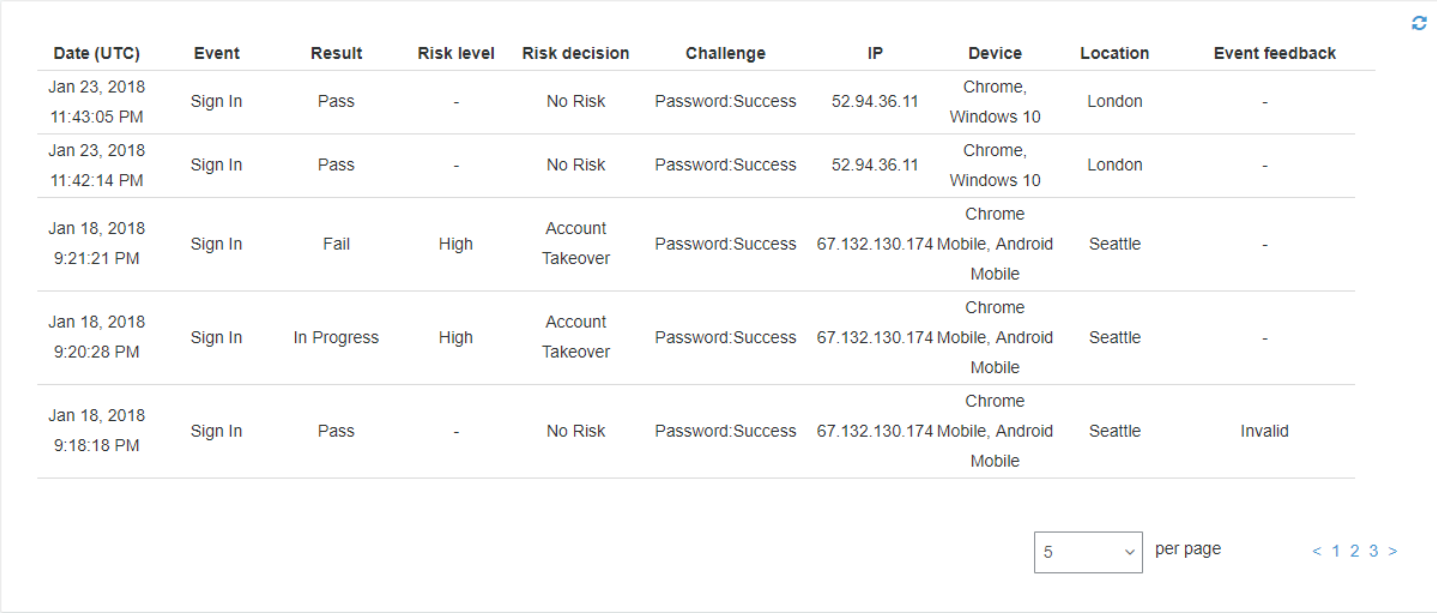
Recursos **`ContextData`** y **`UserContextData`**

- AWS Amplify SDK para Android: GetUserContextData
- AWS Amplify SDK para iOS: userContextData
- JavaScript: amazon-cognito-advanced-security-data.min.js

Visualización del historial de eventos de los usuarios

> **Note**
>
> En la nueva consola de Amazon Cognito, puede ver el historial de eventos del usuario en la pestaña Users (Usuarios).

Para ver el historial de inicios de sesión de un usuario, puede elegir el usuario en Users (Usuarios) en la consola de Amazon Cognito. Amazon Cognito conserva el historial de eventos del usuario durante de dos años.



Date (UTC)	Event	Result	Risk level	Risk decision	Challenge	IP	Device	Location	Event feedback
Jan 23, 2018 11:43:05 PM	Sign In	Pass	-	No Risk	Password:Success	52.94.36.11	Chrome, Windows 10	London	-
Jan 23, 2018 11:42:14 PM	Sign In	Pass	-	No Risk	Password:Success	52.94.36.11	Chrome, Windows 10	London	-
Jan 18, 2018 9:21:21 PM	Sign In	Fail	High	Account Takeover	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	-
Jan 18, 2018 9:20:28 PM	Sign In	In Progress	High	Account Takeover	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	-
Jan 18, 2018 9:18:18 PM	Sign In	Pass	-	No Risk	Password:Success	67.132.130.174	Chrome Mobile, Android Mobile	Seattle	Invalid

5 per page < 1 2 3 >

Cada evento de inicio de sesión tiene un ID de evento. El evento también tiene los datos contextuales correspondientes, como la ubicación, los detalles del dispositivo y los resultados de detección de riesgos. Puede consultar el historial de eventos del usuario con la operación de la API de Amazon Cognito AdminListUserAuthEventso con AWS Command Line Interface (AWS CLI) con admin-list-user-auth -events.

También puede correlacionar el ID de evento con el token que Amazon Cognito emitió en el momento en que registró el evento. El ID y los tokens de acceso incluyen este ID de evento en su carga. Amazon Cognito también correlaciona el uso de tokens de actualización con el ID de evento original. El ID de evento original permite localizar el ID de evento del inicio de sesión que dio lugar a la emisión de los tokens de Amazon Cognito. Esto le permite realizar un seguimiento del uso de un token en su sistema hasta un evento de autenticación en concreto. Para obtener más información, consulte Uso de tokens con grupos de usuarios.

Suministro de comentarios sobre los eventos

Los comentarios sobre los eventos afectan a la evaluación de riesgos en tiempo real y mejoran el algoritmo de evaluación de riesgos a lo largo del tiempo. Puede proporcionar comentarios sobre la validez de los intentos de inicio de sesión a través de las operaciones de API y la consola de Amazon Cognito.

> **Note**
>
> Sus comentarios sobre el evento influyen en el nivel de riesgo que Amazon Cognito asigna a las sesiones de usuario posteriores con las mismas características.

En la consola de Amazon Cognito, elija un usuario en la pestaña Users (Usuarios) y seleccione Provide event feedback (Proporcionar comentarios sobre el evento). Puede revisar los detalles del evento y marcar Set as valid (Establecer como válido) o Set as invalid (Establecer como no válido).

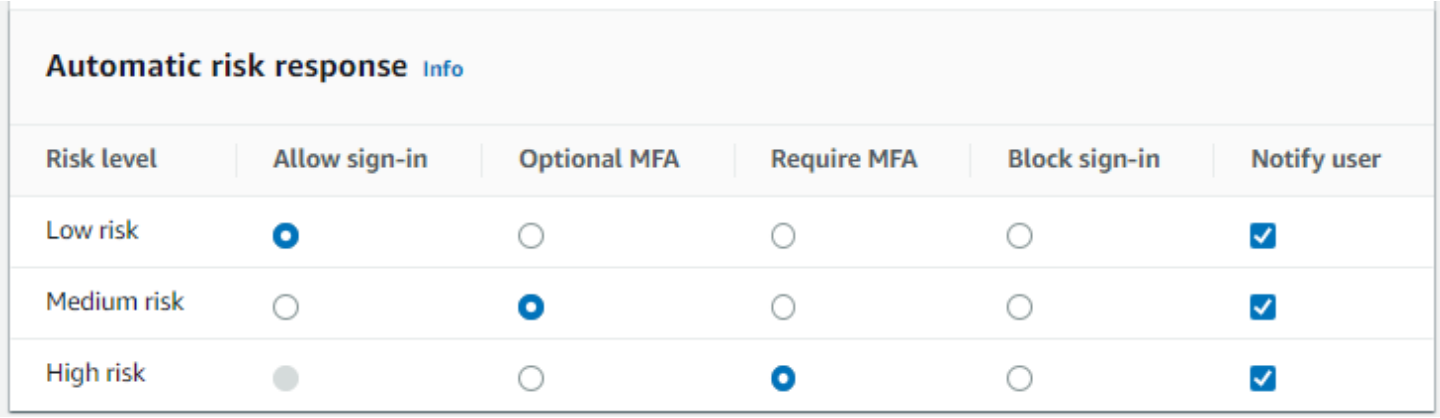
La consola enumera el historial de inicios de sesión en la pestaña Users and groups (Usuarios y grupos). Si selecciona una entrada, puede marcar el evento como válido o no válido. También puede enviar comentarios a través de la operación AdminUpdateAuthEventFeedbackde la API del grupo de usuarios y mediante el AWS CLI comando admin-update-auth-event -feedback.

Al seleccionar Set as valid (Establecer como válido) en la consola de Amazon Cognito o proporcionar un valor `FeedbackValue` de `valid` en la API, le está indicando a Amazon Cognito que confía en una sesión de usuario en la que Amazon Cognito ha determinado que hay cierto nivel de riesgo. Al seleccionar Set as invalid (Definir como no válido) en la consola de Amazon Cognito o proporcionar un valor `FeedbackValue` de `invalid` en la API, le está indicando a Amazon Cognito que no confía en una sesión de usuario o no cree que Amazon Cognito haya determinado que tiene un nivel de riesgo suficiente.

Envío de mensajes de notificación

Con protecciones de seguridad avanzadas, Amazon Cognito puede notificar a los usuarios los intentos de inicio de sesión de riesgo. Amazon Cognito también puede solicitar a los usuarios que seleccionen enlaces para indicar si el inicio de sesión es válido o no. Amazon Cognito utiliza estos comentarios para mejorar la precisión de la detección de riesgos de su grupo de usuarios.

En la sección Automatic risk response (Respuesta automática al riesgo) elija Notify Users (Notificar a los usuarios) para los casos de riesgo bajo, medio y alto.



Amazon Cognito envía notificaciones por correo electrónico a sus usuarios independientemente de si han verificado su dirección de correo electrónico.

Puede personalizar los mensajes de correo electrónico de notificación y proporcionar versiones de texto sin formato y HTML de dichos mensajes. Para personalizar las notificaciones por correo electrónico, abra Email templates (Plantillas de correo electrónico) desde Adaptive authentication messages (Mensajes de autenticación flexible) en su configuración de seguridad avanzada. Para obtener más información sobre las plantillas de correo electrónico, consulte Plantillas de mensaje.

Visualización de las métricas de seguridad avanzadas

Amazon Cognito publica métricas para funciones de seguridad avanzadas en su cuenta de Amazon. CloudWatch Amazon Cognito agrupa las métricas de seguridad avanzadas por nivel de riesgo y también por nivel de solicitud.

Para ver las métricas en la consola CloudWatch

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Metrics (Métricas).
3. Elija Amazon Cognito.
4. Elija un grupo de métricas agregadas, como By Risk Classification (Por clasificación de riesgos).
5. La pestaña All metrics (Todas las métricas) muestra todas las métricas para esa opción. Puede hacer lo siguiente:

   - Para ordenar la tabla, utilice el encabezado de columna.
   - Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella. Para seleccionar todas las métricas, seleccione la casilla de verificación en la fila de encabezado de la tabla.

- Para filtrar por recurso, elija el ID de recurso y, a continuación, elija Add to search (Añadir a la búsqueda).
- Para filtrar por métrica, elija el nombre de la métrica y, a continuación, elija Add to search (Añadir a la búsqueda).

Métrica	Descripción	Dimensiones de la métrica
CompromisedCredentialRisk	Solicitudes en las que Amazon Cognito detectó credenciales comprometidas.	Operation: tipo de operación. `PasswordChange` , `SignIn` o `SignUp` son las únicas dimensiones. UserPoolId: el identificador del grupo de usuarios. RiskLevel: alto (predeter minado), medio o bajo.
AccountTakeoverRisk	Solicitudes en las que Amazon Cognito detectó riesgo de usurpación de la cuenta.	Operation: tipo de operación. `PasswordChange` , `SignIn` o `SignUp` son las únicas dimensiones. UserPoolId: el identificador del grupo de usuarios. RiskLevel: alto, medio o bajo.
OverrideBlock	Solicitudes que Amazon Cognito bloqueó debido a la configuración que proporcionó el desarrollador.	Operation: tipo de operación. `PasswordChange` , `SignIn` o `SignUp` son las únicas dimensiones. UserPoolId: el identificador del grupo de usuarios. RiskLevel: alto, medio o bajo.

Métrica	Descripción	Dimensiones de la métrica
Riesgo	Solicitudes que Amazon Cognito marcó como riesgosas.	Operation: tipo de operación como, por ejemplo, `PasswordChange` , `SignIn` o `SignUp`. UserPoolId: el identificador del grupo de usuarios.
NoRisk	Solicitudes en las que Amazon Cognito no identificó ningún riesgo.	Operation: tipo de operación como, por ejemplo, `PasswordChange` , `SignIn` o `SignUp`. UserPoolId: El identificador del grupo de usuarios.

Amazon Cognito le ofrece dos grupos predefinidos de métricas para facilitar el análisis. CloudWatch By Risk Classification (Por clasificación de riesgo) identifica el grado de detalle del nivel de riesgo para las solicitudes que Amazon Cognito identifica como arriesgadas. By Request Classification (Por clasificación de solicitud) refleja las métricas agregadas por nivel de solicitud.

Grupo de métricas agregadas	Descripción
By Risk Classification	Solicitudes que Amazon Cognito identifica como arriesgadas.
By Request Classification	Métricas agregadas por solicitud.

Activación de la seguridad avanzada del grupo de usuarios desde la aplicación

Después de configurar las características de seguridad avanzadas para el grupo de usuarios, debe activarlas en la aplicación móvil o web.

Uso de seguridad avanzada con JavaScript

1. Añada el SDK de Amazon Cognito Identity JavaScript a su aplicación.

2. En CognitoUserPool.js, `AdvancedSecurityDataCollectionFlag` establézcalo en. `true` Establezca `UserPoolId` en el ID del grupo de usuarios.
3. Agrega esta referencia de origen al JavaScript archivo de tu aplicación. `<region>`Sustitúyala por una Región de AWS de la siguiente lista: `us-east-1` `us-east-2us-west-2`,`eu-west-1`,,`eu-west-2`, o`eu-central-1`.

```
<script src="https://amazon-cognito-assets.<region>.amazoncognito.com/amazon-cognito-advanced-security-data.min.js"></script>
```

Uso de seguridad avanzada con Android

1. Crea tu aplicación AWS Amplify para Android. Para obtener más información, consulte Configuración del proyecto en el AWS Amplify Dev Center.
2. Con `userContextDataProvider`, Incluya la información del usuario y del dispositivo en las solicitudes de autenticación.

Para obtener información sobre cómo agregar datos de contexto de usuario en el SDK de Android anterior, consulte aws-android-sdk-cognitoidentityprovider-asf.

Uso de seguridad avanzada con iOS

1. Crea tu aplicación AWS Amplify para Swift o Flutter. Para obtener más información, consulte Configuración de proyecto de Swift y Configuración de proyecto de Flutter en el AWS Amplify Dev Center.
2. Incluya la información del usuario y del dispositivo en las solicitudes de autenticación. Para ver un ejemplo para usarlo con la operación de la InitiateAuthAPI, consulta `userContextData` InitiateAuthInput+Amplify.swift on. GitHub

Para obtener información sobre cómo agregar datos de contexto de usuario en el SDK de iOS anterior, consulte AWSCognitoIdentityProviderASF.

Asociar una ACL AWS WAF web a un grupo de usuarios

AWS WAF es un firewall de aplicaciones web. Con una lista de control de acceso AWS WAF web (ACL web), puede proteger su grupo de usuarios de solicitudes no deseadas a su interfaz de usuario alojada y a los puntos de enlace del servicio de la API de Amazon Cognito. Una ACL web le

proporciona un control detallado sobre todas las solicitudes web HTTPS a las que responde el grupo de usuarios. Para obtener más información sobre las ACL AWS WAF web, consulte Administración y uso de una lista de control de acceso web (ACL web) en la AWS WAF Guía para desarrolladores.

Cuando tiene una ACL AWS WAF web asociada a un grupo de usuarios, Amazon Cognito reenvía los encabezados no confidenciales seleccionados y el contenido de las solicitudes de sus usuarios a. AWS WAF AWS WAF inspecciona el contenido de la solicitud, la compara con las reglas que especificó en su ACL web y devuelve una respuesta a Amazon Cognito.

Lo que debe saber sobre las ACL AWS WAF web y Amazon Cognito

- Las solicitudes bloqueadas por AWS WAF no se incluyen en la cuota de solicitudes de ningún tipo de solicitud. Se llama al AWS WAF controlador antes que a los controladores de regulación a nivel de API.
- Al crear una ACL web, pasa una pequeña cantidad de tiempo antes de que la ACL web se haya propagado por completo y esté disponible para Amazon Cognito. El tiempo de propagación puede oscilar entre unos segundos y varios minutos. AWS WAF devuelve a `WAFUnavailableEntityException`cuando intenta asociar una ACL web antes de que se haya propagado por completo.
- Puede asociar una ACL web con un grupo de usuarios.
- Es posible que la solicitud dé lugar a una carga útil superior a los límites de lo que AWS WAF puede inspeccionar. Consulte Gestión de componentes de solicitudes de gran tamaño en la Guía para AWS WAF desarrolladores para obtener información sobre cómo configurar el modo en que AWS WAF gestiona las solicitudes de gran tamaño de Amazon Cognito.
- No puede asociar una ACL web que utilice la prevención de apropiación de cuentas (ATP) de AWS WAF Fraud Control con un grupo de usuarios de Amazon Cognito. Debe implementar la característica ATP cuando agregue el grupo de reglas administrado `AWS-AWSManagedRulesATPRuleSet`. Antes de asociarlo a un grupo de usuarios, asegúrese de que la ACL web no utilice este grupo de reglas administrado.
- Si tiene una ACL AWS WAF web asociada a un grupo de usuarios y una regla de su ACL web presenta un CAPTCHA, esto puede provocar un error irrecuperable en el registro del TOTP de la interfaz de usuario alojada. Para crear una regla que tenga una acción CAPTCHA y no afecte al TOTP de la IU alojada, consulte Configuración de la ACL AWS WAF web para la interfaz de usuario alojada en el MFA.

AWS WAF inspecciona las solicitudes a los siguientes puntos finales.

IU alojada

Solicitudes a todos los puntos de conexión en Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada.

Operaciones de la API públicas

Solicitudes de su aplicación a la API de Amazon Cognito que no utilizan AWS credenciales para autorizar. Esto incluye operaciones de API como InitiateAuthRespondToAuthChallenge, y GetUser. Las operaciones de la API que están dentro del ámbito de aplicación AWS WAF no requieren autenticación con AWS credenciales. No están autenticadas o están autorizadas con una cadena de sesión o un token de acceso. Para obtener más información, consulte Operaciones de API autenticadas y no autenticadas de los grupos de usuarios de Amazon Cognito.

Puede configurar las reglas en la ACL web con acciones de reglas como Count (Recuento), Allow (Permiso), Block (Bloque) o presentar un CAPTCHA en respuesta a una solicitud que coincide con una regla. Para obtener más información, consulte Reglas de AWS WAF en la Guía para desarrolladores de AWS WAF . Según la acción de la regla, puede personalizar la respuesta que Amazon Cognito devuelve a los usuarios.

Important

Las opciones para personalizar la respuesta de error dependen de la forma en que realice una solicitud a la API.

- Puede personalizar el código de error y el cuerpo de respuesta de las solicitudes de IU alojadas. Solo puede presentar un CAPTCHA para que el usuario lo resuelva en la IU alojada.
- Para las solicitudes que realice con la API de grupos de usuarios de Amazon Cognito, puede personalizar el cuerpo de la respuesta de una solicitud que recibe una respuesta de Bloque. También puede especificar un código de error personalizado en el intervalo de 400 a 499.
- Los AWS Command Line Interface (AWS CLI) y los AWS SDK devuelven un `ForbiddenException` error a las solicitudes que generan una respuesta de bloqueo o CAPTCHA.

Asociación de una ACL web con un grupo de usuarios

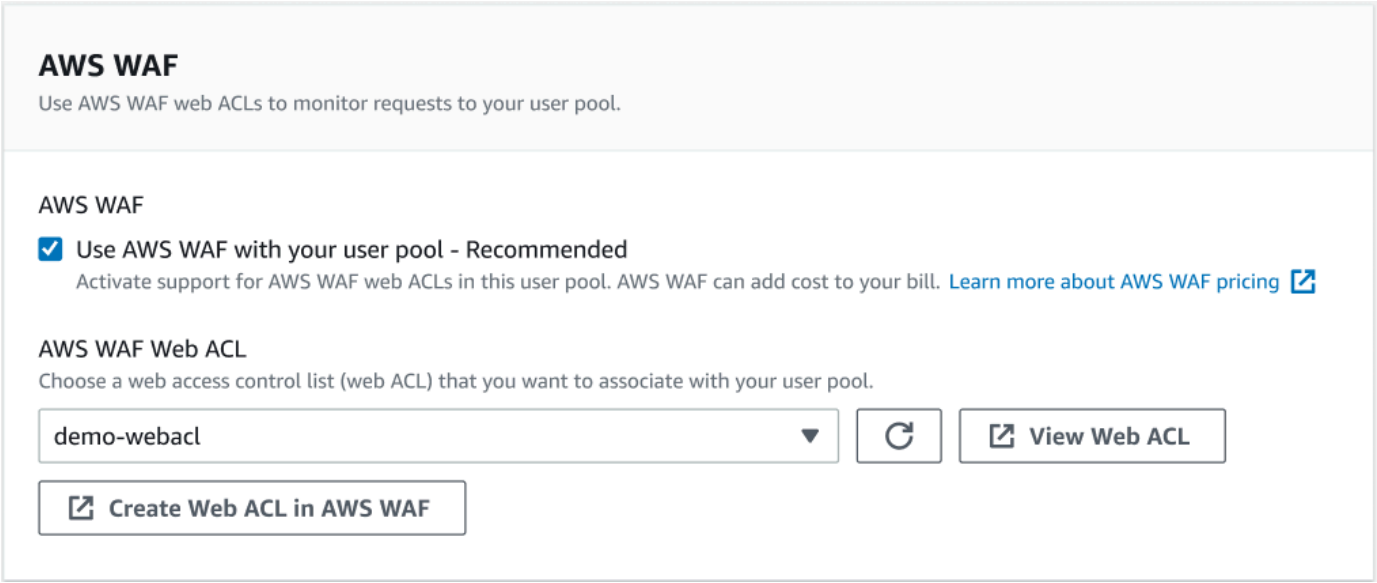
Para trabajar con una ACL web en su grupo de usuarios, su director AWS Identity and Access Management (IAM) debe tener los siguientes permisos de Amazon Cognito. Para obtener información sobre AWS WAF los permisos, consulte los permisos de la AWS WAF API en la Guía AWS WAF para desarrolladores.

- `cognito-idp:AssociateWebACL`
- `cognito-idp:DisassociateWebACL`
- `cognito-idp:GetWebACLForResource`
- `cognito-idp:ListResourcesForWebACL`

Si bien debe conceder permisos de IAM, las acciones enumeradas son solo con permisos y no corresponden a una operación de la API.

Para activarlos AWS WAF para su grupo de usuarios y asociar una ACL web

1. Inicie sesión en la consola de Amazon Cognito.
2. En el panel de navegación, elija User Pools (Grupos de usuarios) y elija el grupo de usuarios que desea editar.
3. Elija la pestaña User pool properties (Propiedades del grupo de usuarios).
4. Elija Edit (Editar) junto a AWS WAF.
5. En AWS WAF, seleccione Usar AWS WAF con su grupo de usuarios.



6. Elija una ACL AWS WAF web que ya haya creado o elija Crear ACL web en AWS WAF para crear una en una nueva AWS WAF sesión del AWS Management Console.
7. Elija Guardar cambios.

Para asociar mediante programación una ACL web a su grupo de usuarios en el SDK AWS Command Line Interface o en un SDK, utilice la AssociateWebACL de la AWS WAF API. Amazon Cognito no tiene una operación de API independiente que asocie una ACL web.

Probar y registrar las ACL web AWS WAF

Al configurar una acción de regla como Recuento en su ACL web, AWS WAF agrega la solicitud a un recuento de solicitudes que coinciden con la regla. Para probar una ACL web con el grupo de usuarios, establezca las acciones de reglas para Count (Recuento) y tenga en cuenta el volumen de solicitudes que coinciden con cada regla. Por ejemplo, si una regla que desea establecer en una acción de Block (Bloque) coincide con un gran número de solicitudes que usted determina que son tráfico de usuario normal, es posible que tenga que volver a configurar la regla. Para obtener más información, consulte Probar y ajustar sus AWS WAF protecciones en la Guía para AWS WAF desarrolladores.

También puede configurarlo AWS WAF para registrar los encabezados de las solicitudes en un grupo de CloudWatch registros de Amazon Logs, un bucket de Amazon Simple Storage Service (Amazon S3) o un Amazon Data Firehose. Puede identificar las solicitudes de Amazon Cognito que realiza con la API de grupos de usuarios mediante `x-amzn-cognito-client-id` y `x-amzn-cognito-operation-name`. Las solicitudes de IU alojada solo incluyen el encabezado `x-amzn-cognito-client-id`. Para obtener más información, consulte Registro del tráfico de la ACL web en la Guía para desarrolladores de AWS WAF .

AWS WAF Las ACL web no están sujetas a los precios de las funciones de seguridad avanzadas de Amazon Cognito. Las funciones de seguridad AWS WAF complementan las funciones de seguridad avanzadas de Amazon Cognito. Puede activar ambas funciones en un grupo de usuarios. AWS WAF factura por separado la inspección de las solicitudes del grupo de usuarios. Para obtener más información, consulte AWS WAF Precios.

El registro de los datos de las AWS WAF solicitudes está sujeto a una facturación adicional por parte del servicio al que dirijas tus registros. Para obtener más información, consulte Precios para registrar información de tráfico de ACL web en la Guía para desarrolladores de AWS WAF .

Sensibilidad de mayúsculas y minúsculas en el grupo de usuarios

De forma predeterminada, los grupos de usuarios de Amazon Cognito que cree en ellos AWS Management Console no distinguen mayúsculas de minúsculas. Cuando un grupo de usuarios no distingue entre mayúsculas y minúsculas, user@example.com y User@example.com hacen referencia al mismo usuario. Cuando los nombres de usuario de un grupo de usuarios no distinguen entre mayúsculas y minúsculas, los atributos `preferred_username` y `email` tampoco distinguen entre mayúsculas y minúsculas.

Para tener en cuenta la configuración de distinción entre mayúsculas y minúsculas del grupo de usuarios, identifique a los usuarios en el código de la aplicación en función del atributo de usuario alternativo. Porque el caso de un nombre de usuario, un nombre de usuario preferido o un atributo de dirección de correo electrónico puede variar en diferentes perfiles de usuario, se recomienda que haga referencia al atributo `sub` en su lugar. También puede crear un atributo personalizado inmutable en su grupo de usuarios y asignar su propio valor de identificador único al atributo en cada nuevo perfil de usuario. Al crear un usuario por primera vez, puede escribir un valor en un atributo personalizado inmutable que haya creado.

> **Note**
>
> Independientemente de la configuración de distinción entre mayúsculas y minúsculas de su grupo de usuarios, Amazon Cognito requiere que un usuario federado de un proveedor de identidades (IdP) SAML u OIDC pase una única notificación `NameId` o `sub` única con distinción entre mayúsculas y minúsculas. Para obtener más información sobre el identificador único que distingue entre mayúsculas y minúsculas y el SAML IdPs, consulte. Uso del inicio de sesión SAML iniciado por SPI

Creación de un grupo de usuarios que distingue entre minúsculas y mayúsculas

Si crea recursos con AWS Command Line Interface (AWS CLI) y operaciones de API como CreateUserPool, por ejemplo, debe establecer el parámetro booleano `CaseSensitive` en. `false` Esta configuración crea un grupo de usuarios sin distinción entre mayúsculas y minúsculas Si no especifica ningún valor, la `CaseSensitive` utiliza `true` de forma predeterminada. Esto es lo contrario al comportamiento predeterminado de los grupos de usuarios que se crean en AWS Management Console. Antes del 12 de febrero de 2020, los grupos de usuarios distinguían entre mayúsculas y minúsculas de forma predeterminada, independientemente de la plataforma.

Puedes usar la pestaña Experiencia de inicio de sesión de la operación AWS Management Console o de la DescribeUserPoolAPI para revisar la configuración de distinción entre mayúsculas y minúsculas de cada grupo de usuarios de tu cuenta.

Migrar a un nuevo grupo de usuarios

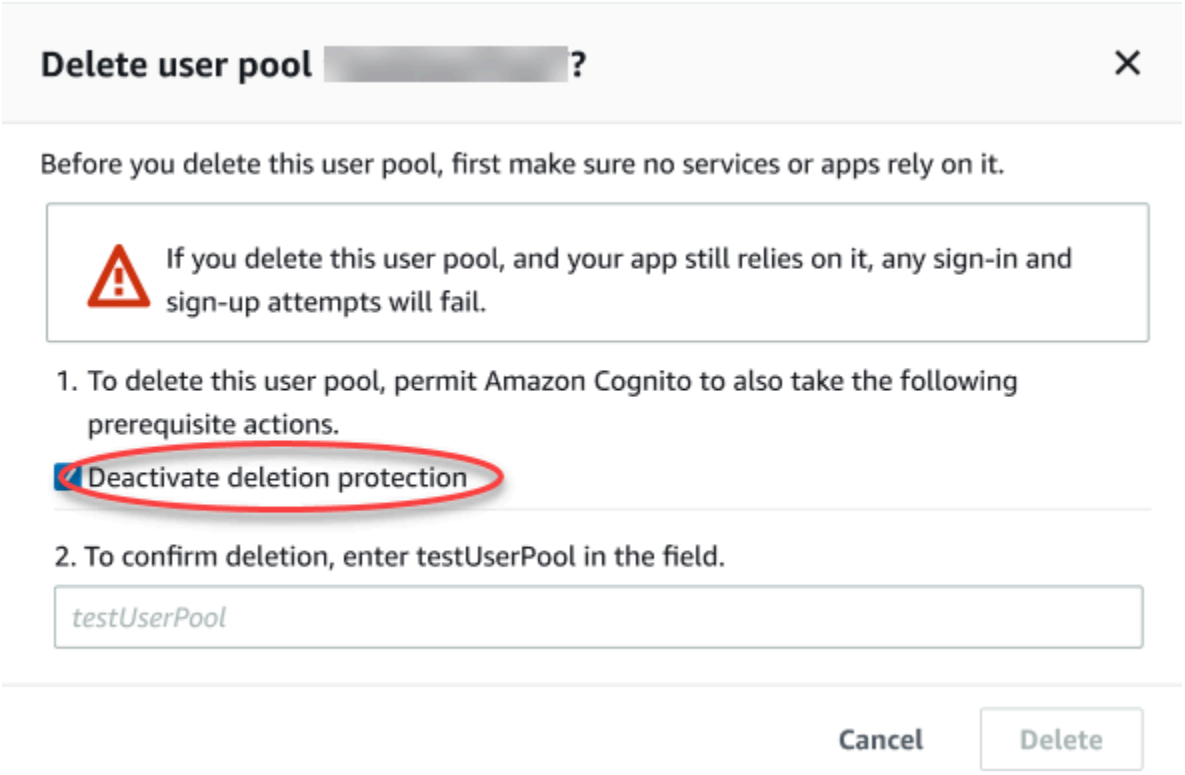
Debido a los posibles conflictos entre los perfiles de usuario, no se puede cambiar un grupo de usuarios de Amazon Cognito de una configuración que distingue entre mayúsculas y minúsculas a una que no hace la distinción. En su lugar, se deben migrar los usuarios a un nuevo grupo de usuarios. Debe crear código de migración para resolver conflictos relacionados con la distinción entre mayúsculas y minúsculas. Este código debe devolver un nuevo usuario único o rechazar el intento de inicio de sesión cuando detecta un conflicto. En un nuevo grupo de usuarios que no distingue entre mayúsculas y minúsculas, asigne un Migración del desencadenador de Lambda del usuario. La AWS Lambda función puede crear usuarios en el nuevo grupo de usuarios que no distingue entre mayúsculas y minúsculas. Cuando el usuario no logra iniciar sesión con el grupo de usuarios que no distingue entre mayúsculas y minúsculas, la función de Lambda busca y duplica al usuario desde el grupo de usuarios que distingue entre mayúsculas y minúsculas. También puede activar un activador Lambda de migración de usuarios en ForgotPasswordlos eventos. Amazon Cognito transfiere información de usuario y metadatos de eventos de la acción de inicio de sesión o de recuperación de contraseña a la función Lambda. Puede utilizar los datos de eventos para administrar conflictos entre nombres de usuario y direcciones de correo electrónico cuando la función cree el nuevo usuario en el grupo de usuarios que no distingue entre mayúsculas y minúsculas. Estos conflictos se producen entre nombres de usuario y direcciones de correo electrónico que serían únicos en un grupo de usuarios que no distingue entre mayúsculas y minúsculas, pero idénticos en un grupo de usuarios que sí distingue entre mayúsculas y minúsculas.

Para obtener más información sobre cómo utilizar un activador Lambda de migración de usuarios entre grupos de usuarios de Amazon Cognito, consulte Migración de usuarios a grupos de usuarios de Amazon Cognito en el blog. AWS

Protección de eliminación de grupo de usuarios

Para que los administradores no eliminen accidentalmente el grupo de usuarios, active la protección de eliminación. Con la protección de eliminación activa, debe confirmar que desea eliminar el grupo de usuarios antes de eliminarlo. Al eliminar un grupo de usuarios en la AWS Management Console, puede desactivar la protección de eliminación al mismo tiempo. Cuando acepta la solicitud para

desactivar la protección de eliminación y confirma su intención de eliminarla, como se muestra en la siguiente imagen, Amazon Cognito elimina el grupo de usuarios.



Si desea eliminar un grupo de usuarios con una solicitud a la API de Amazon Cognito, primero debe cambiar `DeletionProtection` a `Inactive` en una solicitud de UpdateUserPool. Si no desactiva la protección de eliminación, Amazon Cognito devuelve un error `InvalidParameterException`. Después de desactivar la protección de eliminación, puede eliminar el grupo de usuarios en una solicitud de DeleteUserPool.

Amazon Cognito activa Deletion protection (Protección de eliminación) de forma predeterminada al crear un grupo de usuarios nuevo en la AWS Management Console. Al crear un grupo de usuarios con la API `CreateUserPool`, la protección de eliminación está inactiva de forma predeterminada. Para utilizar esta función en los grupos de usuarios que cree con la AWS CLI o con un SDK de AWS, defina el parámetro `DeletionProtection` como `True`.

Puede activar o desactivar el estado de la protección de eliminación en el contenedor de Deletion protection (Protección de eliminación) de la pestaña de User pool settings (Configuración del grupo de usuarios) de la consola de Amazon Cognito.

Para configurar la protección de eliminación

1. Diríjase a la consola de Amazon Cognito. Es posible que se le soliciten sus credenciales de AWS.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña User pool settings (Configuración del grupo de usuarios). Busque Deletion Protection (Protección de eliminación) y seleccione Active (Activar) o Deactivate (Desactivar).
5. Confirme su elección en el siguiente cuadro de diálogo.

Administración de las respuestas de error de existencia de usuarios

Amazon Cognito es compatible con la personalización de las respuestas de error que devuelven los grupos de usuarios. Existen respuestas de error personalizadas para las operaciones de creación y autenticación de usuarios, recuperación de contraseñas y confirmación.

Utilice la configuración de `PreventUserExistenceErrors` de un cliente de aplicaciones del grupo de usuarios para habilitar o desactivar errores relacionados con la existencia del usuario. Cuando crea un nuevo cliente de aplicaciones con la API de grupos de usuarios de Amazon Cognito`LEGACY`, `PreventUserExistenceErrors` está o deshabilitado de forma predeterminada. En la consola de Amazon Cognito, la opción Evitar errores de existencia del usuario (un ajuste de `ENABLED` for`PreventUserExistenceErrors`) está seleccionada de forma predeterminada. Para actualizar la `PreventUserExistenceErrors` configuración, realice una de las siguientes acciones:

- Cambia el valor `PreventUserExistenceErrors` entre `ENABLED` y `LEGACY` en una solicitud UpdateUserPoolClientde API.
- Edite el cliente de la aplicación en la consola de Amazon Cognito y cambie el estado de Evitar errores de existencia de usuarios entre selected (`ENABLED`) y deselected (). `LEGACY`

Si esta propiedad tiene un valor de`LEGACY`, el cliente de la aplicación devuelve una respuesta de `UserNotFoundException` error cuando un usuario intenta iniciar sesión con un nombre de usuario que no existe en su grupo de usuarios.

Cuando esta propiedad tiene un valor de`ENABLED`, el cliente de la aplicación no revela con un `UserNotFoundException` error la inexistencia de una cuenta de usuario en el grupo de usuarios. Una `PreventUserExistenceErrors` configuración de `ENABLED` tiene los siguientes efectos:

- Amazon Cognito responde con información no específica a las solicitudes de API cuando, de lo contrario, su respuesta podría revelar la existencia de un usuario válido.
- Las API de inicio de sesión y contraseña olvidada de Amazon Cognito devuelven una respuesta genérica a un error de autenticación. Con la respuesta de error, se indica que el nombre de usuario o la contraseña es incorrecto.
- Las API de confirmación de cuenta y recuperación de contraseñas de Amazon Cognito devuelven una respuesta que indica que se envió un código a un medio de entrega simulado, en lugar de una representación parcial de la información de contacto del usuario.

La siguiente información detalla el comportamiento de las operaciones del grupo de usuarios cuando `PreventUserExistenceErrors` está configurado en. `ENABLED`

Operaciones de autenticación y creación de usuarios

Puede configurar las respuestas de error tanto en la autenticación de nombre de usuario y contraseña como en la de contraseña remota segura (SRP). También puede personalizar los errores que devuelve con la autenticación personalizada. Las siguientes API realizan estas operaciones de autenticación:

- `AdminInitiateAuth`
- `AdminRespondToAuthChallenge`
- `InitiateAuth`
- `RespondToAuthChallenge`

En la siguiente lista se muestra cómo puede personalizar las respuestas de error en las operaciones de autenticación de usuarios.

Autenticación de nombre de usuario y contraseña

Para iniciar la sesión de un usuario con `ADMIN_USER_PASSWORD_AUTH` y `USER_PASSWORD_AUTH`, incluya el nombre de usuario y la contraseña en una solicitud de API `AdminInitiateAuth` o `InitiateAuth`. Amazon Cognito devuelve un error `NotAuthorizedException` genérico en el que se indica que el nombre de usuario o la contraseña no son correctos.

Autenticación basada en contraseña remota segura (SRP)

Para iniciar la sesión de un usuario con `USER_SRP_AUTH`, incluya un nombre de usuario y un parámetro `SRP_A` en una solicitud de API `AdminInitiateAuth` o `InitiateAuth`. En respuesta, Amazon Cognito devuelve `SRP_B` y sal para el usuario. Cuando no se encuentra ningún usuario, Amazon Cognito devuelve una respuesta simulada en el primer paso, tal y como se describe en RFC 5054. Amazon Cognito devuelve la misma "sal" y un ID de usuario interno en formato de identificador único universal (UUID) para la misma combinación de nombre de usuario y grupo de usuarios. Al enviar una solicitud de API `RespondToAuthChallenge` con prueba de contraseña, Amazon Cognito devuelve un error genérico `NotAuthorizedException` cuando el nombre de usuario o la contraseña son incorrectos.

> **Note**
>
> Puede simular una respuesta genérica con autenticación de nombre de usuario y contraseña si utiliza atributos de alias basados en la verificación y el nombre de usuario inmutable no tiene formato de UUID.

Desencadenador de Lambda de desafío de autenticación personalizado

Si utiliza el desencadenador de Lambda de desafío de autenticación personalizado y habilita las respuestas de error, `LambdaChallenge` devuelve un parámetro booleano llamado `UserNotFound`. A continuación se pasa en la solicitud de los desencadenadores de Lambda `DefineAuthChallenge`, `VerifyAuthChallenge`, y `CreateAuthChallenge`. Puede utilizar este desencadenador a fin de simular desafíos de autenticación personalizados para un usuario que no exista. Si llama al desencadenador de Lambda de autenticación previa para un usuario que no existe, Amazon Cognito devuelve `UserNotFound`.

En la siguiente lista se muestra cómo puede personalizar las respuestas de error en las operaciones de creación de usuarios.

SignUp

La `SignUp` operación siempre `UsernameExistsException` se devuelve cuando ya se ha utilizado un nombre de usuario. Si no desea que Amazon Cognito devuelva un error `UsernameExistsException` para las direcciones de correo electrónico y los números de teléfono cuando registre usuarios en su aplicación, utilice atributos de alias basados en

verificación. Para obtener más información acerca de los alias, consulte Personalización de los atributos de inicio de sesión.

Para obtener un ejemplo de cómo Amazon Cognito puede evitar que se utilicen las solicitudes de API `SignUp` para descubrir usuarios en su grupo de usuarios, consulte Prevención de errores `UsernameExistsException` en las direcciones de correo electrónico y los números de teléfono al registrarse.

Usuarios importados

Durante la autenticación de los usuarios importados, si se habilita `PreventUserExistenceErrors`, se devuelve un error `NotAuthorizedException` genérico en el que se indica que el nombre de usuario o la contraseña eran incorrectos, en lugar de devolver `PasswordResetRequiredException`. Para obtener más información, consulte Requisito para que los usuarios importados restablezcan sus contraseñas.

Migración del desencadenador de Lambda del usuario

Amazon Cognito devolverá una respuesta simulada para los usuarios que no existan cuando el desencadenador de Lambda establezca una respuesta vacía en el contexto del evento original. Para obtener más información, consulte Migración del desencadenador de Lambda del usuario.

Prevención de errores **`UsernameExistsException`** en las direcciones de correo electrónico y los números de teléfono al registrarse

En el siguiente ejemplo se demuestra cómo, al configurar los atributos de alias en su grupo de usuarios, puede evitar que las direcciones de correo electrónico y los números de teléfono duplicados generen errores `UsernameExistsException` en respuesta a las solicitudes de API `SignUp`. Debe haber creado su grupo de usuarios con la dirección de correo electrónico o el número de teléfono como atributo de alias. Para obtener más información, consulte la sección Customizing sign-in attributes (Personalización de atributos de inicio de sesión) de User pool attributes (Atributos de grupo de usuarios).

1. Jie se registra para obtener un nuevo nombre de usuario y también proporciona la dirección de correo electrónico `jie@example.com`. Amazon Cognito envía un código a su dirección de correo electrónico.

   Ejemplo de AWS CLI comando

```
aws cognito-idp sign-up --client-id 1234567890abcdef0 --username jie --password
 PASSWORD --user-attributes Name="email",Value="jie@example.com"
```

Ejemplo de respuesta

```
{
    "UserConfirmed": false,
    "UserSub": "<subId>",
    "CodeDeliveryDetails": {
        "AttributeName": "email",
        "Destination": "j****@e****",
        "DeliveryMedium": "EMAIL"
    }
}
```

2. Jie proporciona el código que se le envió para confirmar su propiedad de la dirección de correo electrónico. Esto completa su registro como usuario.

   Ejemplo de AWS CLI comando

   ```
   aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=jie --
   confirmation-code xxxxxx
   ```

3. Shirley registra una nueva cuenta de usuario y proporciona la dirección de correo electrónico `jie@example.com`. Amazon Cognito no devuelve ningún error `UsernameExistsException` y envía un código de confirmación a la dirección de correo electrónico de Jie.

   Ejemplo de AWS CLI comando

   ```
   aws cognito-idp sign-up --client-id 1234567890abcdef0 --username shirley --password
    PASSWORD --user-attributes Name="email",Value="jie@example.com"
   ```

   Ejemplo de respuesta

   ```
   {
       "UserConfirmed": false,
       "UserSub": "<new subId>",
       "CodeDeliveryDetails": {
           "AttributeName": "email",
           "Destination": "j****@e****",
   ```

```
        "DeliveryMedium": "EMAIL"
    }
}
```

4. En un escenario diferente, Shirley tiene la propiedad de `jie@example.com`. Shirley recupera el código que Amazon Cognito envió a la dirección de correo electrónico de Jie e intenta confirmar la cuenta.

   Ejemplo de AWS CLI comando

   ```
   aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=shirley --
   confirmation-code xxxxxx
   ```

   Ejemplo de respuesta

   ```
   An error occurred (AliasExistsException) when calling the ConfirmSignUp operation: An
    account with the email already exists.
   ```

Amazon Cognito no devuelve un error a la solicitud de `aws cognito-idp sign-up` de Shirley, a pesar de que `jie@example.com` se asigne a un usuario existente. Shirley debe demostrar la propiedad de la dirección de correo electrónico antes de que Amazon Cognito devuelva una respuesta de error. En un grupo de usuarios con atributos de alias, este comportamiento impide utilizar la API `SignUp` pública para comprobar si existe un usuario con una dirección de correo electrónico o un número de teléfono determinados.

Este comportamiento es diferente de la respuesta que Amazon Cognito devuelve a la solicitud `SignUp` con un nombre de usuario existente, como se muestra en el siguiente ejemplo. Aunque Shirley sabe por esta respuesta que ya existe un usuario con el nombre de usuario `jie`, no conoce ninguna dirección de correo electrónico o número de teléfono asociados al usuario.

Comando de la CLI de ejemplo

```
aws cognito-idp sign-up --client-id 1example23456789 --username jie --password PASSWORD
      --user-attributes Name="email",Value="shirley@example.com"
```

Ejemplo de respuesta

```
An error occurred (UsernameExistsException) when calling the SignUp operation: User
 already exists
```

Operaciones de restablecimiento de contraseña

Amazon Cognito devuelve las siguientes respuestas a las operaciones de restablecimiento de la contraseña del usuario cuando se evitan los errores de existencia de usuarios.

ForgotPassword

Cuando un usuario no se encuentra, está desactivado o no dispone de un mecanismo de entrega verificado para recuperar su contraseña, Amazon Cognito siempre devuelve `CodeDeliveryDetails` con un medio de entrega simulado para un usuario. El medio de entrega simulado vendrá determinado por el formato del nombre de usuario de entrada y la configuración de verificación del grupo de usuarios.

ConfirmForgotPassword

Amazon Cognito devuelve el error `CodeMismatchException` para los usuarios que no existen o que están inhabilitados. Si no se solicita un código al utilizar `ForgotPassword`, Amazon Cognito devuelve el error `ExpiredCodeException`.

Operaciones de confirmación

Amazon Cognito devuelve las siguientes respuestas a las operaciones de confirmación y verificación de usuarios cuando se evitan los errores de existencia de usuarios.

ResendConfirmationCode

Amazon Cognito devuelve `CodeDeliveryDetails` para un usuario inhabilitado o que no existe. Amazon Cognito envía un código de confirmación al correo electrónico o al número de teléfono del usuario existente.

ConfirmSignUp

Se devuelve `ExpiredCodeException` si un código se ha vencido. Amazon Cognito devuelve `NotAuthorizedException` cuando un usuario no está autorizado. Si el código no coincide con lo que el servidor espera, Amazon Cognito devuelve `CodeMismatchException`.

Grupos de identidades de Amazon Cognito

Un grupo de identidades de Amazon Cognito es un directorio de identidades federadas que puede intercambiar por credenciales de AWS . Los grupos de identidades generan AWS credenciales temporales para los usuarios de tu aplicación, tanto si han iniciado sesión como si aún no los has identificado. Con las funciones y políticas AWS Identity and Access Management (de IAM), puedes elegir el nivel de permiso que quieres conceder a tus usuarios. Los usuarios pueden empezar como invitados y recuperar los activos que mantiene en Servicios de AWS. A continuación, pueden iniciar sesión con un proveedor de identidades de terceros para desbloquear el acceso a los activos que pone a disposición de los miembros registrados. El proveedor de identidades de terceros puede ser un proveedor de OAuth 2.0 de consumo (social) como Apple o Google, un proveedor de identidades SAML u OIDC personalizado o un esquema de autenticación personalizado, también denominado proveedor de desarrolladores, diseñado por usted mismo.

Características de los grupos de identidades de Amazon Cognito

Firma las solicitudes de Servicios de AWS

> Firme solicitudes de API Servicios de AWS como Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB. Analice la actividad de los usuarios con servicios como Amazon Pinpoint y Amazon. CloudWatch

Filtrar las solicitudes con políticas basadas en recursos

> Ejerza un control detallado sobre el acceso de los usuarios a los recursos. Transforme las reclamaciones de los usuarios en Etiquetas de sesión de IAM y cree políticas de IAM que concedan acceso a los recursos a distintos subconjuntos de los usuarios.

Asignar acceso como invitado

> Para los usuarios que aún no hayan iniciado sesión, configure el grupo de identidades para generar credenciales de AWS con un alcance de acceso limitado. Autentique a los usuarios mediante un único proveedor de inicio de sesión para aumentar el acceso.

Asignar roles de IAM en función de las características del usuario

> Asigne un solo rol de IAM a todos los usuarios autenticados o elija el rol en función de las reclamaciones de cada usuario.

Aceptar una variedad de proveedores de identidad

Cambie un identificador o un token de acceso, un token de grupo de usuarios, una afirmación de SAML o un token de OAuth de un proveedor social por credenciales. AWS

Validar las propias identidades

Realiza tu propia validación de usuario y usa tus credenciales de desarrollador para emitir AWS credenciales para tus usuarios.

Es posible que ya disponga de un grupo de usuarios de Amazon Cognito que proporcione servicios de autenticación y autorización para la aplicación. Puede configurar el grupo de usuarios como proveedor de identidades (IdP) para el grupo de identidades. Cuando lo haga, sus usuarios podrán autenticarse a través de su grupo de usuarios IdPs, consolidar sus afirmaciones en un token de identidad común del OIDC e intercambiar ese token por credenciales. AWS A continuación, el usuario puede presentar las credenciales en una solicitud firmada dirigida a los Servicios de AWS.

También puede presentar las reclamaciones autenticadas de cualquiera de los proveedores de identidad directamente al grupo de identidades. Amazon Cognito personaliza las reclamaciones de los usuarios de los proveedores de SAML, OAuth y OIDC en una solicitud de API para credenciales a corto plazo. AssumeRoleWithWebIdentity

Los grupos de usuarios de Amazon Cognito son como los proveedores de identidades de OIDC para las aplicaciones habilitadas para SSO. Los grupos de identidades actúan como un proveedor de identidades de AWS para cualquier aplicación cuyas dependencias de recursos funcionen mejor con la autorización de IAM.

Los grupos de identidades de Amazon Cognito admiten los siguientes proveedores de identidad:

- Proveedores públicos: Configuración de Login with Amazon como un IdP de grupos de identidades, Configurar Facebook como un IdP de grupos de identidades, Configurar Google como un IdP de grupo de identidades, Configurar el inicio de sesión con Apple como un IdP de grupo de identidades, Twitter.
- Grupos de usuarios de Amazon Cognito
- Configuración de un proveedor de OIDC como un IdP de grupo de identidades
- Configurar un proveedor de SAML como un IdP de grupo de identidades
- Identidades autenticadas por el desarrollador (grupos de identidades)

Para obtener información sobre la disponibilidad regional de los grupos de identidades de Amazon Cognito, consulte Disponibilidad regional del servicio de AWS.

Para obtener más información sobre los grupos de identidades de Amazon Cognito, consulte los siguientes temas.

Temas

Uso de grupos de identidades (identidades federadas)

Los grupos de identidades de Amazon Cognito proporcionan AWS credenciales temporales para los usuarios que son invitados (sin autenticar) y para los usuarios que se han autenticado y han recibido un token. Un grupo de identidades es un almacén de datos de identidades de usuarios específicos de su cuenta.

Para crear un grupo de identidades nuevo en la consola

1. Inicie sesión en la consola de Amazon Cognito y seleccione Grupos de identidades.
2. Elija Crear grupo de identidades.
3. En Configurar confianza de grupo de identidades, elija configurar el grupo de identidades para el acceso autenticado, el acceso de invitado o ambos.

    - Si elige Acceso autenticado, seleccione uno o más tipos de identidades que desee establecer como origen de identidades autenticadas en el grupo de identidades. Si configura un Proveedor de desarrolladores personalizado, no podrá modificarlo ni eliminarlo después de crear el grupo de identidades.

4. En Configurar permisos, elija un rol de IAM predeterminado para los usuarios autenticados o invitados del grupo de identidades.

   a. Elija Crear un nuevo rol de IAM si desea que Amazon Cognito cree uno nuevo para usted con permisos básicos y una relación de confianza con el grupo de identidades. Ingrese un Nombre de rol de IAM para identificar el nuevo rol, por ejemplo `myidentitypool_authenticatedrole`. Seleccione Ver documento de política para revisar los permisos que Amazon Cognito asignará al nuevo rol de IAM.

   b. Puede optar por utilizar una función de IAM existente si ya tiene una función Cuenta de AWS que desee utilizar. Debe configurar la política de confianza del rol de IAM para incluir `cognito-identity.amazonaws.com`. Configure la política de confianza del rol para que solo permita que Amazon Cognito asuma el rol cuando presente pruebas de que la solicitud proviene de un usuario autenticado del grupo de identidades específico. Para obtener más información, consulte Confianza y permisos de rol.

5. En Connect Identity Providers, introduzca los detalles de los proveedores de identidad (IdPs) que eligió en Configurar la confianza del grupo de identidades. Es posible que se le pida que proporcione información del cliente de la aplicación OAuth, elija un grupo de usuarios de Amazon Cognito, elija un IdP de IAM o ingrese un identificador personalizado para un proveedor de desarrolladores.

   a. Elija la Configuración del rol para cada IdP. Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas. Con un IdP del grupo de usuarios de Amazon Cognito, también puede Elegir un rol con preferred_role en los tokens. Para obtener más información acerca de la reclamación de `cognito:preferred_role`, consulte Asignación de valores de prioridad a los grupos.

      i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

      ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

   b. Configure Atributos para el control de acceso para cada IdP. Los atributos del control de acceso asignan las reclamaciones de los usuarios a las Etiquetas de las entidades principales que Amazon Cognito aplica a la sesión temporal. Puede crear políticas de IAM para filtrar el acceso de los usuarios en función de las etiquetas que aplique a la sesión.

i. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

ii. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

iii. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

6. En Configurar propiedades, ingrese un Nombre en Nombre de grupo de identidades.
7. En Autenticación básica (clásica), elija si desea Activar el flujo básico. Con el flujo básico activo, puede omitir las funciones que ha seleccionado para usted IdPs y llamar AssumeRoleWithWebIdentitydirectamente. Para obtener más información, consulte Flujo de autenticación de grupos de identidades (identidades federadas).
8. En Etiquetas, elija Agregar etiqueta si quiere aplicar etiquetas al grupo de identidades.
9. En Revisar y crear, confirme las selecciones que realizó para el nuevo grupo de identidades. Seleccione Editar para volver al asistente y cambiar cualquier configuración. Cuando haya acabado, seleccione Crear grupo de identidades.

Roles de IAM de usuario

Un rol de IAM define los permisos para que los usuarios accedan a los AWS recursos, por ejemploAmazon Cognito Sync. Los usuarios de su aplicación asumirán los roles que cree. Puede especificar otros roles para usuarios autenticados y sin autenticar. Para obtener más información acerca de los roles de IAM, consulte Roles de IAM.

Identidades autenticadas y sin autenticar

Los grupos de identidades de Amazon Cognito admiten tanto las identidades autenticadas como las no autenticadas. Las identidades autenticadas pertenecen a los usuarios que se han autenticado mediante un proveedor de identidad compatible. En cuanto a las identidades sin autenticar normalmente corresponden a usuarios invitados.

- Para configurar identidades autenticadas en un proveedor de inicio de sesión público, consulte Proveedores de identidad externos de grupos de identidades.
- Para configurar su propio proceso de autenticación de backend, consulte Identidades autenticadas por el desarrollador (grupos de identidades).

Activar o desactivar el acceso de invitados

El acceso de invitado a los grupos de identidades de Amazon Cognito (identidades no autenticadas) proporciona un identificador y AWS credenciales únicos para los usuarios que no se autentican con un proveedor de identidad. Si la aplicación permite usuarios que no inician sesión, puede activar el acceso de identidades sin autenticar. Para obtener más información, consulte Introducción a los grupos de identidades de Amazon Cognito.

Para actualizar el acceso de invitado en un grupo de identidades

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Busque el Acceso de invitado. En un grupo de identidades que actualmente no admite el acceso de invitado, el Estado es Inactivo.

    a. Si el Acceso de invitado está Activo y quiere desactivarlo, seleccione Desactivar.

    b. Si el Acceso de invitado está Inactivo y quiere activarlo, seleccione Editar.

    - Elija un rol de IAM predeterminado para los usuarios invitados del grupo de identidades.

        A. Elija Crear un nuevo rol de IAM si desea que Amazon Cognito cree uno nuevo para usted con permisos básicos y una relación de confianza con el grupo de identidades. Ingrese un Nombre de rol de IAM para identificar el nuevo rol, por ejemplo `myidentitypool_authenticatedrole`. Seleccione Ver documento de política para revisar los permisos que Amazon Cognito asignará al nuevo rol de IAM.

        B. Puede optar por utilizar una función de IAM existente si ya tiene una función Cuenta de AWS que desee utilizar. Debe configurar la política de confianza del rol de IAM para incluir `cognito-identity.amazonaws.com`. Configure la política de confianza del rol para que solo permita que Amazon Cognito asuma el rol cuando presente pruebas de que la solicitud proviene de un usuario autenticado del grupo de identidades específico. Para obtener más información, consulte Confianza y permisos de rol.

        C. Seleccione Guardar cambios.

        D. Para activar el acceso como invitado, seleccione Activar en la pestaña Acceso de usuario.

Cambio del rol asociado a un tipo de identidad

Cada identidad del grupo de identidades es autenticada o sin autenticar. Las identidades autenticadas pertenecen a usuarios que se han autenticado mediante un proveedor de inicio de sesión público (grupos de usuarios de Amazon Cognito, Login with Amazon, Sign in with Apple, Facebook, Google, SAML o cualquier proveedor de OpenID Connect) o un proveedor de desarrolladores (su propio proceso de autenticación backend). En cuanto a las identidades sin autenticar normalmente corresponden a usuarios invitados.

Cada tipo de identidad tiene un rol asignado. Este rol tiene una política adjunta que determina a qué rol puede acceder Servicios de AWS ese rol. Cuando Amazon Cognito recibe una solicitud, el servicio determina el tipo de identidad y el rol asignado a dicho tipo de identidad, y utiliza la política adjunta a ese rol para responder. Al modificar una política o asignar un rol diferente a un tipo de identidad, puede controlar a qué tipo Servicios de AWS de identidad puede acceder. Para ver o modificar las políticas asociadas a los roles en su grupo de identidades, consulte la consola de AWS IAM.

Para cambiar el rol predeterminado autenticado o no autenticado del grupo de identidades

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Busque el Acceso de invitado o el Acceso autenticado. En un grupo de identidades que no esté configurado actualmente para ese tipo de acceso, el Estado es Inactivo. Seleccione Editar.
4. Elija un rol de IAM predeterminado para los usuarios autenticados o invitados del grupo de identidades.

    a. Elija Crear un nuevo rol de IAM si desea que Amazon Cognito cree uno nuevo para usted con permisos básicos y una relación de confianza con el grupo de identidades. Ingrese un Nombre de rol de IAM para identificar el nuevo rol, por ejemplo `myidentitypool_authenticatedrole`. Seleccione Ver documento de política para revisar los permisos que Amazon Cognito asignará al nuevo rol de IAM.

    b. Puede optar por utilizar una función de IAM existente si ya tiene una función Cuenta de AWS que desee utilizar. Debe configurar la política de confianza del rol de IAM para incluir `cognito-identity.amazonaws.com`. Configure la política de confianza del rol para que solo permita que Amazon Cognito asuma el rol cuando presente pruebas de que

la solicitud proviene de un usuario autenticado del grupo de identidades específico. Para obtener más información, consulte Confianza y permisos de rol.

5. Seleccione Guardar cambios.

Editar proveedores de identidad

Si permite que los usuarios se autentiquen mediante proveedores de identidad de los clientes (por ejemplo, grupos de usuarios de Amazon Cognito, Login with Amazon, Sign in with Apple, Facebook o Google), puede especificar los identificadores de su aplicación en la consola de grupos de identidades de Amazon Cognito (identidades federadas). De esta forma, asociará el ID de la aplicación (proporcionado por el proveedor de inicio de sesión público) a su grupo de identidades.

También puede configurar en esta página reglas de autenticación para cada proveedor. Cada proveedor permite un máximo de 25 reglas. Las reglas se aplican en el orden que ha seguido para guardarlas para cada proveedor. Para obtener más información, consulte Uso del control de acceso basado en roles.

> **Warning**
>
> Cambiar el ID de aplicación de IdP enlazado en el grupo de identidades evita que los usuarios existentes se puedan autenticar con dicho grupo de identidades. Para obtener más información, consulte Proveedores de identidad externos de grupos de identidades.

Para actualizar un proveedor de identidades (IdP) de un grupo de identidades

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Localice proveedores de identidades. Elija el proveedor de identidades que desea editar. Si quiere agregar un nuevo IdP, seleccione Agregar proveedor de identidades.
   - Si elige Agregar proveedor de identidades, elija uno de los tipos de identidad que desee agregar.
4. Para cambiar el ID de la aplicación, seleccione Editar en la Información del proveedor de identidades.

5. Para cambiar el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, elija Editar en la Configuración del rol.

   - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas. Con un IdP del grupo de usuarios de Amazon Cognito, también puede Elegir un rol con preferred_role en los tokens. Para obtener más información acerca de la reclamación de `cognito:preferred_role`, consulte Asignación de valores de prioridad a los grupos.

     i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

     ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

6. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, elija Editar en Atributos para el control de acceso.

   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

   c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

7. Seleccione Guardar cambios.

Eliminación de un grupo de identidades

No puede deshacer la eliminación de un grupo de identidades. Tras eliminar un grupo de identidades, todas las aplicaciones y los usuarios que dependen de él dejan de funcionar.

Para eliminar un grupo de identidades

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione el botón de opción situado junto al grupo de identidades que desea eliminar.

2. Seleccione Eliminar.
3. Ingrese o pegue el nombre del grupo de identidades y seleccione Eliminar.

> **Warning**
>
> Si selecciona botón Delete (eliminar), eliminará permanentemente el grupo de identidades y todos los datos de usuarios que dicho grupo contiene. La eliminación de un grupo de identidades hace que las aplicaciones y los demás servicios que usan el grupo dejen de funcionar.

Eliminación de una identidad de un grupo de identidades

Al eliminar una identidad de un grupo de identidades, se elimina la información de identificación que Amazon Cognito ha almacenado para ese usuario federado. Cuando el usuario vuelva a solicitar las credenciales, recibirá un nuevo ID de identidad si el grupo de identidades sigue confiando en el proveedor de identidades. No podrá deshacer esta operación.

Para eliminar una identidad

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Navegador de identidades.
3. Seleccione las casillas de verificación situadas junto a las identidades que desea eliminar y elija Eliminar. Confirme que desea eliminar las identidades y elija Eliminar.

Uso de Amazon Cognito Sync con grupos de identidades

Amazon Cognito Sync es una Servicio de AWS biblioteca de clientes que permite sincronizar los datos de usuario relacionados con las aplicaciones en todos los dispositivos. Amazon Cognito Sync puede sincronizar los datos de los perfiles de usuario entre los dispositivos móviles y la web sin necesidad de utilizar su propio backend. Las bibliotecas de cliente almacenan los datos localmente en la caché para que su aplicación pueda leer y escribir datos, sin importar el estado de conexión del dispositivo. Cuando el dispositivo esté en línea, podrá sincronizar los datos. Cuando el dispositivo esté en línea, podrá notificar inmediatamente a otros dispositivos que hay una actualización disponible.

Administración de conjuntos de datos

Si ha implementado la funcionalidad de Amazon Cognito Sync en la aplicación, la consola de grupos de identidades de Amazon Cognito permite crear y eliminar de forma manual conjuntos de datos y registros de identidades individuales. Los cambios que efectúe en el conjunto de datos o los registros de una identidad en la consola de grupos de identidades de Amazon Cognito no se guardarán hasta que no haya seleccionado Sincronize (Sincronizar) en la consola. El cambio no es visible para el usuario final hasta que la identidad llama a Sincronize (Sincronizar). Los datos que se sincronizando desde otros dispositivos para identidades individuales son visibles cuando se actualiza la página de conjuntos de datos de lista de una identidad determinada.

Creación de un conjunto de datos para una identidad

Amazon Cognito Sync asocia un conjunto de datos a una identidad. Puede rellenar el conjunto de datos con información de identificación sobre el usuario que representa la identidad y, a continuación, sincronizar esa información con todos los dispositivos del usuario.

Para agregar un conjunto de datos y los registros de un conjunto de datos a una identidad

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Navegador de identidades.
3. Seleccione la identidad que desea editar.
4. En Conjuntos de datos, elija Crear conjunto de datos.
5. Ingrese un Nombre de conjunto de datos y seleccione Crear conjunto de datos.
6. Si quiere agregar registros al conjunto de datos, elija el conjunto de datos entre los detalles de identidad. En Registros, seleccione Crear registro.
7. Ingrese una Clave y un Valor para el registro. Elija Confirmar. Repita el procedimiento para agregar más registros.

Eliminación de un conjunto de datos asociado a una identidad

Para eliminar un conjunto de datos y sus registros de una identidad

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Navegador de identidades.

3. Seleccione la identidad que contiene el conjunto de datos que desea eliminar.
4. En Conjuntos de datos, elija el botón de opción situado junto al conjunto de datos que desea eliminar.
5. Seleccione Eliminar. Revise la elección y vuelva a seleccionar Eliminar.

Publicación en masa de datos

La publicación en masa se puede utilizar para exportar datos que ya se encuentren en un almacén de Amazon Cognito Sync a una transmisión de Amazon Kinesis. Para obtener instrucciones sobre cómo publicar en masa todos los flujos, consulte Amazon Cognito Streams.

Activar sincronización mediante inserción

Amazon Cognito realiza seguimiento de forma automática de la asociación entre la identidad y los dispositivos. El uso de la característica de sincronización por inserción puede asegurar que todas las instancias de una determinada identidad reciban una notificación cuando cambien los datos de identidad. La sincronización mediante inserción hace que, cuando el conjunto de datos cambia para una identidad, todos los dispositivos asociados con esa identidad recibirán una notificación de inserción silenciosa que informa del cambio.

Puede activar la sincronización mediante inserción en la consola de Amazon Cognito.

Para activar sincronización mediante inserción

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Propiedades del grupo de identidades.
3. En Sincronización mediante inserción, seleccione Editar
4. Seleccione Activar la sincronización mediante inserción con el grupo de identidades.
5. Elija una de las aplicaciones de la plataforma de Amazon Simple Notification Service (Amazon SNS) que ha creado en la Región de AWS actual. Amazon Cognito publica notificaciones push en la aplicación de la plataforma. Seleccione Crear aplicación de plataforma para ir a la consola de Amazon SNS y crear una nueva.
6. Para publicar en la aplicación de plataforma, Amazon Cognito asume un rol de IAM en la Cuenta de AWS. Elija Crear un nuevo rol de IAM si desea que Amazon Cognito cree uno nuevo para usted con permisos básicos y una relación de confianza con el

grupo de identidades. Ingrese un Nombre de rol de IAM para identificar el nuevo rol, por ejemplo `myidentitypool_authenticatedrole`. Seleccione Ver documento de política para revisar los permisos que Amazon Cognito asignará al nuevo rol de IAM.

7. Puede optar por utilizar una función de IAM existente si ya tiene una función Cuenta de AWS que desee utilizar. Debe configurar la política de confianza del rol de IAM para incluir `cognito-identity.amazonaws.com`. Configure la política de confianza del rol para que solo permita que Amazon Cognito asuma el rol cuando presente pruebas de que la solicitud proviene de un usuario autenticado del grupo de identidades específico. Para obtener más información, consulte Confianza y permisos de rol.
8. Seleccione Guardar cambios.

Configuración de Amazon Cognito Streams

Amazon Cognito Streams ofrece a los desarrolladores control e información de los datos almacenados en Amazon Cognito Sync. Ahora, los desarrolladores pueden configurar un flujo de Kinesis para recibir eventos como datos. Amazon Cognito puede enviar cada cambio en el conjunto de datos a un flujo de Kinesis de su propiedad en tiempo real. Para obtener instrucciones acerca de cómo configurar Amazon Cognito Streams en la consola de Amazon Cognito, consulte Amazon Cognito Streams.

Configuración de Amazon Cognito Events

Amazon Cognito Events le permite ejecutar una AWS Lambda función en respuesta a eventos importantes en Amazon Cognito Sync. Amazon Cognito Sync lanza el evento desencadenador de sincronización cuando se sincroniza un conjunto de datos. Puede utilizar el evento disparador de la sincronización para actuar cuando un usuario actualiza los datos. Para obtener instrucciones sobre cómo configurar Amazon Cognito Events desde la consola, consulte Amazon Cognito Events.

Para obtener más información AWS Lambda, consulte. AWS Lambda

Conceptos de grupos de identidades

Puede usar los grupos de identidades de Amazon Cognito para crear identidades únicas para sus usuarios y autenticarlos con proveedores de identidad. Con una identidad, puede obtener AWS credenciales temporales con privilegios limitados para acceder a otras. Servicios de AWS Los grupos de identidades de Amazon Cognito son compatibles con proveedores de identidad públicos (como

Amazon, Apple, Facebook y Google) y con identidades no autenticadas. También es compatible con las identidades autenticadas de desarrollador, que permiten registrar y autenticar a los usuarios mediante su propio proceso de autenticación backend.

Para obtener información sobre la disponibilidad regional de los grupos de identidades de Amazon Cognito, consulte Disponibilidad regional del servicio de AWS. Para obtener más información sobre los grupos de identidades de Amazon Cognito, consulte los siguientes temas.

Temas

- Flujo de autenticación de grupos de identidades (identidades federadas)
- Roles de IAM
- Confianza y permisos de rol

Flujo de autenticación de grupos de identidades (identidades federadas)

Amazon Cognito sirve de ayuda a fin de crear identificadores únicos para los usuarios finales, que se mantienen homogéneos en todos los dispositivos y plataformas. Amazon Cognito también proporciona credenciales temporales con privilegios limitados a su aplicación para acceder a los recursos. AWS En esta página, se describen los aspectos básicos de cómo funciona la autenticación en Amazon Cognito y se explica el ciclo de vida de una identidad dentro del grupo de identidades.

Flujo de autenticación con proveedores externos

Un usuario que se autentique con Amazon Cognito pasa por varias etapas para iniciar el proceso de arranque de las credenciales. Amazon Cognito tiene dos flujos diferentes para la autenticación con proveedores públicos: el flujo básico y el mejorado.

Una vez que complete uno de estos flujos, podrá acceder a otros Servicios de AWS según lo definan las políticas de acceso de su función. De forma predeterminada, la consola de Amazon Cognito crea roles con acceso al almacén de Amazon Cognito Sync y a Amazon Mobile Analytics. Para obtener más información sobre cómo conceder acceso adicional, consulte Roles de IAM.

Los grupos de identidades aceptan los siguientes artefactos de los proveedores:

Proveedor	Artefacto de autenticación
Grupos de usuarios de Amazon Cognito	Token de ID

Proveedor	Artefacto de autenticación
OpenID Connect (OIDC)	Token de ID
SAML 2.0	Aserción de SAML
Proveedor social	Token de acceso

Flujo de autenticación mejorado (simplificado)

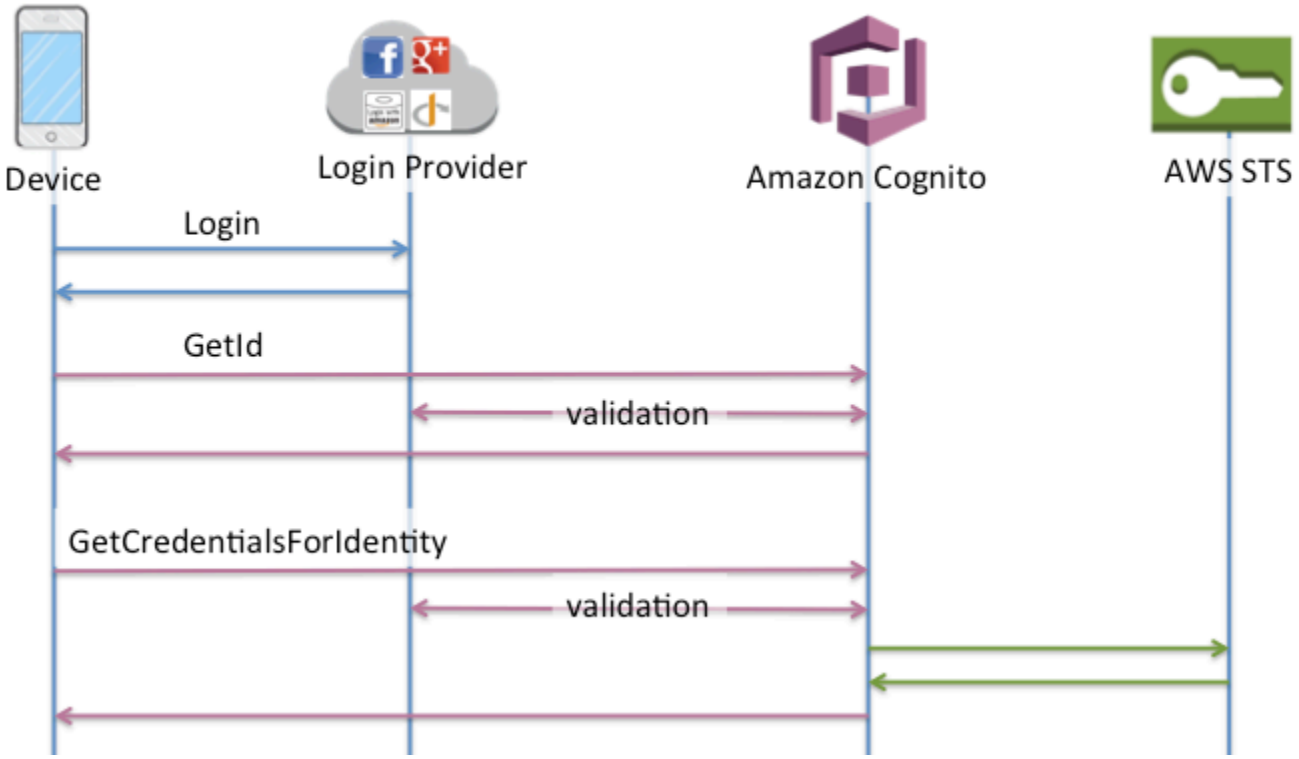
Cuando utilizas el flujo de autenticación mejorado, tu aplicación presenta primero en una solicitud una prueba de autenticación de un grupo de usuarios autorizado de Amazon Cognito o de un proveedor de identidad externo. GetId

1. En una solicitud de GetID, su aplicación presenta una prueba de autenticación (un token web JSON o una afirmación de SAML) de un grupo de usuarios autorizado de Amazon Cognito o de un proveedor de identidad externo.
2. Su conjunto de identidades devuelve un identificador de identidad.
3. La aplicación combina el identificador de identidad con la misma prueba de autenticación en una GetCredentialsForIdentitysolicitud.
4. Su grupo de identidades devuelve AWS las credenciales.
5. Su aplicación firma las solicitudes de AWS API con las credenciales temporales.

La autenticación mejorada gestiona la lógica de la selección de roles de IAM y la recuperación de credenciales en la configuración del grupo de identidades. Puede configurar su grupo de identidades para seleccionar un rol predeterminado y aplicar los principios del control de acceso basado en atributos (ABAC) o del control de acceso basado en roles (RBAC) a la selección de roles. Las AWS credenciales de la autenticación mejorada son válidas durante una hora.

Orden de las operaciones en la autenticación mejorada

1. `GetId`
2. `GetCredentialsForIdentity`



Flujo de autenticación básico (clásico)

Al utilizar el flujo de autenticación básico,

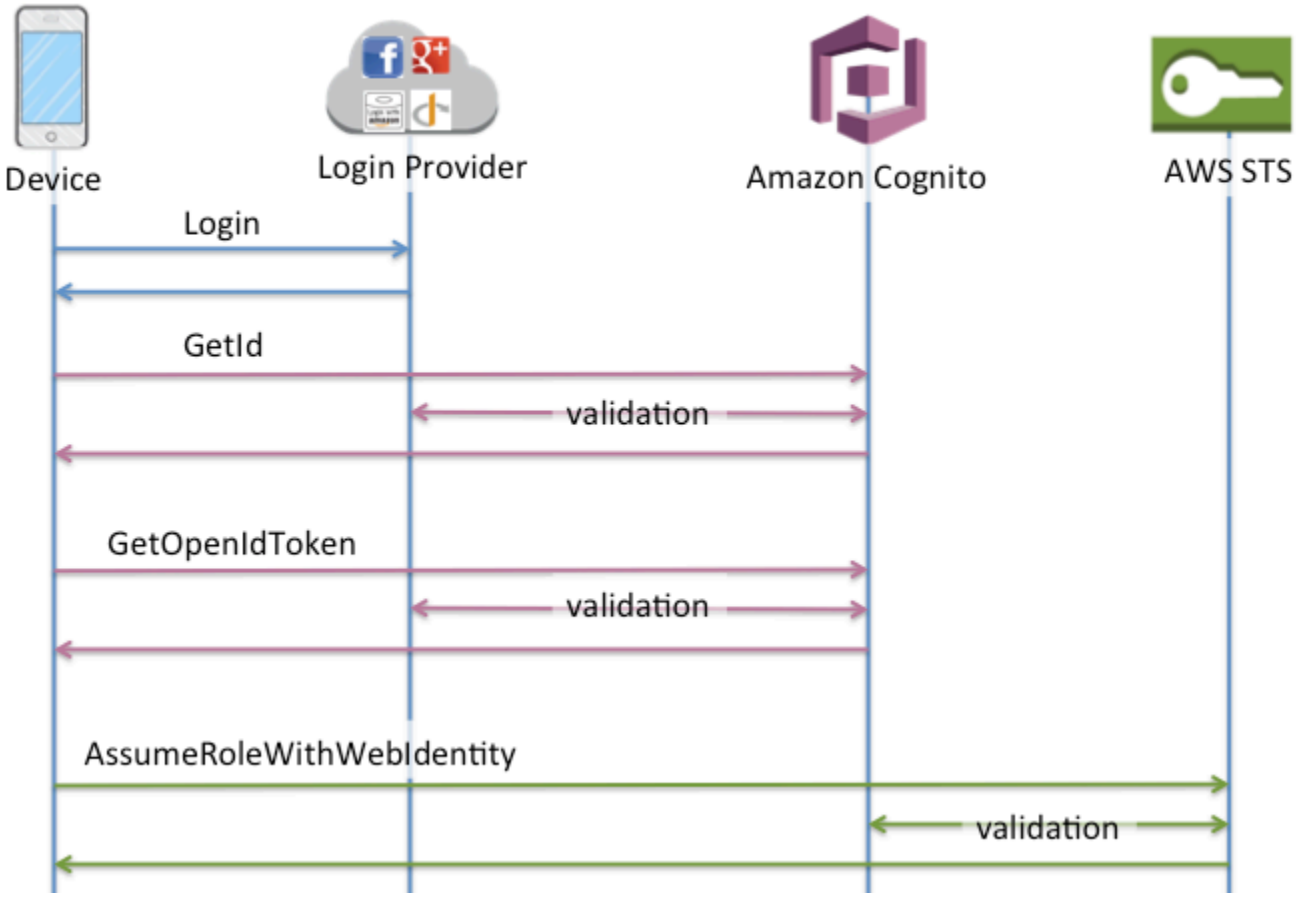
1. En una solicitud de GetID, su aplicación presenta una prueba de autenticación (un token web JSON o una afirmación de SAML) de un grupo de usuarios autorizado de Amazon Cognito o de un proveedor de identidad externo.
2. Su conjunto de identidades devuelve un identificador de identidad.
3. La aplicación combina el identificador de identidad con la misma prueba de autenticación en una GetOpenIdTokensolicitud.
4. `GetOpenIdToken`devuelve un nuevo token de OAuth 2.0 emitido por tu grupo de identidades.
5. Tu aplicación presenta el nuevo token en una AssumeRoleWithWebIdentitysolicitud.
6. AWS Security Token Service AWS STS) devuelve AWS las credenciales.
7. Su aplicación firma las solicitudes de AWS API con las credenciales temporales.

El flujo de trabajo básico le proporciona un control más pormenorizado sobre las credenciales que distribuye a los usuarios. La solicitud `GetCredentialsForIdentity` del flujo de autenticación mejorado solicita un rol basado en el contenido de un token de acceso. La

`AssumeRoleWithWebIdentity` solicitud del flujo de trabajo clásico otorga a tu aplicación una mayor capacidad para solicitar credenciales para cualquier AWS Identity and Access Management rol que hayas configurado con una política de confianza suficiente. También puede solicitar una duración de sesión de rol personalizada.

Orden de las operaciones en la autenticación básica

1. `GetId`
2. `GetOpenIdToken`
3. `AssumeRoleWithWebIdentity`



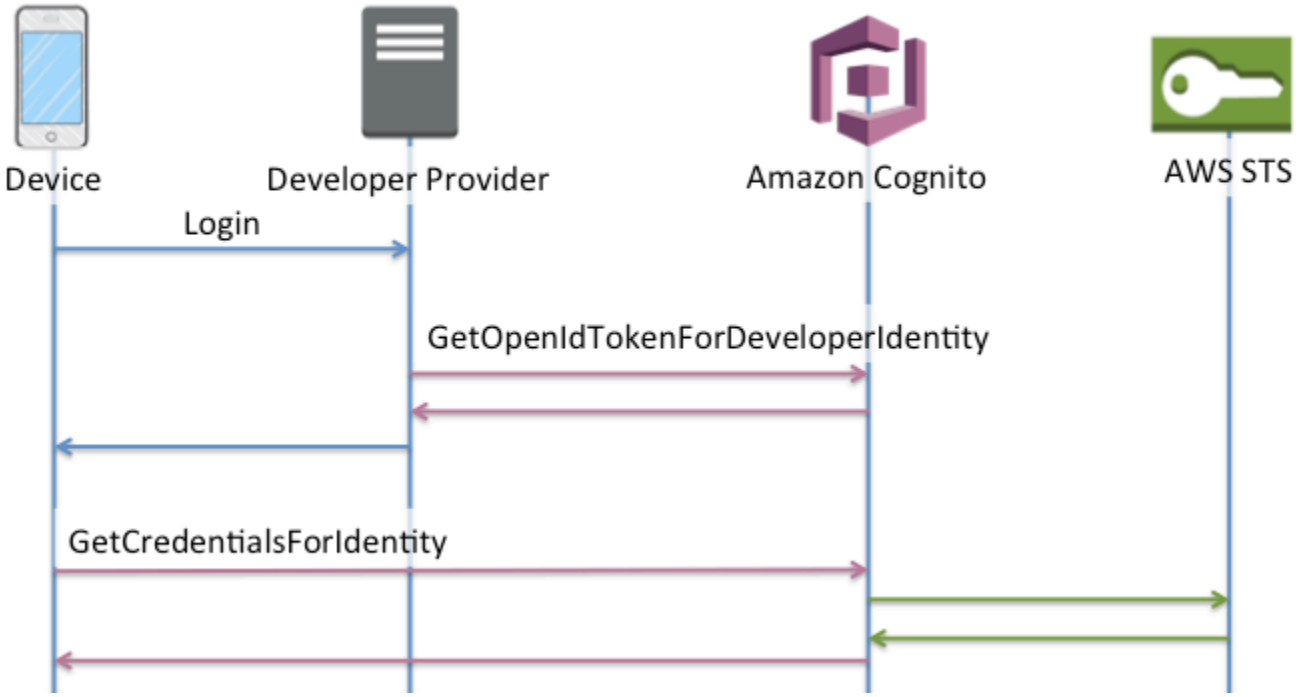
Flujo de autenticación de identidades autenticadas por el desarrollador

Cuando se utiliza Identidades autenticadas por el desarrollador (grupos de identidades), el cliente emplea otro flujo de autenticación, que incluye código fuera de Amazon Cognito para validar al usuario en el propio sistema de autenticación del desarrollador. El código por fuera de Amazon Cognito se indica como tal.

Flujo de autenticación mejorado

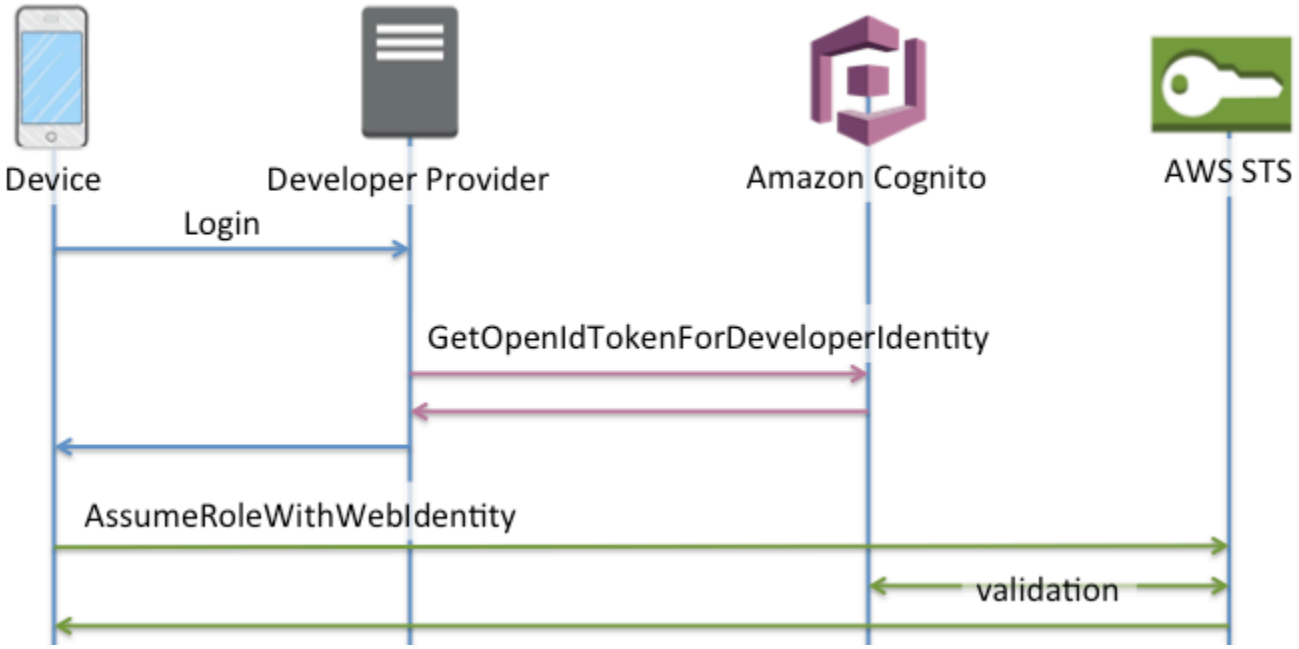
Orden de las operaciones en la autenticación mejorada con un proveedor desarrollador

1. Inicio de sesión a mediante un proveedor de desarrollador (código por fuera de Amazon Cognito)
2. Validación del inicio de sesión de usuario (código por fuera de Amazon Cognito)
3. GetOpenIdTokenForDeveloperIdentity
4. GetCredentialsForIdentity



Orden de las operaciones en la autenticación básica con un proveedor desarrollador

1. Implemente una lógica fuera del grupo de identidades para iniciar sesión y generar un identificador entre el desarrollador y el proveedor.
2. Recupera las credenciales almacenadas del lado del servidor. AWS
3. Envía el identificador del proveedor desarrollador en una solicitud GetOpenIdTokenForDeveloperIdentityde API firmada con credenciales autorizadas AWS .
4. Solicita las credenciales de la aplicación con AssumeRoleWithWebIdentity.



¿Qué flujo de autenticación debo usar?

El flujo mejorado es la opción más segura con el nivel más bajo de esfuerzo del desarrollador:

- El flujo mejorado reduce la complejidad, el tamaño y la velocidad de las solicitudes de API.
- No es necesario que tu aplicación realice solicitudes de API adicionales a AWS STS.
- Tu grupo de identidades evalúa a tus usuarios para determinar las credenciales de rol de IAM que deben recibir. No necesita integrar la lógica para la selección de roles en su cliente.

> **Important**
>
> Cuando cree un nuevo grupo de identidades, no active la autenticación básica (clásica) de forma predeterminada, como práctica recomendada. Para implementar la autenticación básica, primero evalúe las relaciones de confianza de sus funciones de IAM para las identidades web. A continuación, incorpore la lógica para la selección de roles en su cliente y proteja al cliente contra cualquier modificación por parte de los usuarios.

El flujo de autenticación básico delega la lógica de selección de roles de IAM en su aplicación. En este flujo, Amazon Cognito valida la sesión autenticada o no autenticada del usuario y emite un token con el que puede intercambiar credenciales. AWS STS Los usuarios pueden intercambiar los tokens

de la autenticación básica por cualquier función de IAM que confíe en su conjunto de identidades o en el estado autenticado o no autenticado. `amr`

Del mismo modo, tenga en cuenta que la autenticación del desarrollador es un método abreviado para validar la autenticación del proveedor de identidad. Amazon Cognito confía en las AWS credenciales que autorizan una GetOpenIdTokenForDeveloperIdentitysolicitud sin necesidad de validar adicionalmente el contenido de la solicitud. Proteja los secretos que autorizan la autenticación de los desarrolladores para que no puedan acceder a ellos los usuarios.

Resumen de las API

GetId

La llamada a la GetIdAPI es la primera llamada necesaria para establecer una nueva identidad en Amazon Cognito.

Acceso sin autenticar

Amazon Cognito permite acceder a sus aplicaciones como invitado no autenticado. Si esta característica está habilitada en el grupo de identidades, los usuarios pueden solicitar un ID de identidad nuevo en cualquier momento mediante la API `GetId`. Se espera que la aplicación almacene en caché este ID de identidad para realizar llamadas posteriores a Amazon Cognito. Los SDK para AWS dispositivos móviles y el AWS SDK para el navegador tienen proveedores de credenciales que se encargan de este almacenamiento JavaScript en caché por usted.

Acceso autenticado

Si configura su aplicación para que sea compatible con un proveedor de inicio de sesión público (Facebook, Google+, Login with Amazon o Sign in with Apple), los usuarios también pueden suministrar tokens (OAuth u OpenID Connect) que los identifiquen en dichos proveedores. Cuando se utiliza en una llamada a `GetId`, Amazon Cognito crea una identidad autenticada nueva o devuelve la identidad ya asociada a ese inicio de sesión en particular. Para ello, Amazon Cognito valida el token con el proveedor y se asegura de que se cumpla lo siguiente:

- El token es válido y del proveedor configurado.
- El token no está caducado.
- El token coincide con el identificador de aplicaciones creado en dicho proveedor (por ejemplo, el ID de aplicación de Facebook).
- El token coincide con el identificador de usuario.

GetCredentialsForIdentity

Se puede llamar a la GetCredentialsForIdentityAPI después de establecer un ID de identidad. Por AssumeRoleWithWebIdentitylo tanto, esta operación es funcionalmente equivalente a llamar GetOpenIdToken.

Para que Amazon Cognito llame a `AssumeRoleWithWebIdentity` en su nombre, el grupo de identidades debe tener roles de IAM asociados. Puede hacerlo a través de la consola Amazon Cognito o manualmente mediante la SetIdentityPoolRolesoperación.

GetOpenIdToken

Realice una solicitud a la GetOpenIdTokenAPI después de establecer un identificador de identidad. Guarde en caché los ID de identidad después de la primera solicitud e inicie las sesiones básicas (clásicas) posteriores para esa identidad con `GetOpenIdToken`.

La respuesta a una solicitud de la API `GetOpenIdToken` es un token que genera Amazon Cognito. Puedes enviar este token como `WebIdentityToken` parámetro en una AssumeRoleWithWebIdentitysolicitud.

Antes de enviar el token de OpenID, verifíquelo en su aplicación. Puede utilizar las bibliotecas OIDC del SDK o una biblioteca como aws-jwt-verify para confirmar que Amazon Cognito ha emitido el token. El ID de clave de firma, o `kid`, del token de OpenID es uno de los que figuran en el documento jwks_uri† de Amazon Cognito Identity. Estas claves están sujetas a cambios. La función que verifica los tokens de Amazon Cognito Identity debe actualizar periódicamente su lista de claves desde el documento jwks_uri. Amazon Cognito establece la duración de la actualización en el encabezado de respuesta de cache-control jwks_uri, que actualmente está establecido `max-age` en 30 días.

Acceso sin autenticar

Para obtener un token para una identidad sin autenticar, solo necesita el ID de identidad. No es posible obtener un token sin autenticar para identidades autenticadas o que se han desactivado.

Acceso autenticado

Si tiene una identidad autenticada, debe transmitir al menos un token válido para un inicio de sesión que ya esté asociado a dicha identidad. Todos los tokens que se transmitan durante la llamada `GetOpenIdToken` deben pasar la misma validación mencionada anteriormente; si alguno de los tokens falla, toda la llamada fallará. La respuesta de la llamada

`GetOpenIdToken` también incluye el ID de identidad. Esto se debe a que el ID de identidad que pasa puede que no sea el que se devuelve.

Vinculación de inicios de sesión

Si envía un token para un inicio de sesión que todavía no tiene ninguna identidad asociada, se considerará que el inicio de sesión está “vinculado" a la identidad asociada. Solo puede vincular un inicio de sesión por proveedor público. Los intentos de vincular más de un inicio de sesión a un proveedor público generan una respuesta de error `ResourceConflictException`. Si un inicio de sesión solo está vinculado a una identidad existente, el ID de identidad que `GetOpenIdToken` devuelva será el mismo que el que se pasó.

Combinación de identidades

Si pasa un token para un inicio de sesión que no está vinculado a la identidad determinada, pero está vinculado a otra identidad, las dos identidades se combinan. Una vez combinadas, una identidad se convertirá en la identidad principal/propietaria de todos los inicios de sesión asociados mientras que la otra se deshabilitará. En este caso, se devuelve el ID de identidad principal/propietario. Debe actualizar la caché local si este valor difiere. Los proveedores de los SDK AWS móviles o AWS del SDK para JavaScript el navegador realizan esta operación por ti.

GetOpenIdTokenForDeveloperIdentity

La GetOpenIdTokenForDeveloperIdentityoperación reemplaza el uso GetOpenIdTokendesde GetIdy desde el dispositivo cuando se utilizan identidades autenticadas por el desarrollador. Dado que su aplicación firma las solicitudes a esta operación de API con AWS credenciales, Amazon Cognito confía en que el identificador de usuario proporcionado en la solicitud sea válido. La autenticación de desarrolladores reemplaza la validación de token que Amazon Cognito realiza con proveedores externos.

La carga útil de esta API incluye un `logins` mapa. Este mapa debe contener la clave del proveedor desarrollador y un valor como identificador del usuario en el sistema. Si el identificador de usuario todavía no está vinculado a una identidad existente, Amazon Cognito crea una identidad nueva y devuelve el ID de la identidad nueva y un token de OpenID Connect para dicha entidad. Si el identificador del usuario ya está vinculado, Amazon Cognito devuelve el ID de identidad preexistente y un token de OpenID Connect. Guarde en caché los ID de identidad del desarrollador después de la primera solicitud e inicie las sesiones básicas (clásicas) posteriores para esa identidad con `GetOpenIdTokenForDeveloperIdentity`.

La respuesta a una solicitud de la API `GetOpenIdTokenForDeveloperIdentity` es un token que genera Amazon Cognito. Puede enviar este token como parámetro `WebIdentityToken` en una solicitud `AssumeRoleWithWebIdentity`.

Antes de enviar el token de OpenID Connect, verifíquelo en su aplicación. Puede utilizar las bibliotecas OIDC del SDK o una biblioteca como aws-jwt-verify para confirmar que Amazon Cognito ha emitido el token. El ID de clave de firma, o `kid`, del token de OpenID Connect es uno de los que figuran en el documento jwks_uri de Amazon Cognito Identity†. Estas claves están sujetas a cambios. La función que verifica los tokens de Amazon Cognito Identity debe actualizar periódicamente su lista de claves desde el documento jwks_uri. Amazon Cognito establece la duración de la actualización en el encabezado de respuesta de `cache-control` jwks_uri, que actualmente tiene establecido `max-age` en 30 días.

Vinculación de inicios de sesión

De igual modo que ocurre con los proveedores externos, si se suministran inicios de sesión adicionales que todavía no están asociados a una identidad, los inicios de sesión se vincularán implícitamente a dicha identidad. Si enlaza un inicio de sesión de proveedor externo a una identidad, el usuario puede utilizar el flujo de autenticación del proveedor externo con ese proveedor. Sin embargo, no pueden usar el nombre del proveedor de desarrolladores en el mapa de inicios de sesión al ejecutar `GetId` o `GetOpenIdToken`.

Combinación de identidades

Con las identidades autenticadas por el desarrollador, Amazon Cognito admite tanto la fusión implícita como la fusión explícita a través de la API. MergeDeveloperIdentities La combinación explícita le permite marcar dos identidades con los identificadores de usuario de su sistema como una identidad única. Tan solo debe proporcionar los identificadores de usuario de origen y de destino, y Amazon Cognito los combinará. La siguiente vez que solicite un token de OpenID Connect para cada una de las identidades de usuario, se devolverá el mismo ID de identidad.

AssumeRoleWithWebIdentity

Una vez que tengas un token de OpenID Connect, puedes cambiarlo por AWS credenciales temporales mediante la solicitud de AssumeRoleWithWebIdentityAPI a AWS Security Token Service ()AWS STS.

Dado que no hay restricciones en cuanto al número de identidades que se pueden crear, es importante comprender los permisos que va a conceder a los usuarios. Configura diferentes

funciones de IAM para tu aplicación: una para los usuarios no autenticados y otra para los usuarios autenticados. La consola de Amazon Cognito puede crear funciones predeterminadas al configurar el grupo de identidades por primera vez. En efecto, estos roles no tienen permisos concedidos. Modifíquelos para que se adapten a sus necesidades.

Obtener más información sobre Confianza y permisos de rol.

† El documento jwks_uri predeterminado de Amazon Cognito Identity contiene información sobre las claves que firman los tokens de los grupos de identidades en la mayoría de las Regiones de AWS. Las siguientes regiones tienen diferentes documentos jwks_uri.

Amazon Cognito Identity JSON web key URIs in other Regiones de AWS

Región de AWS	Ruta al documento jwks_uri
AWS GovCloud (EE. UU.-Oeste)	`https://cognito-identity.us-gov-west-1.amazonaws.com/.well-known/jwks_uri`
China (Pekín)	`https://cognito-identity.cn-north-1.amazonaws.com.cn/.well-known/jwks_uri`
Regiones de suscripción voluntaria como Europa (Milán) y África (Ciudad del Cabo)	`https://cognito-identity.` *`Region`*`.amazonaws.com/.well-known/jwks_uri`

También puede extrapolar el jwks_uri del emisor o el `iss` que recibe en el token de OpenID desde Amazon Cognito. El punto de conexión de detección estándar de OIDC `<issuer>/.well-known/openid-configuration` muestra una ruta al jwks_uri para su token.

Roles de IAM

En el proceso de creación de un grupo de identidades, se le solicita que actualice los roles de IAM que asumen sus usuarios. Las funciones de IAM funcionan así: cuando un usuario inicia sesión en su aplicación, Amazon Cognito genera credenciales AWS temporales para el usuario. Estas credenciales temporales se asocian a un rol de IAM específico. Con la función de IAM, puede definir un conjunto de permisos para acceder a sus recursos. AWS

Puede especificar los roles de IAM predeterminados para usuarios autenticados y sin autenticar. Asimismo, puede definir reglas para elegir el rol de cada usuario en función de las notificaciones contenidas en el token de ID. Para obtener más información, consulte Uso del control de acceso basado en roles.

De forma predeterminada, la consola de Amazon Cognito crea roles de IAM que brindan acceso a Amazon Mobile Analytics y Amazon Cognito Sync. O bien, puede optar por utilizar los de IAM existentes.

Modifique los roles de IAM para permitir o restringir el acceso a otros servicios. Para ello, inicie sesión en la consola de IAM. A continuación, seleccione Roles (Roles) y seleccione un rol. Las políticas adjuntas al rol seleccionado se indican en la pestaña Permissions (Permisos). Puede personalizar una política de acceso mediante la selección del enlace Manage Policy (Administrar política) correspondiente. Para obtener más información sobre el uso y la definición de políticas, consulte lainformación general sobre las políticas de IAM.

> **Note**
>
> Como práctica recomendada, defina políticas que sigan el principio de concesión de privilegios mínimos. En otras palabras, las políticas incluyen solo los permisos que los usuarios necesitan para llevar a cabo sus tareas. Para obtener más información, consulte Concesión de mínimos privilegios en la Guía del usuario de IAM.
> Recuerde que las identidades sin autenticar las asumen los usuarios que no inician sesión en su aplicación. Normalmente, los permisos que asigna para las identidades sin autenticar deben ser más restrictivas que los de las identidades autenticadas.

Temas

- Configuración de una política de confianza
- Políticas de acceso

Configuración de una política de confianza

Amazon Cognito aprovecha los roles de IAM para generar credenciales temporales para los usuarios de su aplicación. El acceso a los permisos se controla mediante las relaciones de confianza de un rol. Obtener más información sobre Confianza y permisos de rol.

El token que se presenta lo genera un grupo de identidades, que traduce un token de grupo de usuarios, una red social o de un proveedor de OIDC, o una afirmación de SAML, en su propio token. AWS STS El token del grupo de identidades contiene una reclamación `aud` que es el ID del grupo de identidades.

El siguiente ejemplo de política de confianza de roles permite al director `cognito-identity.amazonaws.com` del servicio federado llamar a la API. AWS STS `AssumeRoleWithWebIdentity` La solicitud solo se realizará correctamente si el token del grupo de identidades de la solicitud de la API contiene las siguientes reclamaciones.

1. Una reclamación `aud` del ID del grupo de identidades `us-west-2:abcdefg-1234-5678-910a-0e8443553f95`.
2. Esta reclamación de `amr` de `authenticated` que se agrega cuando el usuario ha iniciado sesión y no es un usuario invitado.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "cognito-identity.amazonaws.com"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "cognito-identity.amazonaws.com:aud": "us-
west-2:abcdefg-1234-5678-910a-0e8443553f95"
                },
                "ForAnyValue:StringLike": {
                    "cognito-identity.amazonaws.com:amr": "authenticated"
                }
            }
        }
    ]
}
```

Políticas de confianza para las funciones de IAM en la autenticación básica (clásica)

Debe aplicar al menos una condición que limite las políticas de confianza para los roles que utilice con los grupos de identidades. Al crear o actualizar políticas de confianza de roles para los grupos de identidades, IAM devuelve un error si intenta guardar los cambios sin al menos una clave de condición que limite las identidades de origen. AWS STS no permite AssumeRoleWithWebIdentityrealizar operaciones entre cuentas desde grupos de identidades hasta roles de IAM que no cuenten con una condición de este tipo.

En este tema se incluyen varias condiciones que limitan las identidades de origen de los grupos de identidades. Para obtener una lista completa, consulte Claves disponibles para la federación de identidades AWS web.

En la autenticación básica o clásica con un grupo de identidades, puede asumir cualquier función de IAM AWS STS si cuenta con la política de confianza adecuada. Los roles de IAM para los grupos de identidades de Amazon Cognito confían en que la entidad principal `cognito-identity.amazonaws.com` del servicio asuma el rol. Esta configuración no es suficiente para proteger sus funciones de IAM contra el acceso no deseado a los recursos. Los roles de este tipo deben aplicar una condición adicional a la política de confianza de los roles. No puede crear ni modificar roles para grupos de identidades sin al menos una de las siguientes condiciones.

`cognito-identity.amazonaws.com:aud`

Restringe la función a las operaciones de uno o más grupos de identidades. Amazon Cognito indica el conjunto de identidades de origen en la `aud` declaración del token del grupo de identidades.

`cognito-identity.amazonaws.com:amr`

Restringe la función a uno de los usuarios `authenticated` o a los `unauthenticated` usuarios (invitados). Amazon Cognito indica el estado de autenticación en la `amr` declaración del token del grupo de identidades.

`cognito-identity.amazonaws.com:sub`

Restringe la función a uno o más usuarios mediante el UUID. Este UUID es el identificador de identidad del usuario en el grupo de identidades. Este valor no es el `sub` valor del proveedor de identidad original del usuario. Amazon Cognito indica este UUID en la `sub` declaración del token del grupo de identidades.

La autenticación de flujo mejorado requiere que la función de IAM sea la Cuenta de AWS misma que la del grupo de identidades, pero este no es el caso de la autenticación básica.

Se aplican consideraciones adicionales a los grupos de identidades de Amazon Cognito que asumen roles de IAM entre cuentas. Las políticas de confianza de esas funciones deben aceptar el principio del `cognito-identity.amazonaws.com` servicio y deben contener la condición específica. `cognito-identity.amazonaws.com:aud` Para evitar el acceso no deseado a AWS los recursos, la clave de `aud` condición restringe la función a los usuarios de los grupos de identidades del valor de la condición.

El token que un grupo de identidades emite para una identidad contiene información sobre el origen del grupo Cuenta de AWS de identidades. Cuando presentas un token de grupo de identidades en una solicitud de AssumeRoleWithWebIdentityAPI, AWS STS comprueba si el grupo de identidades de origen es el Cuenta de AWS mismo que el rol de IAM. Si AWS STS determina que la solicitud es multicuenta, comprueba si la política de confianza de roles tiene alguna condición. `aud` La llamada a asumir un rol falla si no se dan tales condiciones en la política de confianza del rol. Si la solicitud no es multicuenta, AWS STS no aplica esta restricción. Como práctica recomendada, aplique siempre una condición de este tipo a las políticas de confianza de las funciones de su grupo de identidades.

Condiciones adicionales de la política de confianza

Reutilización de roles en los grupos de identidades

Para reutilizar un rol en varios grupos de identidades que comparten un conjunto de permisos comunes, puede incluir varios grupos de identidades, como se indica a continuación:

```
"StringEquals": {
    "cognito-identity.amazonaws.com:aud": [
        "us-east-1:12345678-abcd-abcd-abcd-123456790ab",
        "us-east-1:98765432-dcba-dcba-dcba-123456790ab"
    ]
}
```

Restricción del acceso a identidades concretas

Para crear una política limitada a un conjunto específico de usuarios de la aplicación, compruebe el valor de `cognito-identity.amazonaws.com:sub`:

```
"StringEquals": {
    "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-abcd-abcd-
abcd-123456790ab",
```

```
    "cognito-identity.amazonaws.com:sub": [
        "us-east-1:12345678-1234-1234-1234-123456790ab",
        "us-east-1:98765432-1234-1234-1243-123456790ab"
    ]
}
```

Restricción del acceso a proveedores concretos

Para crear una política limitada a los usuarios que han iniciado sesión con un proveedor específico (quizás su propio proveedor de inicios de sesión), compruebe el valor de `cognito-identity.amazonaws.com:amr`:

```
"ForAnyValue:StringLike": {
    "cognito-identity.amazonaws.com:amr": "login.myprovider.myapp"
}
```

Por ejemplo, una aplicación que solo confía en Facebook tendría la siguiente cláusula amr:

```
"ForAnyValue:StringLike": {
    "cognito-identity.amazonaws.com:amr": "graph.facebook.com"
}
```

Políticas de acceso

Los permisos que adjunte a un rol se aplican a todos los usuarios que asuman ese rol. Para particionar el acceso de sus usuarios, utilice condiciones y variables de política. Para obtener más información, consulte Elementos de la política de IAM: variables y etiquetas. Puede utilizar la condición `sub` para restringir las acciones a los ID de identidad de Amazon Cognito en sus políticas de acceso. Utilice esta opción con precaución, sobre todo en el caso de las identidades no autenticadas, que carecen de un ID de usuario coherente. Para obtener más información sobre las variables de política de IAM para la federación web con Amazon Cognito, consulte IAM AWS STS y claves de contexto de condición en AWS Identity and Access Management la Guía del usuario.

Para ofrecer protección de seguridad adicional, Amazon Cognito aplica una política de ámbito reducido a las credenciales que asigna a sus usuarios no autenticados en el flujo mejorado, mediante `GetCredentialsForIdentity`. La política de ámbito reducido añade una Política de sesión en línea y una AWS política de sesiones gestionadas a las políticas de IAM que aplica a su rol no autenticado. Dado que debe conceder acceso tanto en las políticas de IAM para el rol como en las políticas de sesión, la política de ámbito reducido limita el acceso de los usuarios a los servicios que no sean los que se muestran en la siguiente lista.

> **Note**
>
> En el flujo básico (clásico), realiza su propia solicitud de API AssumeRoleWithWebIdentity y puede aplicar estas restricciones a la solicitud. Como práctica recomendada de seguridad, no asigne ningún permiso por encima de esta política de ámbito reducido a usuarios no autenticados.

Amazon Cognito también impide que los usuarios autenticados y no autenticados realicen solicitudes de la API a los grupos de identidades de Amazon Cognito y a Amazon Cognito Sync. Otras Servicios de AWS podrían imponer restricciones al acceso a los servicios desde las identidades web.

En una solicitud correcta con el flujo mejorado, Amazon Cognito realiza una solicitud de API `AssumeRoleWithWebIdentity` en segundo plano. Entre los parámetros de esta solicitud, Amazon Cognito incluye los siguientes.

1. El ID de identidad de su usuario.
2. El ARN del rol de IAM que el usuario desea asumir.
3. Un parámetro `policy` que agrega una política de sesión en línea.
4. `PolicyArns.member.N`Parámetro cuyo valor es una política AWS gestionada que concede permisos adicionales en Amazon CloudWatch.

Servicios a los que pueden acceder los usuarios no autenticados

Cuando utiliza el flujo mejorado, las políticas de ámbito reducido que Amazon Cognito aplica a la sesión del usuario impiden que utilice otros servicios que no sean los que se muestran en la siguiente tabla. Para un subconjunto de servicios, solo se permiten acciones específicas.

Categoría	Servicio
Análisis	Amazon Data Firehose Amazon Managed Service para Apache Flink
Integración de aplicaciones	Amazon Simple Queue Service
Realidad aumentada y realidad virtual	Amazon Sumerian[1]

Categoría	Servicio
Aplicaciones empresariales	Amazon Mobile Analytics Amazon Simple Email Service
Cálculo	AWS Lambda
Criptografía y PKI	AWS Key Management Service[1]
Base de datos	Amazon DynamoDB Amazon SimpleDB
Web y móvil front-end	AWS AppSync Amazon Location Service Amazon Simple Notification Service Amazon Pinpoint
Desarrollo de juegos	Amazon GameLift
Internet de las cosas (IoT)	AWS IoT

Categoría	Servicio
Machine Learning	Amazon CodeWhisperer
	Amazon Comprehend
	Amazon Lex
	Amazon Machine Learning
	Amazon Personalize
	Amazon Polly
	Amazon Rekognition
	Amazon SageMaker [1]
	Amazon Textract[1]
	Amazon Transcribe
	Amazon Translate
Administración y gobernanza	Amazon CloudWatch
	Amazon CloudWatch Logs
Redes y entrega de contenido	Amazon API Gateway
Seguridad, identidad y conformidad	Grupos de usuarios de Amazon Cognito
Almacenamiento	Amazon Simple Storage Service

[1] Servicios de AWS En el caso de la siguiente tabla, la política en línea concede un subconjunto de acciones. En la tabla se muestran las acciones disponibles en cada uno.

Servicio de AWS	Permisos máximos para usuarios de flujo mejorado no autenticados

AWS Key Management Service	`Encrypt`
	`Decrypt`
	`ReEncrypt`
	`GenerateDataKey`

Amazon SageMaker	`InvokeEndpoint`
Amazon Textract	`DetectDocumentText` `AnalyzeDocument`
Amazon Sumerian	`View*`

Para conceder acceso Servicios de AWS más allá de esta lista, active el flujo de autenticación básico (clásico) en su grupo de identidades. Si los usuarios ven errores `NotAuthorizedException` de Servicios de AWS que están permitidos por las políticas asignadas al rol de IAM para usuarios no autenticados, evalúe si puede eliminar ese servicio del caso de uso. Si no puede, cambie al flujo básico.

La política de sesión en línea

La política de sesión integrada impide que los permisos efectivos de tu usuario incluyan el acceso a cualquier permiso Servicios de AWS ajeno a los de la siguiente lista. También debe concederles permisos Servicios de AWS en las políticas que aplique a la función de IAM del usuario. Los permisos efectivos de un usuario para una sesión de rol asumido son la intersección de las políticas asignadas a su rol y su política de sesión. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de AWS Identity and Access Management .

Amazon Cognito agrega la siguiente política insertada en las sesiones de los usuarios en Regiones de AWS que están habilitadas de forma predeterminada.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:*",
                "logs:*",
                "dynamodb:*",
                "kinesis:*",
                "mobileanalytics:*",
                "s3:*",
                "ses:*",
                "sns:*",
                "sqs:*",
                "lambda:*",
                "machinelearning:*",
                "execute-api:*",
                "iot:*",
                "gamelift:*",
                "scs:*",
                "cognito-identity:*",
                "cognito-idp:*",
                "lex:*",
                "polly:*",
                "comprehend:*",
                "translate:*",
                "transcribe:*",
                "rekognition:*",
                "mobiletargeting:*",
                "firehose:*",
                "appsync:*",
                "personalize:*",
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "sagemaker:InvokeEndpoint",
                "cognito-sync:*",
                "sumerian:View*",
                "codewhisperer:*",
                "textract:DetectDocumentText",
                "textract:AnalyzeDocument",
                "sdb:*"
            ],
            "Resource": [
                "*"
```

```
            ]
        }
    ]
}
```

Para todas las demás regiones, la política de ámbito reducido insertada incluye todo lo que se muestra en las regiones predeterminadas, excepto las siguientes instrucciones `Action`.

```
                "cognito-sync:*",
                "sumerian:View*",
                "codewhisperer:*",
                "textract:DetectDocumentText",
                "textract:AnalyzeDocument",
                "sdb:*"
```

La política de sesiones AWS gestionadas

Amazon Cognito también limita el alcance de los permisos de los usuarios no autenticados con la política administrada de AWS `AmazonCognitoUnAuthedIdentitiesSessionPolicy` a los usuarios no autenticados en el flujo mejorado. También debe conceder este permiso en las políticas que asocie a su rol de IAM no autenticado.

La política administrada por `AmazonCognitoUnAuthedIdentitiesSessionPolicy` contiene los permisos siguientes.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "rum:PutRumEvents",
            "polly:*",
            "comprehend:*",
            "translate:*",
            "transcribe:*",
            "rekognition:*",
            "mobiletargeting:*",
            "firehose:*",
            "personalize:*",
            "sagemaker:InvokeEndpoint"
        ],
```

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

Ejemplos de políticas de acceso

En esta sección, puede encontrar políticas de acceso de Amazon Cognito de ejemplo que conceden a los usuarios los permisos mínimos necesarios para realizar operaciones específicas. Puede limitar aún más los permisos de un determinado ID de identidad utilizando variables de política siempre que sea posible. Por ejemplo, utilizando ${cognito-identity.amazonaws.com:sub}. Para obtener más información, consulte Información sobre la parte 3 de la autenticación de Amazon Cognito: roles y políticas en el blog de AWS Mobile.

> **Note**
>
> Como práctica recomendada de seguridad, las políticas deben incluir únicamente los permisos que los usuarios necesitan para realizar sus tareas. Esto significa que debe intentar siempre el acceso a una identidad individual para objetos cuando sea posible.

Otorgar a una identidad acceso de lectura a un único objeto en Amazon S3

La siguiente política de acceso concede permisos de lectura a una identidad para recuperar un único objeto de un determinado bucket de S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/assets/my_picture.jpg"]
    }
  ]
}
```

Otorgar a una identidad acceso de lectura y escritura a rutas específicas de identidad en Amazon S3

La siguiente política de acceso concede permisos de lectura y escritura para obtener acceso a una "carpeta" de prefijo específico en un bucket de S3 mapeando el prefijo a la variable `${cognito-identity.amazonaws.com:sub}`.

Con esta política, una identidad como `us-east-1:12345678-1234-1234-1234-123456790ab` insertada a través de `${cognito-identity.amazonaws.com:sub}` puede obtener, colocar y enumerar objetos en `arn:aws:s3:::mybucket/us-east-1:12345678-1234-1234-1234-123456790ab`. Sin embargo, la identidad no concedería acceso a otros objetos en `arn:aws:s3:::mybucket`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket"],
      "Condition": {"StringLike": {"s3:prefix": ["${cognito-
identity.amazonaws.com:sub}/*"]}}
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/${cognito-identity.amazonaws.com:sub}/*"]
    }
  ]
}
```

Asignar a identidades acceso detallado a Amazon DynamoDB

La siguiente política de acceso proporciona un control de acceso minucioso a los recursos de DynamoDB mediante variables de entorno de Amazon Cognito. Estas variables otorgan acceso a los elementos de DynamoDB por ID de identidad: Para obtener más información, consulte Uso de condiciones de políticas de IAM para control de acceso preciso en la Guía para desarrolladores de Amazon DynamoDB.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource": [
        "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"
      ],
      "Condition": {
        "ForAllValues:StringEquals": {
          "dynamodb:LeadingKeys":  ["${cognito-identity.amazonaws.com:sub}"]
        }
      }
    }
  ]
}
```

Otorgar a una identidad permiso para llamar a una función de Lambda

La siguiente política de acceso concede un permiso de identidad para invocar una función de Lambda.

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
             "Effect": "Allow",
             "Action": "lambda:InvokeFunction",
             "Resource": [
                 "arn:aws:lambda:us-west-2:123456789012:function:MyFunction"
             ]
         }
     ]
 }
```

Otorgar a una identidad permiso para publicar registros en Kinesis Data Streams

La siguiente política de acceso permite a una identidad utilizar la operación `PutRecord` con cualquiera de los Kinesis Data Streams. Se puede aplicar a los usuarios que necesitan añadir registros de datos a todos los flujos de una cuenta. Para obtener más información, consulte Control del acceso a los recursos de Amazon Kinesis Data Streams por medio de IAM en la Guía para desarrolladores de Amazon Kinesis Data Streams.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kinesis:PutRecord",
            "Resource": [
                "arn:aws:kinesis:us-east-1:111122223333:stream/stream1"
            ]
        }
    ]
}
```

Otorgar a una identidad acceso a sus datos en el almacén de Amazon Cognito Sync

La siguiente política de acceso solo concede permisos de identidad para acceder a sus propios datos en el almacén de Amazon Cognito Sync.

```
{
  "Version": "2012-10-17",
  "Statement":[{
      "Effect":"Allow",
      "Action":"cognito-sync:*",
      "Resource":["arn:aws:cognito-sync:us-east-1:123456789012:identitypool/${cognito-
identity.amazonaws.com:aud}/identity/${cognito-identity.amazonaws.com:sub}/*"]
      }]
  }
```

Confianza y permisos de rol

Estos roles se diferencian en sus relaciones de confianza. Este es un ejemplo de política de confianza para roles no autenticados:

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-corner-
cafe-123456790ab"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "unauthenticated"
        }
      }
    }
  ]
}
```

Esta política permite a los usuarios federados de `cognito-identity.amazonaws.com` (el emisor del token de OpenID Connect) asumir este rol. Además, la política restringe el `aud` del token, en este caso, el ID del grupo de identidades para adaptarse al grupo de identidades. Por último, la política especifica que uno de los miembros de la matriz de la notificación multivalor `amr` del token emitido por la operación de la API `GetOpenIdToken` de Amazon Cognito tiene el valor `unauthenticated`.

Cuando Amazon Cognito crea un token, establece el `amr` del token como `unauthenticated` o `authenticated`. Si `amr` está `authenticated`, el token incluye todos los proveedores utilizados durante la autenticación. Esto significa que puede crear un rol que confíe solo en los usuarios que iniciaron sesión a través de Facebook, cambiando la condición `amr`, como en el ejemplo siguiente:

```
"ForAnyValue:StringLike": {
  "cognito-identity.amazonaws.com:amr": "graph.facebook.com"
}
```

Sea prudente cuando cambie las relaciones de confianza de sus roles o cuando intente utilizar roles en todos los grupos de identidades. Si el rol no está configurado correctamente para confiar en su grupo de identidades, se visualizará una excepción de STS como la siguiente:

```
AccessDenied -- Not authorized to perform sts:AssumeRoleWithWebIdentity
```

Si ve este mensaje, compruebe que está utilizando un rol adecuado para el grupo de identidades y el tipo de autenticación.

Prácticas recomendadas de seguridad para los grupos de identidades de Amazon Cognito

Los grupos de identidades de Amazon Cognito proporcionan AWS credenciales temporales para su aplicación. Cuentas de AWS suelen contener tanto los recursos que necesitan los usuarios de la aplicación como los recursos de back-end privados. Las funciones y políticas de IAM que componen las AWS credenciales pueden conceder acceso a cualquiera de estos recursos.

La mejor práctica principal a la hora de configurar el grupo de identidades es garantizar que la aplicación pueda realizar su trabajo sin privilegios excesivos o involuntarios. Para evitar errores de configuración de seguridad, revise estas recomendaciones antes del lanzamiento de cada aplicación que desee lanzar a producción.

Temas

Prácticas recomendadas de configuración de IAM

Cuando un usuario invitado o autenticado inicia una sesión en la aplicación que requiere credenciales del grupo de identidades, la aplicación recupera las AWS credenciales temporales para una función de IAM. Las credenciales pueden ser para un rol predeterminado, un rol elegido según las reglas de la configuración del grupo de identidades o para un rol personalizado elegido por la aplicación. Con los permisos asignados a cada rol, tu usuario obtiene acceso a tus AWS recursos.

Para obtener más información sobre las prácticas recomendadas generales de IAM, consulte las mejores prácticas de IAM en la Guía del AWS Identity and Access Management usuario.

Utilice condiciones de política de confianza en las funciones de IAM

La IAM exige que las funciones de los grupos de identidades tengan al menos una condición de política de confianza. Esta condición puede, por ejemplo, establecer el ámbito del rol solo para los usuarios autenticados. AWS STS también requiere que las solicitudes de autenticación básica

multicuenta tengan dos condiciones específicas: `cognito-identity.amazonaws.com:aud` y. `cognito-identity.amazonaws.com:amr` Como práctica recomendada, aplique estas dos condiciones a todas las funciones de IAM que confíen en el principal servicio de grupos de identidades. `cognito-identity.amazonaws.com`

- `cognito-identity.amazonaws.com:aud`: La afirmación aud del token del grupo de identidades debe coincidir con un ID de grupo de identidades confiable.
- `cognito-identity.amazonaws.com:amr`: La afirmación amr del token del conjunto de identidades debe estar autenticada o no autenticada. Con esta condición, puedes reservar el acceso a un rol solo a los invitados no autenticados o solo a los usuarios autenticados. Puede restringir aún más el valor de esta condición para restringir el rol a los usuarios de un proveedor específico, por ejemplo. `graph.facebook.com`

El siguiente ejemplo de política de confianza de roles otorga acceso a un rol en las siguientes condiciones:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
```

Elementos relacionados con los grupos de identidades

- `"Federated": "cognito-identity.amazonaws.com"`: Los usuarios deben provenir de un grupo de identidades.
- `"cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-cdef-example11111"`: Los usuarios deben provenir del grupo de identidades específicous-east-1:a1b2c3d4-5678-90ab-cdef-example11111.
- `"cognito-identity.amazonaws.com:amr": "authenticated"`: Los usuarios deben estar autenticados. Los usuarios invitados no pueden asumir el rol.

Aplica permisos con privilegios mínimos

Al establecer permisos con las políticas de IAM para el acceso autenticado o el acceso de invitados, conceda únicamente los permisos específicos necesarios para realizar tareas específicas, o los permisos con privilegios mínimos. El siguiente ejemplo de política de IAM, cuando se aplica a un rol, otorga acceso de solo lectura a un único archivo de imagen en un bucket de Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/assets/my_picture.jpg"]
    }
  ]
}
```

Prácticas recomendadas para la configuración del grupo de identidades

Los grupos de identidades tienen opciones flexibles para la generación de AWS credenciales. No utilice atajos de diseño cuando su aplicación puede funcionar con los métodos más seguros.

Comprenda los efectos del acceso de invitados

El acceso de invitados no autenticado permite a los usuarios recuperar tus datos Cuenta de AWS antes de iniciar sesión. Cualquier persona que conozca el ID de su grupo de identidades

puede solicitar credenciales no autenticadas. El ID de su grupo de identidades no es información confidencial. Al activar el acceso como invitado, los AWS permisos que concedas a las sesiones no autenticadas están disponibles para todos.

Como práctica recomendada, deja el acceso como invitado desactivado y recupera los recursos necesarios solo después de que los usuarios se hayan autenticado. Si su aplicación requiere acceso a los recursos antes de iniciar sesión, tome las siguientes precauciones.

- Familiarícese con las limitaciones automáticas que se imponen a los roles no autenticados.
- Supervise y ajuste los permisos de sus funciones de IAM no autenticadas para que se adapten a las necesidades específicas de su aplicación.
- Conceda acceso a recursos específicos.
- Proteja la política de confianza de su función de IAM no autenticada predeterminada.
- Active el acceso como invitado solo cuando esté seguro de que va a conceder los permisos de su función de IAM a cualquier usuario de Internet.

Usa la autenticación mejorada de forma predeterminada

Con la autenticación básica (clásica), Amazon Cognito delega la selección del rol de IAM en su aplicación. Por el contrario, el flujo mejorado utiliza la lógica centralizada del grupo de identidades para determinar la función de IAM. También proporciona seguridad adicional para las identidades no autenticadas con una política de alcance reducido que establece un límite máximo para los permisos de IAM. El flujo mejorado es la opción más segura con el nivel más bajo de esfuerzo de los desarrolladores. Para obtener más información sobre estas opciones, consulteFlujo de autenticación de grupos de identidades (identidades federadas).

El flujo básico puede exponer la lógica del lado del cliente que interviene en la selección de funciones y en el ensamblaje de la solicitud de credenciales de la API de AWS STS. El flujo mejorado oculta tanto la lógica como la solicitud de asumir el rol que hay detrás de la automatización del grupo de identidades.

Al configurar la autenticación básica, aplique las mejores prácticas de IAM a sus funciones de IAM y a sus permisos.

Utilice los proveedores de desarrolladores de forma segura

Las identidades autenticadas por los desarrolladores son una característica de los grupos de identidades para las aplicaciones del lado del servidor. La única prueba de autenticación que

los grupos de identidades requieren para la autenticación de los desarrolladores son las AWS credenciales de un desarrollador de grupos de identidades. Los grupos de identidades no imponen ninguna restricción a la validez de los identificadores entre desarrolladores y proveedores que se presentan en este flujo de autenticación.

Como práctica recomendada, implementa proveedores de desarrolladores únicamente en las siguientes condiciones:

- Para crear responsabilidad por el uso de las credenciales autenticadas por el desarrollador, diseñe el nombre y los identificadores del proveedor desarrollador para indicar la fuente de autenticación. Por ejemplo: `"Logins" : {"MyCorp provider" : "[provider application ID]"}`.
- Evite las credenciales de usuario de larga duración. Configure su cliente del lado del servidor para solicitar identidades con funciones vinculadas a servicios, como perfiles de instancia EC2 y funciones de ejecución de Lambda.
- Evite mezclar fuentes de confianza internas y externas en el mismo grupo de identidades. Agregue su proveedor de desarrolladores y sus proveedores de inicio de sesión único (SSO) en grupos de identidades separados.

Uso de atributos para el control de acceso

Los atributos para el control de acceso es la implementación de los grupos de identidades de Amazon Cognito del control de acceso basado en atributos (ABAC). Puede utilizar políticas de IAM para controlar el acceso a los recursos de AWS a través de los grupos de identidades de Amazon Cognito en función de los atributos del usuario. Estos atributos pueden extraerse de los proveedores de identidad social y corporativa. Puede mapear atributos dentro de los tokens de acceso e ID de los proveedores o de las aserciones SAML a etiquetas a las que se puede hacer referencia en las políticas de permisos de IAM.

Puede elegir mapeos predeterminados o crear sus propios mapeos personalizados en grupos de identidades de Amazon Cognito. Los mapeos predeterminados permiten escribir políticas de IAM basadas en un conjunto fijo de atributos de usuario. Los mapeos personalizados permiten seleccionar un conjunto personalizado de atributos de usuario a los que se hace referencia en las políticas de permisos de IAM. Los nombres de atributos de la consola de Amazon Cognito se mapean en la clave de etiqueta del principal, que son las etiquetas a las que se hace referencia en la política de permisos de IAM.

Por ejemplo, supongamos que tiene un servicio de streaming multimedia con una pertenencia gratuita y otra de pago. Almacena los archivos multimedia en Amazon S3 y los etiqueta con etiquetas gratuitas o prémium. Puede utilizar atributos de control de acceso para permitir el acceso a contenido gratuito y de pago basado en el nivel de pertenencia del usuario, que es parte del perfil del usuario. Puede mapear el atributo de la membresía a una clave de etiqueta para que el principal pase a la política de permisos de IAM. De esta forma, puede crear una única política de permisos y permitir condicionalmente el acceso a los contenidos prémium en función del valor del nivel de membresía y de la etiqueta de los archivos de contenido.

Temas

El uso de atributos para controlar el acceso aporta varios beneficios:

- La administración de permisos es más fácil cuando se utilizan atributos para el control de acceso. Puede crear una política de permisos básica en la que se utilicen atributos de usuario, en lugar de crear varias políticas para diferentes funciones de trabajo.
- No es necesario que actualice las políticas cada vez que agregue o quite recursos o usuarios de la aplicación. La política de permisos solo concederá el acceso a los usuarios con los atributos de usuario coincidentes. Por ejemplo, es posible que deba controlar el acceso a determinados buckets de S3 en función del título de trabajo de los usuarios. En ese caso, puede crear una política de permisos para permitir que solo los usuarios dentro del título de trabajo definido accedan a estos archivos. Para obtener más información, consulte Tutorial de IAM: Uso de etiquetas de sesión SAML para ABAC.
- Los atributos se pueden pasar como etiquetas principales a una política que permite o rechaza los permisos en función de los valores de esos atributos.

Uso de atributos para el control de acceso con grupos de identidades de Amazon Cognito

Antes de utilizar atributos para el control de acceso, asegúrese de cumplir los siguientes requisitos previos:

- Una cuenta de AWS.
- Grupo de usuarios
- Grupo de identidades
- Configurar un SDK
- Integración de proveedores de identidad
- Credenciales

Para utilizar los atributos para el control de acceso, la Reclamación que establece como origen de datos establece el valor de la Clave de etiqueta que elija. Amazon Cognito aplica la clave y el valor de la etiqueta a la sesión del usuario. Las políticas de IAM pueden evaluar el acceso del usuario a partir de la condición `${aws:PrincipalTag/tagkey}`. IAM evalúa el valor de la etiqueta del usuario en función de la política.

Debe preparar los roles de IAM cuyas credenciales desee transmitir a los usuarios. La política de confianza de estos roles debe permitir a Amazon Cognito asumir el rol para el usuario. Para los atributos para el control de acceso, también debe permitir que Amazon Cognito aplique las etiquetas de las entidades principales a la sesión temporal del usuario. Conceda permiso para asumir el rol con la acción AssumeRoleWithWebIdentity. Conceda permiso para etiquetar las sesiones de los usuarios con la acción de solo permiso `sts:TagSession`. Para obtener más información, consulte Transferencia de etiquetas de sesión en AWS Security Token Service en la Guía del usuario de AWS Identity and Access Management. Para una política de confianza de ejemplo que concede permisos `sts:AssumeRoleWithWebIdentity` y `sts:TagSession` a la entidad principal de servicio de Amazon Cognito `cognito-identity.amazonaws.com`, consulte Ejemplo de política de uso de atributos para el control de acceso.

Para configurar atributos para el control de acceso en la consola

1. Inicie sesión en la consola de Amazon Cognito y seleccione Grupos de identidades. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Localice proveedores de identidades. Elija el proveedor de identidades que desea editar. Si quiere agregar un nuevo IdP, seleccione Agregar proveedor de identidades.
4. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, elija Editar en Atributos para el control de acceso.

a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

5. Seleccione Save changes (Guardar cambios).

Ejemplo de política de uso de atributos para el control de acceso

Piense en una situación en la que un empleado del departamento legal de una empresa necesita enumerar todos los archivos en buckets que pertenecen a su departamento y están clasificados con su nivel de seguridad. Supongamos que el token que este empleado obtiene del proveedor de identidad contiene las siguientes notificaciones.

Notificaciones

```
{ .
  .
"sub" : "57e7b692-4f66-480d-98b8-45a6729b4c88",
"department" : "legal",
"clearance" : "confidential",
 .
 .
}
```

Estos atributos pueden mapearse a etiquetas y hacerse referencia en las políticas de permisos de IAM como etiquetas principales. Ahora puede administrar el acceso si cambia el perfil de usuario al final del proveedor de identidades. Como alternativa, puede cambiar atributos en el lado del recurso mediante nombres o etiquetas sin cambiar la propia política.

La siguiente política de permisos realiza dos tareas:

- Permite el acceso a la lista a todos los buckets de S3 que terminan con un prefijo que coincide con el nombre del departamento del usuario.

- Permite el acceso de lectura en los archivos de estos buckets, siempre y cuando la etiqueta de autorización del archivo coincida con el atributo de autorización del usuario.

Política de permisos

```
   {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:List*",
            "Resource": "arn:aws:s3:::*-${aws:PrincipalTag/department}"
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject*",
            "Resource": "arn:aws:s3:::*-${aws:PrincipalTag/department}/*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/clearance": "${aws:PrincipalTag/clearance}"
                }
            }
        }
    ]
}
```

La política de confianza determina quién puede asumir este rol. La política de relación de confianza permite el uso de `sts:AssumeRoleWithWebIdentity` y `sts:TagSession` para permitir el acceso. Agrega condiciones para restringir la política al grupo de identidades que ha creado y se asegura de que es para un rol autenticado.

Política de confianza

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRoleWithWebIdentity",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "IDENTITY-POOL-ID"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
```

Desactivar atributos para el control de acceso (consola)

Siga este procedimiento para desactivar los atributos para el control de acceso.

Para desactivar atributos para el control de acceso en la consola

1. Inicie sesión en la consola de Amazon Cognito y seleccione Grupos de identidades. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Localice proveedores de identidades. Elija el proveedor de identidades que desea editar.
4. Elija Editar en Atributos para el control de acceso.
5. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.
6. Seleccione Save changes (Guardar cambios).

Mapeos de proveedores predeterminados

En la siguiente tabla, se encuentra la información de mapeo predeterminado para los proveedores de autenticación que admite Amazon Cognito.

Proveedor	Tipo de token	Valores de etiquetas del principal	Ejemplo
Grupos de usuarios de Amazon Cognito	Token de ID	aud(ID de cliente) y sub(ID de usuario)	"6jk8ltokc7ac9es6jrtg9q572f", "57e7b692-4f66-480d-98b8-45a6729b4c88"
Facebook	Token de acceso	aud(app_id), sub(user_id)	"492844718097981", "112177216992379"
Google	Token de ID	aud(ID de cliente) y sub(ID de usuario)	"620493171733-eebk7c0hcp5lj3e1tlqp1gntt3k0rncv.apps.googleusercontent.com", "109220063452404746097"
SAML	Aserciones	"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier" , "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"	"auth0|5e28d196f8f55a0eaaa95de3", "user123@gmail.com"
Apple	Token de ID	aud(ID de cliente) y sub(ID de usuario)	"com.amazonaws.ec2-54-80-172-243.compute-1.client", "001968.a6ca34e9c1e742458a26cf8005854be9.0733"
Amazon	Token de acceso	aud(ID de cliente en Amzn Dev Ac), user_id(ID de usuario)	"amzn1.application-oa2-client.9d70d9382d3446108aaee3dd763a0fa6", "amzn1.account.AGH

Proveedor	Tipo de token	Valores de etiquetas del principal	Ejemplo
			NIFJQMFSB G3G6XCPVB 35ORQAA"
Proveedores estándar de OIDC	Tokens de ID y de acceso	aud (como client_id), sub (como user ID)	"620493171733-eebk 7c0hcp5lj3e1tlqp1g ntt3k0rncv.apps.go ogleusercontent.com", "10922006345240474 6097"
Twitter	Token de acceso	aud (ID de la aplicació n; secreto de la aplicación), sub (ID de usuario)	"DfwifTtKEX1FiIBRn OTIR0CFK; Xgj5xb8xIrIVCPjXgL IdkW7fXmw cJJrFvnoK9gwZkLexo 1y5z1", "12690038 84292222976"
DevAuth	Mapeo	No aplicable	"tag1", "tag2"

> **Note**
>
> La opción de los mapeos de atributos predeterminados se completa de forma automática en los nombres Tag Key for Principal (Clave de etiquetas del principal) y Attribute (Atributo). No se pueden cambiar los mapeos predeterminados.

Uso del control de acceso basado en roles

Los grupos de identidades de Amazon Cognito asignan a los usuarios autenticados un conjunto de credenciales temporales con privilegios limitados para acceder a sus recursos. AWS Los permisos de cada usuario se controlan mediante los roles de IAM que cree. Puede definir reglas para elegir el rol de cada usuario en función de las notificaciones contenidas en el token de ID. Puede definir un rol

predeterminado para los usuarios autenticados. También puede definir un rol de IAM independiente con permisos limitados para los usuarios invitados que no estén autenticados.

Creación de roles para la asignación de roles

Es importante agregar la política de confianza adecuada para cada rol de forma que Amazon Cognito solo lo pueda asumir para los usuarios autenticados del grupo de identidades. A continuación se muestra un ejemplo de política de confianza:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:12345678-corner-
cafe-123456790ab"
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
```

Con esta política, los usuarios federados de `cognito-identity.amazonaws.com` (el emisor del token de OpenID Connect) pueden asumir este rol. Además, la política restringe el `aud` del token, en este caso, el ID del grupo de identidades para adaptarse al grupo de identidades. Por último, la política especifica que uno de los miembros de la matriz de la notificación multivalor `amr` del token emitido por la acción de la API `GetOpenIdToken` de Amazon Cognito tiene el valor `authenticated`.

Concesión del permiso para transmitir roles

Para permitir que un usuario establezca roles con permisos superiores a los permisos existentes del usuario en un grupo de identidades, concédale el permiso `iam:PassRole` para pasar el rol a la API `set-identity-pool-roles`. Por ejemplo, si el usuario no puede escribir en Amazon S3, pero el rol de IAM que el usuario establece en el grupo de identidades concede permiso de escritura en Amazon S3, el usuario solo podrá definir ese rol si el rol tiene concedido el permiso `iam:PassRole`. En el ejemplo de política siguiente se muestra cómo conceder el permiso `iam:PassRole`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::123456789012:role/myS3WriteAccessRole"
            ]
        }
    ]
}
```

En este ejemplo de política, se concede el permiso `iam:PassRole` para el rol `myS3WriteAccessRole`. El rol se especifica mediante el nombre de recurso de Amazon (ARN) del rol. También debe adjuntar esta política a su usuario. Para obtener más información, consulte Uso de políticas administradas.

> **Note**
>
> Las funciones de Lambda utilizan una política basada en recursos. Esta política está directamente asociada a la función de Lambda en sí. Cuando crea una regla que invoca una función de Lambda, no transmite un rol, por lo que el usuario que crea la regla no necesita el permiso `iam:PassRole`. Para obtener más información sobre las autorizaciones de funciones de Lambda, consulte Administración de permisos: uso de una política de funciones de Lambda.

Uso de tokens para asignar roles a usuarios

En el caso de los usuarios que inicien sesión mediante los grupos de usuarios de Amazon Cognito, los roles se pueden pasar en el token de ID que asignó el grupo de usuarios. Los roles aparecen en las siguientes notificaciones del token de ID:

- La notificación `cognito:preferred_role` es el ARN del rol.
- La notificación `cognito:roles` es una cadena separada por comas que contiene un conjunto de ARN de roles permitidos.

Las notificaciones se establecen como sigue:

- La notificación `cognito:preferred_role` se establece en el rol del grupo con el mejor valor `Precedence` (menor). Si solo se permite un rol, `cognito:preferred_role` se establece en dicho rol. Si hay varios roles y ninguno tiene la mejor prioridad, esta notificación no se establece.
- La notificación `cognito:roles` se establece si hay al menos un rol.

Cuando se utilizan tokens para asignar roles, si hay varios roles que se pueden asignar al usuario, el grupo de identidades de Amazon Cognito (identidades federadas) elige el rol de la siguiente manera:

- Utilice el GetCredentialsForIdentity`CustomRoleArn`parámetro si está establecido y coincide con un rol de la reclamación. `cognito:roles` Si este parámetro no coincide con un rol de `cognito:roles`, deniegue el acceso.
- Si la notificación `cognito:preferred_role` está establecida, utilícela.
- Si la `cognito:preferred_role` afirmación no está establecida, se establece y no `CustomRoleArn` se especifica en la llamada a`GetCredentialsForIdentity`, se utiliza la configuración de resolución de roles de la consola o del `AmbiguousRoleResolution` campo (en el `RoleMappings` parámetro de la SetIdentityPoolRolesAPI) para determinar la función que se va a asignar. `cognito:roles`

Uso de la asignación basada en reglas para la asignación de roles a los usuarios

Con las reglas, se pueden mapear notificaciones de un token de proveedor de identidad a roles de IAM.

Cada regla especifica una notificación de token (como un atributo de usuario en el token de ID de un grupo de usuarios de Amazon Cognito), el tipo de coincidencia, un valor y un rol de IAM. El tipo de asociación puede ser `Equals`, `NotEqual`, `StartsWitho` `Contains`. Si un usuario tiene un valor coincidente con la notificación, dicho usuario puede asumir ese rol cuando obtenga las credenciales. Por ejemplo, puede crear una regla con la que se asigne un rol de IAM específico a usuarios que tengan un valor de atributo personalizado `custom:dept` de `Sales`.

> **Note**
>
> En la configuración de una regla, los atributos personalizados deben tener el prefijo `custom:` para diferenciarse de los atributos estándar.

Las reglas se evalúan en orden y se usa el rol de IAM para la primera regla de coincidencia, a menos que se haya especificado `CustomRoleArn` para anular el orden. Para obtener más información sobre los atributos de usuario en los grupos de usuarios de Amazon Cognito, consulte Custom pool attributes (.

Puede configurar varias reglas para un proveedor de autenticación en la consola del grupo de identidades (identidades federadas). Las reglas se aplican en orden. Si quiere cambiar el orden, puede arrastrar las reglas. La primera regla coincidente tiene prioridad. Si el tipo de asociación es `NotEqual` y la notificación no existe, no se evaluará la regla. Si no hay reglas que coincidan, el ajuste de Resolución de rol se aplica a Usar rol autenticado predeterminado o Denegar solicitud.

En la API y la CLI, puede especificar el rol que se asignará cuando ninguna regla coincida en el `AmbiguousRoleResolution` campo del RoleMappingtipo, que se especifica en el `RoleMappings` parámetro de la SetIdentityPoolRolesAPI.

Puede configurar el mapeo basado en reglas para los proveedores de identidad de OpenID Connect (OIDC) y SAML en la API AWS CLI o con el campo del tipo. `RulesConfiguration` RoleMapping Puede especificar este campo en el parámetro de la API. `RoleMappings` SetIdentityPoolRoles Por el AWS Management Console momento, no permite añadir reglas para los proveedores de OIDC o SAML.

Por ejemplo, el siguiente AWS CLI comando agrega una regla que asigna la función `arn:aws:iam::123456789012:role/Sacramento_team_S3_admin` a los usuarios de su ubicación de Sacramento que fueron autenticados por el IdP de OIDC: `arn:aws:iam::123456789012:oidc-provider/myOIDCIdP`

```
aws cognito-identity set-identity-pool-roles --region us-east-1 --cli-input-json
 file://role-mapping.json
```

Contenido de **`role-mapping.json`**:

```
{
    "IdentityPoolId": "us-east-1:12345678-corner-cafe-123456790ab",
    "Roles": {
        "authenticated": "arn:aws:iam::123456789012:role/myS3WriteAccessRole",
        "unauthenticated": "arn:aws:iam::123456789012:role/myS3ReadAccessRole"
    },
    "RoleMappings": {
        "arn:aws:iam::123456789012:oidc-provider/myOIDCIdP": {
            "Type": "Rules",
            "AmbiguousRoleResolution": "AuthenticatedRole",
            "RulesConfiguration": {
                "Rules": [
                    {
                        "Claim": "locale",
                        "MatchType": "Equals",
                        "Value": "Sacramento",
                        "RoleARN": "arn:aws:iam::123456789012:role/
Sacramento_team_S3_admin"
                    }
                ]
            }
        }
    }
}
```

Por cada grupo de usuarios u otro proveedor de autenticación que configure para un grupo de identidades, se pueden crear hasta 25 reglas. Este límite no se puede ajustar. Para obtener más información, consulte el tema sobre cuotas de Amazon Cognito.

Notificaciones de token para usarlas en una asignación basada en reglas

Amazon Cognito

Un token de ID de Amazon Cognito se representa como un JSON Web Token (JWT). El token contiene notificaciones sobre la identidad del usuario autenticado, como por ejemplo `name`, `family_name`y `phone_number`. Para obtener más información acerca de las notificaciones

estándar, consulte la especificación OpenID Connect. Aparte de las notificaciones estándar, a continuación indicamos otras notificaciones específicas de Amazon Cognito:

- `cognito:groups`
- `cognito:roles`
- `cognito:preferred_role`

Amazon

Las notificaciones siguientes, junto con los valores posibles de dichas notificaciones, se pueden utilizar con Login with Amazon:

- `iss`: www.amazon.com
- `aud`: ID de aplicación
- `sub`: `sub` desde el token de Login with Amazon

Facebook

Las notificaciones siguientes, junto con los valores posibles de dichas notificaciones, se pueden utilizar con Facebook:

- `iss`: graph.facebook.com
- `aud`: ID de aplicación
- `sub`: `sub` del token de Facebook

Google

Un token de Google contiene notificaciones estándar de la especificación OpenID Connect. Todas las notificaciones del token de OpenID están disponibles para el mapeo basado en reglas. Consulte el sitio de OpenID Connect de Google para obtener información sobre las notificaciones disponibles en el token de Google.

Apple

Un token de Apple contiene notificaciones estándar de la Especificación OpenID Connect. Consulte Autenticación de usuarios con Sign in with Apple en la documentación de Apple para obtener más información sobre la notificación disponible del token de Apple. El token de Apple no contiene siempre `email`.

OpenID

Todas las notificaciones del token de Open ID están disponibles para el mapeo basado en reglas. Para obtener más información acerca de las notificaciones estándar, consulte la especificación OpenID Connect. Consulte la documentación del proveedor de OpenID para obtener información adicional acerca de las notificaciones que están disponibles.

SAML

Las notificaciones se analizan en la aserción de SAML recibida. Todas las notificaciones que están disponibles en la aserción de SAML se pueden utilizar en el mapeo basado en reglas.

Prácticas recomendadas para el control de acceso basado en roles

> **Important**
>
> Si la notificación que está mapeando a un rol la puede modificar el usuario final, cualquier usuario final puede asumir su rol y definir la política en consecuencia. Asigne únicamente las notificaciones que el usuario final no puede establecer directamente a los roles con permisos elevados. En un grupo de usuarios de Amazon Cognito, puede establecer permisos de lectura y escritura por aplicación para cada atributo de usuario.

> **Important**
>
> Si establece roles para grupos en un grupo de usuarios de Amazon Cognito, estos roles se transfieren por medio del token de ID del usuario. Para utilizar estos roles, también debe establecer Choose role from token (Elegir rol a partir de un token) para la selección de roles autenticados para el grupo de identidades.
> Puedes usar la configuración de resolución de roles de la consola y el `RoleMappings` parámetro de la SetIdentityPoolRolesAPI para especificar cuál es el comportamiento predeterminado cuando no se puede determinar el rol correcto a partir del token.

Obtención de credenciales

Puede usar Amazon Cognito para entregar credenciales temporales con privilegios limitados a su aplicación, de modo que sus usuarios puedan acceder a los recursos. AWS En esta sección, se

describe cómo obtener credenciales y cómo recuperar una identidad de Amazon Cognito de un grupo de identidades.

Amazon Cognito es compatible con las identidades autenticadas y no autenticadas. La identidad de los usuarios sin autenticar no se verifica, lo que hace que este rol sea adecuado para los usuarios invitados de la aplicación o para cuando no importa si se verifica la identidad de los usuarios. Los usuarios autenticados inician sesión en la aplicación a través de un proveedor de identidad externo, o un grupo de usuarios, que verifica su identidad. Asegúrese de asignar los permisos de los recursos de forma adecuada, para no conceder acceso a ellos a los usuarios no autenticados.

Las identidades de Amazon Cognito no son credenciales. Se intercambian por credenciales mediante el soporte de federación de identidades web en (). AWS Security Token Service AWS STS La forma recomendada de obtener credenciales de AWS para los usuarios de la aplicación es utilizar `AWS.CognitoIdentityCredentials`. A continuación, la identidad del objeto de credenciales se intercambia por las credenciales que se utilizan AWS STS.

> **Note**
>
> Si creó el grupo de identidades antes de febrero de 2015, debe volver a asociar los roles al grupo de identidades para utilizar el constructor `AWS.CognitoIdentityCredentials` sin los roles como parámetros. Para ello, abra la consola de Amazon Cognito, elija Manage Identity Pools (Administrar grupos de identidades), seleccione su grupo de identidades, elija Edit Identity Pool (Editar grupo de identidades), especifique los roles autenticados y sin autenticar, y guarde los cambios.

Los proveedores de credenciales de identidad web forman parte de la cadena de proveedores de credenciales predeterminada en los AWS SDK. Para configurar el token de su grupo de identidades en un `config` archivo local para un AWS SDK o el AWS CLI, añada una entrada `web_identity_token_file` de perfil. Consulte Asumir el rol de proveedor de credenciales en la Guía de referencia de AWS SDK y herramientas.

Para obtener más información sobre cómo rellenar las credenciales de identidad web en el SDK, consulte la guía para desarrolladores del SDK. Para obtener los mejores resultados, comience su proyecto con la integración del grupo de identidades integrada en. AWS Amplify

AWS Recursos del SDK para obtener y establecer credenciales con grupos de identidades

- Federación del grupo de identidades (Android) en el Amplify Dev Center

- Federación del grupo de identidades (iOS) en el Amplify Dev Center
- Uso de Amazon Cognito Identity para autenticar a los usuarios en la Guía para desarrolladores AWS SDK for JavaScript
- El proveedor de credenciales de Amazon Cognito en la AWS SDK for .NET Guía para desarrolladores
- Especifique las credenciales mediante programación en la guía para desarrolladores AWS SDK for Go
- Proporcione las credenciales temporales en código en la AWS SDK for Java 2.x Guía para desarrolladores
- assumeRoleWithWebIdentityCredentialProviderproveedor en la Guía AWS SDK for PHP para desarrolladores
- Asumir el rol con el proveedor de identidades web en la documentación de AWS SDK for Python (Boto3)
- Especificar las credenciales y la región predeterminada en la Guía para AWS SDK para Rust desarrolladores

En las siguientes secciones se proporciona un ejemplo de código de algunos AWS SDK antiguos.

Android

Puede usar Amazon Cognito para entregar credenciales temporales con privilegios limitados a su aplicación, de modo que sus usuarios puedan acceder a los recursos. AWS Amazon Cognito es compatible con las identidades autenticadas y no autenticadas. Para proporcionar AWS credenciales a su aplicación, siga los pasos que se indican a continuación.

Para usar un grupo de identidades de Amazon Cognito en una aplicación de Android, configure. AWS Amplify Para obtener más información, consulte Autenticación en el Amplify Dev Center.

Recuperación de una identidad de Amazon Cognito

Si admite usuarios no autenticados, puede recuperar un identificador único de Amazon Cognito (ID de identidad) para el usuario final de forma inmediata. Si está autenticando usuarios, puede recuperar el ID de identidad después de definir los tokens de inicio de sesión en el proveedor de credenciales:

```
String identityId = credentialsProvider.getIdentityId();
```

```
Log.d("LogTag", "my ID is " + identityId);
```

> **Note**
>
> No llame a `getIdentityId()`, `refresh()` o `getCredentials()` en el subproceso principal de la aplicación. A partir de Android 3.0 (nivel de API 11), tu aplicación fallará automáticamente y generará un error NetworkOnMainThreadExceptionsi realizas operaciones de E/S de red en el subproceso principal de la aplicación. Debe mover el código a un subproceso en segundo plano usando `AsyncTask`. Para obtener más información, consulte la documentación de Android. También puede llamar a `getCachedIdentityId()` para recuperar un ID, pero solo si ya hay uno almacenado localmente en la caché. De lo contrario, el método devolverá un valor nulo.

iOS - Objective-C

Puede usar Amazon Cognito para entregar credenciales temporales con privilegios limitados a su aplicación, de modo que sus usuarios puedan acceder a los recursos. AWS Los grupos de identidades de Amazon Cognito admiten tanto las identidades autenticadas como las no autenticadas. Para proporcionar AWS credenciales a su aplicación, complete los siguientes pasos.

Para usar un grupo de identidades de Amazon Cognito en una aplicación de iOS, configure. AWS Amplify Para obtener más información, consulte Autenticación de Swift y Autenticación de Flutter en el Amplify Dev Center.

Recuperación de una identidad de Amazon Cognito

Puede recuperar un identificador único de Amazon Cognito (ID de identidad) para el usuario final de inmediato si admite usuarios no autenticados o después de que haya establecido los tokens de inicio de sesión en el proveedor de credenciales si autentica a usuarios:

```
// Retrieve your Amazon Cognito ID
[[credentialsProvider getIdentityId] continueWithBlock:^id(AWSTask *task) {
    if (task.error) {
        NSLog(@"Error: %@", task.error);
    }
    else {
        // the task result will contain the identity id
        NSString *cognitoId = task.result;
```

```
    }
    return nil;
}];
```

> **Note**
>
> `getIdentityId` es una llamada asíncrona. Si ya hay un ID de identidad definido en el proveedor, puede llamar a `credentialsProvider.identityId` para recuperar la identidad, que está almacenada localmente en la caché. Sin embargo, si no hay un ID de identidad definido en el proveedor, la llamada a `credentialsProvider.identityId` devolverá `nil`. Para obtener más información, consulte Referencia de la API del SDK para móviles para iOS.

iOS - Swift

Puede usar Amazon Cognito para entregar credenciales temporales con privilegios limitados a su aplicación para que los usuarios puedan acceder a los recursos. AWS Amazon Cognito es compatible con las identidades autenticadas y no autenticadas. Para proporcionar AWS credenciales a su aplicación, siga los pasos que se indican a continuación.

Para usar un grupo de identidades de Amazon Cognito en una aplicación de iOS, configure. AWS Amplify Para obtener más información, consulte Autenticación de Swift en el Amplify Dev Center.

Recuperación de una identidad de Amazon Cognito

Puede recuperar un identificador único de Amazon Cognito (ID de identidad) para el usuario final de inmediato si admite usuarios no autenticados o después de que haya establecido los tokens de inicio de sesión en el proveedor de credenciales si autentica a usuarios:

```
// Retrieve your Amazon Cognito ID
credentialsProvider.getIdentityId().continueWith(block: { (task) -> AnyObject? in
    if (task.error != nil) {
        print("Error: " + task.error!.localizedDescription)
    }
    else {
        // the task result will contain the identity id
        let cognitoId = task.result!
        print("Cognito id: \(cognitoId)")
    }
```

```
    return task;
})
```

> **Note**
>
> `getIdentityId` es una llamada asíncrona. Si ya hay un ID de identidad definido en el proveedor, puede llamar a `credentialsProvider.identityId` para recuperar la identidad, que está almacenada localmente en la caché. Sin embargo, si no hay un ID de identidad definido en el proveedor, la llamada a `credentialsProvider.identityId` devolverá `nil`. Para obtener más información, consulte Referencia de la API del SDK para móviles para iOS.

JavaScript

Si todavía no lo ha hecho, cree un grupo de identidades en la consola de Amazon Cognito antes de usar `AWS.CognitoIdentityCredentials`.

Después de configurar un grupo de identidades con sus proveedores de identidad, puede utilizar `AWS.CognitoIdentityCredentials` para autenticar a los usuarios. Para configurar las credenciales de la aplicación para utilizar `AWS.CognitoIdentityCredentials`, establezca la propiedad `credentials` de `AWS.Config` o una configuración específica para cada servicio. El siguiente ejemplo utiliza `AWS.Config`:

```
// Set the region where your identity pool exists (us-east-1, eu-west-1)
AWS.config.region = 'us-east-1';

// Configure the credentials provider to use your identity pool
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
    IdentityPoolId: 'IDENTITY_POOL_ID',
    Logins: { // optional tokens, used for authenticated login
        'graph.facebook.com': 'FBTOKEN',
        'www.amazon.com': 'AMAZONTOKEN',
        'accounts.google.com': 'GOOGLETOKEN',
        'appleid.apple.com': 'APPLETOKEN'
    }
});

// Make the call to obtain credentials
AWS.config.credentials.get(function(){
```

```
    // Credentials will be available when this function is called.
    var accessKeyId = AWS.config.credentials.accessKeyId;
    var secretAccessKey = AWS.config.credentials.secretAccessKey;
    var sessionToken = AWS.config.credentials.sessionToken;

});
```

La propiedad opcional `Logins` es un mapeo entre los nombres de los proveedores de identidad y los tokens de identidad de los proveedores. La forma de obtener el token del proveedor de identidad depende del proveedor que se utilice. Por ejemplo, si Facebook es uno de sus proveedores de identidad, puede utilizar la función `FB.login` del SDK de Facebook para obtener un token de proveedor de identidad:

```
FB.login(function (response) {
    if (response.authResponse) { // logged in
        AWS.config.credentials = new AWS.CognitoIdentityCredentials({
          IdentityPoolId: 'us-east-1:1699ebc0-7900-4099-b910-2df94f52a030',
          Logins: {
            'graph.facebook.com': response.authResponse.accessToken
          }
        });

        console.log('You are now logged in.');
    } else {
        console.log('There was a problem logging you in.');
    }
});
```

Recuperación de una identidad de Amazon Cognito

Puede recuperar un identificador único de Amazon Cognito (ID de identidad) para el usuario final de inmediato si admite usuarios no autenticados o después de que haya establecido los tokens de inicio de sesión en el proveedor de credenciales si autentica a usuarios:

```
var identityId = AWS.config.credentials.identityId;
```

Unity

Puede usar Amazon Cognito para entregar credenciales temporales con privilegios limitados a su aplicación, de modo que sus usuarios puedan acceder a los recursos. AWS Amazon Cognito es

compatible con las identidades autenticadas y no autenticadas. Para proporcionar AWS credenciales a su aplicación, siga los pasos que se indican a continuación.

El AWS SDK para Unity ahora forma parte de AWS SDK for .NET. Para empezar a utilizar Amazon Cognito en AWS SDK for .NET, consulte el proveedor de credenciales de Amazon Cognito en AWS SDK for .NET la Guía para desarrolladores. O consulta Amplify Dev Center para ver las opciones con las que crear una aplicación. AWS Amplify

Recuperación de una identidad de Amazon Cognito

Puede recuperar un identificador único de Amazon Cognito (ID de identidad) para el usuario final de inmediato si admite usuarios no autenticados o después de que haya establecido los tokens de inicio de sesión en el proveedor de credenciales si autentica a usuarios:

```
credentials.GetIdentityIdAsync(delegate(AmazonCognitoIdentityResult<string> result) {
    if (result.Exception != null) {
        //Exception!
    }
    string identityId = result.Response;
});
```

Xamarin

Puede usar Amazon Cognito para entregar credenciales temporales con privilegios limitados a su aplicación para que los usuarios puedan acceder a los recursos. AWS Amazon Cognito es compatible con las identidades autenticadas y no autenticadas. Para proporcionar AWS credenciales a su aplicación, siga los pasos que se indican a continuación.

El AWS SDK de Xamarin ahora forma parte de AWS SDK for .NET. Para empezar a utilizar Amazon Cognito en AWS SDK for .NET, consulte el proveedor de credenciales de Amazon Cognito en AWS SDK for .NET la Guía para desarrolladores. O consulta Amplify Dev Center para ver las opciones con las que crear una aplicación. AWS Amplify

> **Note**
>
> Nota: si creó el grupo de identidades antes de febrero de 2015, debe volver a asociar los roles a su grupo de identidades a fin de utilizar este constructor sin los roles como parámetros. Para ello, abra la consola de Amazon Cognito, elija Manage Identity Pools (Administrar grupos de identidades), seleccione su grupo de identidades, elija Edit Identity

Pool (Editar grupo de identidades), especifique los roles autenticados y sin autenticar, y guarde los cambios.

Recuperación de una identidad de Amazon Cognito

Puede recuperar un identificador único de Amazon Cognito (ID de identidad) para el usuario final de inmediato si admite usuarios no autenticados o después de que haya establecido los tokens de inicio de sesión en el proveedor de credenciales si autentica a usuarios:

```
var identityId = await credentials.GetIdentityIdAsync();
```

Acceder a AWS los servicios

Tras configurar el proveedor de credenciales de Amazon Cognito y recuperar AWS las credenciales, puede crear un Servicio de AWS cliente.

AWS Recursos del SDK para crear un cliente

- AWS Configuración del cliente en la Guía para AWS SDK for C++ desarrolladores
- Uso de la AWS SDK for Go V2 con Servicios de AWS la Guía para AWS SDK for Go desarrolladores
- Configuración de clientes HTTP en la Guía para AWS SDK for Java 2.x desarrolladores
- Cómo crear y llamar a objetos de servicio en la Guía para AWS SDK for JavaScript desarrolladores
- Creación de clientes en la AWS SDK for Python (Boto3) documentación
- Creación de un cliente de servicio en la Guía para AWS SDK para Rust desarrolladores
- Uso de clientes en la Guía para AWS SDK para Swift desarrolladores

El siguiente fragmento de código inicializa un cliente de Amazon DynamoDB:

Android

Para usar un grupo de identidades de Amazon Cognito en una aplicación de Android, configure. AWS Amplify Para obtener más información, consulte Autenticación en el Amplify Dev Center.

```
// Create a service client with the provider
```

```
AmazonDynamoDB client = new AmazonDynamoDBClient(credentialsProvider);
```

El proveedor de credenciales se comunica con Amazon Cognito y recupera tanto el identificador único de los usuarios autenticados como los no autenticados, así como las credenciales temporales con privilegios AWS limitados para el SDK móvil. AWS Las credenciales recuperadas son válidas durante una hora y el proveedor las actualiza cuando caducan.

iOS - Objective-C

Para usar un grupo de identidades de Amazon Cognito en una aplicación de iOS, configure. AWS Amplify Para obtener más información, consulte Autenticación de Swift y Autenticación de Flutter en el Amplify Dev Center.

```
// create a configuration that uses the provider
AWSServiceConfiguration *configuration = [AWSServiceConfiguration
 configurationWithRegion:AWSRegionUSEast1 provider:credentialsProvider];
// get a client with the default service configuration
AWSDynamoDB *dynamoDB = [AWSDynamoDB defaultDynamoDB];
```

El proveedor de credenciales se comunica con Amazon Cognito y recupera tanto el identificador único de los usuarios autenticados como los no autenticados, así como las credenciales temporales con privilegios AWS limitados para el SDK móvil. AWS Las credenciales recuperadas son válidas durante una hora y el proveedor las actualiza cuando caducan.

iOS - Swift

Para usar un grupo de identidades de Amazon Cognito en una aplicación de iOS, configure. AWS Amplify Para obtener más información, consulte Autenticación de Swift en el Amplify Dev Center.

```
// get a client with the default service configuration
let dynamoDB = AWSDynamoDB.default()

// get a client with a custom configuration
AWSDynamoDB.register(with: configuration!, forKey: "USWest2DynamoDB");
let dynamoDBCustom = AWSDynamoDB(forKey: "USWest2DynamoDB")
```

El proveedor de credenciales se comunica con Amazon Cognito y recupera tanto el identificador único de los usuarios autenticados como los no autenticados, así como las credenciales temporales con privilegios AWS limitados para el SDK móvil. AWS Las credenciales recuperadas son válidas durante una hora y el proveedor las actualiza cuando caducan.

JavaScript

```
// Create a service client with the provider
var dynamodb = new AWS.DynamoDB({region: 'us-west-2'});
```

El proveedor de credenciales se comunica con Amazon Cognito y recupera tanto el identificador único de los usuarios autenticados como los no autenticados, así como las credenciales temporales con privilegios limitados AWS para el SDK móvil. AWS Las credenciales recuperadas son válidas durante una hora y el proveedor las actualiza cuando caducan.

Unity

El AWS SDK para Unity ahora forma parte de AWS SDK for .NET. Para empezar a utilizar Amazon Cognito en AWS SDK for .NET, consulte el proveedor de credenciales de Amazon Cognito en AWS SDK for .NET la Guía para desarrolladores. O consulta Amplify Dev Center para ver las opciones con las que crear una aplicación. AWS Amplify

```
// create a service client that uses credentials provided by Cognito
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials, REGION);
```

El proveedor de credenciales se comunica con Amazon Cognito y recupera tanto el identificador único de los usuarios autenticados como los no autenticados, así como las credenciales temporales con privilegios limitados AWS para el SDK móvil. AWS Las credenciales recuperadas son válidas durante una hora y el proveedor las actualiza cuando caducan.

Xamarin

El AWS SDK de Xamarin ahora forma parte de AWS SDK for .NET. Para empezar a utilizar Amazon Cognito en AWS SDK for .NET, consulte el proveedor de credenciales de Amazon Cognito en AWS SDK for .NET la Guía para desarrolladores. O consulta Amplify Dev Center para ver las opciones con las que crear una aplicación. AWS Amplify

```
// create a service client that uses credentials provided by Cognito
var client = new AmazonDynamoDBClient(credentials, REGION)
```

El proveedor de credenciales se comunica con Amazon Cognito y recupera tanto el identificador único de los usuarios autenticados como los no autenticados, así como las credenciales temporales

con privilegios limitados AWS para el SDK móvil. AWS Las credenciales recuperadas son válidas durante una hora y el proveedor las actualiza cuando caducan.

Proveedores de identidad externos de grupos de identidades

La propiedad `logins` le permite configurar las credenciales recibidas de un proveedor de identidad (IdP). Además, puede asociar un grupo de identidades a varios IdPs. Por ejemplo, puede definir los tokens de Facebook y Google en la propiedad `logins` para que la identidad única de Amazon Cognito se asocie a los inicios de sesión de ambos IdP. El usuario puede autenticarse en cualquiera de las cuentas, pero Amazon Cognito devuelve el mismo identificador de usuario.

Las siguientes instrucciones le guían a través de la autenticación con IdPs los grupos de identidades de Amazon Cognito compatibles.

Temas

Configurar Facebook como un IdP de grupos de identidades

Los grupos de identidades de Amazon Cognito se integran en Facebook para ofrecer una autenticación federada a los usuarios de aplicaciones móviles. En esta sección se explica cómo registrar y configurar su aplicación con Facebook como IdP.

Configuración de Facebook

Registre su aplicación en Facebook para poder empezar a autenticar usuarios de Facebook e interactuar con las API de Facebook.

El portal para desarrolladores de Facebook le ayuda a configurar la aplicación. Siga este procedimiento antes de integrar Facebook en su grupo de identidades de Amazon Cognito:

Configuración de Facebook

1. En el portal de desarrolladores de Facebook, inicie sesión con sus credenciales de Facebook.
2. En el menú Apps (Aplicaciones), seleccione Add a New App (Añadir una nueva aplicación).
3. Seleccione una plataforma y complete el proceso de inicio rápido.

Android

Para obtener más información sobre cómo integrar aplicaciones Android con el inicio de sesión de Facebook, consulte la guía de introducción a Facebook.

iOS - Objective-C

Para obtener más información sobre cómo integrar aplicaciones iOS Objective-C con el inicio de sesión de Facebook, consulte la guía de introducción a Facebook.

iOS - Swift

Para obtener más información sobre cómo integrar aplicaciones iOS Swift con el inicio de sesión de Facebook, consulte la guía de introducción a Facebook.

JavaScript

Para obtener más información sobre cómo integrar aplicaciones JavaScript web con el inicio de sesión de Facebook, consulta la Guía de introducción de Facebook.

Unity

Para obtener más información sobre cómo integrar aplicaciones Unity con el inicio de sesión de Facebook, consulte la guía de introducción a Facebook.

Xamarin

Para añadir la autenticación de Facebook, empiece por seguir el flujo adecuado a continuación para integrar el SDK de Facebook en su aplicación. En los grupos de identidades de Amazon Cognito, se utiliza el token de acceso de Facebook para generar un identificador de usuario único asociado a una identidad de Amazon Cognito.

- SDK de Facebook iOS por Xamarin

- SDK de Facebook Android por Xamarin

Configurar un proveedor de identidades en la consola de grupos de identidades de Amazon Cognito

Utilice el siguiente procedimiento para configurar el proveedor de identidades.

Para agregar un proveedor de identidades (IdP) de Facebook

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Elija Facebook.
5. Ingrese el ID de aplicación del proyecto de OAuth que creó en Meta para desarrolladores. Para obtener más información, consulte Inicio de sesión de Facebook en Meta para documentos de desarrolladores.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.
    - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.
        i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.
        ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.
7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.
    a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Uso de Facebook

Android

Para añadir la autenticación de Facebook, empiece por seguir la guía de Facebook para integrar el SDK de Facebook en su aplicación. A continuación, añada un botón Iniciar sesión con Facebook a la interfaz de usuario de Android. El SDK de Facebook utiliza un objeto de sesión para realizar seguimiento de su estado. Amazon Cognito utiliza el token de acceso de este objeto de sesión para autenticar al usuario, generar el identificador único y, si es necesario, conceder al usuario acceso a otros recursos. AWS

Una vez que haya autenticado a su usuario con el SDK de Facebook, agregue el token de sesión al proveedor de credenciales de Amazon Cognito.

SDK de Facebook 4.0 o posterior:

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", AccessToken.getCurrentAccessToken().getToken());
credentialsProvider.setLogins(logins);
```

SDK de Facebook antes de la versión 4.0:

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", Session.getActiveSession().getAccessToken());
credentialsProvider.setLogins(logins);
```

El proceso de inicio de sesión de Facebook inicializa una sesión singleton en su SDK. El objeto de sesión de Facebook contiene un token de OAuth que Amazon Cognito utiliza para AWS generar las credenciales del usuario final autenticado. Amazon Cognito también utiliza el token para buscar la existencia de un usuario que corresponda a esta identidad de Facebook en concreto en su base de datos. Si el usuario ya existe, la API devuelve el identificador existente. De lo contrario, la API

devuelve un identificador nuevo. El SDK cliente guarda en caché los identificadores automáticamente en el dispositivo local.

> **Note**
>
> Después de configurar el mapa de inicios de sesión, realice una llamada `refresh` o `get` recupere las credenciales. AWS

iOS - Objective-C

Para añadir la autenticación de Facebook, empiece por seguir la guía de Facebook para integrar el SDK de Facebook en su aplicación. A continuación, añada un botón "Iniciar sesión con Facebook" a la interfaz de usuario. El SDK de Facebook utiliza un objeto de sesión para realizar seguimiento de su estado. Amazon Cognito utiliza el token de acceso de este objeto de sesión para autenticar al usuario y vincularlo a un grupo de identidades único de Amazon Cognito (identidades federadas).

Para proporcionar el token de acceso de Facebook a Amazon Cognito, implemente el protocolo AWSIdentityProviderManager.

Al implementar el método `logins`, devuelva un diccionario que contiene `AWSIdentityProviderFacebook`. Este diccionario sirve como la clave, y el token de acceso actual del usuario autenticado de Facebook actúa como valor, como se muestra en el ejemplo de código siguiente.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    FBSDKAccessToken* fbToken = [FBSDKAccessToken currentAccessToken];
    if(fbToken){
        NSString *token = fbToken.tokenString;
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook : token }];
    }else{
            return [AWSTask taskWithError:[NSError errorWithDomain:@"Facebook Login"
                                                              code:-1
                                                          userInfo:@{@"error":@"No current
 Facebook access token"}]];
    }
}
```

Cuando cree instancias de `AWSCognitoCredentialsProvider`, transmita la clase que implementa `AWSIdentityProviderManager` como valor de

`identityProviderManager` en el constructor. Para obtener más información, vaya a la página de AWSCognitoCredentialsProviderreferencia y elija initWithRegionTipo:identityPoolId: identityProviderManager.

iOS - Swift

Para añadir la autenticación de Facebook, empiece por seguir la guía de Facebook para integrar el SDK de Facebook en su aplicación. A continuación, añada un botón "Iniciar sesión con Facebook" a la interfaz de usuario. El SDK de Facebook utiliza un objeto de sesión para realizar seguimiento de su estado. Amazon Cognito utiliza el token de acceso de este objeto de sesión para autenticar al usuario y vincularlo a un grupo de identidades único de Amazon Cognito (identidades federadas).

Para proporcionar el token de acceso de Facebook a Amazon Cognito, implemente el protocolo AWSIdentityProviderManager.

Al implementar el método `logins`, devuelva un diccionario que contenga `AWSIdentityProviderFacebook`. Este diccionario sirve como la clave, y el token de acceso actual del usuario autenticado de Facebook actúa como valor, como se muestra en el ejemplo de código siguiente.

```
class FacebookProvider: NSObject, AWSIdentityProviderManager {
    func logins() -> AWSTask<NSDictionary> {
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
 ["Facebook" : "No current Facebook access token"]))
    }
}
```

Cuando cree instancias de `AWSCognitoCredentialsProvider`, transmita la clase que implementa `AWSIdentityProviderManager` como valor de `identityProviderManager` en el constructor. Para obtener más información, vaya a la página de AWSCognitoCredentialsProviderreferencia y elija initWithRegionTipo:identityPoolId: identityProviderManager.

JavaScript

Para proporcionar autenticación de Facebook, siga el inicio de sesión con Facebook para web para añadir el botón Iniciar sesión con Facebook a su sitio web. El SDK de Facebook utiliza un objeto de

sesión para realizar seguimiento de su estado. Amazon Cognito utiliza el token de acceso de este objeto de sesión para autenticar al usuario, generar el identificador único y, si es necesario, conceder al usuario acceso a otros recursos. AWS

Una vez que haya autenticado a su usuario con el SDK de Facebook, agregue el token de sesión al proveedor de credenciales de Amazon Cognito.

```
FB.login(function (response) {

  // Check if the user logged in successfully.
  if (response.authResponse) {

    console.log('You are now logged in.');

    // Add the Facebook access token to the Amazon Cognito credentials login map.
    AWS.config.credentials = new AWS.CognitoIdentityCredentials({
      IdentityPoolId: 'IDENTITY_POOL_ID',
      Logins: {
        'graph.facebook.com': response.authResponse.accessToken
      }
    });

    // Obtain AWS credentials
    AWS.config.credentials.get(function(){
        // Access AWS resources here.
    });

  } else {
    console.log('There was a problem logging you in.');
  }

});
```

El SDK de Facebook obtiene un token de OAuth que Amazon Cognito utiliza para generar AWS credenciales para el usuario final autenticado. Amazon Cognito también utiliza el token para buscar la existencia de un usuario que corresponda a esta identidad de Facebook en concreto en su base de datos. Si el usuario ya existe, la API devuelve el identificador existente. De lo contrario, devuelve un identificador nuevo. El SDK cliente guarda los identificadores automáticamente en caché los en el dispositivo local.

> **Note**
>
> Después de configurar la asignación de inicios de sesión, deberá hacer una llamada a `refresh` o `get` para obtener las credenciales. Para ver un ejemplo de código, consulte el «Caso de uso 17, Integración de grupos de usuarios con Cognito Identity», en el JavaScript archivo README.

Unity

Para añadir la autenticación de Facebook, empiece por seguir la guía de Facebook para integrar el SDK de Facebook en su aplicación. Amazon Cognito utiliza el token de acceso de Facebook del objeto FB para generar un identificador de usuario único asociado a una identidad de Amazon Cognito.

Una vez que haya autenticado a su usuario con el SDK de Facebook, agregue el token de sesión al proveedor de credenciales de Amazon Cognito:

```
void Start()
{
    FB.Init(delegate() {
        if (FB.IsLoggedIn) { //User already logged in from a previous session
            AddFacebookTokenToCognito();
        } else {
            FB.Login ("email", FacebookLoginCallback);
        }
    });
}

void FacebookLoginCallback(FBResult result)
{
    if (FB.IsLoggedIn)
    {
        AddFacebookTokenToCognito();
    }
    else
    {
        Debug.Log("FB Login error");
    }
}

void AddFacebookTokenToCognito()
```

```
{
    credentials.AddLogin ("graph.facebook.com",
 AccessToken.CurrentAccessToken.TokenString);
}
```

Antes de usar `FB.AccessToken`, llame a `FB.Login()` y asegúrese de que `FB.IsLoggedIn` es verdadero.

Xamarin

Xamarin para Android:

```
public void InitializeFacebook() {
    FacebookSdk.SdkInitialize(this.ApplicationContext);
    callbackManager = CallbackManagerFactory.Create();
    LoginManager.Instance.RegisterCallback(callbackManager, new FacebookCallback &lt;
 LoginResult &gt; () {
      HandleSuccess = loginResult = &gt; {
        var accessToken = loginResult.AccessToken;
        credentials.AddLogin("graph.facebook.com", accessToken.Token);
        //open new activity
      },
      HandleCancel = () = &gt; {
        //throw error message
      },
      HandleError = loginError = &gt; {
        //throw error message
      }
    });
    LoginManager.Instance.LogInWithReadPermissions(this, new List &lt; string &gt; {
      "public_profile"
    });
  }
```

Xamarin para iOS:

```
public void InitializeFacebook() {
  LoginManager login = new LoginManager();
  login.LogInWithReadPermissions(readPermissions.ToArray(),
 delegate(LoginManagerLoginResult result, NSError error) {
    if (error != null) {
      //throw error message
    } else if (result.IsCancelled) {
```

```
      //throw error message
    } else {
      var accessToken = loginResult.AccessToken;
      credentials.AddLogin("graph.facebook.com", accessToken.Token);
      //open new view controller
    }
  });
}
```

Configuración de Login with Amazon como un IdP de grupos de identidades

Amazon Cognito se integra en Login with Amazon con el fin de ofrecer autenticación federada para sus usuarios de las aplicaciones móviles y web. En esta sección se explica cómo registrar y configurar su aplicación con Login with Amazon como proveedor de identidad (IdP).

Configuración de Login with Amazon para que funcione con Amazon Cognito en el portal para desarrolladores. Para obtener más información, consulte Configuración de Login with Amazon en las preguntas frecuentes sobre Login with Amazon.

> **Note**
>
> Para integrar Login with Amazon en una aplicación de Xamarin, siga la guía de introducción a Xamarin.

> **Note**
>
> No puede integrar de forma nativa Login with Amazon en la plataforma Unity. En su lugar, utilice una vista web y siga el flujo de inicio de sesión del navegador.

Configuración de Login with Amazon

Implementar Login with Amazon

En el portal para desarrolladores de Amazon puede configurar una aplicación OAuth para integrarla con su grupo de identidades, buscar documentación de Login with Amazon y descargar SDK. Elija Developer console (Consola para desarrolladores) y luego Login with Amazon (Inicio de sesión con Amazon) en el portal para desarrolladores. Puede crear un perfil de seguridad para tu su y, a

continuación, crear mecanismos de autenticación Login with Amazon en ella. Consulte Obtención de credenciales para ver más información sobre cómo integrar la autenticación Login with Amazon con la aplicación.

Amazon emite un ID de cliente de OAuth 2.0 para su nuevo perfil de seguridad. Puede encontrar el ID de cliente en la pestaña Web Settings (Configuración web) del perfil de seguridad. Ingrese el ID de perfil de seguridad en el campo ID de aplicación del IdP de Login with Amazon en el grupo de identidades.

> **Note**
>
> Se ingresa el ID de perfil de seguridad en el campo ID de aplicación del IdP de Login with Amazon en el grupo de identidades. Esto difiere de los grupos de usuarios, que utilizan el ID de cliente.

Configuración del proveedor externo en la consola de Amazon Cognito

Para agregar un inicio de sesión con el proveedor de identidades (IdP) de Amazon

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Elija Iniciar sesión con Amazon.
5. Ingrese el ID de aplicación del proyecto de OAuth que creó en Inicio de sesión con Amazon. Para obtener más información, consulte la Documentación de Login with Amazon.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.

    - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.

        i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.

   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

   c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Utilización de Login with Amazon: Android

Tras autenticar el inicio de sesión en Amazon, puede pasar el token al proveedor de credenciales de Amazon Cognito en el método onSuccess de la interfaz. TokenListener El código tiene este aspecto:

```
@Override
public void onSuccess(Bundle response) {
    String token = response.getString(AuthzConstants.BUNDLE_KEY.TOKEN.val);
    Map<String, String> logins = new HashMap<String, String>();
    logins.put("www.amazon.com", token);
    credentialsProvider.setLogins(logins);
}
```

Utilización de Login with Amazon: iOS - Objective-C

Tras autenticar el inicio de sesión en Amazon, puede pasar el token al proveedor de credenciales de Amazon Cognito mediante requestDidSucceed el método AMZN: AccessTokenDelegate

```
- (void)requestDidSucceed:(APIResult \*)apiResult {
    if (apiResult.api == kAPIAuthorizeUser) {
        [AIMobileLib getAccessTokenForScopes:[NSArray arrayWithObject:@"profile"]
 withOverrideParams:nil delegate:self];
    }
    else if (apiResult.api == kAPIGetAccessToken) {
```

```
        credentialsProvider.logins = @{ @(AWSCognitoLoginProviderKeyLoginWithAmazon):
 apiResult.result };
    }
}}
```

Utilización de Login with Amazon: iOS - Swift

Una vez que haya autenticado el inicio de sesión con Amazon, puede transmitir el token al proveedor de credenciales de Amazon Cognito con el método `requestDidSucceed` de la interfaz `AMZNAccessTokenDelegate`:

```
func requestDidSucceed(apiResult: APIResult!) {
    if apiResult.api == API.AuthorizeUser {
        AIMobileLib.getAccessTokenForScopes(["profile"], withOverrideParams: nil,
 delegate: self)
    } else if apiResult.api == API.GetAccessToken {
        credentialsProvider.logins =
 [AWSCognitoLoginProviderKey.LoginWithAmazon.rawValue: apiResult.result]
    }
}
```

Usa Login with Amazon: JavaScript

Una vez que el usuario se haya autenticado con Login with Amazon y vuelva a su sitio web, se proporciona el token de acceso de Login with Amazon en la cadena de consulta. Pase este token a la asignación de inicios de sesión de credenciales.

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
   IdentityPoolId: 'IDENTITY_POOL_ID',
   Logins: {
       'www.amazon.com': 'Amazon Access Token'
   }
});
```

Utilización de Login with Amazon: Xamarin

Xamarin para Android

```
AmazonAuthorizationManager manager = new AmazonAuthorizationManager(this,
 Bundle.Empty);

var tokenListener = new APIListener {
```

```
  Success = response => {
    // Get the auth token
    var token = response.GetString(AuthzConstants.BUNDLE_KEY.Token.Val);
    credentials.AddLogin("www.amazon.com", token);
  }
};

// Try and get existing login
manager.GetToken(new[] {
  "profile"
}, tokenListener);
```

Xamarin para iOS

En `AppDelegate.cs`, inserte lo siguiente:

```
public override bool OpenUrl (UIApplication application, NSUrl url, string
 sourceApplication, NSObject annotation)
{
    // Pass on the url to the SDK to parse authorization code from the url
    bool isValidRedirectSignInURL = AIMobileLib.HandleOpenUrl (url, sourceApplication);
    if(!isValidRedirectSignInURL)
        return false;

    // App may also want to handle url
    return true;
}
```

Luego, haga lo siguiente en `ViewController.cs`:

```
public override void ViewDidLoad ()
{
    base.LoadView ();

    // Here we create the Amazon Login Button
    btnLogin = UIButton.FromType (UIButtonType.RoundedRect);
    btnLogin.Frame = new RectangleF (55, 206, 209, 48);
    btnLogin.SetTitle ("Login using Amazon", UIControlState.Normal);
    btnLogin.TouchUpInside += (sender, e) => {
        AIMobileLib.AuthorizeUser (new [] { "profile"}, new AMZNAuthorizationDelegate
 ());
    };
    View.AddSubview (btnLogin);
```

```
}

// Class that handles Authentication Success/Failure
public class AMZNAuthorizationDelegate : AIAuthenticationDelegate
{
  public override void RequestDidSucceed(ApiResult apiResult)
    {
      // Your code after the user authorizes application for requested scopes
      var token = apiResult["access_token"];
      credentials.AddLogin("www.amazon.com",token);
    }

    public override void RequestDidFail(ApiError errorResponse)
    {
      // Your code when the authorization fails
      InvokeOnMainThread(() => new UIAlertView("User Authorization Failed",
 errorResponse.Error.Message, null, "Ok", null).Show());
    }
}
```

Configurar Google como un IdP de grupo de identidades

Amazon Cognito se integra con Google para ofrecer autenticación federada a sus usuarios de aplicación móvil. En esta sección se explica cómo registrar y configurar su aplicación con Google como IdP.

Android

> **Note**
>
> Si su aplicación utiliza Google y está disponible en varias plataformas móviles, debe configurarla como proveedor de OpenID Connect. Agregue todos los ID de cliente creados como valores de audiencia adicionales para permitir una mejor integración. Para obtener más información sobre el modelo de identidad de varios clientes de Google, consulte Cross-client Identity.

Configuración de Google

Para activar el inicio de sesión de Google para Android, cree un proyecto de consola de Google Developers para su aplicación.

1. Vaya a la consola de Google Developers y cree un proyecto.
2. Elija APIs & Services (API y servicios) y luego OAuth consent screen (Pantalla de consentimiento de OAuth). Personalice la información que Google muestra a sus usuarios cuando Google solicita su consentimiento para compartir sus datos de perfil con su aplicación.
3. Elija Credentials (Credenciales) y luego Create credentials (Crear credenciales). Elija OAuth client ID (ID de cliente de OAuth). Seleccione Android en Application type (Tipo de aplicación). Cree un ID de cliente independiente para cada plataforma en la que desarrolle la aplicación.
4. Desde Credentials (Credenciales), elija Manage service accounts (Administrar cuentas de servicios). Elija Create service account (Crear cuenta de servicio). Ingrese los detalles de su cuenta de servicio y, a continuación, seleccione Create and continue (Crear y continuar).
5. Conceda a la cuenta de servicio acceso a su proyecto. Conceda a los usuarios acceso a la cuenta de servicio según lo requiera su aplicación.
6. Elija la nueva cuenta de servicio y luego la pestaña Keys (Claves) y Add key (Agregar clave). Cree y descargue una nueva clave JSON.

Para ver más información acerca de cómo utilizar la consola de Google Developers, consulte Crea y administra proyectos en la documentación de Google Cloud.

Para obtener más información sobre cómo integrar Google en tu aplicación de Android, consulta Cómo autenticar a los usuarios mediante el inicio de sesión con Google en la documentación de Google Identity.

Para agregar un proveedor de identidades (IdP) de Google

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Seleccione Google.
5. Ingrese el ID de cliente del proyecto de OAuth que creó en la Plataforma de Google Cloud. Para obtener más información, consulte Configurar OAuth 2.0 en la Ayuda de la consola de Google Cloud Platform.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.

- Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.

    i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

    ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.

    a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

    b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

    c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Uso de Google

Para habilitar el inicio de sesión con Google en su aplicación, siga las instrucciones de la documentación de Google para Android. Cuando un usuario inicia sesión, solicita un token de autenticación de OpenID Connect a Google. Luego Amazon Cognito utiliza el token para autenticar al usuario y generar un identificador único.

En el siguiente código de muestra se muestra cómo recuperar el token de autenticación de Google Play Service:

```
GooglePlayServicesUtil.isGooglePlayServicesAvailable(getApplicationContext());
AccountManager am = AccountManager.get(this);
Account[] accounts = am.getAccountsByType(GoogleAuthUtil.GOOGLE_ACCOUNT_TYPE);
String token = GoogleAuthUtil.getToken(getApplicationContext(), accounts[0].name,
        "audience:server:client_id:YOUR_GOOGLE_CLIENT_ID");
```

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("accounts.google.com", token);
credentialsProvider.setLogins(logins);
```

iOS - Objective-C

> **Note**
>
> Si su aplicación utiliza Google y está disponible en varias plataformas móviles, configure Google como proveedor de OpenID Connect. Agregue todos los ID de cliente creados como valores de audiencia adicionales para permitir una mejor integración. Para obtener más información sobre el modelo de identidad de varios clientes de Google, consulte Cross-client Identity.

Configuración de Google

Para habilitar el inicio de sesión de Google para iOS, cree un proyecto de consola de Google Developers para su aplicación.

1. Vaya a la consola de Google Developers y cree un proyecto.
2. Elija APIs & Services (API y servicios) y luego OAuth consent screen (Pantalla de consentimiento de OAuth). Personalice la información que Google muestra a sus usuarios cuando Google solicita su consentimiento para compartir sus datos de perfil con su aplicación.
3. Elija Credentials (Credenciales) y luego Create credentials (Crear credenciales). Elija OAuth client ID (ID de cliente de OAuth). Seleccione iOS en Application type (Tipo de aplicación). Cree un ID de cliente independiente para cada plataforma en la que desarrolle la aplicación.
4. Desde Credentials (Credenciales), elija Manage service accounts (Administrar cuentas de servicios). Elija Create service account (Crear cuenta de servicio). Ingrese los detalles de su cuenta de servicio y seleccione Create and continue (Crear y continuar).
5. Conceda a la cuenta de servicio acceso a su proyecto. Conceda a los usuarios acceso a la cuenta de servicio según lo requiera su aplicación.
6. Elija la nueva cuenta de servicio. Elija la pestaña Keys (Claves) y Add key (Agregar clave). Cree y descargue una nueva clave JSON.

Para ver más información acerca de cómo utilizar la consola de Google Developers, consulte Crea y administra proyectos en la documentación de Google Cloud.

Para ver más información acerca de cómo integrar Google en su aplicación iOS, consulte el tema sobre inicio de sesión con Google para iOS en la documentación de Google Identity.

Para agregar un proveedor de identidades (IdP) de Google

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Seleccione Google.
5. Ingrese el ID de cliente del proyecto de OAuth que creó en la Plataforma de Google Cloud. Para obtener más información, consulte Configurar OAuth 2.0 en la Ayuda de la consola de Google Cloud Platform.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.

   - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.

     i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

     ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.
7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.

   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.

   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

   c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Uso de Google

Para habilitar el inicio de sesión con Google en su aplicación, siga la documentación de Google para iOS. Si la autenticación tiene éxito, se genera un token de autenticación de OpenID Connect que Amazon Cognito utiliza para autenticar al usuario y generar un identificador único.

Si la autenticación tiene éxito, se genera un objeto `GTMOAuth2Authentication`, que contiene un `id_token` y que Amazon Cognito utiliza para autenticar al usuario y generar un identificador único:

```
- (void)finishedWithAuth: (GTMOAuth2Authentication *)auth error: (NSError *) error {
        NSString *idToken = [auth.parameters objectForKey:@"id_token"];
        credentialsProvider.logins = @{ @(AWSCognitoLoginProviderKeyGoogle): idToken };
    }
```

iOS - Swift

> **Note**
>
> Si su aplicación utiliza Google y está disponible en varias plataformas móviles, configure Google como proveedor de OpenID Connect. Agregue todos los ID de cliente creados como valores de audiencia adicionales para permitir una mejor integración. Para obtener más información sobre el modelo de identidad de varios clientes de Google, consulte Cross-client Identity.

Configuración de Google

Para habilitar el inicio de sesión de Google para iOS, cree un proyecto de consola de Google Developers para su aplicación.

1. Vaya a la consola de Google Developers y cree un proyecto.
2. Elija APIs & Services (API y servicios) y luego OAuth consent screen (Pantalla de consentimiento de OAuth). Personalice la información que Google muestra a sus usuarios cuando Google solicita su consentimiento para compartir sus datos de perfil con su aplicación.
3. Elija Credentials (Credenciales) y luego Create credentials (Crear credenciales). Elija OAuth client ID (ID de cliente de OAuth). Seleccione iOS en Application type (Tipo de aplicación). Cree un ID de cliente independiente para cada plataforma en la que desarrolle la aplicación.

4. Desde Credentials (Credenciales), elija Manage service accounts (Administrar cuentas de servicios). Elija Create service account (Crear cuenta de servicio). Ingrese los detalles de su cuenta de servicio y seleccione Create and continue (Crear y continuar).
5. Conceda a la cuenta de servicio acceso a su proyecto. Conceda a los usuarios acceso a la cuenta de servicio según lo requiera su aplicación.
6. Elija la nueva cuenta de servicio y luego la pestaña Keys (Claves) y Add key (Agregar clave). Cree y descargue una nueva clave JSON.

Para ver más información acerca de cómo utilizar la consola de Google Developers, consulte Crea y administra proyectos en la documentación de Google Cloud.

Para ver más información acerca de cómo integrar Google en su aplicación iOS, consulte el tema sobre inicio de sesión con Google para iOS en la documentación de Google Identity.

Elija Manage Identity Pools (Administrar grupos de identidades) de la página de inicio de la consola de Amazon Cognito:

Configuración del proveedor externo en la consola de Amazon Cognito

1. Elija el nombre del grupo de identidades donde desee habilitar Google como proveedor externo. Se mostrará la página Dashboard (Panel) de su grupo de identidades.
2. En la esquina superior derecha de la página Dashboard (Panel), elija Edit identity pool (Editar grupo de identidades). Se visualizará la página Edit identity pool.
3. Desplácese hacia abajo y elija Authentication providers (Proveedores de autenticación) para expandir la sección.
4. Elija la pestaña Google .
5. Elija Unlock (Desbloquear).
6. Introduzca el ID de cliente de Google que Google le ha entregado y, a continuación, elija Save Changes (Guardar cambios).

Uso de Google

Para habilitar el inicio de sesión con Google en su aplicación, siga la documentación de Google para iOS. Si la autenticación tiene éxito, se genera un token de autenticación de OpenID Connect que Amazon Cognito utiliza para autenticar al usuario y generar un identificador único.

Una autenticación correcta da como resultado un objeto `GTMOAuth2Authentication` que contiene `unid_token`. Amazon Cognito utiliza este token para autenticar al usuario y generar un identificador único.

```
func finishedWithAuth(auth: GTMOAuth2Authentication!, error: NSError!) {
    if error != nil {
      print(error.localizedDescription)
    }
    else {
      let idToken = auth.parameters.objectForKey("id_token")
      credentialsProvider.logins = [AWSCognitoLoginProviderKey.Google.rawValue:
 idToken!]
    }
}
```

JavaScript

> **Note**
>
> Si su aplicación utiliza Google y está disponible en varias plataformas móviles, debe configurar Google como proveedor de OpenID Connect. Agregue todos los ID de cliente creados como valores de audiencia adicionales para permitir una mejor integración. Para obtener más información sobre el modelo de identidad de varios clientes de Google, consulte Cross-client Identity.

Configuración de Google

Para habilitar el inicio de sesión con Google en una aplicación JavaScript web, crea un proyecto de consola de Google Developers para tu aplicación.

1. Vaya a la consola de Google Developers y cree un proyecto.
2. Elija APIs & Services (API y servicios) y luego OAuth consent screen (Pantalla de consentimiento de OAuth). Personalice la información que Google muestra a sus usuarios cuando Google solicita su consentimiento para compartir sus datos de perfil con su aplicación.
3. Elija Credentials (Credenciales) y luego Create credentials (Crear credenciales). Elija OAuth client ID (ID de cliente de OAuth). Seleccione Web application (Aplicación web) en Application type (Tipo de aplicación). Cree un ID de cliente independiente para cada plataforma en la que desarrolle la aplicación.

4. Desde Credentials (Credenciales), elija Manage service accounts (Administrar cuentas de servicios). Elija Create service account (Crear cuenta de servicio). Ingrese los detalles de su cuenta de servicio y seleccione Create and continue (Crear y continuar).
5. Conceda a la cuenta de servicio acceso a su proyecto. Conceda a los usuarios acceso a la cuenta de servicio según lo requiera su aplicación.
6. Elija la nueva cuenta de servicio y luego la pestaña Keys (Claves) y Add key (Agregar clave). Cree y descargue una nueva clave JSON.

Para ver más información acerca de cómo utilizar la consola de Google Developers, consulte Crea y administra proyectos en la documentación de Google Cloud.

Para ver más información acerca de cómo integrar Google en su aplicación web, consulte el tema sobre inicio de sesión con Google en la documentación de Google Identity.

Configuración del proveedor externo en la consola de Amazon Cognito

Para agregar un proveedor de identidades (IdP) de Google

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Seleccione Google.
5. Ingrese el ID de cliente del proyecto de OAuth que creó en la Plataforma de Google Cloud. Para obtener más información, consulte Configurar OAuth 2.0 en la Ayuda de la consola de Google Cloud Platform.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.

    - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.

        i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.

   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.
   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.
   c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Uso de Google

Para habilitar el inicio de sesión con Google en su aplicación, siga la documentación de Google para la web.

Si la autenticación tiene éxito, se genera un objeto de respuesta que contiene un `id_token` que Amazon Cognito utiliza para autenticar al usuario y generar un identificador único:

```
function signinCallback(authResult) {
  if (authResult['status']['signed_in']) {

     // Add the Google access token to the Amazon Cognito credentials login map.
     AWS.config.credentials = new AWS.CognitoIdentityCredentials({
        IdentityPoolId: 'IDENTITY_POOL_ID',
        Logins: {
           'accounts.google.com': authResult['id_token']
        }
     });

     // Obtain AWS credentials
     AWS.config.credentials.get(function(){
        // Access AWS resources here.
     });
  }
}
```

Unity

Configuración de Google

Para habilitar el inicio de sesión de Google para una aplicación Unity, cree un proyecto de consola de Google Developers para su aplicación.

1. Vaya a la consola de Google Developers y cree un proyecto.
2. Elija APIs & Services (API y servicios) y luego OAuth consent screen (Pantalla de consentimiento de OAuth). Personalice la información que Google muestra a sus usuarios cuando Google solicita su consentimiento para compartir sus datos de perfil con su aplicación.
3. Elija Credentials (Credenciales) y luego Create credentials (Crear credenciales). Elija OAuth client ID (ID de cliente de OAuth). Seleccione Web application (Aplicación web) en Application type (Tipo de aplicación). Cree un ID de cliente independiente para cada plataforma en la que desarrolle la aplicación.
4. Para Unity, crea un ID de cliente de OAuth adicional paraAndroid y otro para iOS.
5. Desde Credentials (Credenciales), elija Manage service accounts (Administrar cuentas de servicios). Elija Create service account (Crear cuenta de servicio). Ingrese los detalles de su cuenta de servicio y seleccione Create and continue (Crear y continuar).
6. Conceda a la cuenta de servicio acceso a su proyecto. Conceda a los usuarios acceso a la cuenta de servicio según lo requiera su aplicación.
7. Elija la nueva cuenta de servicio y luego la pestaña Keys (Claves) y Add key (Agregar clave). Cree y descargue una nueva clave JSON.

Para ver más información acerca de cómo utilizar la consola de Google Developers, consulte Crea y administra proyectos en la documentación de Google Cloud.

Creación de un proveedor de OpenID en la consola de IAM

1. Cree un proveedor de OpenID en la consola de IAM. Para ver instrucciones sobre cómo configurar un proveedor de OpenID, consulte la página sobre uso de proveedores de identidad OpenID Connect.
2. Cuando se le solicite la URL de su proveedor, especifique `"https://accounts.google.com"`.
3. Cuando se le pida que introduzca un valor en el campo Audience (Público), introduzca uno de los tres ID de cliente que ha creado en los pasos anteriores.
4. Elija el nombre del proveedor y agregue dos públicos más con los dos otros ID de cliente.

Configuración del proveedor externo en la consola de Amazon Cognito

Elija Manage Identity Pools (Administrar grupos de identidades) de la página de inicio de la consola de Amazon Cognito:

Para agregar un proveedor de identidades (IdP) de Google

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Seleccione Google.
5. Ingrese el ID de cliente del proyecto de OAuth que creó en la Plataforma de Google Cloud. Para obtener más información, consulte Configurar OAuth 2.0 en la Ayuda de la consola de Google Cloud Platform.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.
   - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.
     i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.
     ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.
7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.
   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.
   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Instalación del complemento Unity Google

1. Añada el complemento Google Play Games para Unity al proyecto de Unity.
2. En Unity, en el menú de Windows, use los tres ID para las plataformas Android e iOS para configurar el complemento.

Uso de Google

En el siguiente código de muestra se muestra cómo recuperar el token de autenticación de Google Play Service:

```
void Start()
{
  PlayGamesClientConfiguration config = new
 PlayGamesClientConfiguration.Builder().Build();
  PlayGamesPlatform.InitializeInstance(config);
  PlayGamesPlatform.DebugLogEnabled = true;
  PlayGamesPlatform.Activate();
  Social.localUser.Authenticate(GoogleLoginCallback);
}

void GoogleLoginCallback(bool success)
{
  if (success)
  {
    string token = PlayGamesPlatform.Instance.GetIdToken();
    credentials.AddLogin("accounts.google.com", token);
  }
  else
  {
    Debug.LogError("Google login failed. If you are not running in an actual Android/
iOS device, this is expected.");
  }
}
```

Xamarin

> **Note**
>
> Amazon Cognito no es compatible con Google de forma nativa en la plataforma Xamarin. Actualmente, necesita el uso de una vista de web para seguir el flujo de inicio de sesión del navegador. Para obtener información acerca de cómo la integración de Google funciona con otros SDK, seleccione otra plataforma.

Para habilitar el inicio de sesión con Google en su aplicación, autentique a sus usuarios y obtenga un token de OpenID Connect para ellos. Amazon Cognito utiliza este token para generar un identificador de usuario único asociado a una identidad de Amazon Cognito. Puesto que el SDK de Google para Xamarin no le permite recuperar el token de OpenID Connect, utilice un cliente alternativo o el flujo web en una vista web.

Una vez que tenga el token, puede establecerlo en sus `CognitoAWSCredentials`:

```
credentials.AddLogin("accounts.google.com", token);
```

> **Note**
>
> Si su aplicación utiliza Google y está disponible en varias plataformas móviles, debe configurar Google como proveedor de OpenID Connect. Agregue todos los ID de cliente creados como valores de audiencia adicionales para permitir una mejor integración. Para obtener más información sobre el modelo de identidad de varios clientes de Google, consulte Cross-client Identity.

Configurar el inicio de sesión con Apple como un IdP de grupo de identidades

Amazon Cognito se integra en Sign in with Apple con el fin de ofrecer autenticación federada para sus usuarios de las aplicaciones móvil y web. En esta sección se explica cómo registrar y configurar su aplicación con Inicio de sesión con Apple como proveedor de identidad (IdP).

Para agregar Iniciar sesión con Apple como proveedor de autenticación a un grupo de identidades debe completar dos procedimientos. En primer lugar, integrar Inicio de sesión con Apple en una

aplicación y, a continuación, configurar Inicio de sesión con Apple en grupos de identidades. Para up-to-date obtener más información sobre cómo configurar el inicio de sesión con Apple, consulta Cómo configurar tu entorno para iniciar sesión con Apple en la documentación para desarrolladores de Apple.

Configurar SignInWithApple

Para configurar Inicio de sesión con Apple como IdP, debe registrar su aplicación en Apple para recibir el ID de cliente.

1. Cree una cuenta de desarrollador en Apple.
2. Inicie sesión con las credenciales de Apple.
3. En el panel de navegación de la izquierda, elija Certificates, IDs & Profiles (Certificados, identificadores y perfiles).
4. En el panel de navegación izquierdo, elija Identifiers (Identificadores).
5. En la página Identifiers (Identificadores), elija el icono +.
6. En la página Register a New Identifier (Registrar un nuevo identificador), elija App IDs (ID de aplicaciones) y, a continuación, Continue (Continuar).
7. En la página Register an App ID (Registrar un ID de aplicación), haga lo siguiente:
    a. En Description (Descripción), escriba una descripción.
    b. En Bundle ID (Identificador de paquete), escriba un identificador. Anote este ID de paquete, ya que necesitará este valor para definir a Apple como proveedor en el grupo de identidades.
    c. En Capabilities (Funcionalidades), elija SignInWithApple y, a continuación, elija Edit (Editar).
    d. En la página Inicio de sesión con Apple: configuración de ID de Apple, seleccione la configuración adecuada para la aplicación. A continuación, elija Guardar.
    e. Elija Continue (Continuar).
8. En la página Confirm your App ID (Confirmar ID de Apple), elija Register (Registrarse).
9. Continúe con el paso 10 si desea integrar Sign in with Apple en una aplicación nativa de iOS. El paso 11 es para aplicaciones que desea integrar con Inicio de sesión con Apple JS.
10. En la página Identifiers (Identificadores), elija el menú App IDs (ID de aplicaciones) y luego Services IDs (ID de servicios). Elija el icono +.
11. En la página Register a New Identifier (Registrar un nuevo identificador), elija Services IDs (ID de servicios) y, a continuación, Continue (Continuar).

12. En la página Register a Services ID (Registrar un ID de servicio), haga lo siguiente:

    a. En Description (Descripción), escriba una descripción.

    b. En Identifier (Identificador), escriba un identificador. Anote el ID de servicios ya que necesita este valor para configurar Apple como proveedor en su grupo de identidades.

    c. Seleccione Sign In with Apple (Inicio de sesión con Apple) y luego elija Configure (Configurar).

    d. En la página Web Authentication Configuration (Configuración de autenticación web), elija Primary App ID (ID de aplicación principal). En Website URLs (URL del sitio web), seleccione el icono +. En Domains and Subdomains (Dominios y subdominios), introduzca el nombre de dominio de su aplicación. En Return URLs (URL de devolución), introduzca la URL de devolución de llamada a la que redirige al usuario la autorización después autenticarse con Iniciar sesión con Apple.

    e. Elija Siguiente.

    f. Seleccione Continue (Continuar) y, a continuación, Register (Registrarse).

13. En el panel de navegación izquierdo, elija Keys (Claves).

14. En la página Keys (Claves), elija el icono +.

15. En la página Register a New Key (Registrar una nueva clave), haga lo siguiente:

    a. En Key Name (Nombre de clave), escriba un nombre de clave.

    b. Elija Sign In with Apple y, a continuación, Configure (Configurar).

    c. En la página Configure Key (Configurar clave), elija un Primary App ID (ID de aplicación principal) y luego elija Save (Guardar).

    d. Seleccione Continue (Continuar) y, a continuación, Register (Registrarse).

> **Note**
>
> Para integrar Inicio de sesión con Apple con una aplicación iOS nativa, consulte Implementación de la autenticación de usuario con Inicio de sesión con Apple.
> Para integrar Inicio de sesión con Apple en una plataforma que no sea nativa de iOS, consulte Inicio de sesión con Apple JS.

Configuración del proveedor externo en la consola de identidades federadas de Amazon Cognito

Utilice el siguiente procedimiento para configurar su proveedor externo.

Para agregar un inicio de sesión con el proveedor de identidades (IdP) de Apple

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Elija Iniciar sesión con Apple.
5. Ingrese el ID de servicios del proyecto de OAuth que creó con el Desarrollador de Apple. Para obtener más información, consulte Autenticación de usuarios con inicio de sesión con Apple en Iniciar sesión con documentación de Apple.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.
   - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.
     i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.
     ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.
7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.
   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.
   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.

c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

SignInWithApple como proveedor en los ejemplos de la CLI de identidades federadas de Amazon Cognito

En este ejemplo se crea un grupo de identidades llamado `MyIdentityPool` con Inicio de sesión con Apple como IdP.

```
aws cognito-identity create-identity-pool --identity-
pool-name MyIdentityPool --supported-login-providers
appleid.apple.com="sameple.apple.clientid"
```

Para obtener más información, vea Crear grupo de identidades

Generación de un ID de identidad Amazon Cognito

En este ejemplo se genera (o se recupera) un ID de Amazon Cognito. Esta API es pública, por lo que no se necesita ninguna credencial para llamar a esta API.

```
aws cognito-identity get-id --identity-pool-id SampleIdentityPoolId --
logins appleid.apple.com="SignInWithAppleIdToken"
```

Para obtener más información, consulte get-id.

Obtención de credenciales para un ID de identidad de Amazon Cognito

En este ejemplo se devuelven las credenciales del ID de identidad proporcionado e Inicio de sesión con inicio de sesión de Apple. Esta API es pública, por lo que no se necesita ninguna credencial para llamar a esta API.

```
aws cognito-identity get-credentials-for-identity --identity-id
SampleIdentityId --logins appleid.apple.com="SignInWithAppleIdToken"
```

Para obtener más información, consulta get-credentials-for-identity

Usar Sign in with Apple: Android

Apple no proporciona un SDK que admita Inicio de sesión con Apple para Android. Puede utilizar el flujo web en una vista web en su lugar.

- Para configurar Inicio de sesión con Apple en su aplicación, consulte el tema Configurar su página web para Inicio de sesión con Apple en la documentación de Apple.
- Para agregar un botón Sign in with Apple (Iniciar sesión con Apple) a la interfaz de usuario de Android, consulte el tema Displaying and Configuring Sign In with Apple Buttons en la documentación de Apple.
- Para autenticar de forma segura a los usuarios con Inicio de sesión con Apple, siga Autenticación de usuarios con Inicio de sesión con Apple en la documentación de Apple.

En Sign in with Apple se utiliza un objeto de sesión para realizar seguimiento de su estado. Amazon Cognito utiliza el token de ID de este objeto de sesión para autenticar al usuario, generar el identificador único y, si es necesario, conceder al usuario acceso a otros recursos. AWS

```
@Override
public void onSuccess(Bundle response) {
    String token = response.getString("id_token");
    Map<String, String> logins = new HashMap<String, String>();
    logins.put("appleid.apple.com", token);
    credentialsProvider.setLogins(logins);
}
```

Usar Sign in with Apple: iOS - Objective-C

Apple proporcionó compatibilidad de SDK para Sign in with Apple en aplicaciones iOS nativas. Para implementar la autenticación de usuario con Inicio de sesión con Apple en dispositivos iOS nativos, consulte el tema Implementar la autenticación de usuarios con Inicio de sesión con Apple en la documentación de Apple.

Amazon Cognito utiliza el token de ID para autenticar al usuario, generar el identificador único y, si es necesario, conceder al usuario acceso a otros recursos. AWS

```
(void)finishedWithAuth: (ASAuthorizationAppleIDCredential *)auth error: (NSError *)
 error {
        NSString *idToken = [ASAuthorizationAppleIDCredential
 objectForKey:@"identityToken"];
```

```
        credentialsProvider.logins = @{ "appleid.apple.com": idToken };
    }
```

Usar Sign in with Apple: iOS - Swift

Apple proporcionó compatibilidad de SDK para Sign in with Apple en aplicaciones iOS nativas. Para implementar la autenticación de usuario con Inicio de sesión con Apple en dispositivos iOS nativos, consulte el tema Implementar la autenticación de usuarios con Inicio de sesión con Apple en la documentación de Apple.

Amazon Cognito utiliza el token de ID para autenticar al usuario, generar el identificador único y, si es necesario, conceder al usuario acceso a otros recursos. AWS

Para obtener más información sobre cómo configurar Inicio de sesión con Apple en iOS, consulte Configurar Inicio de sesión con Apple.

```
func finishedWithAuth(auth: ASAuthorizationAppleIDCredential!, error: NSError!) {
    if error != nil {
      print(error.localizedDescription)
    }
    else {
      let idToken = auth.identityToken,
      credentialsProvider.logins = ["appleid.apple.com": idToken!]
    }
}
```

Utilice Iniciar sesión con Apple: JavaScript

Apple no proporciona un SDK que permita iniciar sesión con Apple para JavaScript. Puede utilizar el flujo web en una vista web en su lugar.

- Para configurar Inicio de sesión con Apple en su aplicación, consulte el tema Configurar su página web para Inicio de sesión con Apple en la documentación de Apple.
- Para añadir un botón de inicio de sesión con Apple a tu interfaz de JavaScript usuario, consulta Cómo mostrar y configurar el inicio de sesión con los botones de Apple en la documentación de Apple.
- Para autenticar de forma segura a los usuarios mediante Inicio de sesión con Apple, consulte el tema Configurar su página web para Inicio de sesión con Apple en la documentación de Apple.

En Sign in with Apple se utiliza un objeto de sesión para realizar seguimiento de su estado. Amazon Cognito utiliza el token de ID de este objeto de sesión para autenticar al usuario, generar el identificador único y, si es necesario, conceder al usuario acceso a otros recursos. AWS

```
function signinCallback(authResult) {
     // Add the apple's id token to the Amazon Cognito credentials login map.
     AWS.config.credentials = new AWS.CognitoIdentityCredentials({
        IdentityPoolId: 'IDENTITY_POOL_ID',
        Logins: {
           'appleid.apple.com': authResult['id_token']
        }
     });

     // Obtain AWS credentials
     AWS.config.credentials.get(function(){
        // Access AWS resources here.
     });
}
```

Usar Sign in with Apple: Xamarin

No tenemos un SDK que admita Inicio de sesión con Apple para Xamarin. Puede utilizar el flujo web en una vista web en su lugar.

- Para configurar Inicio de sesión con Apple en su aplicación, consulte el tema Configurar su página web para Inicio de sesión con Apple en la documentación de Apple.
- Para agregar un botón Sign in with Apple (Iniciar sesión con Apple) a la interfaz de usuario de Xamarin, consulte el tema Displaying and Configuring Sign In with Apple Buttons en la documentación de Apple.
- Para autenticar de forma segura a los usuarios mediante Inicio de sesión con Apple, consulte el tema Configurar su página web para Inicio de sesión con Apple en la documentación de Apple.

En Sign in with Apple se utiliza un objeto de sesión para realizar seguimiento de su estado. Amazon Cognito utiliza el token de ID de este objeto de sesión para autenticar al usuario, generar el identificador único y, si es necesario, conceder al usuario acceso a otros recursos. AWS

Una vez que tenga el token, puede establecerlo en sus `CognitoAWSCredentials`:

```
credentials.AddLogin("appleid.apple.com", token);
```

Configuración de un proveedor de OIDC como un IdP de grupo de identidades

OpenID Connect es un estándar abierto de autenticación compatible con varios proveedores de inicio de sesión. Amazon Cognito le ayuda a vincular identidades con los proveedores de OpenID Connect que configura mediante AWS Identity and Access Management.

Adición de un proveedor OpenID Connect

Para obtener más información acerca de cómo crear un proveedor de OpenID Connect, consulte Creación de proveedores de identidades de OpenID Connect (OIDC) en la Guía del usuario de AWS Identity and Access Management .

Asociación de un proveedor con Amazon Cognito

Para agregar un proveedor de identidades (IdP) de OIDC

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Elija OpenID Connect (OIDC).
5. Elija un proveedor de identidad OIDC del IAM de su. IdPs Cuenta de AWS Si desea agregar un nuevo proveedor de SAML, elija Crear nuevo proveedor para navegar hasta la consola de IAM.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.

    - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.

        i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

        ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.

   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.
   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.
   c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.
8. Seleccione Guardar cambios.

Puede asociar varios proveedores OpenID Connect a un único grupo de identidades.

Uso de OpenID Connect

Consulte la documentación de su proveedor para informarse de cómo iniciar sesión y recibir un token de ID.

Una vez que tenga el token, añádalo a la asignación de inicios de sesión. Utilice el URI de su proveedor como clave.

Validación de un token de OpenID Connect

En la primera integración con Amazon Cognito puede que reciba una excepción `InvalidToken`. Es importante entender cómo Amazon Cognito valida los tokens de OpenID Connect (OIDC).

> **Note**
>
> Como se especifica aquí (https://tools.ietf.org/html/rfc7523), con Amazon Cognito ofrece un periodo de gracia de 5 minutos para controlar cualquier sesgo del reloj entre sistemas.

1. El parámetro `iss` debe coincidir con la clave que utiliza la asignación de inicios de sesión (por ejemplo, login.provider.com).
2. La firma debe ser válida. Debe poder verificarse mediante una clave pública RSA.
3. La huella digital de la clave pública del certificado coincide con la huella digital que estableció en IAM cuando creó su proveedor OIDC.

4. Si el parámetro `azp` está presente, compruebe este valor comparándolo con la lista de los ID de cliente del proveedor OIDC.
5. Si el parámetro `azp` no está presente, compruebe el parámetro `aud` comparándolo con la lista de los ID de cliente del proveedor OIDC.

El sitio web jwt.io es un recurso valioso que puede usar para descodificar tokens para verificar estos valores.

Android

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("login.provider.com", token);
credentialsProvider.setLogins(logins);
```

iOS - Objective-C

```
credentialsProvider.logins = @{ "login.provider.com": token }
```

iOS - Swift

Para proporcionar el token de ID de OIDC a Amazon Cognito, implemente el protocolo `AWSIdentityProviderManager`.

En la implementación del método `logins`, devuelva un diccionario que contenga el nombre del proveedor OIDC configurado. Este diccionario actúa como la clave y el token de ID actual del usuario autenticado actúa como valor, tal como se muestra en el siguiente ejemplo de código.

```
class OIDCProvider: NSObject, AWSIdentityProviderManager {
    func logins() -> AWSTask<NSDictionary> {
        let completion = AWSTaskCompletionSource<NSString>()
        getToken(tokenCompletion: completion)
        return completion.task.continueOnSuccessWith { (task) -> AWSTask<NSDictionary>?
 in
            //login.provider.name is the name of the OIDC provider as setup in the
 Amazon Cognito console
            return AWSTask(result:["login.provider.name":task.result!])
        } as! AWSTask<NSDictionary>

    }
```

```
    func getToken(tokenCompletion: AWSTaskCompletionSource<NSString>) -> Void {
        //get a valid oidc token from your server, or if you have one that hasn't
 expired cached, return it

        //TODO code to get token from your server
        //...

        //if error getting token, set error appropriately
        tokenCompletion.set(error:NSError(domain: "OIDC Login", code: -1 , userInfo:
 ["Unable to get OIDC token" : "Details about your error"]))
        //else
        tokenCompletion.set(result:"result from server id token")
    }
}
```

Al crear una instancia`AWSCognitoCredentialsProvider`, pase la clase que se implementa AWSIdentityProviderManager como el valor de en el constructor. identityProviderManager Para obtener más información, vaya a la página de AWSCognitoCredentialsProviderreferencia y elija initWithRegionType:: identityPoolId. identityProviderManager

JavaScript

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
 IdentityPoolId: 'IDENTITY_POOL_ID',
 Logins: {
    'login.provider.com': token
 }
});
```

Unity

```
credentials.AddLogin("login.provider.com", token);
```

Xamarin

```
credentials.AddLogin("login.provider.com", token);
```

Configurar un proveedor de SAML como un IdP de grupo de identidades

Amazon Cognito admite la autenticación con proveedores de identidad (IdPs) mediante Security Assertion Markup Language 2.0 (SAML 2.0). Puede utilizar un IdP que admita el lenguaje SAML

con Amazon Cognito para proporcionar un flujo de incorporación sencillo a sus usuarios. El IdP compatible con SAML especifica los roles de IAM que los usuarios pueden asumir. De esta forma, distintos usuarios pueden recibir distintos conjuntos de permisos.

Configuración de un grupo de identidades para un IdP SAML

En los pasos siguientes se describe cómo configurar su grupo de identidades para utilizar un IdP SAML.

> **Note**
>
> A fin de poder configurar un grupo de identidades para que admita un proveedor SAML, primero tiene que configurar el IdP SAML en la consola de IAM. Para obtener más información, consulte Integración de proveedores de soluciones SAML externos con AWS en la guía del usuario de IAM.

Para agregar un proveedor de identidades (IdP) de SAML

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Elija SAML.
5. Elija un proveedor de identidad SAML del IAM de su. IdPs Cuenta de AWS Si desea agregar un nuevo proveedor de SAML, elija Crear nuevo proveedor para navegar hasta la consola de IAM.
6. Para establecer el rol que Amazon Cognito solicita cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Configuración del rol.

    - Puede asignar a los usuarios de ese IdP el Rol predeterminado que configuró al configurar el Rol autenticado o puede Elegir el rol con reglas.

        i. Si eligió Elegir rol con reglas, ingrese la Reclamación de origen de la autenticación del usuario, el Operador con el que desea comparar la afirmación, el Valor que hará que coincida con esta elección de rol y el Rol que desea asignar cuando la Asignación del rol coincida. Seleccione Agregar otra para crear una regla adicional en función de una condición diferente.

ii. Elija una Resolución de rol. Cuando las reclamaciones del usuario no coinciden con las reglas, puede denegar las credenciales o emitir credenciales para el Rol autenticado.

7. Para cambiar las etiquetas de la entidad principal que Amazon Cognito asigna cuando emite credenciales a los usuarios que se han autenticado con este proveedor, configure Atributos para el control de acceso.

   a. Para no aplicar ninguna etiqueta de entidad principal, elija Inactivo.
   b. Para aplicar etiquetas de entidades principales en función de las reclamaciones `sub` y `aud`, elija Usar mapeos predeterminados.
   c. Para crear su propio esquema personalizado de atributos para las etiquetas de la entidades principales, elija Usar mapeos personalizados. A continuación, ingrese una Clave de etiqueta que desee obtener de cada Reclamación que desee representar en una etiqueta.

8. Seleccione Guardar cambios.

Configuración del IdP SAML

Después de crear el proveedor SAML, configure su IdP SAML para añadir una relación de confianza entre el IdP y AWS. Con muchos IdPs, puede especificar una URL que el IdP pueda usar para leer la información y los certificados de la parte que confía en un documento XML. Para AWS ello, puede utilizar https://signin.aws.amazon.com/static/saml-metadata.xml. El siguiente paso consiste en configurar la respuesta de aserción SAML de su IdP para completar las notificaciones que necesita. AWS Para obtener más información sobre la configuración de notificaciones, consulte Configuración de aserciones SAML para la respuesta de autenticación.

Cuando el IdP de SAML incluye más de un certificado de firma en los metadatos de SAML, al iniciar sesión, el grupo de usuarios determina que la afirmación de SAML es válida si coincide con algún certificado de los metadatos de SAML.

Personalización de un rol de usuario con SAML

Al usar SAML con Identidad de Amazon Cognito se puede personalizar el rol para el usuario final. Amazon Cognito solo admite el flujo mejorado con el IdP basado en SAML. Para que el grupo de identidades utilice un IdP basado en SAML, no es necesario especificar un rol autenticado o sin autenticar. El atributo `https://aws.amazon.com/SAML/Attributes/Role` de la notificación especifica uno o varios pares compuestos por un ARN de proveedor y un ARN de rol, y delimitado con comas. Estos son los roles que el usuario puede asumir. El IdP SAML se puede configurar para rellenar los atributos de rol en función de la información de atributo de usuario que el IdP tiene

disponible. Si la aserción SAML recibe varios roles, rellene el parámetro `customRoleArn` opcional debe al llamar a `getCredentialsForIdentity`. El usuario asume este `customRoleArn` si el rol coincide con uno de la reclamación de la aserción SAML.

Autenticación de usuarios con un IdP SAML

Para federarse con el IdP basado en SAML, determine la URL en la que el usuario inicia el inicio de sesión. AWS la federación utiliza el inicio de sesión iniciado por el IdP. En AD FS 2.0, la URL adopta la forma `https://<fqdn>/adfs/ls/IdpInitiatedSignOn.aspx?loginToRp=urn:amazon:webservices`.

A fin de agregar compatibilidad para el IdP SAML en Amazon Cognito, primero autentique a los usuarios con el proveedor de identidad SAML a partir de su aplicación de iOS o Android. El código que utiliza para integrar y autenticar con el IdP SAML es específico de los proveedores SAML. Una vez que autentique al usuario, puede proporcionar la aserción SAML resultante a Identidad de Amazon Cognito mediante las API de Amazon Cognito.

No puede repetir ni reproducir una aserción de SAML en la asignación de `Logins` de la solicitud de API de grupo de identidades. Una aserción de SAML reproducida tiene un ID de aserción que duplica el ID de una solicitud de API anterior. Entre las operaciones de la API que pueden aceptar una afirmación de SAML en el `Logins` mapa se incluyen GetId, GetCredentialsForIdentity, GetOpenIdTokeny el ID. GetOpen TokenForDeveloperIdentity Puede reproducir un ID de aserción de SAML una vez por solicitud de API en un flujo de autenticación de grupo de identidades. Por ejemplo, puede proporcionar la misma aserción de SAML en una solicitud `GetId` y en una solicitud `GetCredentialsForIdentity` posterior, pero no en una segunda solicitud `GetId`.

Android

Si utiliza el SDK para Android, puede rellenar la asignación de inicios de sesión con la aserción SAML como se indica a continuación.

```
Map logins = new HashMap();
logins.put("arn:aws:iam::aws account id:saml-provider/name", "base64 encoded assertion
 response");
// Now this should be set to CognitoCachingCredentialsProvider object.
CognitoCachingCredentialsProvider credentialsProvider = new
 CognitoCachingCredentialsProvider(context, identity pool id, region);
credentialsProvider.setLogins(logins);
// If SAML assertion contains multiple roles, resolve the role by setting the custom
 role
```

```
credentialsProvider.setCustomRoleArn("arn:aws:iam::aws account id:role/
customRoleName");
// This should trigger a call to the Amazon Cognito service to get the credentials.
credentialsProvider.getCredentials();
```

iOS

Si utiliza el SDK de iOS puede proporcionar la aserción SAML en `AWSIdentityProviderManager` como se indica a continuación.

```
- (AWSTask<NSDictionary<NSString*,NSString*> *> *) logins {
    //this is hardcoded for simplicity, normally you would asynchronously go to your
 SAML provider
    //get the assertion and return the logins map using a AWSTaskCompletionSource
    return [AWSTask taskWithResult:@{@"arn:aws:iam::aws account id:saml-provider/
name":@"base64 encoded assertion response"}];
}

// If SAML assertion contains multiple roles, resolve the role by setting the custom
 role.
// Implementing this is optional if there is only one role.
- (NSString *)customRoleArn {
    return @"arn:aws:iam::accountId:role/customRoleName";
}
```

Identidades autenticadas por el desarrollador (grupos de identidades)

Amazon Cognito es compatible con las identidades autenticadas por el desarrollador y con la federación de identidades web mediante Configurar Facebook como un IdP de grupos de identidades, Configurar Google como un IdP de grupo de identidades, Configuración de Login with Amazon como un IdP de grupos de identidades y Configurar el inicio de sesión con Apple como un IdP de grupo de identidades. Con las identidades autenticadas por el desarrollador, puede registrar y autenticar a los usuarios mediante su propio proceso de autenticación existente y, al mismo tiempo, utilizar Amazon Cognito para sincronizar los datos de los usuarios y acceder a los recursos. AWS El uso de las identidades autenticadas por el desarrollador implica una interacción entre el dispositivo del usuario final, el backend para la autenticación y Amazon Cognito. Para obtener más información, consulte Descripción de la autenticación de Amazon Cognito, parte 2: Identidades autenticadas por desarrolladores, en el blog. AWS

Descripción del flujo de autenticación

La operación de la GetOpenIdTokenForDeveloperIdentityAPI puede iniciar la autenticación del desarrollador tanto para la autenticación básica como para la mejorada. Esta API autentica una solicitud con credenciales administrativas. El `Logins` mapa es el nombre de un desarrollador y proveedor de un grupo de identidades, `login.mydevprovider` junto con un identificador personalizado.

Ejemplo:

```
"Logins": {
        "login.mydevprovider": "my developer identifier"
    }
```

Autenticación mejorada

Llame a la operación de la GetCredentialsForIdentityAPI con un `Logins` mapa con el nombre `cognito-identity.amazonaws.com` y el valor del token desde`GetOpenIdTokenForDeveloperIdentity`.

Ejemplo:

```
"Logins": {
        "cognito-identity.amazonaws.com": "eyJra12345EXAMPLE"
    }
```

`GetCredentialsForIdentity`con identidades autenticadas por el desarrollador, devuelve credenciales temporales para el rol autenticado predeterminado del grupo de identidades.

Autenticación básica

Llama a la operación de la AssumeRoleWithWebIdentityAPI y solicita la `RoleArn` de cualquier rol de IAM que tenga definida una relación de confianza adecuada. Defina el valor del token del `WebIdentityToken` que se obtuvo`GetOpenIdTokenForDeveloperIdentity`.

Para obtener información sobre el flujo de autenticación de las identidades autenticadas por el desarrollador y en qué se diferencian de las identidades de los proveedores externos, consulte. Flujo de autenticación de grupos de identidades (identidades federadas)

Definición de un nombre de proveedor de desarrollador y asociación de dicho nombre a un grupo de identidades

Para utilizar las identidades autenticadas por el desarrollador, es preciso que el proveedor de desarrollador tenga un grupo de identidades asociado. Para ello, siga estos pasos:

Para agregar un proveedor de desarrolladores personalizado

1. Elija Grupos de identidades en la consola de Amazon Cognito. Seleccione un grupo de identidades.
2. Elija la pestaña Acceso de usuario.
3. Seleccione Agregar proveedor de identidades.
4. Elija un Proveedor de desarrolladores personalizado.
5. Ingrese un Nombre de proveedor de desarrolladores. No puede cambiar ni eliminar el proveedor de desarrolladores después de agregarlo.
6. Seleccione Guardar cambios.

Nota: una vez que se haya definido el nombre del proveedor, este no podrá modificarse.

Para obtener más indicaciones sobre cómo trabajar con la consola de Amazon Cognito, consulte Uso de la consola de Amazon Cognito.

Implementación de un proveedor de identidad

Android

Para usar las identidades autenticadas por el desarrollador, implemente su propia clase de proveedor de identidad que amplía `AWSAbstractCognitoIdentityProvider`. La clase de proveedor de identidad debe devolver un objeto de respuesta que contenga el token como un atributo.

El siguiente es un ejemplo básico de un proveedor de identidades.

```
public class DeveloperAuthenticationProvider extends
 AWSAbstractCognitoDeveloperIdentityProvider {

  private static final String developerProvider = "<Developer_provider_name>";
```

```
 public DeveloperAuthenticationProvider(String accountId, String identityPoolId,
Regions region) {
   super(accountId, identityPoolId, region);
   // Initialize any other objects needed here.
 }

 // Return the developer provider name which you choose while setting up the
 // identity pool in the &COG; Console

 @Override
 public String getProviderName() {
   return developerProvider;
 }

 // Use the refresh method to communicate with your backend to get an
 // identityId and token.

 @Override
 public String refresh() {

   // Override the existing token
   setToken(null);

   // Get the identityId and token by making a call to your backend
   // (Call to your backend)

   // Call the update method with updated identityId and token to make sure
   // these are ready to be used from Credentials Provider.

   update(identityId, token);
   return token;

 }

 // If the app has a valid identityId return it, otherwise get a valid
 // identityId from your backend.

 @Override
 public String getIdentityId() {

   // Load the identityId from the cache
   identityId = cachedIdentityId;

   if (identityId == null) {
```

```
       // Call to your backend
    } else {
       return identityId;
    }

  }
}
```

Para utilizar este proveedor de identidad, tiene que pasarlo en `CognitoCachingCredentialsProvider`. A continuación se muestra un ejemplo:

```
DeveloperAuthenticationProvider developerProvider = new
 DeveloperAuthenticationProvider( null, "IDENTITYPOOLID", context, Regions.USEAST1);
CognitoCachingCredentialsProvider credentialsProvider = new
 CognitoCachingCredentialsProvider( context, developerProvider, Regions.USEAST1);
```

iOS - Objective-C

Para usar las identidades autenticadas por el desarrollador, implemente su propia clase de proveedor de identidad que amplía AWSCognitoCredentialsProviderHelper. La clase de proveedor de identidad debe devolver un objeto de respuesta que contenga el token como un atributo.

```
@implementation DeveloperAuthenticatedIdentityProvider
/*
 * Use the token method to communicate with your backend to get an
 * identityId and token.
 */

- (AWSTask <NSString*> *) token {
    //Write code to call your backend:
    //Pass username/password to backend or some sort of token to authenticate user
    //If successful, from backend call getOpenIdTokenForDeveloperIdentity with logins
 map
    //containing "your.provider.name":"enduser.username"
    //Return the identity id and token to client
    //You can use AWSTaskCompletionSource to do this asynchronously

    // Set the identity id and return the token
    self.identityId = response.identityId;
    return [AWSTask taskWithResult:response.token];
}
```

```
@end
```

Para utilizar este proveedor de identidad, tiene que pasarlo en `AWSCognitoCredentialsProvider`, como se muestra en el ejemplo siguiente:

```
DeveloperAuthenticatedIdentityProvider * devAuth =
 [[DeveloperAuthenticatedIdentityProvider alloc]
 initWithRegionType:AWSRegionYOUR_IDENTITY_POOL_REGION
                                          identityPoolId:@"YOUR_IDENTITY_POOL_ID"
                                         useEnhancedFlow:YES
                                identityProviderManager:nil];
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider
 alloc]

 initWithRegionType:AWSRegionYOUR_IDENTITY_POOL_REGION
                                                                identityProvider:devAuth];
```

Si quiere dar soporte a las identidades sin autenticar y a las identidades autenticadas por el desarrollador, anule el método `logins` en la implementación de `AWSCognitoCredentialsProviderHelper`.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else{
        return [super logins];
    }
}
```

Si quiere dar soporte a las identidades autenticadas por el desarrollador y a los proveedores sociales, debe administrar quién es el proveedor actual en la implementación `logins` de `AWSCognitoCredentialsProviderHelper`.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else if (/*logic to determine if user is Facebook*/){
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook :
 [FBSDKAccessToken currentAccessToken] }];
    }else {
        return [super logins];
```

```
    }
}
```

iOS - Swift

Para usar las identidades autenticadas por el desarrollador, implemente su propia clase de proveedor de identidad que amplía AWSCognitoCredentialsProviderHelper. La clase de proveedor de identidad debe devolver un objeto de respuesta que contenga el token como un atributo.

```
import AWSCore
/*
 * Use the token method to communicate with your backend to get an
 * identityId and token.
 */
class DeveloperAuthenticatedIdentityProvider : AWSCognitoCredentialsProviderHelper {
    override func token() -> AWSTask<NSString> {
    //Write code to call your backend:
    //pass username/password to backend or some sort of token to authenticate user, if
 successful,
    //from backend call getOpenIdTokenForDeveloperIdentity with logins map containing
 "your.provider.name":"enduser.username"
    //return the identity id and token to client
    //You can use AWSTaskCompletionSource to do this asynchronously

    // Set the identity id and return the token
    self.identityId = resultFromAbove.identityId
    return AWSTask(result: resultFromAbove.token)
}
```

Para utilizar este proveedor de identidad, tiene que pasarlo en `AWSCognitoCredentialsProvider`, como se muestra en el ejemplo siguiente:

```
let devAuth =
 DeveloperAuthenticatedIdentityProvider(regionType: .YOUR_IDENTITY_POOL_REGION,
 identityPoolId: "YOUR_IDENTITY_POOL_ID", useEnhancedFlow: true,
 identityProviderManager:nil)
let credentialsProvider =
 AWSCognitoCredentialsProvider(regionType: .YOUR_IDENTITY_POOL_REGION,
 identityProvider:devAuth)
let configuration = AWSServiceConfiguration(region: .YOUR_IDENTITY_POOL_REGION,
 credentialsProvider:credentialsProvider)
AWSServiceManager.default().defaultServiceConfiguration = configuration
```

Si quiere dar soporte a las identidades sin autenticar y a las identidades autenticadas por el desarrollador, anule el método `logins` en la implementación de `AWSCognitoCredentialsProviderHelper`.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else {
        return super.logins()
    }
}
```

Si quiere dar soporte a las identidades autenticadas por el desarrollador y a los proveedores sociales, debe administrar quién es el proveedor actual en la implementación `logins` de `AWSCognitoCredentialsProviderHelper`.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else if (/*logic to determine if user is Facebook*/){
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
 ["Facebook" : "No current Facebook access token"]))
    }else {
        return super.logins()
    }
}
```

JavaScript

Una vez que obtenga un ID de identidad y un token de sesión del backend, deberá pasarlos al proveedor de `AWS.CognitoIdentityCredentials`. A continuación se muestra un ejemplo.

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
   IdentityPoolId: 'IDENTITY_POOL_ID',
   IdentityId: 'IDENTITY_ID_RETURNED_FROM_YOUR_PROVIDER',
   Logins: {
      'cognito-identity.amazonaws.com': 'TOKEN_RETURNED_FROM_YOUR_PROVIDER'
   }
```

```
});
```

Unity

Para usar las identidades autenticadas por el desarrollador, debe ampliar `CognitoAWSCredentials` y anular el método `RefreshIdentity` para recuperar el ID y el token de identidad del usuario del backend y devolverlos. A continuación, se muestra un ejemplo sencillo de un proveedor de identidad que se pone en contacto con un hipotético backend en "example.com":

```
using UnityEngine;
using System.Collections;
using Amazon.CognitoIdentity;
using System.Collections.Generic;
using ThirdParty.Json.LitJson;
using System;
using System.Threading;

public class DeveloperAuthenticatedCredentials : CognitoAWSCredentials
{
    const string PROVIDER_NAME = "example.com";
    const string IDENTITY_POOL = "IDENTITY_POOL_ID";
    static readonly RegionEndpoint REGION = RegionEndpoint.USEast1;

    private string login = null;

    public DeveloperAuthenticatedCredentials(string loginAlias)
        : base(IDENTITY_POOL, REGION)
    {
        login = loginAlias;
    }

    protected override IdentityState RefreshIdentity()
    {
        IdentityState state = null;
        ManualResetEvent waitLock = new ManualResetEvent(false);
        MainThreadDispatcher.ExecuteCoroutineOnMainThread(ContactProvider((s) =>
        {
            state = s;
            waitLock.Set();
        }));
        waitLock.WaitOne();
        return state;
    }
```

```
    IEnumerator ContactProvider(Action<IdentityState> callback)
    {
        WWW www = new WWW("http://example.com/?username="+login);
        yield return www;
        string response = www.text;

        JsonData json = JsonMapper.ToObject(response);

        //The backend has to send us back an Identity and a OpenID token
        string identityId = json["IdentityId"].ToString();
        string token = json["Token"].ToString();

        IdentityState state = new IdentityState(identityId, PROVIDER_NAME, token,
 false);
        callback(state);
    }
}
```

El código anterior utiliza un objeto distribuidor de subprocesos para llamar a una corutina. Si no dispone de una forma de hacerlo en su proyecto, puede utilizar el script siguiente en sus escenas:

```
using System;
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class MainThreadDispatcher : MonoBehaviour
{
    static Queue<IEnumerator> _coroutineQueue = new Queue<IEnumerator>();
    static object _lock = new object();

    public void Update()
    {
        while (_coroutineQueue.Count > 0)
        {
            StartCoroutine(_coroutineQueue.Dequeue());
        }
    }

    public static void ExecuteCoroutineOnMainThread(IEnumerator coroutine)
    {
        lock (_lock) {
```

```
            _coroutineQueue.Enqueue(coroutine);
        }
    }
}
```

Xamarin

Para usar las identidades autenticadas por el desarrollador, debe ampliar `CognitoAWSCredentials` y anular el método `RefreshIdentity` para recuperar el ID y el token de identidad del usuario del backend y devolverlos. A continuación, se muestra un ejemplo sencillo de un proveedor de identidades que contacta con un hipotético backend en "example.com":

```
public class DeveloperAuthenticatedCredentials : CognitoAWSCredentials
{
    const string PROVIDER_NAME = "example.com";
    const string IDENTITY_POOL = "IDENTITY_POOL_ID";
    static readonly RegionEndpoint REGION = RegionEndpoint.USEast1;
    private string login = null;

    public DeveloperAuthenticatedCredentials(string loginAlias)
        : base(IDENTITY_POOL, REGION)
    {
        login = loginAlias;
    }

    protected override async Task<IdentityState> RefreshIdentityAsync()
    {
        IdentityState state = null;
        //get your identity and set the state
        return state;
    }
}
```

Actualización de la asignación de inicios de sesión (solo Android e iOS)

Android

Después de autenticar correctamente al usuario con su propio sistema de autenticación, actualice la asignación de inicios de sesión con el nombre del proveedor de desarrollador y un identificador de usuario de desarrollador. Se trata de una cadena alfanumérica que identifica de forma exclusiva

a un usuario en el sistema de autenticación. Asegúrese de llamar al método `refresh` después de actualizar la asignación de inicios de sesión, ya que `identityId` podría haber cambiado:

```
HashMap<String, String> loginsMap = new HashMap<String, String>();
loginsMap.put(developerAuthenticationProvider.getProviderName(),
 developerUserIdentifier);

credentialsProvider.setLogins(loginsMap);
credentialsProvider.refresh();
```

iOS - Objective-C

El SDK para iOS solo llama al método `logins` para obtener la última asignación de inicios de sesión si no hay credenciales o estas han caducado. Si quiere obligar al SDK a obtener nuevas credenciales (por ejemplo, el usuario final ha pasado de no estar autenticado a estar autenticado y usted quiere credenciales sobre el usuario autenticado), llame a `clearCredentials` en su `credentialsProvider`.

```
[credentialsProvider clearCredentials];
```

iOS - Swift

El SDK para iOS solo llama al método `logins` para obtener la última asignación de inicios de sesión si no hay credenciales o estas han caducado. Si quiere obligar al SDK a obtener nuevas credenciales (por ejemplo, el usuario final ha pasado de no estar autenticado a estar autenticado y usted quiere credenciales sobre el usuario autenticado), llame a `clearCredentials` en su `credentialsProvider`.

```
credentialsProvider.clearCredentials()
```

Obtención de un token (lado del servidor)

Para obtener un token, llama. GetOpenIdTokenForDeveloperIdentity Esta API debe invocarse desde tu backend con las credenciales de AWS desarrollador. No debe invocarse desde el SDK de cliente. La API recibe el ID del grupo de identidades de Cognito, una asignación de inicios de sesión que contiene el nombre del proveedor de identidad como la clave y el identificador como el valor y, opcionalmente, un ID de identidad de Cognito (por ejemplo, está convirtiendo a un usuario sin autenticar en autenticado). El identificador puede ser el nombre de usuario del usuario, una dirección

de correo electrónico o un valor numérico. La API responde a la llamada con un ID de Cognito único para el usuario y un token de OpenID Connect para el usuario final.

Tenga en cuenta los siguientes puntos sobre el token devuelto por `GetOpenIdTokenForDeveloperIdentity`:

- Puede especificar una duración de vencimiento personalizada para el token para poder almacenarlo en la caché. Si no la proporciona, el token será válido durante 15 minutos.
- La duración máxima del token que puede definir es de 24 horas.
- Piense en las implicaciones para la seguridad que supone aumentar el token de duración. Si un atacante obtiene este token, puede cambiarlo por AWS credenciales para el usuario final durante el tiempo que dure el token.

En el siguiente fragmento Java, se muestra cómo inicializar un cliente de Amazon Cognito y recuperar un token para una identidad autenticada por el desarrollador.

```
// authenticate your end user as appropriate
// ....

// if authenticated, initialize a cognito client with your AWS developer credentials
AmazonCognitoIdentity identityClient = new AmazonCognitoIdentityClient(
  new BasicAWSCredentials("access_key_id", "secret_access_key")
);

// create a new request to retrieve the token for your end user
GetOpenIdTokenForDeveloperIdentityRequest request =
  new GetOpenIdTokenForDeveloperIdentityRequest();
request.setIdentityPoolId("YOUR_COGNITO_IDENTITY_POOL_ID");

request.setIdentityId("YOUR_COGNITO_IDENTITY_ID"); //optional, set this if your client
 has an
                                                   //identity ID that you want to link
 to this
                                                   //developer account

// set up your logins map with the username of your end user
HashMap<String,String> logins = new HashMap<>();
logins.put("YOUR_IDENTITY_PROVIDER_NAME","YOUR_END_USER_IDENTIFIER");
request.setLogins(logins);

// optionally set token duration (in seconds)
```

```
request.setTokenDuration(60 * 15l);
GetOpenIdTokenForDeveloperIdentityResult response =
  identityClient.getOpenIdTokenForDeveloperIdentity(request);

// obtain identity id and token to return to your client
String identityId = response.getIdentityId();
String token = response.getToken();

//code to return identity id and token to client
//...
```

Si sigue los pasos anteriores, debe tener la posibilidad de integrar las identidades autenticadas por el desarrollador en la aplicación. Si tiene algún problema o alguna pregunta, no dude en publicar en nuestros foros.

Conexión con una identidad social existente

Toda vinculación de proveedores efectuada durante el uso de identidades autenticadas por el desarrollador se debe realizar desde el backend. Para conectar una identidad personalizada a la identidad social de un usuario (Login with Amazon, Sign in with Apple, Facebook o Google), añade el token del proveedor de identidad al mapa de inicios de sesión cuando llames GetOpenIdTokenForDeveloperIdentity. Para ello, cuando llame al backend desde el SDK de cliente para autenticar al usuario final, pase además el token de proveedor social del usuario final.

Por ejemplo, si está intentando vincular una identidad personalizada a Facebook, añada el token de Facebook, además del identificador del proveedor de identidad, a la asignación de inicios de sesión cuando llame a `GetOpenIdTokenForDeveloperIdentity`.

```
logins.put("YOUR_IDENTITY_PROVIDER_NAME","YOUR_END_USER_IDENTIFIER");
logins.put("graph.facebook.com","END_USERS_FACEBOOK_ACCESSTOKEN");
```

Compatibilidad con la transición entre proveedores

Android

Es posible que la aplicación requiera admitir identidades sin autenticar o autenticadas mediante proveedores públicos (Login with Amazon, Sign in with Apple, Facebook o Google) junto con identidades autenticadas por el desarrollador. La diferencia fundamental entre las identidades autenticadas por el desarrollador y otras identidades (identidades sin autenticar e identidades

autenticadas mediante un proveedor público) radica en la forma de obtener identityId y el token. En el caso de las demás identidades, la aplicación móvil interactúa directamente con Amazon Cognito, en lugar de contactar con el sistema de autenticación. Por lo tanto, la aplicación móvil debería poder admitir dos flujos diferentes en función de la elección realizada por el usuario de la aplicación. Para esto, tendrá que realizar algunos cambios en el proveedor de identidades personalizado.

El método `refresh` comprueba el mapa de inicios de sesión. Si el mapa no está vacío y tiene una clave con el nombre del proveedor de desarrolladores, llame al backend. De lo contrario, llama al getIdentityId método y devuelve null.

```
public String refresh() {

   setToken(null);

   // If the logins map is not empty make a call to your backend
   // to get the token and identityId
   if (getProviderName() != null &&
      !this.loginsMap.isEmpty() &&
      this.loginsMap.containsKey(getProviderName())) {

      /**
       * This is where you would call your backend
       **/

      // now set the returned identity id and token in the provider
      update(identityId, token);
      return token;

   } else {
      // Call getIdentityId method and return null
      this.getIdentityId();
      return null;
   }
}
```

Igualmente, el método `getIdentityId` tendrá dos flujos, en función del contenido de la asignación de inicios de sesión:

```
public String getIdentityId() {

   // Load the identityId from the cache
   identityId = cachedIdentityId;
```

```
   if (identityId == null) {

      // If the logins map is not empty make a call to your backend
      // to get the token and identityId

      if (getProviderName() != null && !this.loginsMap.isEmpty()
         && this.loginsMap.containsKey(getProviderName())) {

         /**
           * This is where you would call your backend
          **/

         // now set the returned identity id and token in the provider
         update(identityId, token);
         return token;

      } else {
         // Otherwise call &COG; using getIdentityId of super class
         return super.getIdentityId();
      }

   } else {
      return identityId;
   }

}
```

iOS - Objective-C

Es posible que la aplicación requiera admitir identidades sin autenticar o autenticadas mediante proveedores públicos (Login with Amazon, Sign in with Apple, Facebook o Google) junto con identidades autenticadas por el desarrollador. Para ello, anule el AWSCognitoCredentialsProviderHelper`logins`método para poder devolver el mapa de inicios de sesión correcto en función del proveedor de identidad actual. En este ejemplo se muestra cómo puede moverse entre identidades sin autenticar e identidades autenticadas mediante Facebook o por el desarrollador.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else if (/*logic to determine if user is Facebook*/){
```

```
        return [AWSTask taskWithResult: @{ AWSIdentityProviderFacebook :
 [FBSDKAccessToken currentAccessToken] }];
    }else {
        return [super logins];
    }
}
```

Cuando pase de identidades sin autenticar a identidades autenticadas, llame a `[credentialsProvider clearCredentials];` para obligar al SDK a obtener credenciales autenticadas nuevas. Cuando cambie entre dos proveedores autenticados y no esté intentando vincularlos (por ejemplo, no proporciona tokens para varios proveedores en el diccionario de inicios de sesión), llame a `[credentialsProvider clearKeychain];`. Esto borrará las credenciales y la identidad y obligará al SDK a obtener otras nuevas.

iOS - Swift

Es posible que la aplicación requiera admitir identidades sin autenticar o autenticadas mediante proveedores públicos (Login with Amazon, Sign in with Apple, Facebook o Google) junto con identidades autenticadas por el desarrollador. Para ello, anule el AWSCognitoCredentialsProviderHelper`logins`método para poder devolver el mapa de inicios de sesión correcto en función del proveedor de identidad actual. En este ejemplo se muestra cómo puede moverse entre identidades sin autenticar e identidades autenticadas mediante Facebook o por el desarrollador.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else if (/*logic to determine if user is Facebook*/){
        if let token = AccessToken.current?.authenticationToken {
            return AWSTask(result: [AWSIdentityProviderFacebook:token])
        }
        return AWSTask(error:NSError(domain: "Facebook Login", code: -1 , userInfo:
 ["Facebook" : "No current Facebook access token"]))
    }else {
        return super.logins()
    }
}
```

Cuando pase de identidades sin autenticar a identidades autenticadas, llame a `credentialsProvider.clearCredentials()` para obligar al SDK a obtener credenciales

autenticadas nuevas. Cuando cambie entre dos proveedores autenticados y no esté intentando vincularlos (por ejemplo, no proporciona tokens para varios proveedores en el diccionario de inicios de sesión), llame a `credentialsProvider.clearKeychain()`. Esto borrará las credenciales y la identidad y obligará al SDK a obtener otras nuevas.

Unity

Es posible que la aplicación requiera admitir identidades sin autenticar o autenticadas mediante proveedores públicos (Login with Amazon, Sign in with Apple, Facebook o Google) junto con identidades autenticadas por el desarrollador. La diferencia fundamental entre las identidades autenticadas por el desarrollador y otras identidades (identidades sin autenticar e identidades autenticadas mediante un proveedor público) radica en la forma de obtener identityId y el token. En el caso de las demás identidades, la aplicación móvil interactúa directamente con Amazon Cognito, en lugar de contactar con el sistema de autenticación. La aplicación móvil debería poder admitir dos flujos diferentes en función de la elección realizada por el usuario de la aplicación. Para ello, tendrá que realizar algunos cambios en el proveedor de identidad personalizado.

La forma recomendada de hacerlo en Unity es ampliar el proveedor de identidad desde AmazonCognitoEnhancedIdentityProvide y llamar al RefreshAsync método principal en lugar del tuyo propio en caso de que el usuario no esté autenticado con tu propio servidor. AbstractCognitoIdentityProvider Si el usuario está autenticado, puede utilizar el mismo flujo explicado anteriormente.

Xamarin

Es posible que la aplicación requiera admitir identidades sin autenticar o autenticadas mediante proveedores públicos (Login with Amazon, Sign in with Apple, Facebook o Google) junto con identidades autenticadas por el desarrollador. La diferencia fundamental entre las identidades autenticadas por el desarrollador y otras identidades (identidades sin autenticar e identidades autenticadas mediante un proveedor público) radica en la forma de obtener identityId y el token. En el caso de las demás identidades, la aplicación móvil interactúa directamente con Amazon Cognito, en lugar de contactar con el sistema de autenticación. La aplicación móvil debería poder admitir dos flujos diferentes en función de la elección realizada por el usuario de la aplicación. Para esto, tendrá que realizar algunos cambios en el proveedor de identidades personalizado.

Cambio de los usuarios sin autenticar a los usuarios autenticados (Grupos de identidades)

Los grupos de identidades de Amazon Cognito admiten usuarios no autenticados y autenticados. Los usuarios no autenticados reciben acceso a sus recursos AWS incluso si no han iniciado sesión con ninguno de sus proveedores de identidad (IdP). Este grado de acceso es útil para mostrar contenido a los usuarios antes de que inicien sesión. Cada usuario sin autenticar tiene una identidad única en el grupo de identidades, aunque no hayan iniciado sesión y se hayan autenticado individualmente.

En esta sección se describe el caso en el que su usuario decide cambiar y en lugar de iniciar sesión con una identidad sin autenticar usa una identidad autenticada.

Android

Los usuarios pueden iniciar sesión en su aplicación como invitados sin autenticar. Con el tiempo podrían decidir iniciar sesión con uno de los IdP compatibles. Amazon Cognito se asegura de que en la identidad anterior se conserve el mismo identificador único que en la nueva y de que los datos del perfil se fusionen de manera automática.

Se informará a la aplicación de una fusión de perfil a través de la interfaz `IdentityChangedListener`. Implemente el método `identityChanged` en la interfaz para recibir estos mensajes:

```
@override
public void identityChanged(String oldIdentityId, String newIdentityId) {
    // handle the change
}
```

iOS - Objective-C

Los usuarios pueden iniciar sesión en su aplicación como invitados sin autenticar. Con el tiempo podrían decidir iniciar sesión con uno de los IdP compatibles. Amazon Cognito se asegura de que en la identidad anterior se conserve el mismo identificador único que en la nueva y de que los datos del perfil se fusionen de manera automática.

`NSNotificationCenter` informa a la aplicación de que se ha producido una fusión de perfil:

```
[[NSNotificationCenter defaultCenter] addObserver:self
                                         selector:@selector(identityIdDidChange:)
```

```
                                        name:AWSCognitoIdentityIdChangedNotification
                                        object:nil];

-(void)identityDidChange:(NSNotification*)notification {
    NSDictionary *userInfo = notification.userInfo;
    NSLog(@"identity changed from %@ to %@",
        [userInfo objectForKey:AWSCognitoNotificationPreviousId],
        [userInfo objectForKey:AWSCognitoNotificationNewId]);
}
```

iOS - Swift

Los usuarios pueden iniciar sesión en su aplicación como invitados sin autenticar. Con el tiempo podrían decidir iniciar sesión con uno de los IdP compatibles. Amazon Cognito se asegura de que en la identidad anterior se conserve el mismo identificador único que en la nueva y de que los datos del perfil se fusionen de manera automática.

`NSNotificationCenter` informa a la aplicación de que se ha producido una fusión de perfil:

```
[NSNotificationCenter.defaultCenter().addObserver(observer: self
   selector:"identityDidChange"
   name:AWSCognitoIdentityIdChangedNotification
   object:nil)

func identityDidChange(notification: NSNotification!) {
  if let userInfo = notification.userInfo as? [String: AnyObject] {
    print("identity changed from: \(userInfo[AWSCognitoNotificationPreviousId])
    to: \(userInfo[AWSCognitoNotificationNewId])")
  }
}
```

JavaScript

Usuario sin autenticar inicialmente

Los usuarios suelen comenzar con el rol sin autenticar. Para este rol, usted establece la propiedad de las credenciales del objeto de configuración sin una propiedad de inicio de sesión. En este caso, la configuración predeterminada podría tener el siguiente aspecto:

```

// set the default config object
```

```
var creds = new AWS.CognitoIdentityCredentials({
    IdentityPoolId: 'us-east-1:1699ebc0-7900-4099-b910-2df94f52a030'
});
AWS.config.credentials = creds;
```

Cambie a usuario autenticado

Cuando un usuario sin autenticar inicia sesión en un proveedor de identidad y tiene un token, puede cambiar el usuario de no estar autenticado a estar autenticado llamando a una función personalizada que actualiza el objeto de las credenciales y añade el token de inicio de sesión:

```
// Called when an identity provider has a token for a logged in user
function userLoggedIn(providerName, token) {
    creds.params.Logins = creds.params.Logins || {};
    creds.params.Logins[providerName] = token;

    // Expire credentials to refresh them on the next request
    creds.expired = true;
}
```

También puede crear un objeto de `CognitoIdentityCredentials`. Si lo hace, debe restablecer las propiedades de las credenciales de cualquier objeto de servicio existente para reflejar la información de configuración de las credenciales actualizadas. Consulte la sección relativa al uso del objeto de configuración global.

Para obtener más información sobre el objeto `CognitoIdentityCredentials`, consulte .CognitoIdentityCredentials de AWS en la referencia de la API de AWS SDK for JavaScript.

Unity

Los usuarios pueden iniciar sesión en su aplicación como invitados sin autenticar. Con el tiempo podrían decidir iniciar sesión con uno de los IdP compatibles. Amazon Cognito se asegura de que en la identidad anterior se conserve el mismo identificador único que en la nueva y de que los datos del perfil se fusionen de manera automática.

Puede suscribirse a `IdentityChangedEvent` para que se le notifiquen las fusiones de perfil:

```
credentialsProvider.IdentityChangedEvent += delegate(object sender,
 CognitoAWSCredentials.IdentityChangedArgs e)
```

```
{
    // handle the change
    Debug.log("Identity changed from " + e.OldIdentityId + " to " + e.NewIdentityId);
};
```

Xamarin

Los usuarios pueden iniciar sesión en su aplicación como invitados sin autenticar. Con el tiempo podrían decidir iniciar sesión con uno de los IdP compatibles. Amazon Cognito se asegura de que en la identidad anterior se conserve el mismo identificador único que en la nueva y de que los datos del perfil se fusionen de manera automática.

```
credentialsProvider.IdentityChangedEvent += delegate(object sender,
 CognitoAWSCredentials.IdentityChangedArgs e){
    // handle the change
    Console.WriteLine("Identity changed from " + e.OldIdentityId + " to " +
 e.NewIdentityId);
};
```

Amazon Cognito Sync

Si es la primera vez que utiliza Amazon Cognito Sync, utilice AWS AppSync. Como Amazon Cognito Sync, AWS AppSync es un servicio destinado a sincronizar los datos de las aplicaciones entre dispositivos.
Con este, se pueden sincronizar los datos de usuarios, como las preferencias de aplicación o el estado del juego. También amplía estas funcionalidades, ya que permite que varios usuarios se sincronicen y colaboren en tiempo real sobre los datos compartidos.

Amazon Cognito Sync es un Servicio de AWS y una biblioteca de cliente que permite la sincronización entre dispositivos de datos de usuarios relacionados con la aplicación. Amazon Cognito Sync puede sincronizar los datos de los perfiles de usuario entre los dispositivos móviles y la web sin necesidad de utilizar su propio backend. Las bibliotecas de cliente almacenan los datos localmente en la caché para que su aplicación pueda leer y escribir datos, sin importar el estado de conexión del dispositivo. Cuando el dispositivo esté en línea, podrá sincronizar los datos. Cuando el dispositivo esté en línea, podrá notificar inmediatamente a otros dispositivos que hay una actualización disponible.

Para obtener información acerca de la disponibilidad regional de Amazon Cognito Identity, consulte Disponibilidad regional del servicio de AWS.

Para obtener más información sobre Amazon Cognito Sync, consulte los siguientes temas.

Temas

Introducción a Amazon Cognito Sync

> ⚠ Si es la primera vez que usa Amazon Cognito Sync, utilice AWS AppSync. Como Amazon Cognito Sync, AWS AppSync es un servicio destinado a sincronizar los datos de las aplicaciones entre dispositivos.
> Con este, se pueden sincronizar los datos de usuarios, como las preferencias de aplicación o el estado del juego. También amplía estas funcionalidades, ya que permite que varios usuarios se sincronicen y colaboren en tiempo real sobre los datos compartidos.

Amazon Cognito Sync es un servicio de AWS y una biblioteca cliente que permite la sincronización entre dispositivos de los datos de usuarios relacionados con la aplicación. Puede usarlo para sincronizar los datos de perfiles de usuario en los dispositivos móviles y aplicaciones web. Las bibliotecas de cliente almacenan los datos localmente en la caché para que su aplicación pueda leer y escribir datos, sea cual sea el estado de conexión del dispositivo. Cuando el dispositivo está online, puede sincronizar los datos y, si configura una sincronización por inserción, notifique inmediatamente a los demás dispositivos que hay una actualización disponible.

Configuración de un grupo de identidades de Amazon Cognito

Amazon Cognito Sync exige un grupo de identidades de Amazon Cognito para proporcionar identidades de usuario. Antes de usar Amazon Cognito Sync, primero debe configurar un grupo de identidades. Para crear un grupo de identidades e instalar el SDK, consulte Introducción a los grupos de identidades de Amazon Cognito.

Almacenamiento y sincronización de datos

Una vez que haya configurado el grupo de identidades e instalado el SDK, puede comenzar a almacenar y sincronizar datos entre dispositivos. Para obtener más información, consulte Sincronización de datos.

Sincronización de datos

> ⚠ Si es la primera vez que usa Amazon Cognito Sync, utilice AWS AppSync. Como Amazon Cognito Sync, AWS AppSync es un servicio destinado a sincronizar los datos de las aplicaciones entre dispositivos.

Con este, se pueden sincronizar los datos de usuarios, como las preferencias de aplicación o el estado del juego. También amplía estas funcionalidades, ya que permite que varios usuarios se sincronicen y colaboren en tiempo real sobre los datos compartidos.

Con Amazon Cognito, puede guardar los datos de usuarios en conjuntos de datos que contienen pares clave-valor. Amazon Cognito asocia estos datos a una entidad en el grupo de identidades, de modo que su aplicación puede acceder a ella a través de los inicios de sesión y los dispositivos. Para sincronizar estos datos entre el servicio de Amazon Cognito y los dispositivos de un usuario final, invoque el método de sincronización. Cada conjunto de datos puede tener un tamaño máximo de 1 MB. Puede asociar hasta 20 conjuntos de datos a una identidad.

El cliente de Amazon Cognito Sync crea una memoria caché local para los datos de identidad. Cuando la aplicación lee y escribe claves se comunica con la memoria caché local. Esta comunicación garantiza que todos los cambios realizados en el dispositivo estén disponibles inmediatamente en el dispositivo, incluso si está sin conexión. Cuando se llama al método de sincronización, los cambios que provienen del servicio se envían al dispositivo, mientras que todos los cambios locales se transmiten al servicio. Ahora los cambios ya se podrán sincronizar con otros dispositivos.

Inicialización del cliente de Amazon Cognito Sync

Para inicializar el cliente de Amazon Cognito Sync, primero debe crear un proveedor de credenciales. El proveedor de credenciales recibe credenciales temporales de AWS para que la aplicación pueda acceder a sus recursos de AWS. También debe importar los archivos de encabezado necesarios. Ejecute los pasos siguientes para inicializar el cliente de Amazon Cognito Sync.

Android

1. Cree un proveedor de credenciales siguiendo las instrucciones descritas en Obtención de credenciales.
2. Importe el paquete de Amazon Cognito del siguiente modo: `import com.amazonaws.mobileconnectors.cognito.*;`
3. Inicialice Amazon Cognito Sync. Transfiera el contexto de la aplicación para Android, el ID del grupo de identidades, un Región de AWS y un proveedor de credenciales de Amazon Cognito inicializado de la siguiente manera:

```
CognitoSyncManager client = new CognitoSyncManager(
    getApplicationContext(),
    Regions.YOUR_REGION,
    credentialsProvider);
```

iOS - Objective-C

1. Cree un proveedor de credenciales siguiendo las instrucciones descritas en Obtención de credenciales.
2. Importe `AWSCore` y `Cognito` e inicialice `AWSCognito` de la siguiente manera:

```
#import <AWSiOSSDKv2/AWSCore.h>
#import <AWSCognitoSync/Cognito.h>

AWSCognito *syncClient = [AWSCognito defaultCognito];
```

3. Si utiliza CocoaPods, sustituya `<AWSiOSSDKv2/AWSCore.h>` por `AWSCore.h`. Siga la misma sintaxis para la importación de Amazon Cognito.

iOS - Swift

1. Cree un proveedor de credenciales siguiendo las instrucciones descritas en Obtención de credenciales.
2. Importe e inicialice `AWSCognito` de la siguiente manera:

```
import AWSCognito
let syncClient = AWSCognito.default()!
```

JavaScript

1. Descargue el administrador de Amazon Cognito Sync para JavaScript.
2. Incluya la biblioteca del administrador de sincronización en el proyecto.
3. Cree un proveedor de credenciales siguiendo las instrucciones descritas en Obtención de credenciales.
4. Inicialice el administrador de sincronizaciones de la siguiente manera:

```
var syncManager = new AWS.CognitoSyncManager();
```

Unity

1. Cree una instancia de `CognitoAWSCredentials`, siguiendo las instrucciones descritas en Obtención de credenciales.
2. Cree una instancia de `CognitoSyncManager`. Transfiera el objeto `CognitoAwsCredentials` y un `AmazonCognitoSyncConfig`, e incluya al menos el conjunto Región, de la siguiente manera:

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =
 REGION };
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Xamarin

1. Cree una instancia de `CognitoAWSCredentials`, siguiendo las instrucciones descritas en Obtención de credenciales.
2. Cree una instancia de `CognitoSyncManager`. Transfiera el objeto `CognitoAwsCredentials` y un `AmazonCognitoSyncConfig`, e incluya al menos el conjunto Región, de la siguiente manera:

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =
 REGION };
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Descripción de los conjuntos de datos

Amazon Cognito organiza los datos del perfil del usuario en conjuntos de datos. Cada conjunto de datos puede contener hasta 1 MB de datos en forma de pares de clave-valor. Un conjunto de datos es la entidad más precisa que puede sincronizar. Las operaciones de lectura y escritura realizadas en un conjunto de datos solo afectan al almacén local mientras no se invoque el método de sincronización. Amazon Cognito identifica un conjunto de datos mediante una cadena única. Puede crear un conjunto de datos nuevo o abrir uno existente, tal como se muestra a continuación.

Android

```
Dataset dataset = client.openOrCreateDataset("datasetname");
```

Para eliminar un conjunto de datos, primero tiene que llamar al método que lo eliminará del almacenamiento local y, a continuación, al método `synchronize` a fin de eliminar el conjunto de datos de Amazon Cognito del siguiente modo:

```
dataset.delete();
dataset.synchronize(syncCallback);
```

iOS - Objective-C

```
AWSCognitoDataset *dataset = [syncClient openOrCreateDataset:@"myDataSet"];
```

Para eliminar un conjunto de datos, primero tiene que llamar al método que lo eliminará del almacenamiento local y, a continuación, al método `synchronize` a fin de eliminar el conjunto de datos de Amazon Cognito del siguiente modo:

```
[dataset clear];
[dataset synchronize];
```

iOS - Swift

```
let dataset = syncClient.openOrCreateDataset("myDataSet")!
```

Para eliminar un conjunto de datos, primero tiene que llamar al método que lo eliminará del almacenamiento local y, a continuación, al método `synchronize` a fin de eliminar el conjunto de datos de Amazon Cognito del siguiente modo:

```
dataset.clear()
dataset.synchronize()
```

JavaScript

```
syncManager.openOrCreateDataset('myDatasetName', function(err, dataset) {
   // ...
```

```
});
```

Unity

```
string myValue = dataset.Get("myKey");
dataset.Put("myKey", "newValue");
```

Para eliminar una clave de un conjunto de datos, utilice `Remove` de la siguiente manera:

```
dataset.Remove("myKey");
```

Xamarin

```
Dataset dataset = syncManager.OpenOrCreateDataset("myDatasetName");
```

Para eliminar un conjunto de datos, primero tiene que llamar al método que lo eliminará del almacenamiento local y, a continuación, al método `synchronize` a fin de eliminar el conjunto de datos de Amazon Cognito del siguiente modo:

```
dataset.Delete();
dataset.SynchronizeAsync();
```

Lectura y escritura de datos en conjuntos de datos

Los conjuntos de datos de Amazon Cognito funcionan como diccionarios, con valores accesibles por clave: Puede leer, añadir o modificar las claves y los valores de un conjunto de datos como si el conjunto de datos fuese un diccionario, tal como se muestra en el ejemplo.

Tenga en cuenta que los valores escritos en un conjunto de datos solo afectan a la copia local de los datos almacenada en la caché hasta que llame al método de sincronización.

Android

```
String value = dataset.get("myKey");
dataset.put("myKey", "my value");
```

iOS - Objective-C

```
[dataset setString:@"my value" forKey:@"myKey"];
```

```
NSString *value = [dataset stringForKey:@"myKey"];
```

iOS - Swift

```
dataset.setString("my value", forKey:"myKey")
let value = dataset.stringForKey("myKey")
```

JavaScript

```
dataset.get('myKey', function(err, value) {
  console.log('myRecord: ' + value);
});

dataset.put('newKey', 'newValue', function(err, record) {
  console.log(record);
});

dataset.remove('oldKey', function(err, record) {
  console.log(success);
});
```

Unity

```
string myValue = dataset.Get("myKey");
dataset.Put("myKey", "newValue");
```

Xamarin

```
//obtain a value
string myValue = dataset.Get("myKey");

// Create a record in a dataset and synchronize with the server
dataset.OnSyncSuccess += SyncSuccessCallback;
dataset.Put("myKey", "myValue");
dataset.SynchronizeAsync();

void SyncSuccessCallback(object sender, SyncSuccessEventArgs e) {
  // Your handler code here
```

```
}
```

Android

Para eliminar claves de un conjunto de datos, utilice el método `remove` del siguiente modo:

```
dataset.remove("myKey");
```

iOS - Objective-C

Para eliminar una clave de un conjunto de datos, utilice `removeObjectForKey` de la siguiente manera:

```
[dataset removeObjectForKey:@"myKey"];
```

iOS - Swift

Para eliminar una clave de un conjunto de datos, utilice `removeObjectForKey` de la siguiente manera:

```
dataset.removeObjectForKey("myKey")
```

Unity

Para eliminar una clave de un conjunto de datos, utilice `Remove` de la siguiente manera:

```
dataset.Remove("myKey");
```

Xamarin

Puede utilizar `Remove` para eliminar una clave de un conjunto de datos:

```
dataset.Remove("myKey");
```

Sincronización de datos locales con el almacén de sincronización

Android

Mediante el método `synchronize`, se comparan los datos locales almacenados en la memoria caché con los datos guardados en el almacén de Amazon Cognito Sync. Los cambios remotos se extraen del almacén de Amazon Cognito Sync. Si se produce algún conflicto, se invoca la resolución de conflictos y los valores actualizados en el dispositivo se envían al servicio. Para sincronizar un conjunto de datos, llame a su método `synchronize`:

```
dataset.synchronize(syncCallback);
```

El método `synchronize` recibe una implementación de la interfaz `SyncCallback`, tratada a continuación.

El método `synchronizeOnConnectivity()` intenta realizar la sincronización cuando la conectividad está disponible. Si la conectividad está disponible inmediatamente, `synchronizeOnConnectivity()` se comporta como `synchronize()`. De lo contrario, supervisará los cambios de conectividad y realizará la sincronización cuando la conectividad esté disponible. Si se llama varias veces a `synchronizeOnConnectivity()`, solo se mantendrá la última solicitud de sincronización y solo se desencadenará la última devolución de llamada. Si el conjunto de datos o la devolución de llamada se limpia de la memoria, este método no realizará una sincronización, y la devolución de llamada no se iniciará.

Para obtener más información acerca de la sincronización de datos y las diversas devoluciones de llamada, consulte Gestión de la devolución de llamadas.

iOS - Objective-C

Mediante el método `synchronize`, se comparan los datos locales almacenados en la memoria caché con los datos guardados en el almacén de Amazon Cognito Sync. Los cambios remotos se extraen del almacén de Amazon Cognito Sync. Si se produce algún conflicto, se invoca la resolución de conflictos y los valores actualizados en el dispositivo se envían al servicio. Para sincronizar un conjunto de datos, llame a su método `synchronize`:

El método `synchronize` es asíncrono y devuelve un objeto `AWSTask` para tratar la respuesta:

```
[[dataset synchronize] continueWithBlock:^id(AWSTask *task) {
    if (task.isCancelled) {
        // Task cancelled.
    } else if (task.error) {
        // Error while executing task.
```

```
    } else {
        // Task succeeded. The data was saved in the sync store.
    }
    return nil;
}];
```

El método `synchronizeOnConnectivity` intenta realizar la sincronización cuando el dispositivo dispone de conectividad. En primer lugar, `synchronizeOnConnectivity` comprueba la conectividad y, si el dispositivo está online, invoca inmediatamente a synchronize y devuelve el objeto `AWSTask` asociado al intento.

Si el dispositivo está sin conexión, `synchronizeOnConnectivity` 1) programa una sincronización para la siguiente vez que el dispositivo esté online y 2) devuelve un `AWSTask` con un resultado nulo. La sincronización programada solo es válida durante el ciclo de vida del objeto del conjunto de datos. Los datos no se sincronizan si se cierra la aplicación antes de recuperar la conectividad. Si quiere recibir una notificación cuando se producen eventos en la sincronización programada, debe añadir observadores de notificaciones encontradas en `AWSCognito`.

Para obtener más información acerca de la sincronización de datos y las diversas devoluciones de llamada, consulte Gestión de la devolución de llamadas.

iOS - Swift

Mediante el método `synchronize`, se comparan los datos locales almacenados en la memoria caché con los datos guardados en el almacén de Amazon Cognito Sync. Los cambios remotos se extraen del almacén de Amazon Cognito Sync. Si se produce algún conflicto, se invoca la resolución de conflictos y los valores actualizados en el dispositivo se envían al servicio. Para sincronizar un conjunto de datos, llame a su método `synchronize`:

El método `synchronize` es asíncrono y devuelve un objeto `AWSTask` para tratar la respuesta:

```
dataset.synchronize().continueWith(block: { (task) -> AnyObject? in

            if task.isCancelled {
                // Task cancelled.
            } else if task.error != nil {
                // Error while executing task
            } else {
                // Task succeeded. The data was saved in the sync store.
            }
            return task
```

```
})
```

El método `synchronizeOnConnectivity` intenta realizar la sincronización cuando el dispositivo dispone de conectividad. En primer lugar, `synchronizeOnConnectivity` comprueba la conectividad y, si el dispositivo está online, invoca inmediatamente a `synchronize` y devuelve el objeto `AWSTask` asociado al intento.

Si el dispositivo está sin conexión, `synchronizeOnConnectivity` 1) programa una sincronización para la siguiente vez que el dispositivo esté online y 2) devuelve un objeto `AWSTask` con un resultado nulo. La sincronización programada solo es válida durante el ciclo de vida del objeto del conjunto de datos. Los datos no se sincronizan si se cierra la aplicación antes de recuperar la conectividad. Si quiere recibir una notificación cuando se producen eventos en la sincronización programada, debe añadir observadores de notificaciones encontradas en `AWSCognito`.

Para obtener más información acerca de la sincronización de datos y las diversas devoluciones de llamada, consulte Gestión de la devolución de llamadas.

JavaScript

Mediante el método `synchronize`, se comparan los datos locales almacenados en la memoria caché con los datos guardados en el almacén de Amazon Cognito Sync. Los cambios remotos se extraen del almacén de Amazon Cognito Sync. Si se produce algún conflicto, se invoca la resolución de conflictos y los valores actualizados en el dispositivo se envían al servicio. Para sincronizar un conjunto de datos, llame a su método `synchronize`:

```
dataset.synchronize();
```

Para obtener más información acerca de la sincronización de datos y las diversas devoluciones de llamada, consulte Gestión de la devolución de llamadas.

Unity

Con el método synchronize, se comparan los datos locales almacenados en la memoria caché con los datos guardados en el almacén de Amazon Cognito Sync. Los cambios remotos se extraen del almacén de Amazon Cognito Sync. Si se produce algún conflicto, se invoca la resolución de conflictos y los valores actualizados en el dispositivo se envían al servicio. Para sincronizar un conjunto de datos, llame a su método `synchronize`:

```
dataset.Synchronize();
```

La sincronización se ejecutará de forma asíncrona y acabará llamando a una de las diversas devoluciones de llamadas que puede especificar en el conjunto de datos.

Para obtener más información acerca de la sincronización de datos y las diversas devoluciones de llamada, consulte Gestión de la devolución de llamadas.

Xamarin

Mediante el método `synchronize`, se comparan los datos locales almacenados en la memoria caché con los datos guardados en el almacén de Amazon Cognito Sync. Los cambios remotos se extraen del almacén de Amazon Cognito Sync. Si se produce algún conflicto, se invoca la resolución de conflictos y los valores actualizados en el dispositivo se envían al servicio. Para sincronizar un conjunto de datos, llame a su método `synchronize`:

```
dataset.SynchronizeAsync();
```

Para obtener más información acerca de la sincronización de datos y las diversas devoluciones de llamada, consulte Gestión de la devolución de llamadas.

Gestión de la devolución de llamadas

> ⚠ Si es la primera vez que usa Amazon Cognito Sync, utilice AWS AppSync. Como Amazon Cognito Sync, AWS AppSync es un servicio destinado a sincronizar los datos de las aplicaciones entre dispositivos.
> Con este, se pueden sincronizar los datos de usuarios, como las preferencias de aplicación o el estado del juego. También amplía estas funcionalidades, ya que permite que varios usuarios se sincronicen y colaboren en tiempo real sobre los datos compartidos.

En esta sección se describe cómo tratar las devoluciones de llamadas.

Android

Interfaz SyncCallback

Si implementa la interfaz `SyncCallback`, puede recibir en la aplicación notificaciones acerca de la sincronización del conjunto de datos. De esta manera, la aplicación puede tomar decisiones activas

acerca de la eliminación de datos locales, la combinación o no de perfiles autenticados y la solución de los conflictos de sincronización. Debe aplicar los métodos siguientes, obligatorios en la interfaz:

- `onSuccess()`
- `onFailure()`
- `onConflict()`
- `onDatasetDeleted()`
- `onDatasetsMerged()`

Tenga en cuenta que, si no quiere especificar todas las devoluciones de llamadas, también puede utilizar la clase `DefaultSyncCallback`, que proporciona implementaciones vacías de forma predeterminada para todos ellas.

onSuccess

La devolución de llamada `onSuccess()` se activa cuando se descarga correctamente un conjunto de datos desde el almacén de sincronización.

```
@Override
public void onSuccess(Dataset dataset, List<Record> newRecords) {
}
```

onFailure

Se llama a onFailure() si se produce una excepción durante la sincronización.

```
@Override
public void onFailure(DataStorageException dse) {
}
```

onConflict

Pueden producirse conflictos si la misma clave se ha modificado en el almacén local y en el almacén de sincronización. El método `onConflict()` se encarga de la resolución de conflictos. Si no implementa este método, el cliente de Amazon Cognito Sync utiliza de forma predeterminada el cambio más reciente.

```
@Override
```

```
public boolean onConflict(Dataset dataset, final List<SyncConflict> conflicts) {
    List<Record> resolvedRecords = new ArrayList<Record>();
    for (SyncConflict conflict : conflicts) {
        /* resolved by taking remote records */
        resolvedRecords.add(conflict.resolveWithRemoteRecord());

        /* alternately take the local records */
        // resolvedRecords.add(conflict.resolveWithLocalRecord());

        /* or customer logic, say concatenate strings */
        // String newValue = conflict.getRemoteRecord().getValue()
        //     + conflict.getLocalRecord().getValue();
        // resolvedRecords.add(conflict.resolveWithValue(newValue);
    }
    dataset.resolve(resolvedRecords);

    // return true so that synchronize() is retried after conflicts are resolved
    return true;
}
```

onDatasetDeleted

Cuando se elimina un conjunto de datos, el cliente de Amazon Cognito utiliza la interfaz `SyncCallback` para confirmar si la copia del conjunto de datos que está almacenada en la memoria caché local se tiene que eliminar también. Implemente el método `onDatasetDeleted()` para decirle al SDK de cliente qué debe hacer con los datos locales.

```
@Override
public boolean onDatasetDeleted(Dataset dataset, String datasetName) {
    // return true to delete the local copy of the dataset
    return true;
}
```

onDatasetMerged

Cuando se vinculan dos identidades que anteriormente no estaban conectadas, todos sus conjuntos de datos se combinan. Las aplicaciones reciben una notificación de la combinación a través del método `onDatasetsMerged()`:

```
@Override
public boolean onDatasetsMerged(Dataset dataset, List<String> datasetNames) {
    // return false to handle Dataset merge outside the synchronization callback
```

```
    return false;
}
```

iOS - Objective-C

Notificaciones de sincronización

El cliente de Amazon Cognito generará una serie de eventos `NSNotification` durante una llamada de sincronización. Puede registrar la supervisión de dichas notificaciones mediante el `NSNotificationCenter` estándar:

```
[NSNotificationCenter defaultCenter]
  addObserver:self
  selector:@selector(myNotificationHandler:)
  name:NOTIFICATION_TYPE
  object:nil];
```

Amazon Cognito es compatible con cinco tipos de notificaciones, que se indican a continuación.

AWSCognitoDidStartSynchronizeNotification

Se llamada cuando se inicia una operación se sincronización. El objeto `userInfo` contendrá el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado.

AWSCognitoDidEndSynchronizeNotification

Se llama cuando finaliza una operación de sincronización (correctamente o no). El objeto `userInfo` contendrá el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado.

AWSCognitoDidFailToSynchronizeNotification

Se llama cuando una operación de sincronización falla. El objeto `userInfo` contendrá el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado, y el error de clave que contiene el error que ha provocado el error.

AWSCognitoDidChangeRemoteValueNotification

Se llama cuando los cambios locales se envían de forma correcta a Amazon Cognito. El objeto `userInfo` contiene el conjunto de datos de la clave, que corresponde al nombre del conjunto de

datos sincronizado, y las claves que contienen un NSArray de las claves de registro que se han transmitido.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Se llama cuando un valor local cambia debido a una operación de sincronización. El objeto `userInfo` contiene el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado, y las claves que contienen un NSArray de las claves de registro que han cambiado.

Gestor de resolución de conflictos

Durante una operación de sincronización, pueden producirse conflictos si se ha modificado la misma clave en el almacén local y en el almacén de sincronización. Si no ha definido un gestor de resolución de conflictos, Amazon Cognito elige de forma predeterminada la actualización más reciente.

La implementación y la asignación de un gestor AWSCognitoRecordConflictHandler le permite modificar la resolución de conflictos predeterminada. El conflicto del parámetro de entrada AWSCognitoConflict contiene un objeto AWSCognitoRecord para los datos almacenados en la memoria caché local y para el registro de conflicto en el almacén de sincronización. Con AWSCognitoConflict puede solucionar el conflicto con el registro local: [conflict resolveWithLocalRecord], el registro remoto: [conflict resolveWithRemoteRecord] o un valor nuevo: [conflict resolveWithValue: value]. La devolución de un valor nulo a partir de este método, impide que prosiga la sincronización y los conflictos volverán a producirse la siguiente vez que se inicie el proceso de sincronización.

Puede configurar el gestor de resolución de conflictos en el nivel de cliente:

```
client.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
 AWSCognitoConflict *conflict) {
    // always choose local changes
    return [conflict resolveWithLocalRecord];
};
```

O en el nivel de conjunto de datos:

```
dataset.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
 AWSCognitoConflict *conflict) {
```

```
    // override and always choose remote changes
    return [conflict resolveWithRemoteRecord];
};
```

Gestor de supresión de conjuntos de datos

Cuando se elimina un conjunto de datos, el cliente de Amazon Cognito utiliza el `AWSCognitoDatasetDeletedHandler` para confirmar si la copia del conjunto de datos que está almacenada en la memoria caché local se tiene que eliminar también. Si no hay un `AWSCognitoDatasetDeletedHandler` implementado, los datos locales se purgarán automáticamente. Implemente un `AWSCognitoDatasetDeletedHandler` si desea conservar una copia de los datos locales antes de borrar o si desea conservar los datos locales.

Puede configurar el gestor de supresión del conjunto de datos en el nivel de cliente:

```
client.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // make a backup of the data if you choose
    ...
    // delete the local data (default behavior)
    return YES;
};
```

O en el nivel de conjunto de datos:

```
dataset.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // override default and keep the local data
    return NO;
};
```

Gestor de combinación del conjuntos de datos

Cuando se vinculan dos identidades que anteriormente no estaban conectadas, todos sus conjuntos de datos se combinan. Las aplicaciones reciben una notificación de la combinación mediante `DatasetMergeHandler`. El gestor recibirá el nombre del conjunto de datos raíz, así como una gama de nombres de conjuntos de datos que están marcados como combinaciones del conjunto de datos raíz.

Si el `DatasetMergeHandler` no se implementa, no se tendrán en cuenta estos conjuntos de datos, pero se seguirá usando espacio hasta un máximo de 20 conjuntos de datos en total.

Puede configurar el gestor de combinación de conjuntos de datos en el nivel de cliente:

```
client.datasetMergedHandler = ^(NSString *datasetName, NSArray *datasets) {
    // Blindly delete the datasets
    for (NSString *name in datasets) {
        AWSCognitoDataset *merged = [[AWSCognito defaultCognito]
 openOrCreateDataset:name];
       [merged clear];
       [merged synchronize];
    }
};
```

O en el nivel de conjunto de datos:

```
dataset.datasetMergedHandler = ^(NSString *datasetName, NSArray *datasets) {
    // Blindly delete the datasets
    for (NSString *name in datasets) {
        AWSCognitoDataset *merged = [[AWSCognito defaultCognito]
 openOrCreateDataset:name];
        // do something with the data if it differs from existing dataset
        ...
        // now delete it
        [merged clear];
        [merged synchronize];
    }
};
```

iOS - Swift

Notificaciones de sincronización

El cliente de Amazon Cognito generará una serie de eventos `NSNotification` durante una llamada de sincronización. Puede registrar la supervisión de dichas notificaciones mediante el `NSNotificationCenter` estándar:

```
NSNotificationCenter.defaultCenter().addObserver(observer: self,
   selector: "myNotificationHandler",
   name:NOTIFICATION_TYPE,
   object:nil)
```

Amazon Cognito es compatible con cinco tipos de notificaciones, que se indican a continuación.

AWSCognitoDidStartSynchronizeNotification

Se llamada cuando se inicia una operación se sincronización. El objeto `userInfo` contendrá el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado.

AWSCognitoDidEndSynchronizeNotification

Se llama cuando finaliza una operación de sincronización (correctamente o no). El objeto `userInfo` contendrá el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado.

AWSCognitoDidFailToSynchronizeNotification

Se llama cuando una operación de sincronización falla. El objeto `userInfo` contendrá el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado, y el error de clave que contiene el error que ha provocado el error.

AWSCognitoDidChangeRemoteValueNotification

Se llama cuando los cambios locales se envían de forma correcta a Amazon Cognito. El objeto `userInfo` contiene el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado, y las claves que contienen un NSArray de las claves de registro que se han transmitido.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Se llama cuando un valor local cambia debido a una operación de sincronización. El objeto `userInfo` contiene el conjunto de datos de la clave, que corresponde al nombre del conjunto de datos sincronizado, y las claves que contienen un NSArray de las claves de registro que han cambiado.

Gestor de resolución de conflictos

Durante una operación de sincronización, pueden producirse conflictos si se ha modificado la misma clave en el almacén local y en el almacén de sincronización. Si no ha definido un gestor de resolución de conflictos, Amazon Cognito elige de forma predeterminada la actualización más reciente.

La implementación y la asignación de un gestor `AWSCognitoRecordConflictHandler` le permite modificar la resolución de conflictos predeterminada. El conflicto del parámetro de entrada

`AWSCognitoConflict` contiene un objeto `AWSCognitoRecord` para los datos almacenados en la memoria caché local y para el registro de conflicto en el almacén de sincronización. Con `AWSCognitoConflict` puede solucionar el conflicto con el registro local: [conflict resolveWithLocalRecord], el registro remoto: [conflict resolveWithRemoteRecord] o un valor nuevo: [conflict resolveWithValue: value]. La devolución de un valor nulo a partir de este método, impide que prosiga la sincronización y los conflictos volverán a producirse la siguiente vez que se inicie el proceso de sincronización.

Puede configurar el gestor de resolución de conflictos en el nivel de cliente:

```
client.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
 AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
}
```

O en el nivel de conjunto de datos:

```
dataset.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
 AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
}
```

Gestor de supresión de conjuntos de datos

Cuando se elimina un conjunto de datos, el cliente de Amazon Cognito utiliza el `AWSCognitoDatasetDeletedHandler` para confirmar si la copia del conjunto de datos que está almacenada en la memoria caché local se tiene que eliminar también. Si no hay un `AWSCognitoDatasetDeletedHandler` implementado, los datos locales se purgarán automáticamente. Implemente un `AWSCognitoDatasetDeletedHandler` si desea conservar una copia de los datos locales antes de borrar o si desea conservar los datos locales.

Puede configurar el gestor de supresión del conjunto de datos en el nivel de cliente:

```
client.datasetDeletedHandler = {
      (datasetName: String!) -> Bool in
      // make a backup of the data if you choose
      ...
      // delete the local data (default behaviour)
```

```
      return true
}
```

O en el nivel de conjunto de datos:

```
dataset.datasetDeletedHandler = {
      (datasetName: String!) -> Bool in
      // make a backup of the data if you choose
      ...
      // delete the local data (default behaviour)
      return true
}
```

Gestor de combinación del conjuntos de datos

Cuando se vinculan dos identidades que anteriormente no estaban conectadas, todos sus conjuntos de datos se combinan. Las aplicaciones reciben una notificación de la combinación mediante `DatasetMergeHandler`. El gestor recibirá el nombre del conjunto de datos raíz, así como una gama de nombres de conjuntos de datos que están marcados como combinaciones del conjunto de datos raíz.

Si el `DatasetMergeHandler` no se implementa, no se tendrán en cuenta estos conjuntos de datos, pero se seguirá usando espacio hasta un máximo de 20 conjuntos de datos en total.

Puede configurar el gestor de combinación de conjuntos de datos en el nivel de cliente:

```
client.datasetMergedHandler = {
    (datasetName: String!, datasets: [AnyObject]!) -> Void in
    for nameObject in datasets {
        if let name = nameObject as? String {
            let merged = AWSCognito.defaultCognito().openOrCreateDataset(name)
            merged.clear()
            merged.synchronize()
        }
    }
}
```

O en el nivel de conjunto de datos:

```
dataset.datasetMergedHandler = {
    (datasetName: String!, datasets: [AnyObject]!) -> Void in
```

```
    for nameObject in datasets {
        if let name = nameObject as? String {
            let merged = AWSCognito.defaultCognito().openOrCreateDataset(name)
            // do something with the data if it differs from existing dataset
              ...
              // now delete it
            merged.clear()
            merged.synchronize()
        }
    }
}
```

JavaScript

Devoluciones de llamadas de sincronización

Cuando ejecute synchronize() en un conjunto de datos, tiene la posibilidad de especificar devoluciones de llamadas para abordar cada uno de los estados siguientes:

```
dataset.synchronize({

   onSuccess: function(dataset, newRecords) {
      //...
   },

   onFailure: function(err) {
      //...
   },

   onConflict: function(dataset, conflicts, callback) {
      //...
   },

   onDatasetDeleted: function(dataset, datasetName, callback) {
      //...
   },

  onDatasetMerged: function(dataset, datasetNames, callback) {
      //...
  }

});
```

onSuccess()

La devolución de llamada `onSuccess()` se activa cuando se actualiza correctamente un conjunto de datos desde el almacén de sincronización. Si no define una devolución de llamada, la sincronización se logrará silenciosamente.

```
onSuccess: function(dataset, newRecords) {
   console.log('Successfully synchronized ' + newRecords.length + ' new records.');
}
```

onFailure()

Se llama a `onFailure()` si se produce una excepción durante la sincronización. Si no define una devolución de llamada, la sincronización fallará silenciosamente.

```
onFailure: function(err) {
   console.log('Synchronization failed.');
   console.log(err);
}
```

onConflict()

Pueden producirse conflictos si la misma clave se ha modificado en el almacén local y en el almacén de sincronización. El método `onConflict()` se encarga de la resolución de conflictos. Si no implementa este método, la sincronización se anulará cuando exista un conflicto.

```
onConflict: function(dataset, conflicts, callback) {

   var resolved = [];

   for (var i=0; i<conflicts.length; i++) {

      // Take remote version.
      resolved.push(conflicts[i].resolveWithRemoteRecord());

      // Or... take local version.
      // resolved.push(conflicts[i].resolveWithLocalRecord());

      // Or... use custom logic.
      // var newValue = conflicts[i].getRemoteRecord().getValue() +
 conflicts[i].getLocalRecord().getValue();
```

```
      // resolved.push(conflicts[i].resovleWithValue(newValue);

   }

   dataset.resolve(resolved, function() {
      return callback(true);
   });

   // Or... callback false to stop the synchronization process.
   // return callback(false);

}
```

onDatasetDeleted()

Cuando se elimina un conjunto de datos, el cliente de Amazon Cognito utiliza la devolución de llamada `onDatasetDeleted()` para decidir si la copia del conjunto de datos que está almacenada en la memoria caché local se tiene que eliminar también. De forma predeterminada, no se eliminará el conjunto de datos.

```
onDatasetDeleted: function(dataset, datasetName, callback) {

   // Return true to delete the local copy of the dataset.
   // Return false to handle deleted datasets outside the synchronization callback.

   return callback(true);

}
```

onDatasetMerged()

Cuando se vinculan dos identidades que anteriormente no estaban conectadas, todos sus conjuntos de datos se combinan. Las aplicaciones reciben una notificación de la combinación mediante la devolución de llamada `onDatasetsMerged()`.

```
onDatasetMerged: function(dataset, datasetNames, callback) {

   // Return true to continue the synchronization process.
   // Return false to handle dataset merges outside the synchronization callback.

   return callback(false);
```

```
}
```

Unity

Después de abrir o crear un conjunto de datos, puede configurar diferentes devoluciones de llamadas al conjunto de datos, que se activarán cuando use el método Synchronize. A continuación, indicamos la forma de registrar las devoluciones de llamadas en ellos:

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Tenga en cuenta que `SyncSuccess` y `SyncFailure` usan **+=** en vez de **=** para que les pueda suscribir más de una devolución de llamada.

OnSyncSuccess

La devolución de llamada `OnSyncSuccess` se activa cuando se actualiza correctamente un conjunto de datos desde la nube. Si no define una devolución de llamada, la sincronización se logrará silenciosamente.

```
private void HandleSyncSuccess(object sender, SyncSuccessEvent e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

Se llama a `OnSyncFailure` si se produce una excepción durante la sincronización. Si no define una devolución de llamada, la sincronización fallará silenciosamente.

```
private void HandleSyncFailure(object sender, SyncFailureEvent e)
{
    Dataset dataset = sender as Dataset;
    if (dataset.Metadata != null) {
        Debug.Log("Sync failed for dataset : " + dataset.Metadata.DatasetName);
    } else {
        Debug.Log("Sync failed");
    }
```

```
    // Handle the error
    Debug.LogException(e.Exception);
}
```

OnSyncConflict

Pueden producirse conflictos si la misma clave se ha modificado en el almacén local y en el almacén de sincronización. La devolución de llamada `OnSyncConflict` se encarga de la resolución de conflictos. Si no implementa este método, la sincronización se anulará cuando exista un conflicto.

```
private bool HandleSyncConflict(Dataset dataset, List < SyncConflict > conflicts)
{
  if (dataset.Metadata != null) {
    Debug.LogWarning("Sync conflict " + dataset.Metadata.DatasetName);
  } else {
    Debug.LogWarning("Sync conflict");
  }
  List < Amazon.CognitoSync.SyncManager.Record > resolvedRecords = new List <
 Amazon.CognitoSync.SyncManager.Record > ();
  foreach(SyncConflict conflictRecord in conflicts) {
    // SyncManager provides the following default conflict resolution methods:
    //      ResolveWithRemoteRecord - overwrites the local with remote records
    //      ResolveWithLocalRecord - overwrites the remote with local records
    //      ResolveWithValue - to implement your own logic
    resolvedRecords.Add(conflictRecord.ResolveWithRemoteRecord());
  }
  // resolves the conflicts in local storage
  dataset.Resolve(resolvedRecords);
  // on return true the synchronize operation continues where it left,
  //      returning false cancels the synchronize operation
  return true;
}
```

OnDatasetDeleted

Cuando se elimina un conjunto de datos, el cliente de Amazon Cognito utiliza la devolución de llamada `OnDatasetDeleted` para decidir si la copia del conjunto de datos que está almacenada en la memoria caché local se tiene que eliminar también. De forma predeterminada, no se eliminará el conjunto de datos.

```
private bool HandleDatasetDeleted(Dataset dataset)
  {
```

```
      Debug.Log(dataset.Metadata.DatasetName + " Dataset has been deleted");
      // Do clean up if necessary
      // returning true informs the corresponding dataset can be purged in the local
 storage and return false retains the local dataset
      return true;
  }
```

OnDatasetMerged

Cuando se vinculan dos identidades que anteriormente no estaban conectadas, todos sus conjuntos de datos se combinan. Las aplicaciones reciben una notificación de la combinación mediante la devolución de llamada `OnDatasetsMerged`.

```
public bool HandleDatasetMerged(Dataset localDataset, List<string> mergedDatasetNames)
{
    foreach (string name in mergedDatasetNames)
    {
        Dataset mergedDataset = syncManager.OpenOrCreateDataset(name);
        //Lambda function to delete the dataset after fetching it
        EventHandler<SyncSuccessEvent> lambda;
        lambda = (object sender, SyncSuccessEvent e) => {
            ICollection<string> existingValues = localDataset.GetAll().Values;
            ICollection<string> newValues = mergedDataset.GetAll().Values;

            //Implement your merge logic here

            mergedDataset.Delete(); //Delete the dataset locally
            mergedDataset.OnSyncSuccess -= lambda; //We don't want this callback to be
 fired again
            mergedDataset.OnSyncSuccess += (object s2, SyncSuccessEvent e2) => {
                localDataset.Synchronize(); //Continue the sync operation that was
 interrupted by the merge
            };
            mergedDataset.Synchronize(); //Synchronize it as deleted, failing to do so
 will leave us in an inconsistent state
        };
        mergedDataset.OnSyncSuccess += lambda;
        mergedDataset.Synchronize(); //Asnchronously fetch the dataset
    }

    // returning true allows the Synchronize to continue and false stops it
    return false;
}
```

Xamarin

Después de abrir o crear un conjunto de datos, puede configurar diferentes devoluciones de llamadas al conjunto de datos, que se activarán cuando use el método Synchronize. A continuación, indicamos la forma de registrar las devoluciones de llamadas en ellos:

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Tenga en cuenta que `SyncSuccess` y `SyncFailure` usan **+=** en vez de **=** para que les pueda suscribir más de una devolución de llamada.

OnSyncSuccess

La devolución de llamada `OnSyncSuccess` se activa cuando se actualiza correctamente un conjunto de datos desde la nube. Si no define una devolución de llamada, la sincronización se logrará silenciosamente.

```
private void HandleSyncSuccess(object sender, SyncSuccessEventArgs e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

Se llama a `OnSyncFailure` si se produce una excepción durante la sincronización. Si no define una devolución de llamada, la sincronización fallará silenciosamente.

```
private void HandleSyncFailure(object sender, SyncFailureEventArgs e)
{
    Dataset dataset = sender as Dataset;
    if (dataset.Metadata != null) {
        Console.WriteLine("Sync failed for dataset : " + dataset.Metadata.DatasetName);
    } else {
        Console.WriteLine("Sync failed");
    }
}
```

OnSyncConflict

Pueden producirse conflictos si la misma clave se ha modificado en el almacén local y en el almacén de sincronización. La devolución de llamada `OnSyncConflict` se encarga de la resolución de conflictos. Si no implementa este método, la sincronización se anulará cuando exista un conflicto.

```
private bool HandleSyncConflict(Dataset dataset, List < SyncConflict > conflicts)
{
  if (dataset.Metadata != null) {
    Console.WriteLine("Sync conflict " + dataset.Metadata.DatasetName);
  } else {
    Console.WriteLine("Sync conflict");
  }
  List < Amazon.CognitoSync.SyncManager.Record > resolvedRecords = new List <
 Amazon.CognitoSync.SyncManager.Record > ();
  foreach(SyncConflict conflictRecord in conflicts) {
    // SyncManager provides the following default conflict resolution methods:
    //      ResolveWithRemoteRecord - overwrites the local with remote records
    //      ResolveWithLocalRecord - overwrites the remote with local records
    //      ResolveWithValue - to implement your own logic
    resolvedRecords.Add(conflictRecord.ResolveWithRemoteRecord());
  }
  // resolves the conflicts in local storage
  dataset.Resolve(resolvedRecords);
  // on return true the synchronize operation continues where it left,
  //      returning false cancels the synchronize operation
  return true;
}
```

OnDatasetDeleted

Cuando se elimina un conjunto de datos, el cliente de Amazon Cognito utiliza la devolución de llamada `OnDatasetDeleted` para decidir si la copia del conjunto de datos que está almacenada en la memoria caché local se tiene que eliminar también. De forma predeterminada, no se eliminará el conjunto de datos.

```
private bool HandleDatasetDeleted(Dataset dataset)
{
    Console.WriteLine(dataset.Metadata.DatasetName + " Dataset has been deleted");
    // Do clean up if necessary
    // returning true informs the corresponding dataset can be purged in the local
 storage and return false retains the local dataset
```

```
    return true;
}
```

OnDatasetMerged

Cuando se vinculan dos identidades que anteriormente no estaban conectadas, todos sus conjuntos de datos se combinan. Las aplicaciones reciben una notificación de la combinación mediante la devolución de llamada `OnDatasetsMerged`.

```
public bool HandleDatasetMerged(Dataset localDataset, List<string> mergedDatasetNames)
{
    foreach (string name in mergedDatasetNames)
    {
        Dataset mergedDataset = syncManager.OpenOrCreateDataset(name);

            //Implement your merge logic here

        mergedDataset.OnSyncSuccess += lambda;
        mergedDataset.SynchronizeAsync(); //Asnchronously fetch the dataset
    }

    // returning true allows the Synchronize to continue and false stops it
    return false;
}
```

Sincronización mediante inserción

⚠ Si es la primera vez que usa Amazon Cognito Sync, utilice AWS AppSync. Como Amazon Cognito Sync, AWS AppSync es un servicio destinado a sincronizar los datos de las aplicaciones entre dispositivos.
Con este, se pueden sincronizar los datos de usuarios, como las preferencias de aplicación o el estado del juego. También amplía estas funcionalidades, ya que permite que varios usuarios se sincronicen y colaboren en tiempo real sobre los datos compartidos.

Amazon Cognito realiza seguimiento de forma automática de la asociación entre la identidad y los dispositivos. El uso de la sincronización mediante inserción puede garantizar que todas las instancias de una determinada identidad reciban una notificación cuando cambien los datos de identidad. La

sincronización por inserción garantiza que, siempre que los datos del almacén de sincronización cambien para una identidad determinada, todos los dispositivos asociados recibirán una notificación de inserción silenciosa que informe del cambio.

> **Note**
>
> La sincronización mediante inserción no es compatible con JavaScript, Unity o Xamarin.

Para poder utilizar la sincronización mediante inserción, primero debe configurar su cuenta para que se sincronice mediante inserción en la consola de Amazon Cognito.

Creación de una aplicación de Amazon Simple Notification Service (Amazon SNS)

Cree y configure una aplicación de Amazon SNS para sus plataformas compatibles, tal como se describe en la Guía para desarrolladores de SNS.

Activación de la sincronización mediante inserción en la consola de Amazon Cognito

Puede habilitar la sincronización mediante inserción mediante la consola de Amazon Cognito. En la página de inicio de la consola:

1. Haga clic en el nombre del grupo de identidades para el que desea habilitar la sincronización por inserción. Se mostrará la página Dashboard (Panel) de su grupo de identidades.
2. En la esquina superior derecha de la página Dashboard (Panel), haga clic en Manage Identity Pools (Administrar grupos de identidades). Se visualizará la página Federated Identities (Identidades federadas).
3. Desplácese hacia abajo y haga clic en Push synchronization (Insertar sincronización) para expandirlo.
4. En el menú desplegable Service role (Rol de servicio), seleccione el rol de IAM que concede a Cognito permiso para enviar una notificación de SNS. Haga clic en Create role (Crear rol) para crear o modificar los roles asociados a su grupo de identidades en la consola de IAM de AWS.
5. Seleccione una aplicación de plataforma y, a continuación, haga clic en Save Changes (Guardar cambios).
6. Autorice a SNS acceso a su aplicación

En la consola de AWS Identity and Access Management, configure los roles de IAM para que tengan pleno acceso de Amazon SNS o cree un rol nuevo que cuente con acceso completo de Amazon SNS. En el ejemplo siguiente de política de confianza de rol se concede a Amazon Cognito Sync una capacidad limitada para que adopte un rol de IAM. Amazon Cognito Sync solo puede adoptar el rol cuando lo hace en nombre del grupo de identidades en la condición `aws:SourceArn` y en la cuenta en la condición `aws:SourceAccount`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-sync.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "AWS:SourceArn": "arn:aws:cognito-identity:us-
east-1:123456789012:identitypool/us-east-1:177a950c-2c08-43f0-9983-28727EXAMPLE"
                }
            }
        }
    ]
}
```

Para obtener más información acerca de los roles de IAM, consulte la sección Roles (delegación y federación).

Uso de la sincronización mediante inserción en su aplicación: Android

Su aplicación deberá importar los servicios de Google Play. Puede descargar la versión más reciente del SDK de Google Play a través del administrador de SDK para Android. Consulte la documentación de Android que se encuentra en Android Implementation para registrar la aplicación y recibir un ID de registro de GCM. Una vez que tenga el ID de registro, deberá registrar el dispositivo con Amazon Cognito, tal como se muestra en el fragmento siguiente:

```
String registrationId = "MY_GCM_REGISTRATION_ID";
```

```
try {
    client.registerDevice("GCM", registrationId);
} catch (RegistrationFailedException rfe) {
    Log.e(TAG, "Failed to register device for silent sync", rfe);
} catch (AmazonClientException ace) {
    Log.e(TAG, "An unknown error caused registration for silent sync to fail", ace);
}
```

Ahora ya puede suscribir un dispositivo para recibir actualizaciones de un conjunto de datos determinado:

```
Dataset trackedDataset = client.openOrCreateDataset("myDataset");
if (client.isDeviceRegistered()) {
    try {
        trackedDataset.subscribe();
    } catch (SubscribeFailedException sfe) {
        Log.e(TAG, "Failed to subscribe to datasets", sfe);
    } catch (AmazonClientException ace) {
        Log.e(TAG, "An unknown error caused the subscription to fail", ace);
    }
}
```

Para dejar de recibir notificaciones de inserción desde un conjunto de datos, solo tiene que llamar al método unsubscribe. Para suscribirse a todos los conjuntos de datos (o a un subconjunto concreto) del objeto `CognitoSyncManager`, utilice `subscribeAll()`:

```
if (client.isDeviceRegistered()) {
    try {
        client.subscribeAll();
    } catch (SubscribeFailedException sfe) {
        Log.e(TAG, "Failed to subscribe to datasets", sfe);
    } catch (AmazonClientException ace) {
        Log.e(TAG, "An unknown error caused the subscription to fail", ace);
    }
}
```

En la implementación del objeto Android BroadcastReceiver, puede comprobar la última versión del conjunto de datos modificado y decidir si su aplicación se debe volver a sincronizar:

```
@Override
public void onReceive(Context context, Intent intent) {
```

```
    PushSyncUpdate update = client.getPushSyncUpdate(intent);

    // The update has the source (cognito-sync here), identityId of the
    // user, identityPoolId in question, the non-local sync count of the
    // data set and the name of the dataset. All are accessible through
    // relevant getters.

    String source = update.getSource();
    String identityPoolId = update.getIdentityPoolId();
    String identityId = update.getIdentityId();
    String datasetName = update.getDatasetName;
    long syncCount = update.getSyncCount;

    Dataset dataset = client.openOrCreateDataset(datasetName);

    // need to access last sync count. If sync count is less or equal to
    // last sync count of the dataset, no sync is required.

    long lastSyncCount = dataset.getLastSyncCount();
    if (lastSyncCount < syncCount) {
        dataset.synchronize(new SyncCallback() {
            // ...
        });
    }

}
```

Las claves siguientes están disponibles en la carga útil de notificaciones de inserción:

- `source`: sincronización de Cognito. Esta clave puede servir de factor de diferenciación entre las notificaciones.
- `identityPoolId`: ID del grupo de identidades. Esta clave se puede utilizar para la validación o para obtener información adicional, aunque desde el punto de vista del receptor no sea integral.
- `identityId`: ID de identidad dentro del grupo.
- `datasetName`: nombre del conjunto de datos que se ha actualizado. Esta clave está disponible para la llamada openOrCreateDataset.
- `syncCount`: número de sincronizaciones para el conjunto de datos remoto. Puede utilizar esta clave como forma de asegurarse de que el conjunto de datos local esté obsoleto y que la sincronización de entrada sea nueva.

Uso de la sincronización mediante inserción en su aplicación: iOS - Objective-C

Para obtener un token de dispositivo para su aplicación, consulte la documentación de Apple en el registro para recibir notificaciones remotas. Una vez que haya recibido el token de dispositivo como objeto NSData desde APN, tendrá que registrar el dispositivo en Amazon Cognito con el método `registerDevice:` del cliente de sincronización, tal como se muestra a continuación:

```
AWSCognito *syncClient = [AWSCognito defaultCognito];
    [[syncClient registerDevice: devToken] continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to registerDevice: %@", task.error);
        } else {
            NSLog(@"Successfully registered device with id: %@", task.result);
        }
        return nil;
      }
    ];
```

En el modo de depuración, el dispositivo se registra en el entorno de pruebas de APN; mientras que en el modo de lanzamiento, se registra en los APN. Para recibir actualizaciones de un conjunto de datos determinado, aplique el método `subscribe`:

```
[[[syncClient openOrCreateDataset:@"MyDataset"] subscribe]
 continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to subscribe to dataset: %@", task.error);
        } else {
            NSLog(@"Successfully subscribed to dataset: %@", task.result);
        }
        return nil;
      }
    ];
```

Para dejar de recibir notificaciones de inserción desde un conjunto de datos, solo tiene que llamar al método `unsubscribe`:

```
[[[syncClient openOrCreateDataset:@”MyDataset”] unsubscribe]
 continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to unsubscribe from dataset: %@", task.error);
```

```
        } else {
            NSLog(@"Successfully unsubscribed from dataset: %@", task.result);
        }
        return nil;
      }
    ];
```

Para suscribirse a todos los conjuntos de datos del objeto `AWSCognito`, llame a `subscribeAll`:

```
[[syncClient subscribeAll] continueWithBlock:^id(AWSTask *task) {
        if(task.error){
            NSLog(@"Unable to subscribe to all datasets: %@", task.error);
        } else {
            NSLog(@"Successfully subscribed to all datasets: %@", task.result);
        }
        return nil;
      }
    ];
```

Antes de llamar a `subscribeAll`, sincronice todos los conjuntos de datos como mínimo una vez, para que dichos conjuntos existan en el servidor.

Para responder a las notificaciones de inserción, debe implementar el método `didReceiveRemoteNotification` en el delgado de la aplicación:

```
- (void)application:(UIApplication *)application didReceiveRemoteNotification:
(NSDictionary *)userInfo
    {
        [[NSNotificationCenter defaultCenter]
 postNotificationName:@"CognitoPushNotification" object:userInfo];
    }
```

Si publica una notificación mediante el controlador de notificaciones, puede responder a la notificación en cualquier punto de la aplicación donde tenga un control sobre el conjunto de datos. Si se suscribe a la notificación de esta forma...

```
[[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(didReceivePushSync:)
    name: :@"CognitoPushNotification" object:nil];
```

... puede actuar sobre la notificación de esta forma:

```
- (void)didReceivePushSync:(NSNotification*)notification
    {
        NSDictionary * data = [(NSDictionary *)[notification object]
 objectForKey:@"data"];
        NSString * identityId = [data objectForKey:@"identityId"];
        NSString * datasetName = [data objectForKey:@"datasetName"];
        if([self.dataset.name isEqualToString:datasetName] && [self.identityId
 isEqualToString:identityId]){
            [[self.dataset synchronize] continueWithBlock:^id(AWSTask *task) {
                if(!task.error){
                    NSLog(@"Successfully synced dataset");
                }
                return nil;
            }];
        }
    }
```

Las claves siguientes están disponibles en la carga útil de notificaciones de inserción:

- `source`: sincronización de Cognito. Esta clave puede servir de factor de diferenciación entre las notificaciones.
- `identityPoolId`: ID del grupo de identidades. Esta clave se puede utilizar para la validación o para obtener información adicional, aunque desde el punto de vista del receptor no sea integral.
- `identityId`: ID de identidad dentro del grupo.
- `datasetName`: nombre del conjunto de datos que se ha actualizado. Esta clave está disponible para la llamada `openOrCreateDataset`.
- `syncCount`: número de sincronizaciones para el conjunto de datos remoto. Puede utilizar esta clave como forma de asegurarse de que el conjunto de datos local esté obsoleto y que la sincronización de entrada sea nueva.

Uso de la sincronización mediante inserción en su aplicación: iOS - Swift

Para obtener un token de dispositivo para su aplicación, consulte la documentación de Apple en el registro para recibir notificaciones remotas. Una vez que haya recibido el token de dispositivo como objeto NSData desde APN, tendrá que registrar el dispositivo en Amazon Cognito con el método registerDevice: del cliente de sincronización, tal como se muestra a continuación:

```
let syncClient = AWSCognito.default()
```

```
syncClient.registerDevice(devToken).continueWith(block: { (task: AWSTask!) ->
 AnyObject! in
    if (task.error != nil) {
        print("Unable to register device: " + task.error.localizedDescription)

    } else {
        print("Successfully registered device with id: \(task.result)")
    }
    return task
})
```

En el modo de depuración, el dispositivo se registra en el entorno de pruebas de APN; mientras que en el modo de lanzamiento, se registra en los APN. Para recibir actualizaciones de un conjunto de datos determinado, aplique el método `subscribe`:

```
syncClient.openOrCreateDataset("MyDataset").subscribe().continueWith(block: { (task:
 AWSTask!) -> AnyObject! in
  if (task.error != nil) {
      print("Unable to subscribe to dataset: " + task.error.localizedDescription)

  } else {
      print("Successfully subscribed to dataset: \(task.result)")
  }
  return task
})
```

Para dejar de recibir notificaciones de inserción desde un conjunto de datos, llame al método `unsubscribe`:

```
syncClient.openOrCreateDataset("MyDataset").unsubscribe().continueWith(block: { (task:
 AWSTask!) -> AnyObject! in
  if (task.error != nil) {
      print("Unable to unsubscribe to dataset: " + task.error.localizedDescription)

  } else {
      print("Successfully unsubscribed to dataset: \(task.result)")
  }
  return task
})
```

Para suscribirse a todos los conjuntos de datos del objeto `AWSCognito`, llame a `subscribeAll`:

```
syncClient.openOrCreateDataset("MyDataset").subscribeAll().continueWith(block: { (task:
 AWSTask!) -> AnyObject! in
  if (task.error != nil) {
      print("Unable to subscribe to all datasets: " + task.error.localizedDescription)

  } else {
      print("Successfully subscribed to all datasets: \(task.result)")
  }
  return task
})
```

Antes de llamar a `subscribeAll`, sincronice todos los conjuntos de datos como mínimo una vez, para que dichos conjuntos existan en el servidor.

Para responder a las notificaciones de inserción, debe implementar el método `didReceiveRemoteNotification` en el delgado de la aplicación:

```
func application(application: UIApplication, didReceiveRemoteNotification userInfo:
 [NSObject : AnyObject],
  fetchCompletionHandler completionHandler: (UIBackgroundFetchResult) -> Void) {

 NSNotificationCenter.defaultCenter().postNotificationName("CognitoPushNotification",
 object: userInfo)
})
```

Si publica una notificación mediante el controlador de notificaciones, puede responder a la notificación en cualquier punto de la aplicación donde tenga un control sobre el conjunto de datos. Si se suscribe a la notificación de esta forma...

```
NSNotificationCenter.defaultCenter().addObserver(observer:self,
   selector:"didReceivePushSync:",
   name:"CognitoPushNotification",
   object:nil)
```

... puede actuar sobre la notificación de esta forma:

```
func didReceivePushSync(notification: NSNotification) {
    if let data = (notification.object as! [String: AnyObject])["data"] as? [String:
 AnyObject] {
        let identityId = data["identityId"] as! String
        let datasetName = data["datasetName"] as! String
```

```
        if self.dataset.name == datasetName && self.identityId == identityId {
          dataset.synchronize().continueWithBlock {(task) -> AnyObject! in
              if task.error == nil {
                print("Successfully synced dataset")
              }
              return nil
          }
        }
    }
}
```

Las claves siguientes están disponibles en la carga útil de notificaciones de inserción:

- `source`: sincronización de Cognito. Esta clave puede servir de factor de diferenciación entre las notificaciones.
- `identityPoolId`: ID del grupo de identidades. Esta clave se puede utilizar para la validación o para obtener información adicional, aunque desde el punto de vista del receptor no sea integral.
- `identityId`: ID de identidad dentro del grupo.
- `datasetName`: nombre del conjunto de datos que se ha actualizado. Esta clave está disponible para la llamada `openOrCreateDataset`.
- `syncCount`: número de sincronizaciones para el conjunto de datos remoto. Puede utilizar esta clave como forma de asegurarse de que el conjunto de datos local esté obsoleto y que la sincronización de entrada sea nueva.

Amazon Cognito Streams

> ⚠ Si es la primera vez que usa Amazon Cognito Sync, utilice AWS AppSync. Como Amazon Cognito Sync, AWS AppSync es un servicio destinado a sincronizar los datos de las aplicaciones entre dispositivos.
> Con este, se pueden sincronizar los datos de usuarios, como las preferencias de aplicación o el estado del juego. También amplía estas funcionalidades, ya que permite que varios usuarios se sincronicen y colaboren en tiempo real sobre los datos compartidos.

Amazon Cognito Streams ofrece a los desarrolladores control e información de los datos almacenados en Amazon Cognito. Ahora los desarrolladores pueden configurar un flujo de Kinesis

para recibir eventos cuando los datos se actualicen y se sincronicen. Amazon Cognito puede enviar cada cambio del conjunto de datos a un flujo de Kinesis de su propiedad en tiempo real.

Con Amazon Cognito Streams, puede mover todos los datos de sincronización a Kinesis, que luego pueden transmitirse a una herramienta de almacenamiento de datos, como Amazon Redshift, para analizarlos en mayor profundidad. Para obtener más información sobre Kinesis, consulte Introducción al uso de Amazon Kinesis.

Configuración de los flujos

Puede configurar Amazon Cognito Streams en la consola de Amazon Cognito. Con el fin de habilitar Amazon Cognito Streams en la consola de Amazon Cognito, debe seleccionar el flujo de Kinesis en el que publicar y un rol de IAM que otorgue permiso a Amazon Cognito para poner eventos en el flujo seleccionado.

En la página de inicio de la consola:

1. Haga clic en el nombre del grupo de identidades para el que desee configurar Amazon Cognito Streams. Se mostrará la página Dashboard (Panel) de su grupo de identidades.
2. En la esquina superior derecha de la página Dashboard (Panel), haga clic en Manage Identity Pools (Administrar grupos de identidades). Se visualizará la página Manage Federated Identities.
3. Desplácese hacia abajo y haga clic en Cognito Streams (Secuencias de Cognito) para expandir esta opción.
4. En el menú desplegable Stream name (Nombre de la secuencia), seleccione el nombre de un flujo de Kinesis ya existente. O bien haga clic en Create stream (Crear secuencia) para crear uno, introduciendo un nombre de secuencia y el número de fragmentos. Para obtener información sobre las particiones y ayuda para calcular la cantidad necesaria de particiones para el flujo, consulte la Guía para desarrolladores de Kinesis.
5. En el menú desplegable Publish role (Publicar rol), seleccione el rol de IAM que concede a Amazon Cognito permiso para publicar su flujo Haga clic en Create role (Crear rol) para crear o modificar los roles asociados a su grupo de identidades en la consola de IAM de AWS.
6. En el menú desplegable Stream status (Estado del flujo), seleccione Enabled (Habilitado) para habilitar las actualizaciones de la secuencia. Haga clic en Save Changes (Guardar cambios).

Después de configurar con éxito los flujos de Amazon Cognito, todas las actualizaciones posteriores aplicadas en conjuntos de datos de este grupo de identidades se enviarán al flujo.

Contenido de los flujos

Cada registro enviado al flujo representa una sincronización única. A continuación se muestra un ejemplo de un registro enviado al flujo:

```
{
    "identityPoolId": "Pool Id",
    "identityId": "Identity Id",
    "dataSetName": "Dataset Name",
    "operation": "(replace|remove)",
    "kinesisSyncRecords": [
        {
            "key": "Key",
            "value": "Value",
            "syncCount": 1,
            "lastModifiedDate": 1424801824343,
            "deviceLastModifiedDate": 1424801824343,
            "op": "(replace|remove)"
        },
        ...
    ],
    "lastModifiedDate": 1424801824343,
    "kinesisSyncRecordsURL": "S3Url",
    "payloadType": "(S3Url|Inline)",
    "syncCount": 1
}
```

En el caso de las actualizaciones que superan el tamaño de carga máximo de Kinesis de 1 MB, Amazon Cognito incluye una URL de Amazon S3 prefirmada con el contenido completo de la actualización.

Después de configurar los flujos de Amazon Cognito, si elimina el flujo de Kinesis o cambia el permiso de confianza del rol para que Amazon Cognito Sync ya no lo pueda asumir, desactivará los flujos de Amazon Cognito. Deberá volver a crear el flujo de Kinesis o arreglar el rol y, a continuación, volver a activar el flujo.

Publicación en masa

Una vez que haya configurado los flujos de Amazon Cognito, podrá ejecutar una operación de publicación en masa de los datos existentes en su grupo de identidades. Después de iniciar una operación de publicación en masa, ya sea a través de la consola o directamente a través de la API, Amazon Cognito comenzará la publicación de estos datos en la misma secuencia que recibe las actualizaciones.

Amazon Cognito no garantiza la exclusividad de los datos enviados al flujo en la operación de publicación en masa. Puede recibir la misma actualización como una actualización o como parte de una publicación en masa. Tenga en mente esta posibilidad cuando proceso los registros de su flujo.

Para publicar en masa todos sus flujos, siga los pasos 1 a 6 de la sección de configuración de los flujos y, a continuación, haga clic en Start bulk publish. Tiene un límite de una operación de publicación en masa en curso en cualquier momento y una solicitud de publicación en masa correcta cada 24 horas.

Amazon Cognito Events

> ⚠ Si es la primera vez que usa Amazon Cognito Sync, utilice AWS AppSync. Como Amazon Cognito Sync, AWS AppSync es un servicio destinado a sincronizar los datos de las aplicaciones entre dispositivos.
> Con este, se pueden sincronizar los datos de usuarios, como las preferencias de aplicación o el estado del juego. También amplía estas funcionalidades, ya que permite que varios usuarios se sincronicen y colaboren en tiempo real sobre los datos compartidos.

Con Amazon Cognito Events, puede ejecutar una función de AWS Lambda como respuesta a eventos importantes de Amazon Cognito. Amazon Cognito lanza el evento desencadenador de sincronización cuando se sincroniza un conjunto de datos. Puede utilizar el evento disparador de la sincronización para actuar cuando un usuario actualiza los datos. La función puede evaluar y, de forma opcional, manipular los datos antes de que estos se almacenen en la nube y se sincronicen con los demás dispositivos del usuario. Es una función útil para validar los datos que vienen del dispositivo antes de que se sincronicen con los demás dispositivos del usuario o actualizar otros valores del conjunto de datos en función de los datos de entrada, como la emisión de un premio cuando un jugador logra un nivel nuevo.

Siga estos pasos para configurar una función de Lambda que se ejecuta cada vez que se sincroniza un conjunto de datos de Amazon Cognito.

> **Note**
>
> Cuando utilice Amazon Cognito Events, solo puede utilizar las credenciales obtenidas de Amazon Cognito Identity. Si tiene una función de Lambda asociada, pero llama a

`UpdateRecords` con las credenciales de la cuenta de AWS (credenciales de desarrollador), la función de Lambda no se invocará.

Creación de una función en AWS Lambda

Para integrar Lambda en Amazon Cognito, primero debe crear una función en Lambda. Para ello:

Selección de la función de Lambda en Amazon Cognito

1. Abra la consola de Lambda.
2. Haga clic en Create a Lambda function (Crear una función de Lambda).
3. En la pantalla Select blueprint, busque "cognito-sync-trigger" y selecciónelo.
4. En la pantalla Configure event sources, deje el tipo de fuente de evento establecido en "Cognito Sync Trigger" y seleccione su grupo de identidades. Haga clic en Next (Siguiente).

> **Note**
>
> Al configurar un desencadenador de Amazon Cognito Sync fuera de la consola, debe agregar permisos basados en recursos de Lambda para permitir que Amazon Cognito invoque la función. Puede agregar este permiso desde la consola de Lambda (consulte Uso de políticas basadas en recursos para AWS Lambda) o mediante el Lambda AddPermission.
>
> Ejemplo de política basada en recursos de Lambda
>
> En la siguiente política basada en recursos de AWS Lambda se otorga a Amazon Cognito una capacidad limitada para invocar una función Lambda. Amazon Cognito solo puede invocar la función en nombre del grupo de identidades en la condición `aws:SourceArn` y en la cuenta en la condición `aws:SourceAccount`.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
        {
            "Sid": "lambda-allow-cognito-my-function",
            "Effect": "Allow",
            "Principal": {
                "Service": "cognito-sync.amazonaws.com"
            },
```

```
            "Action": "lambda:InvokeFunction",
            "Resource": "<your Lambda function ARN>",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "<your account number>"
                },
                "ArnLike": {
                    "AWS:SourceArn": "<your identity pool ARN>"
                }
            }
        }
    ]
}
```

5. En la pantalla de la función Configure, especifique un nombre y una descripción para su función. Deje Runtime establecido en "Node.js". No cambie el código para el ejemplo. El ejemplo predeterminado no modifica los datos que se están sincronizando. Solo registra el hecho de que se ha producido el evento desencadenador de Amazon Cognito Sync. Deje el nombre del controlador establecido en "index.handler". Para la opción de rol, seleccione un rol de IAM que conceda a su código permiso para acceder a AWS Lambda. Para modificar roles, consulte la consola de IAM. Deje la configuración avanzada sin cambiar. Haga clic en Next (Siguiente).
6. En la pantalla Review, revise los detalles y haga clic en Create function. En la página siguiente, se muestra la nueva función de Lambda.

Ahora que ya tiene una función adecuada escrita en Lambda, debe elegir esa función como controlador del evento desencadenador de Amazon Cognito Sync. Los pasos siguientes le guiarán por este proceso.

En la página de inicio de la consola:

1. Haga clic en el nombre del grupo de identidades para el que desee configurar Amazon Cognito Events. Se mostrará la página Dashboard (Panel) de su grupo de identidades.
2. En la esquina superior derecha de la página Dashboard, haga clic en Manage Federated Identities. Se visualizará la página Manage Federated Identities.
3. Desplácese hacia abajo y haga clic en Cognito Events para ampliar esta opción.
4. En el menú desplegable Sync Trigger (Desencadenador de sincronización), seleccione la función de Lambda que desee activar cuando se produzca un evento de sincronización.
5. Haga clic en Save Changes (Guardar cambios).

Ahora su función de Lambda se ejecutará cada vez que se sincronice un conjunto de datos. En la sección siguiente se explica cómo puede leer y modificar los datos de su función mientras se están sincronizando.

Escritura de una función de Lambda para los desencadenadores de sincronización

Los desencadenadores de sincronización respetan el patrón de programación de la interfaz del proveedor de servicios. Amazon Cognito proporciona datos de entrada con el formato JSON siguiente a su función de Lambda.

```
{
  "version": 2,
  "eventType": "SyncTrigger",
  "region": "us-east-1",
  "identityPoolId": "identityPoolId",
  "identityId": "identityId",
  "datasetName": "datasetName",
  "datasetRecords": {
    "SampleKey1": {
      "oldValue": "oldValue1",
      "newValue": "newValue1",
      "op": "replace"
    },
    "SampleKey2": {
      "oldValue": "oldValue2",
      "newValue": "newValue2",
      "op": "replace"
    },..
  }
}
```

Amazon Cognito espera que el valor de retorno de la función tenga el mismo formato que el de entrada.

Al escribir funciones para el evento Sync Trigger, observe lo siguiente:

- Cuando Amazon Cognito llama a la función de Lambda durante una operación UpdateRecords, esta función debe responder en un plazo máximo de 5 segundos. Si no lo hace, el servicio de Amazon Cognito Sync genera una excepción `LambdaSocketTimeoutException`. No puede aumentar este valor de tiempo de espera.

- Si recibe una excepción `LambdaThrottledException`, intente ejecutar la operación de sincronización de nuevo para actualizar los registros.
- Amazon Cognito proporciona todos los registros presentes en el conjunto de datos como datos de entrada para la función.
- Los registros que actualiza el usuario de la aplicación tienen el campo op definido como `replace`. Los registros eliminados tienen el campo op definido como `remove`.
- Puede modificar cualquier registro, aunque el usuario de la aplicación no lo actualice.
- Todos los campos, salvo datasetRecords, son de solo lectura. No los cambie. Si cambia estos campos, no podrá actualizar los registros.
- Para modificar el valor de un registro, actualice el valor y defina op como `replace`.
- Para eliminar un registro, establezca op en `remove` o defina un valor nulo.
- Para añadir un registro, solo tiene que añadir un registro nuevo en la matriz datasetRecords.
- Amazon Cognito ignora cualquier registro omitido en la respuesta cuando Amazon Cognito actualiza el registro.

Función de Lambda de ejemplo

En la siguiente función de Lambda de ejemplo se muestra cómo acceder, modificar y eliminar datos.

```
console.log('Loading function');

exports.handler = function(event, context) {
    console.log(JSON.stringify(event, null, 2));

    //Check for the event type
    if (event.eventType === 'SyncTrigger') {

        //Modify value for a key
        if('SampleKey1' in event.datasetRecords){
            event.datasetRecords.SampleKey1.newValue = 'ModifyValue1';
            event.datasetRecords.SampleKey1.op = 'replace';
        }

        //Remove a key
        if('SampleKey2' in event.datasetRecords){
            event.datasetRecords.SampleKey2.op = 'remove';
        }
```

```
        //Add a key
        if(!('SampleKey3' in event.datasetRecords)){
            event.datasetRecords.SampleKey3={'newValue':'ModifyValue3', 'op' :
 'replace'};
        }

    }
    context.done(null, event);
};
```

Uso de la consola de Amazon Cognito

Puede utilizar la consola de Amazon Cognito para crear y administrar los grupos de usuarios y de identidades.

Esta guía proporciona step-by-step tutoriales sobre las tareas comunes del grupo de usuarios de Amazon Cognito en la consola de Amazon Cognito.

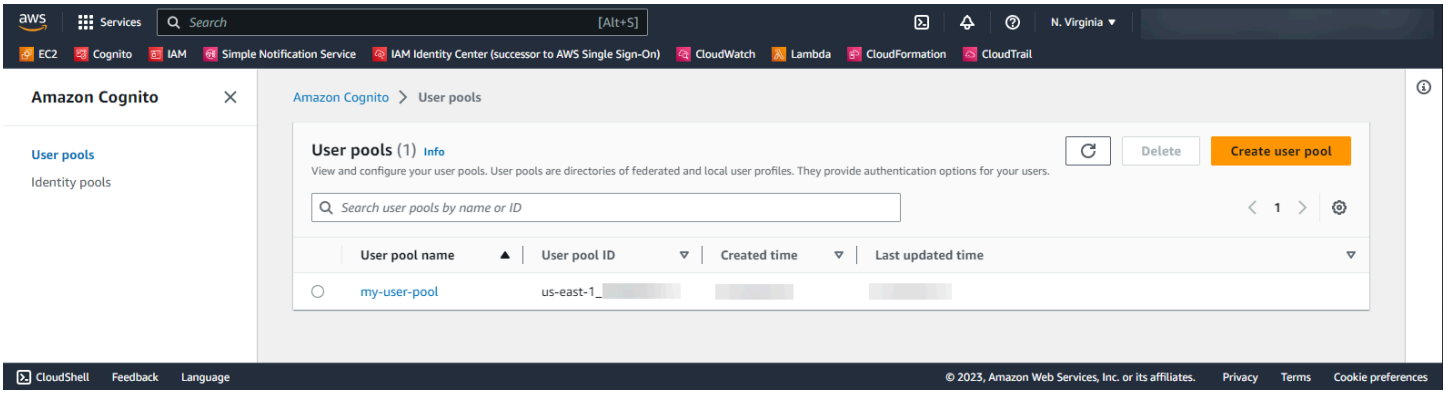
Para utilizar la consola de Amazon Cognito

1. Para utilizar Amazon Cognito, debe crear una AWS cuenta.
2. Vaya a la consola de Amazon Cognito. Es posible que se le pidan sus AWS credenciales.
3. Para crear o editar un grupo de usuarios, elija User Pools (Grupos de usuarios) en el panel de navegación izquierdo.

   Para obtener más información, consulte Introducción a los grupos de usuarios.
4. Para crear o editar un grupo de identidades, elija Grupos de identidades. Se le dirigirá a la consola original para los grupos de identidades de Amazon Cognito.

   Para obtener más información, consulte Introducción a los grupos de identidades de Amazon Cognito.

La consola Amazon Cognito forma parte de la AWS Management Console, que proporciona información sobre su cuenta y facturación. Para obtener más información, consulte Trabajar con la AWS Management Console.



Temas

- La consola de los grupos de usuarios

- La consola de los grupos de identidades

La consola de los grupos de usuarios

Desde la vista de los Grupos de usuarios en la consola de Amazon Cognito, elija un grupo de usuarios de la lista para ver los detalles. En la vista detallada, la Información general del grupo de usuarios en la parte superior de la consola contiene información básica sobre el grupo de usuarios. Las siguientes pestañas organizan la configuración del grupo de usuarios en funciones relacionadas.

Usuarios

La pestaña Usuarios contiene información sobre los usuarios y las importaciones de usuarios desde archivos CSV. Puede agregar, eliminar y editar usuarios en esta pestaña.

Referencias

- Administración de usuarios en el grupo de usuarios
- Importación de usuarios en grupos de usuarios desde un archivo CSV

Grupos

La pestaña Grupos contiene información sobre los grupos de usuarios. Puede agregar, modificar y cambiar la pertenencia a los grupos, así como cambiar los roles de IAM asociados a los grupos para la integración de grupos de identidades.

Referencias

- Agregar grupos a un grupo de usuarios

Experiencia de inicio de sesión

La pestaña Experiencia de inicio de sesión contiene información sobre cómo los usuarios inician sesión en el grupo de usuarios. En esta pestaña aparecen los proveedores de identidad de terceros, las opciones de nombre de usuario, la política de contraseñas, la configuración de la autenticación multifactor (MFA), el comportamiento al olvidar la contraseña y la función de recordar el dispositivo. Puede agregar y modificar proveedores de identidad y cambiar el comportamiento general de inicio de sesión del grupo de usuarios.

Referencias

- Agregar inicio de sesión de grupo de usuarios a través de un tercero

- Personalización de los atributos de inicio de sesión
- Adición de requisitos de contraseña para los grupos de usuarios
- Adición de MFA a un grupo de usuarios.
- Recuperación de cuentas de usuario
- Uso de dispositivos de usuario en el grupos de usuarios

Experiencia de inicio de sesión

La pestaña Experiencia de registro contiene información sobre el registro de autoservicio, los atributos necesarios, la verificación de números de teléfono y direcciones de correo electrónico y los atributos personalizados.

Referencias

- Inscripción y confirmación de cuentas de usuario
- Custom pool attributes (
- Verificación de la información de contacto durante el registro

Mensajería

La pestaña Mensajería contiene información sobre Servicios de AWS que desea utilizar para enviar mensajes de correo electrónico y SMS a los usuarios y el formato de los mensajes que desea enviarles.

Referencias

- Configuración de correo electrónico para grupos de usuarios de Amazon Cognito
- Configuración de mensajes SMS para grupos de usuarios de Amazon Cognito
- Configuración de los mensajes de verificación de correo electrónico y SMS, así como mensajes de invitación al usuario

Integración de la aplicación

La pestaña Integración de aplicaciones contiene información sobre los clientes de aplicaciones del grupo de usuarios, el dominio que se asigna a los puntos de conexión del servicio del grupo de usuarios, los servidores de recursos de la API, la interfaz de usuario alojada y la seguridad avanzada. Puede analizar en detalle cada cliente de aplicación para configurar lo siguiente.

1. Configuración de los tokens
2. Direcciones URL de devolución de llamada

3. Flujos de autenticación
4. Permisos de atributos
5. Configuración avanzada de interfaz de usuario alojada y de seguridad específica de la aplicación
6. Análisis de Amazon Pinpoint

Referencias

- Clientes de aplicación de grupo de usuarios
- Configuración y uso de la interfaz de usuario alojada y los puntos de conexión de federación de Amazon Cognito
- Configuración de un dominio del grupo de usuarios
- Autorización de alcances, M2M y API con servidores de recursos
- Adición de seguridad avanzada a un grupo de usuarios
- Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito

Propiedades del grupo de usuarios

La pestaña de propiedades del grupo de usuarios contiene información sobre la configuración del grupo de usuarios que no está directamente relacionada con los usuarios: activadores Lambda, protección de ACL AWS WAF web, protección de eliminación y etiquetas de recursos.

Referencias

- Personalización de flujos de trabajo de grupos de usuarios con desencadenadores de Lambda
- Asociar una ACL AWS WAF web a un grupo de usuarios
- Protección de eliminación de grupo de usuarios
- Etiquetar sus recursos AWS

La consola de los grupos de identidades

Desde la vista de los Grupos de identidades en la consola de Amazon Cognito, elija un grupo de identidades de la lista para ver los detalles. En la vista detallada, la Información general del grupo de identidades en la parte superior de la consola contiene información básica sobre el grupo de usuarios. Las siguientes pestañas organizan la configuración del grupo de usuarios en funciones relacionadas.

Estadísticas de usuario

La pestaña Estadísticas de los usuarios muestra información estadística sobre los usuarios que han generado identidades en el grupo de identidades. No puede configurar ningún ajuste del grupo de identidades en esta pestaña.

Navegador de identidades

La pestaña Navegador de identidades contiene información sobre las identidades individuales que los usuarios han generado en el grupo de identidades. Puede consultar y eliminar las identidades.

Referencias

- Introducción a los grupos de identidades de Amazon Cognito

Acceso de usuarios

La pestaña Acceso de usuario contiene información sobre los proveedores de identidades que ha vinculado al grupo de identidades, los proveedores de desarrolladores, los roles de IAM predeterminados asignados a las identidades y la configuración del acceso de invitados no autenticados. Puede analizar en detalle cada proveedor de identidades para configurar lo siguiente.

1. Control de acceso basado en roles con Selección de roles de IAM
2. Control de acceso basado en atributos con Atributos para el control de acceso

Referencias

- Proveedores de identidad externos de grupos de identidades
- Roles de IAM
- Identidades autenticadas y sin autenticar
- Identidades autenticadas por el desarrollador (grupos de identidades)
- Uso del control de acceso basado en roles
- Uso de atributos para el control de acceso

Propiedades del grupo de identidades

La pestaña Propiedades del grupo de identidades contiene información sobre diversas configuraciones de grupos de identidades: autenticación básica (clásica) y etiquetas de recursos.

- Flujo de autenticación de grupos de identidades (identidades federadas)
- Etiquetar tus recursos AWS

Seguridad en Amazon Cognito

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de conformidad que se aplican a Amazon Cognito, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad AWS Servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables

Esta documentación sirve de ayuda para comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon Cognito. Muestra cómo configurar Amazon Cognito para satisfacer sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de Amazon Cognito.

Contenido

Protección de datos en Amazon Cognito

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en Amazon Cognito (Amazon Cognito). Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecuta toda la AWS nube. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Este contenido incluye las tareas de configuración y administración de la seguridad de AWS los servicios que utiliza. Para obtener más información sobre la privacidad de datos, consulte Preguntas frecuentes sobre la privacidad de datos.

Con fines de protección de datos, le recomendamos que proteja las credenciales de las AWS cuentas y configure cuentas de usuario individuales con AWS Identity and Access Management (IAM). De esta manera, cada usuario recibe únicamente los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de las siguientes maneras:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados de AWS los servicios.
- Utilice avanzados servicios de seguridad administrados, como Amazon Macie, que lo ayuden a detectar y proteger los datos personales almacenados en Amazon S3.

Le recomendamos encarecidamente que nunca introduzca información de identificación confidencial, como, por ejemplo, números de cuenta de sus clientes, en los campos de formato libre, como el campo Nombre. Esto incluye cuando trabaja con Amazon Cognito u otros AWS servicios mediante la consola, la API o AWS los AWS CLI SDK. Es posible que cualquier dato que ingrese en Amazon Cognito o en otros servicios se incluya en los registros de diagnóstico. Cuando proporcione una URL a un servidor externo, no incluya información de credenciales en la URL para validar la solicitud para ese servidor.

Cifrado de datos

El cifrado de datos normalmente se divide en dos categorías: el cifrado en reposo y el cifrado en tránsito.

Cifrado en reposo

Los datos que se encuentran dentro de Amazon Cognito se cifran en reposo de acuerdo con los estándares del sector.

Cifrado en tránsito

Como servicio gestionado, Amazon Cognito está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Las llamadas a la API AWS publicadas se utilizan para acceder a Amazon Cognito a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM principal. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Los grupos de usuarios y grupos de identidades de Amazon Cognito tienen operaciones de API autenticadas por IAM, no autenticadas y autorizadas por token. Las operaciones de API no autenticadas y autorizadas por token están destinadas a ser utilizadas por sus clientes, los usuarios finales de la aplicación. Las operaciones de API no autenticadas y autorizadas por token están cifradas en reposo y en tránsito. Para obtener más información, consulte Operaciones de API autenticadas y no autenticadas de los grupos de usuarios de Amazon Cognito.

> **Note**
>
> Amazon Cognito cifra el contenido a nivel interno y no admite las claves proporcionadas por el cliente.

Identity and access management para Amazon Cognito

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de Amazon Cognito. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realice en Amazon Cognito.

Usuario de servicio: si utiliza el servicio de Amazon Cognito para realizar el trabajo, el administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amazon Cognito para realizar el trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en Amazon Cognito, consulte Solución de problemas de identidad y acceso de Amazon Cognito.

Administrador de servicio: si está a cargo de los recursos de Amazon Cognito de la empresa, es probable que tenga acceso completo a Amazon Cognito. El trabajo consiste en determinar a qué características y recursos de Amazon Cognito deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo la empresa puede utilizar IAM con Amazon Cognito, consulte Cómo funciona Amazon Cognito con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera obtener más detalles sobre cómo escribir políticas para administrar el acceso a Amazon Cognito. Para consultar ejemplos de políticas basadas en la identidad de Amazon Cognito que puede utilizar en IAM, consulte Ejemplos de políticas basadas en identidades de Amazon Cognito.

Autenticación con identidades

La autenticación es la forma de iniciar sesión para AWS usar sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, asumes un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la

contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute

aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.

- Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar una AWS función a una instancia EC2 y ponerla a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.

- Políticas de control de servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations .
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon Cognito con IAM

Antes de utilizar IAM para administrar el acceso a Amazon Cognito, obtenga información sobre qué características de IAM se pueden utilizar con Amazon Cognito.

Características de IAM que puede utilizar con Amazon Cognito

Característica de IAM	Soporte de Amazon Cognito
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí

Característica de IAM	Soporte de Amazon Cognito
Recursos de políticas	Sí
Claves de condición de política	Sí
ACL	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	No
Roles de servicio	Sí
Roles vinculados al servicio	Sí

Para obtener una visión general de cómo funcionan Amazon Cognito y otros AWS servicios con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas de Amazon Cognito basadas en identidades

Compatibilidad con las políticas basadas en identidad	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en

una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades de Amazon Cognito

Para ver ejemplos de políticas basadas en identidad de Amazon Cognito, consulte Ejemplos de políticas basadas en identidades de Amazon Cognito.

Políticas basadas en recursos de Amazon Cognito

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones de política de Amazon Cognito

Admite acciones de política	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Amazon Cognito, consulte Acciones definidas por Amazon Cognito en la Referencia de autorizaciones de servicio.

En las acciones de políticas de Amazon Cognito, se utiliza el siguiente prefijo antes de la acción:

```
cognito-identity
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "cognito-identity:action1",
      "cognito-identity:action2"
         ]
```

API firmadas y sin firmar

Al firmar las solicitudes de la API de Amazon Cognito con AWS credenciales, puede restringirlas mediante una política AWS Identity and Access Management (IAM). Las solicitudes de API con las que debe firmar las credenciales de AWS incluyen el inicio de sesión en el lado del servidor con `AdminInitiateAuth` y acciones que crean, ven o modifican recursos de Amazon Cognito, como `UpdateUserPool`. Para obtener más información sobre las solicitudes de API firmadas, consulte Firmar solicitudes de AWS API.

Dado que Amazon Cognito es un producto de identidad del consumidor para aplicaciones que desea poner a disposición del público, tiene acceso a las siguientes API sin firmar. La aplicación realiza estas solicitudes de API para los usuarios y los posibles usuarios. Algunas API no requieren autorización previa, como `InitiateAuth` para iniciar una nueva sesión de autenticación. Algunas API utilizan tokens de acceso o claves de sesión para la autorización, como

`VerifySoftwareToken` para completar la configuración de MFA para un usuario que tiene una sesión autenticada existente. Una API de grupo de usuarios de Amazon Cognito no firmada y autorizada admite un parámetro `Session` o `AccessToken` en la sintaxis de la solicitud, tal como se muestra en la Referencia de la API de Amazon. Una API de identidad de Amazon Cognito no firmada admite un parámetro `IdentityId` tal y como se muestra en la Referencia de la API de identidades federadas de Amazon Cognito.

Para obtener más información acerca de los modelos de autorización y roles de las operaciones de la API de grupos de usuarios de Amazon Cognito, consulte Operaciones de API autenticadas y no autenticadas de los grupos de usuarios de Amazon Cognito.

Operaciones de la API de los grupos de identidades de Amazon Cognito

- `GetId`
- `GetOpenIdToken`
- `GetCredentialsForIdentity`
- `UnlinkIdentity`

Operaciones de la API de los grupos de usuarios de Amazon Cognito

- `AssociateSoftwareToken`
- `ChangePassword`
- `ConfirmDevice`
- `ConfirmForgotPassword`
- `ConfirmSignUp`
- `DeleteUser`
- `DeleteUserAttributes`
- `ForgetDevice`
- `ForgotPassword`
- `GetDevice`
- `GetUser`
- `GetUserAttributeVerificationCode`
- `GlobalSignOut`
- `InitiateAuth`

- `ListDevices`
- `ResendConfirmationCode`
- `RespondToAuthChallenge`
- `RevokeToken`
- `SetUserMFAPreference`
- `SetUserSettings`
- `SignUp`
- `UpdateAuthEventFeedback`
- `UpdateDeviceStatus`
- `UpdateUserAttributes`
- `VerifySoftwareToken`
- `VerifyUserAttribute`

Para ver ejemplos de políticas basadas en identidad de Amazon Cognito, consulte Ejemplos de políticas basadas en identidades de Amazon Cognito.

Recursos de políticas de Amazon Cognito

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Nombres de recursos de Amazon (ARN)

ARN para las identidades federadas de Amazon Cognito

En los grupos de identidades de Amazon Cognito (identidades federadas), es posible restringir el acceso de un usuario de IAM a un grupo de identidades específico mediante el formato de nombre de recurso de Amazon (ARN), como se muestra en el ejemplo siguiente. Para obtener más información sobre los ARN, consulte Identificadores de IAM.

```
arn:aws:cognito-identity:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

ARN para Amazon Cognito Sync

En Amazon Cognito Sync, los clientes también pueden restringir el acceso en función del ID del grupo de identidades, el ID de identidad y el nombre del conjunto de datos.

En el caso de las API que operan en un grupo de identidades, el formato ARN del grupo de identidades coincide con el de las identidades federadas de Amazon Cognito, salvo por el hecho de que el nombre del servicio es `cognito-sync`, y no `cognito-identity`:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

En cuanto a las API que operan en una única identidad como, por ejemplo, `RegisterDevice`, puede consultar la identidad individual aplicando el formato de ARN siguiente:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/
identity/IDENTITY_ID
```

En el caso de las API que operan en conjuntos de datos como, por ejemplo, `UpdateRecords` y `ListRecords`puede consultar el conjunto de datos individuales usando el siguiente formato de ARN:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/
identity/IDENTITY_ID/dataset/DATASET_NAME
```

ARN para los grupos de usuarios de Amazon Cognito

En el caso de la característica Sus grupos de usuarios de Amazon Cognito, se puede restringir el acceso de un usuario a un grupo de usuarios específico mediante el formato de ARN siguiente:

```
arn:aws:cognito-idp:REGION:ACCOUNT_ID:userpool/USER_POOL_ID
```

Para ver una lista de los tipos de recursos de Amazon Cognito y los ARN, consulte Recursos definidos por Amazon Cognito en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon Cognito.

Para ver ejemplos de políticas basadas en identidad de Amazon Cognito, consulte Ejemplos de políticas basadas en identidades de Amazon Cognito.

Claves de condición de políticas de Amazon Cognito

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de las claves de condición de Amazon Cognito, consulte Claves de condición para Amazon Cognito en la Referencia de autorizaciones de servicio. Para obtener más información sobre las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon Cognito.

Para ver ejemplos de políticas basadas en identidad de Amazon Cognito, consulte Ejemplos de políticas basadas en identidades de Amazon Cognito.

Listas de control de acceso (ACL) en Amazon Cognito

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Control de acceso basado en atributos (ABAC) con Amazon Cognito

Admite ABAC (etiquetas en las políticas)	Parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con Amazon Cognito

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios de Amazon Cognito

Admite sesiones de acceso directo (FAS)	No

Cuando utilizas un usuario o un rol de IAM para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio

de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio de Amazon Cognito

Compatible con roles de servicio	Sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

Para obtener detalles acerca de los roles de servicio de Amazon Cognito, consulte Activar sincronización mediante inserción y Sincronización mediante inserción.

> **Warning**
>
> Es posible que cambiar los permisos de un rol de servicio interrumpa la funcionalidad de Amazon Cognito. Edite los roles de servicio solo cuando Amazon Cognito proporcione orientación para hacerlo.

Roles vinculados a servicios para Amazon Cognito

Compatible con roles vinculados al servicio	Sí

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para obtener más información sobre cómo crear o administrar roles vinculados a servicios de Amazon Cognito, consulte Uso de roles vinculados a servicios para Amazon Cognito.

Ejemplos de políticas basadas en identidades de Amazon Cognito

De forma predeterminada, los usuarios y los roles no tienen permiso para crear ni modificar los recursos de Amazon Cognito. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por Cognito, incluido el formato de los ARN para cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de Amazon Cognito en la Referencia de autorizaciones de servicio.

Temas

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon Cognito de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso.

Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

> **Note**
>
> La versión original y la nueva de la consola de Amazon Cognito tienen un comportamiento subyacente diferente cuando ve y modifica los recursos de Amazon Cognito. Si concede permiso a las acciones en el prefijo de servicio de `cognito-idp` solo cuando la condición `aws:ViaAWSService` sea cierta, la entidad principal de IAM afectada puede

trabajar con los recursos de Amazon Cognito en la consola original, pero no en la nueva consola. Para trabajar en la consola de Amazon Cognito, no configure una condición `aws:ViaAWSService` en los permisos de Amazon Cognito en la política de IAM.

Uso de la consola de Amazon Cognito

Para acceder a la consola de Amazon Cognito, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de Amazon Cognito que tiene en su cuenta. Cuenta de AWS Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No necesita conceder permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de Amazon Cognito, adjunte también la política gestionada `ReadOnly` AWS o Amazon `ConsoleAccess` Cognito a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. AWS CLI AWS

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

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

Restricción del acceso a la consola a un grupo de identidades concreto

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-identity:ListIdentityPools"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cognito-identity:*"
      ],
      "Resource": "arn:aws:cognito-identity:us-east-1:0123456789:identitypool/us-
east-1:1a1a1a1a-ffff-1111-9999-12345678"
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
        "cognito-sync:*"
      ],
      "Resource": "arn:aws:cognito-sync:us-east-1:0123456789:identitypool/us-
east-1:1a1a1a1a-ffff-1111-9999-12345678"
    }
  ]
}
```

Autorización del acceso a un conjunto de datos concreto para todas las identidades de un grupo

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-sync:ListRecords",
        "cognito-sync:UpdateRecords"
      ],
      "Resource": "arn:aws:cognito-sync:us-east-1:0123456789:identitypool/us-
east-1:1a1a1a1a-ffff-1111-9999-12345678/identity/*/dataset/UserProfile"
    }
  ]
}
```

Solución de problemas de identidad y acceso de Amazon Cognito

Utilice la información siguiente para diagnosticar y solucionar los problemas comunes que es posible que surjan cuando trabaje con Amazon Cognito e IAM.

Temas

- No tengo autorización para realizar una acción en Amazon Cognito
- No estoy autorizado a realizar tareas como: PassRole
- Soy administrador y deseo permitir que otros accedan a Amazon Cognito

- Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amazon Cognito

No tengo autorización para realizar una acción en Amazon Cognito

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `cognito-identity:`*`GetWidget`*.

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

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `cognito-identity:`*`GetWidget`*.

Si necesitas ayuda, ponte en contacto con tu administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas se deben actualizar para permitirle pasar un rol a Amazon Cognito.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon Cognito. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Soy administrador y deseo permitir que otros accedan a Amazon Cognito

Para permitir que otros tengan acceso a Amazon Cognito, debe crear una entidad de IAM (un usuario o rol) para la persona o aplicación que necesita acceso. Esta persona utilizará las credenciales de la entidad para acceder a AWS. A continuación, debe asociar una política a la entidad que les adjudica los permisos correctos en Amazon Cognito.

Para comenzar de inmediato, consulte Creación del primer grupo y usuario delegado de IAM en la Guía del usuario de IAM.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amazon Cognito

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para obtener información acerca de si Amazon Cognito admite estas características, consulte Cómo funciona Amazon Cognito con IAM.
- Para obtener información sobre cómo proporcionar acceso a sus recursos a través de Cuentas de AWS los suyos, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Uso de roles vinculados a servicios para Amazon Cognito

Amazon Cognito utiliza funciones vinculadas a AWS Identity and Access Management servicios (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM con una política de confianza que permite a un usuario asumir el rol. Servicio de AWS Amazon Cognito predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a AWS otros servicios en su nombre.

Un rol vinculado a un servicio simplifica la configuración de Amazon Cognito porque ya no tendrá que agregar de forma manual los permisos necesarios. Amazon Cognito define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Amazon Cognito puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Amazon Cognito, ya que se evita que se puedan eliminar por accidente los permisos de acceso a los recursos.

Para obtener información acerca de otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service Linked Role (Rol vinculado a servicios). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de roles vinculados a servicios para Amazon Cognito

Amazon Cognito utiliza los siguientes roles vinculados a servicios:

- AWSServiceRoleForAmazonCognitoIdpEmailService— Permite que el servicio de grupos de usuarios de Amazon Cognito utilice sus identidades de Amazon SES para enviar correos electrónicos.

- AWSServiceRoleForAmazonCognitoIdp— Permite a los grupos de usuarios de Amazon Cognito publicar eventos y configurar puntos de enlace para sus proyectos de Amazon Pinpoint.

AWSServiceRoleForAmazonCognitoIdpEmailService

El rol vinculado al servicio `AWSServiceRoleForAmazonCognitoIdpEmailService` depende de los siguientes servicios para asumir el rol:

- `email.cognito-idp.amazonaws.com`

La política de permisos del rol permite que Amazon Cognito realice las siguientes acciones en los recursos especificados:

Acciones permitidas para: AWSServiceRoleForAmazonCognitoIdpEmailService

- Acción:`ses:SendEmail` y `ses:SendRawEmail`
- Recurso: *

La política deniega a Amazon Cognito la capacidad para realizar las siguientes acciones en los recursos especificados:

Acciones denegadas

- Acción: `ses:List*`
- Recurso: *

Con estos permisos, Amazon Cognito puede utilizar las direcciones de correo electrónico verificadas en Amazon SES solo para enviar correos electrónicos a los usuarios. Amazon Cognito envía un correo electrónico a los usuarios cuando ejecutan ciertas acciones en la aplicación del cliente para un grupo de usuarios, como registrarse o restablecer una contraseña.

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

AWSServiceRoleForAmazonCognitoIdp

El rol AWSServiceRoleForAmazonCognitoIdp vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `email.cognito-idp.amazonaws.com`

La política de permisos del rol permite que Amazon Cognito realice las siguientes acciones en los recursos especificados:

Acciones permitidas para AWSServiceRoleForAmazonCognitoIdp

- Acción: `cognito-idp:Describe`
- Recurso: *

Con este permiso, Amazon Cognito puede llamar a las operaciones de la API de Amazon Cognito `Describe` por usted.

> **Note**
>
> Cuando integre Amazon Cognito en Amazon Pinpoint mediante `createUserPoolClient` y `updateUserPoolClient`, los permisos de recursos se agregarán a la SLR como una política integrada. La política integrada proporcionará permisos `mobiletargeting:UpdateEndpoint` y `mobiletargeting:PutEvents`. Con estos permisos, Amazon Cognito puede publicar eventos y configurar puntos de conexión para los proyectos Pinpoint que integre con Cognito.

Creación de un rol vinculado a un servicio para Amazon Cognito

No necesita crear manualmente un rol vinculado a servicios. Cuando configura un grupo de usuarios para que utilice su configuración de Amazon SES para gestionar la entrega de correo electrónico en la AWS Management Console AWS CLI, la o la API de Amazon Cognito, Amazon Cognito crea el rol vinculado al servicio automáticamente.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al configurar un grupo de usuarios para utilizar la configuración de Amazon SES a fin de gestionar la entrega de correo electrónico, Amazon Cognito nuevamente crea el rol vinculado a servicios por usted.

Para que Amazon Cognito pueda crear este rol, los permisos de IAM que utilice para configurar su grupo de usuarios deben incluir la acción `iam:CreateServiceLinkedRole`. Para obtener más información acerca de la actualización de permisos en IAM, consulte Cambio de los permisos de un usuario de IAM en la Guía del usuario de IAM.

Edición de un rol vinculado a un servicio para Amazon Cognito

No puede editar los roles AmazonCognitoIdp ni AmazonCognitoIdpEmailService los vinculados al servicio. AWS Identity and Access Management Después de crear un rol vinculado a un servicio,

no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia a él. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado a un servicio para Amazon Cognito

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Si elimina el rol, solo conservará entidades que Amazon Cognito supervisa o mantiene activamente. Antes de poder eliminar los roles AmazonCognitoIdp o los AmazonCognitoIdpEmailService vinculados a un servicio, debe realizar una de las siguientes acciones para cada grupo de usuarios que utilice el rol:

- Eliminar el grupo de usuarios.
- Actualizar la configuración de correo electrónico en el grupo de usuarios para utilizar la funcionalidad de correo electrónico predeterminada. La configuración predeterminada no utiliza el rol vinculado al servicio.

Recuerde realizar la acción en cada uno de ellos Región de AWS con un grupo de usuarios que utilice el rol.

> **Note**
>
> Si el servicio Amazon Cognito utiliza el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar un grupo de usuarios de Amazon Cognito

1. Inicie sesión en la consola de Amazon Cognito AWS Management Console y ábrala en. https://console.aws.amazon.com/cognito
2. Elija Administrar grupos de usuarios.
3. En la página Your User Pools (Sus grupos de usuarios), seleccione el grupo de usuarios que desee eliminar.
4. Elija Delete pool (Eliminar grupo).
5. En la ventana Delete user pool (Eliminar grupo de usuarios), escriba **`delete`** y elija Delete pool (Eliminar grupo).

Para actualizar un grupo de usuarios de Amazon Cognito para utilizar la funcionalidad de correo electrónico predeterminada

1. Inicie sesión en la consola de Amazon Cognito AWS Management Console y ábrala en. https://console.aws.amazon.com/cognito
2. Elija Administrar grupos de usuarios.
3. En la página Your User Pools (Sus grupos de usuarios), seleccione el grupo de usuarios que desee actualizar.
4. En el menú de navegación de la izquierda, elija Message customizations (Personalizaciones de mensajes).
5. En Do you want to send emails through your Amazon SES Configuration? (¿Desea enviar correos electrónicos a través de su configuración de Amazon SES?), elija No - Use Cognito (Default) (No - Usar Cognito [Predeterminado]).
6. Cuando termine de configurar las opciones de su cuenta de correo electrónico, seleccione Save changes (Guardar modificaciones).

Eliminación manual del rol vinculado a servicios mediante IAM

Utilice la consola de IAM AWS CLI, la o la AWS API para eliminar funciones AmazonCognitoIdp o vinculadas a AmazonCognitoIdpEmailService servicios. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones compatibles con los roles vinculados a servicios de Amazon Cognito

Amazon Cognito admite funciones vinculadas a servicios en todos los lugares en los que el servicio Regiones de AWS esté disponible. Para obtener más información, consulte Puntos de conexión y Regiones de AWS.

Registro y monitoreo en Amazon Cognito

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de Amazon Cognito y sus demás AWS soluciones. Actualmente, Amazon Cognito admite los siguientes Servicios de AWS para que pueda supervisar su organización y la actividad que ocurre en ella.

- AWS CloudTrail — Con él CloudTrail puede capturar llamadas a la API desde la consola de Amazon Cognito y desde las llamadas de código a las operaciones de la API de Amazon Cognito.

Por ejemplo, cuando un usuario se autentica, CloudTrail puede registrar detalles como la dirección IP de la solicitud, quién la realizó y cuándo se realizó.

- Amazon CloudWatch Logs: con CloudWatch Logs, puede enviar registros detallados de la actividad de los usuarios a un grupo de registros. Por ejemplo, puede revisar los registros detallados de la actividad de los usuarios para solucionar problemas relacionados con la entrega de mensajes de correo electrónico y SMS a sus usuarios.
- Amazon CloudWatch Metrics: con CloudWatch las métricas, puedes monitorear, informar y tomar acciones automáticas en caso de que se produzca un evento casi en tiempo real. Por ejemplo, puede crear CloudWatch paneles en las métricas proporcionadas para monitorear sus grupos de usuarios de Amazon Cognito, o puede CloudWatch crear alarmas en las métricas proporcionadas para notificarle si se infringe un umbral establecido.
- Amazon CloudWatch Logs Insights: con CloudWatch Logs Insights, puede configurar el envío de eventos CloudTrail a los que supervisar CloudWatch los archivos de CloudTrail registro de Amazon Cognito.

Temas

Costes de supervisión

Amazon Cognito cobra por las siguientes dimensiones de uso.

- Grupo de usuarios activos mensuales (MAU)
- Las MAU del grupo de usuarios que han iniciado sesión con la federación OIDC o SAML
- Las MAU de un grupo de usuarios con funciones de seguridad avanzadas
- Grupo de usuarios activos, clientes de aplicaciones y volumen de solicitudes de autorización de máquina a máquina (M2M) con concesiones de credenciales de cliente
- Se adquirió un uso superior a las cuotas predeterminadas para algunas categorías de API de grupos de usuarios

Además, las funciones de su grupo de usuarios, como los mensajes de correo electrónico, los mensajes SMS y los activadores Lambda, pueden generar costes en servicios dependientes. Para obtener una descripción completa, consulte los precios de Amazon Cognito.

Visualización y previsión de los costos

Puede ver sus AWS costes e informar sobre ellos en la AWS Billing and Cost Management consola. Puedes encontrar los cargos más recientes de Amazon Cognito en la sección Facturación y pagos. En Facturas, cargos por servicio, filtre `Cognito` para ver su consumo. Para obtener más información, consulte Ver su factura en la Guía del usuario de AWS Billing .

Para supervisar las tasas de solicitudes de API, revise la métrica de utilización en la consola Service Quotas. Por ejemplo, las solicitudes de credenciales de los clientes se muestran como Tasa de ClientAuthentication solicitudes. En tu factura, estas solicitudes están asociadas al cliente de la aplicación que las generó. Con esta información, puede asignar los costos de manera equitativa a los inquilinos en una arquitectura con varios inquilinos.

Para hacer un recuento de las solicitudes M2M durante un período de tiempo, también puede enviar los AWS CloudTrail eventos a CloudWatch Logs para su análisis. Consulta tus CloudTrail eventos en busca de `Token_POST` eventos con una concesión de credenciales de cliente. La siguiente consulta de CloudWatch Insights devuelve este recuento.

```
filter eventName = "Token_POST" and @message like '"grant_type":["client_credentials"]'
 | stats count(*)
```

Administración de los costos de

Amazon Cognito factura en función del número de usuarios, el uso de las funciones y el volumen de solicitudes. Los siguientes son algunos consejos para gestionar los costes en Amazon Cognito,

No active a los usuarios inactivos

Las operaciones típicas que hacen que un usuario esté activo son el inicio de sesión, el registro y el restablecimiento de la contraseña. Para obtener una lista más completa, consulte. Monthly active users (Usuarios activos mensuales) Amazon Cognito no incluye a los usuarios inactivos en su factura. Evite cualquier operación que active a un usuario. En lugar de la operación de AdminGetUserAPI, consulta a los usuarios con la ListUsersoperación. No realices pruebas administrativas de gran volumen de operaciones de grupos de usuarios con usuarios inactivos.

Vincula a los usuarios federados

Los usuarios que inician sesión con un proveedor de identidad SAML 2.0 o OpenID Connect (OIDC) tienen un coste mayor que los usuarios locales. Puede vincular estos usuarios a un perfil de usuario local. Un usuario vinculado puede iniciar sesión como usuario local con los atributos y el acceso que vienen con su usuario federado. Los usuarios de SAML u OIDC IdPs que, en el transcurso de un mes, solo inicien sesión con una cuenta local vinculada se facturan como usuarios locales.

Gestione las tarifas de solicitud

Si su grupo de usuarios se acerca al límite superior de su cuota, podría considerar la posibilidad de adquirir capacidad adicional para gestionar el volumen. Es posible que pueda reducir el volumen de solicitudes de su aplicación. Para obtener más información, consulte Optimice las tasas de solicitud para cumplir con los límites de cuota.

Solicita un nuevo token solo cuando lo necesites

La autorización de máquina a máquina (M2M) con la concesión de credenciales de cliente puede alcanzar un gran volumen de solicitudes de token. Cada nueva solicitud de token afecta a tu cuota de solicitudes y al importe de tu factura. Para optimizar los costes, incluya la configuración de caducidad de los tokens y su gestión en el diseño de sus aplicaciones.

- Almacene en caché los tokens de acceso para que, cuando su aplicación solicite un nuevo token, reciba una versión en caché de un token emitido anteriormente. Cuando implementas este método, tu proxy de almacenamiento en caché actúa como protección contra las aplicaciones que solicitan tokens de acceso sin saber que han caducado los tokens adquiridos anteriormente. El almacenamiento en caché de los tokens es ideal para microservicios de corta duración, como las funciones de Lambda y los contenedores de Docker.
- Implemente mecanismos de gestión de tokens en sus aplicaciones que tengan en cuenta la caducidad de los tokens. No solicites un nuevo token hasta que los anteriores hayan caducado. Evalúe las necesidades de confidencialidad y disponibilidad de cada aplicación y configure el cliente de la aplicación del grupo de usuarios para que emita los tokens de acceso con un período de validez adecuado. La duración del token personalizado funciona mejor con API y servidores de larga duración que pueden gestionar de forma persistente la frecuencia de las solicitudes de credenciales.

Elimine las credenciales de cliente (clientes de aplicaciones) no utilizadas

Las autorizaciones M2M se facturan en función de dos factores: la tasa de solicitudes de tokens y la cantidad de clientes de aplicaciones que otorgan credenciales a los clientes. Cuando los clientes de aplicaciones para la autorización M2M no estén en uso, elimínelos o quite su autorización para emitir

credenciales de cliente. Para obtener más información sobre la administración de la configuración de los clientes de aplicaciones, consulteClientes de aplicación de grupo de usuarios.

Gestione la seguridad avanzada

Al configurar funciones de seguridad avanzadas en un grupo de usuarios, la tarifa de facturación de seguridad avanzada se aplica a todas las MAU del grupo de usuarios. Si tiene usuarios que no necesitan funciones de seguridad avanzadas, sepárelos en otro grupo de usuarios.

Seguimiento de las cuotas CloudWatch y el uso en Service Quotas

Puede monitorizar los grupos de usuarios de Amazon Cognito mediante Amazon CloudWatch o Service Quotas. También puede supervisar el uso de los grupos de identidades en Service Quotas. CloudWatch recopila datos sin procesar y los procesa para convertirlos en métricas legibles y prácticamente en tiempo real. En CloudWatch él, puede configurar alarmas que vigilen determinados umbrales y enviar notificaciones o tomar medidas cuando se alcancen esos umbrales. Para crear una CloudWatch alarma para una cuota de servicio, consulta Crear una CloudWatch alarma. Las métricas de Amazon Cognito están disponibles en intervalos de cinco minutos. Para obtener más información sobre los períodos de retención CloudWatch, visita la página de CloudWatch preguntas frecuentes de Amazon.

Puede utilizar Service Quotas para ver y administrar el uso de las cuotas de grupos de usuarios y grupos de identidades de Amazon Cognito. La consola de Service Quotas tiene tres características: ver cuotas de servicio, solicitar un aumento de la cuota de servicio y ver la utilización actual. Puede usar la primera característica para ver las cuotas y si la cuota es ajustable. Puede usar la segunda característica para solicitar un aumento de Service Quotas. Puede usar la última característica para ver la utilización de cuotas. Esta característica solo está disponible después de que la cuenta haya estado activa durante un tiempo. Para obtener más información sobre cómo ver las cuotas en la consola de Service Quotas, consulte Visualización de Service Quotas.

> **Note**
>
> Las métricas de Amazon Cognito están disponibles a intervalos de 5 minutos. Para obtener más información sobre los períodos de retención CloudWatch, visita la página de CloudWatch preguntas frecuentes de Amazon.

Si has iniciado sesión en una cuenta Cuenta de AWS que está configurada como una cuenta de monitorización en el ámbito de la observabilidad CloudWatch multicuenta, puedes usar esa cuenta

de monitorización para visualizar las cuotas de servicio y configurar alarmas para las métricas de las cuentas de origen que están vinculadas a esa cuenta de monitorización. Para obtener más información, consulta la observabilidad CloudWatch entre cuentas.

Temas

- Registro de actividad adicional de grupos de usuarios de Amazon Cognito
- Métricas de los grupos de usuarios de Amazon Cognito
- Dimensiones de los grupos de usuarios de Amazon Cognito
- Uso de la consola de Service Quotas para hacer un seguimiento de las métricas
- Usa la CloudWatch consola para realizar un seguimiento de las métricas
- Cree una CloudWatch alarma para una cuota

Registro de actividad adicional de grupos de usuarios de Amazon Cognito

Puede configurar su grupo de usuarios para enviar registros detallados de alguna actividad adicional a un grupo de CloudWatch registros. Estos registros tienen una granularidad más precisa que los de su grupo de usuarios y pueden resultar útiles para solucionar problemas relacionados con su grupo de usuarios. AWS CloudTrail Al activar esta característica, puede elegir el grupo de registros al que desea que Amazon Cognito envíe los registros. El registro de actividad de los usuarios es útil cuando desea conocer el estado de los mensajes de correo electrónico y SMS que el grupo de usuarios entregó con Amazon SNS y Amazon SES.

Actualmente, solo puede entregar registros de notificación de usuario de nivel Error del grupo de usuarios.

El registro detallado no reemplaza ni cambia las siguientes funciones de registro de los grupos de usuarios.

1. CloudTrail registros de la actividad rutinaria de los usuarios, como el registro y el inicio de sesión.
2. Análisis de la actividad de los usuarios a escala con CloudWatch métricas.

Por separado, también puede encontrar los registros de los trabajos de importación de usuarios y los activadores de Lambda en CloudWatch los registros. Amazon Cognito y Lambda almacenan estos registros en grupos de registros diferentes de los que especifica para obtener registros de actividad detallados.

Puede configurar registros de actividad detallados con la API de grupos de usuarios de Amazon Cognito en una solicitud de SetLogDeliveryConfigurationAPI. Puede ver la configuración de registro de un grupo de usuarios en una solicitud de GetLogDeliveryConfigurationAPI.

Debe autorizar estas solicitudes con AWS credenciales que tengan los siguientes permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
            {
            "Sid": "ManageUserPoolLogs",
            "Action": [
                "cognito-idp:SetLogDeliveryConfiguration",
                "cognito-idp:GetLogDeliveryConfiguration",
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "CognitoLog",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "CognitoLoggingCWL",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
```

```
            ],
            "Effect": "Allow"
        }
    ]
}
```

A continuación, se muestra un evento de ejemplo de un grupo de usuarios. Este esquema de registro está sujeto a cambios. Es posible que algunos campos se registren con valores nulos.

```
{
    "eventTimestamp": "1687297330677",
    "eventSource": "USER_NOTIFICATION",
    "logLevel": "ERROR",
    "message": {
        "details": "String"
    },
    "logSourceId": {
        "userPoolId": "String"
    }
}
```

La entrega de registros desde Amazon Cognito es el mejor esfuerzo. El volumen de registros que entrega su grupo de usuarios y sus cuotas de servicio para CloudWatch los registros pueden afectar a la entrega de registros.

CloudWatch Los cargos por registros se aplican cuando la entrega de registros está habilitada. Para obtener más información, consulta los CloudWatch precios de Vended Logs en Amazon.

Para enviar registros a grupos de registros con una política de recursos de un tamaño superior a 5120 caracteres, configure un grupo de registros con una ruta que comience por `/aws/vendedlogs`. Para obtener más información, consulta Cómo habilitar el registro desde determinados AWS servicios.

Métricas de los grupos de usuarios de Amazon Cognito

En la siguiente tabla, se enumeran las métricas disponibles para grupos de usuarios de Amazon Cognito. El espacio de nombres de las métricas de Amazon CloudWatch para Amazon Cognito es `AWS/Cognito`. Para obtener más información, consulta la Guía del CloudWatch usuario de Namespaces in Amazon.

> **Note**
>
> Las métricas que no han tenido nuevos puntos de datos en las últimas dos semanas no aparecen en la consola. Tampoco aparecen al ingresar el nombre de métrica o los nombres de dimensiones en el cuadro de búsqueda de la pestaña All metrics (Todas las métricas) de la consola. Además, no se devuelven en los resultados de un comando list-metrics. La mejor forma de recuperar estas métricas es con los `get-metric-statistics` comandos `get-metric-data` o de la AWS CLI.

Métrica	Descripción
`SignUpSuccesses`	Proporciona la cantidad total de solicitudes de registro de usuarios correctas realizadas al grupo de usuarios de Amazon Cognito. Una solicitud de registro de usuario correcta produce un valor de 1, mientras que una solicitud incorrecta produce un valor de 0. Una solicitud de limitación controlada también se considera una solicitud incorrecta y, por lo tanto, una solicitud de limitación controlada también producirá un recuento de 0. Para encontrar el porcentaje de solicitudes de registro de usuarios correctas, utilice la estadística `Average` en esta métrica. Para contar el número total de solicitudes de registro de usuarios, utilice la estadística `Sample Count` en esta métrica. Para contar el número total de solicitudes de registro de usuarios correctas, utilice la estadística `Sum` en esta métrica. Para contar el número total de solicitudes de registro de usuarios fallidas, utilice la CloudWatch `Math` expresión y reste la `Sum` estadística de la `Sample Count` estadística.

Métrica	Descripción
	Esta métrica se publica para cada grupo de usuarios de cada cliente de grupo de usuarios. En caso de que el registro del usuario lo realice un administrador, la métrica se publica con el cliente del grupo de usuarios como `Admin`. Tenga en cuenta que esta métrica no se emite en casos de importación de usuarios y migración de usuarios. Dimensión de métrica: `UserPool`, `UserPoolClient` Unidades: recuento
`SignUpThrottles`	Proporciona la cantidad total de solicitudes de registro de usuarios con limitación controlada realizadas al grupo de usuarios de Amazon Cognito. Se publica un recuento de 1 cada vez que se realiza la limitación controlada de una solicitud de registro de usuario. Para contar el número total de solicitudes de registro de usuarios con limitación controlada, utilice la estadística `Sum` para esta métrica. Esta métrica se publica para cada grupo de usuarios de cada cliente. En caso de que la solicitud en la que se haya realizado la limitación controlada fuera realizada por un administrador, la métrica se publica con el cliente del grupo de usuarios como `Admin`. Dimensión de métrica: `UserPool`, `UserPoolClient` Unidades: recuento

Métrica	Descripción
`SignInSuccesses`	Proporciona la cantidad total de solicitudes de autenticación de usuarios correctas realizadas al grupo de usuarios de Amazon Cognito. Una autenticación de usuario se considera correcta cuando se emite un token de autenticación al usuario. Una autenticación correcta produce un valor de 1, mientras que una solicitud incorrect a produce un valor de 0. Una solicitud de limitación controlada también se considera una solicitud incorrecta y, por lo tanto, una solicitud de limitación controlada también producirá un recuento de 0. Para buscar el porcentaje de solicitudes de autenticación de usuario correctas, utilice la estadística `Average` en esta métrica. Para contar el número total de solicitudes de autenticación de usuario, utilice la estadística `Sample Count` en esta métrica. Para contar el número total de solicitudes de autenticación de usuario correctas, utilice la estadística `Sum` en esta métrica. Para contar el número total de solicitudes de autenticación de usuario fallidas, utilice la CloudWatch `Math` expresión y reste la estadística de la `Sum` estadística. `Sample Count` Esta métrica se publica para cada grupo de usuarios de cada cliente. En caso de que se proporcione un cliente de grupo de usuarios no válido con una solicitud, el valor de cliente de grupo de usuarios correspondiente de la métrica contiene un valor fijo `Invalid`, en lugar del valor no válido real enviado en la solicitud.

Métrica	Descripción
	Tenga en cuenta que las solicitudes para actualizar el token de Amazon Cognito no se incluyen en esta métrica. Hay una métrica distinta para proporcionar estadísticas de token de `Refresh`. Dimensión de métrica: `UserPool`, `UserPoolClient` Unidades: recuento

Métrica	Descripción
`SignInThrottles`	Proporciona la cantidad total de solicitudes de autenticación de usuarios con limitación controlada realizadas al grupo de usuarios de Amazon Cognito. Se publica un recuento de 1 cada vez que se realiza la limitación controlada de una solicitud de autenticación. Para contar el número total de solicitudes de autenticación de usuarios con limitación controlada, utilice la estadística `Sum` para esta métrica. Esta métrica se publica para cada grupo de usuarios de cada cliente. En caso de que se proporcione un cliente de grupo de usuarios no válido con una solicitud, el valor de cliente de grupo de usuarios correspondiente de la métrica contiene un valor fijo `Invalid`, en lugar del valor no válido real enviado en la solicitud. Las solicitudes para actualizar el token de Amazon Cognito no se incluyen en esta métrica. Hay una métrica distinta para proporcionar estadísticas de token de `Refresh`. Dimensión de métrica: `UserPool`, `UserPoolClient` Unidades: recuento

Métrica	Descripción
`TokenRefreshSuccesses`	Proporciona la cantidad total de solicitudes correctas para actualizar un token de Amazon Cognito que se realizaron en el grupo de usuarios de Amazon Cognito. Una solicitud de token de Amazon Cognito de actualización correcta produce un valor de 1, mientras que una solicitud incorrecta produce un valor de 0. Una solicitud de limitación controlada también se considera una solicitud incorrecta y, por lo tanto, una solicitud de limitación controlada también producirá un recuento de 0. Para buscar el porcentaje de solicitudes correctas para actualizar un token de Amazon Cognito, utilice la estadística `Average` en esta métrica. Para contar la cantidad total de solicitudes para actualizar un token de Amazon Cognito, utilice la estadística `Sample Count` en esta métrica. Para contar la cantidad total de solicitudes correctas para actualizar un token de Amazon Cognito, utilice la estadísti ca `Sum` en esta métrica. Para contar el número total de solicitudes fallidas para actualizar un token de Amazon Cognito, utilice la CloudWatc h `Math` expresión y reste la `Sum` estadística de la estadística. `Sample Count` Esta métrica se publica por cada cliente del grupo de usuarios. Si un cliente de grupo de usuarios no válido está en una solicitud, el valor de cliente del grupo de usuarios contiene un valor fijo de `Invalid`. Dimensión de métrica: `UserPool`, `UserPoolC lient`

Métrica	Descripción
	Unidades: recuento
`TokenRefreshThrottles`	Proporciona el número total de solicitudes con limitación controlada para actualizar el token de Amazon Cognito que se realizaron en el grupo de usuarios de Amazon Cognito. Se publica un recuento de 1 cada vez que se realiza la limitación controlada de una solicitud de token de actualización de Amazon Cognito. A fin de contar la cantidad total de solicitudes con limitación controlada para actualizar un token de Amazon Cognito, utilice la estadística `Sum` para esta métrica. Esta métrica se publica para cada grupo de usuarios de cada cliente. En caso de que se proporcione un cliente de grupo de usuarios no válido con una solicitud, el valor de cliente del grupo de usuarios correspondiente en la métrica contiene un valor fijo `Invalid`, en lugar del valor no válido real enviado en la solicitud. Dimensión de métrica: `UserPool`, `UserPoolClient` Unidades: recuento

Métrica	Descripción
`FederationSuccesses`	Proporciona la cantidad total de solicitudes de identidad federada correctas al grupo de usuarios de Amazon Cognito. Se considera que una federación de identidades se ha realizado correctamente cuando Amazon Cognito emite tokens de autenticación para el usuario. Una solicitud de identidad federada correcta produce un valor de 1, mientras que una solicitud incorrecta produce un valor de 0. Las solicitudes limitadas y las que generan un código de autorización pero ningún token producen un valor de 0. Para buscar el porcentaje de solicitudes de identidad federada correctas, utilice la estadísti ca `Average` en esta métrica. Para contar el número total de solicitudes de identidad federada, utilice la estadística `Sample Count` en esta métrica. Para contar el número total de solicitudes de identidad federada correctas , utilice la estadística `Sum` en esta métrica. Para contar el número total de solicitudes de federación de identidades fallidas, utilice la CloudWatch `Math` expresión y reste la estadísti ca de la `Sum` estadística. `Sample Count` Dimensión de métrica: `UserPool`, `UserPoolC lient` , `IdentityProvider` Unidades: recuento

Métrica	Descripción
`FederationThrottles`	Proporciona la cantidad total de solicitudes de identidad federada con limitación controlad a al grupo de usuarios de Amazon Cognito. Se publica un recuento de 1 cada vez que se realiza la limitación controlada de una solicitud de identidad federada. Para contar el número total de solicitudes de identidad federada con limitación controlada, utilice la estadística `Sum` para esta métrica. Dimensión de métrica: `UserPool`, `UserPoolClient` , `IdentityProvider` Unidades: recuento
`CallCount`	Proporciona la cantidad total de llamadas que realizan los clientes en relación con una categoría. Esta métrica incluye todas las llamadas, como llamadas con limitació n controlada, llamadas fallidas y llamadas correctas. Esta métrica está disponible en Uso `nameSpace` . La cuota por categorías se aplica a cada AWS cuenta en todos los grupos de usuarios de una cuenta y región. Puede contar la cantidad total de llamadas en una categoría con la estadística de `Sum` para esta métrica. Dimensión métrica: servicio, tipo, recurso, clase Unidades: recuento

Métrica	Descripción
`ThrottleCount`	Proporciona la cantidad total de llamadas con limitación controlada en relación con una categoría. Esta métrica está disponible en Uso `nameSpace` . Esta métrica se publica a nivel de cuenta. Puede contar la cantidad total de llamadas en una categoría con la estadística de `Sum` para esta métrica. Dimensión métrica: servicio, tipo, recurso, clase Unidades: recuento

Dimensiones de los grupos de usuarios de Amazon Cognito

Las siguientes dimensiones se utilizan para ajustar las métricas de uso publicadas por Amazon Cognito. Las dimensiones solo se aplican a las métricas de `CallCount` y `ThrottleCount` .

Dimensión	Descripción
Servicio	El nombre del AWS servicio que contiene el recurso. Para las métricas de uso de Amazon Cognito, el valor de esta dimensión es `Cognito user pool`.
Tipo	El tipo de entidad que se registra. En este momento, el único valor válido para las métricas de uso de Amazon Cognito es API.
Recurso	El tipo de recurso que se está ejecutando. El único valor válido es el nombre de la categoría.

Dimensión	Descripción
Clase	La clase de recurso a la que se realiza el seguimiento. Amazon Cognito no utiliza la dimensión de clase.

Uso de la consola de Service Quotas para hacer un seguimiento de las métricas

Puede utilizar Service Quotas para ver y administrar las cuotas de grupos de usuarios y grupos de identidades de Amazon Cognito desde una ubicación centralizada. Puede utilizar la consola de Service Quotas para ver los detalles de una cuota específica, monitorear la utilización de las cuotas y solicitar un aumento de cuota. Para algunos tipos de cuota, puede crear una CloudWatch alarma para realizar un seguimiento de la utilización de la cuota. Para obtener más información sobre qué métricas de Amazon Cognito puede realizar un seguimiento, consulte Seguimiento del uso de cuotas.

Para consultar el uso de Service Quotas de grupos de usuarios y grupos de identidad de Amazon Cognito, siga estos pasos.

1. Abra la consola de Service Quotas.
2. En el panel de navegación, elija Servicios de AWS .
3. En la lista de servicios de AWS , busque y elija Grupos de usuarios de Amazon Cognito o Identidades federadas de Amazon Cognito. Se mostrará la página de cuotas de servicio.
4. Seleccione una cuota que permita la CloudWatch supervisión. Por ejemplo, elija `Rate of UserAuthentication requests` en los grupos de usuarios de Amazon Cognito.
5. Desplácese hasta Monitoring (Monitoreo). Esta sección solo aparece para las cuotas que admiten la CloudWatch supervisión.
6. En Monitoring (Monitoreo), puede ver la utilización actual de la cuota de servicio en el gráfico.
7. En Monitoring (Monitoreo), seleccione una hora, tres horas, doce horas, un día, tres días o una semana.
8. Seleccione cualquier área dentro del gráfico para ver el porcentaje de utilización de la cuota de servicio. Desde aquí, puedes añadir el gráfico a tu panel de control o usar el menú de acciones para seleccionar Ver en las métricas, lo que te llevará a las métricas relacionadas en la CloudWatch consola.

Usa la CloudWatch consola para realizar un seguimiento de las métricas

Puede realizar un seguimiento y recopilar las métricas de los grupos de usuarios de Amazon Cognito mediante. CloudWatch El CloudWatch panel mostrará las métricas de todos los AWS servicios que utilice. Se puede utilizar CloudWatch para crear alarmas métricas. Las alarmas se pueden configurar para enviar notificaciones o modificar un recurso específico que monitoree. Para ver las métricas de la cuota de servicio CloudWatch, complete los siguientes pasos.

1. Abra la consola de CloudWatch.
2. En el panel de navegación, seleccione Metrics (Métricas).
3. En All metrics (Todas las métricas), seleccione una métrica y una dimensión.
4. Seleccione la casilla de verificación situada junto a una métrica. Las métricas se mostrarán en el gráfico.

> **Note**
>
> Las métricas que no han tenido nuevos puntos de datos en las últimas dos semanas no aparecen en la consola. Tampoco aparecen al ingresar el nombre de la métrica o los nombres de las dimensiones en el cuadro de búsqueda de la pestaña All metrics (Todas las métricas) de la consola, ni aparecen en los resultados de un comando list-metrics. La mejor manera de recuperar estas métricas es con los comandos `get-metric-data` o `get-metric-statistics` en la CLI de AWS .

Cree una CloudWatch alarma para una cuota

Amazon Cognito proporciona métricas de CloudWatch uso que se corresponden con las cuotas de AWS servicio `CallCount` y `ThrottleCount` las API. Para obtener más información sobre el seguimiento del uso en CloudWatch, consulteSeguimiento del uso de cuotas.

En la consola de Service Quotas, puede crear alarmas que le avisen cuándo su uso se acerque a una cuota de servicio. Para obtener información sobre cómo configurar una CloudWatch alarma mediante la consola Service Quotas, consulte Service Quotas and CloudWatch alarm.

Registrar llamadas a la API de Amazon Cognito con AWS CloudTrail

Amazon Cognito está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en Amazon Cognito. CloudTrail

captura un subconjunto de llamadas a la API de Amazon Cognito como eventos, incluidas las llamadas desde la consola de Amazon Cognito y las llamadas en código a las operaciones de la API de Amazon Cognito. Si crea una ruta, puede optar por enviar CloudTrail los eventos a un bucket de Amazon S3, incluidos los eventos de Amazon Cognito. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon Cognito, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, incluido cómo configurarlo y activarlo, consulte la Guía del AWS CloudTrail usuario.

También puedes crear CloudWatch alarmas de Amazon para CloudTrail eventos específicos. Por ejemplo, puede configurarlo CloudWatch para que se active una alarma si se cambia la configuración de un grupo de identidades. Para obtener más información, consulte Creación de CloudWatch alarmas para CloudTrail eventos: ejemplos.

Temas

- Información de Amazon Cognito en CloudTrail
- Explicación de los eventos de inicio de sesión de Amazon Cognito
- Análisis de CloudTrail eventos de Amazon Cognito con Amazon Logs Insights CloudWatch

Información de Amazon Cognito en CloudTrail

CloudTrail se activa al crear su. Cuenta de AWSCuando se produce una actividad de eventos admitida en Amazon Cognito, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los eventos recientes en su AWS cuenta. Para obtener más información, consulta Cómo ver eventos con el historial de CloudTrail eventos.

Para obtener un registro continuo de los eventos de su AWS cuenta, incluidos los eventos de Amazon Cognito, cree una ruta. Un CloudTrail rastro envía los archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configurar las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el elemento CloudTrail UserIdentity.

Datos confidenciales en AWS CloudTrail

Dado que los grupos de usuarios y los grupos de identidades procesan los datos de los usuarios, Amazon Cognito oculta algunos campos privados de sus CloudTrail eventos con el valor. `HIDDEN_FOR_SECURITY_REASONS` Para ver ejemplos de campos que Amazon Cognito no rellena para los eventos, consulte Explicación de los eventos de inicio de sesión de Amazon Cognito. Amazon Cognito solo oculta algunos campos que suelen contener información de usuario, como contraseñas y tokens. Amazon Cognito no detecta ni oculta automáticamente la información de identificación personal que usted rellena en campos no privados en las solicitudes de la API.

Grupos de usuarios de Amazon Cognito

Amazon Cognito admite el registro de todas las acciones que aparecen en la página de acciones del grupo de usuarios como eventos en los archivos de CloudTrail registro. Amazon Cognito registra los eventos del grupo de usuarios CloudTrail como eventos de administración.

El `eventType` campo de una CloudTrail entrada de grupos de usuarios de Amazon Cognito indica si la aplicación realizó la solicitud a la API de grupos de usuarios de Amazon Cognito o a un punto final que proporciona recursos para OpenID Connect, SAML 2.0 o la interfaz de usuario alojada. Las solicitudes de la API tienen un `eventType` de `AwsApiCall` y las solicitudes de punto de conexión tienen un `eventType` de `AwsServiceEvent`.

Amazon Cognito registra las siguientes solicitudes de interfaz de usuario alojada en su interfaz de usuario alojada como eventos en. CloudTrail

Operaciones de interfaz de usuario alojadas en CloudTrail

Operación	Descripción
`Login_GET` , `CognitoAuthentication`	Un usuario ve o envía credenciales a su Punto de conexión Login.
`OAuth2_Authorize_GET` , `Beta_Authorize_GET`	Un usuario ve su Autorizar punto de conexión.
`OAuth2Response_GET` , `OAuth2Response_POST`	Un usuario envía un token de proveedor de identidad a su punto de conexión `/oauth2/idpresponse` .
`SAML2Response_POST` , `Beta_SAML2Response_POST`	Un usuario envía una afirmación de SAML de proveedor de identidad a su punto de conexión `/saml2/idpresponse` .
`Login_OIDC_SAML_POST`	Un usuario introduce un nombre de usuario en su Punto de conexión Login y coincide con un Identificador de proveedor de identidad.
`Token_POST` , `Beta_Token_POST`	Un usuario envía un código de autorización a su Punto de conexión de token.
`Signup_GET` , `Signup_POST`	Un usuario envía la información de registro a su punto de conexión `/signup`.
`Confirm_GET` , `Confirm_POST`	Un usuario envía un código de confirmación en la interfaz de usuario alojada.
`ResendCode_POST`	Un usuario envía una solicitud para volver a enviar un código de confirmación en la interfaz de usuario alojada.

Operación	Descripción
`ForgotPassword_GET` , `ForgotPassword_POST`	Un usuario envía una solicitud para restablecer su contraseña a su punto de conexión `/forgotPassword` .
`ConfirmForgotPassword_GET` , `ConfirmForgotPassword_POST`	Un usuario envía un código a su punto de conexión `/confirmForgotPassword` que confirma su solicitud de `ForgotPassword` .
`ResetPassword_GET` , `ResetPassword_POST`	Un usuario envía una nueva contraseña en la interfaz de usuario alojada.
`Mfa_GET`, `Mfa_POST`	Un usuario envía un código de autenticación multifactor (MFA) en la IU alojada.
`MfaOption_GET` , `MfaOption_POST`	El usuario elige su método preferido para la MFA en la interfaz de usuario alojada.
`MfaRegister_GET` , `MfaRegister_POST`	Un usuario envía un código de autenticación multifactor (MFA) en la IU alojada al registrar la MFA.
`Logout`	Un usuario cierra sesión en su punto de conexión `/logout`.
`SAML2Logout_POST`	Un usuario cierra sesión en su punto de conexión `/saml2/logout` .
`Error_GET`	Un usuario ve una página de error en la interfaz de usuario alojada.
`UserInfo_GET` , `UserInfo_POST`	Un usuario o proveedor de identidad intercambia información con su Punto de conexión de UserInfo.
`Confirm_With_Link_GET`	Un usuario envía una confirmación basada en un enlace que Amazon Cognito envió en un mensaje de correo electrónico.

Operación	Descripción
`Event_Feedback_GET`	Un usuario envía comentarios a Amazon Cognito sobre un evento de características de seguridad avanzadas.

> **Note**
>
> Amazon Cognito registra`UserSub`, pero no `UserName` en CloudTrail los registros, las solicitudes específicas de un usuario. Si desea buscar un usuario para un `UserSub` determinado, llame a la API de `ListUsers` y utilice un filtro para sub.

Grupos de identidades de Amazon Cognito

Eventos de datos

Amazon Cognito registra los siguientes eventos de Amazon Cognito Identity como eventos CloudTrail de datos. Los eventos de datos son operaciones de API del plano de datos de gran volumen que CloudTrail no se registran de forma predeterminada. Se aplican cargos adicionales a los eventos de datos.

- GetCredentialsForIdentity
- GetId
- GetOpenIdToken
- GetOpenIdTokenForDeveloperIdentity
- UnlinkIdentity

Para generar CloudTrail registros para estas operaciones de API, debe activar los eventos de datos en su seguimiento y elegir selectores de eventos para los grupos de identidades de Cognito. Para obtener más información, consulte Registro de eventos de datos para registros de seguimiento en la Guía del usuario de AWS CloudTrail .

También puede añadir selectores de eventos de grupos de identidades a su registro de seguimiento con el siguiente comando de la CLI.

```
aws cloudtrail put-event-selectors --trail-name <trail name> --advanced-event-selectors
 \
"{\
   \"Name\": \"Cognito Selector\",\
   \"FieldSelectors\": [\
      {\
         \"Field\": \"eventCategory\",\
         \"Equals\": [\
            \"Data\"\
         ]\
      },\
      {\
         \"Field\": \"resources.type\",\
         \"Equals\": [\
            \"AWS::Cognito::IdentityPool\"\
         ]\
      }\
   ]\
}"
```

Eventos de administración

Amazon Cognito registra el resto de las operaciones de la API de los grupos de identidades de Amazon Cognito como eventos de administración. CloudTrail registra las operaciones de la API de eventos de administración de forma predeterminada.

Para obtener una lista de las operaciones de la API de grupos de identidades de Amazon Cognito en las que Amazon Cognito inicia sesión, consulte la referencia de CloudTrail la API de grupos de identidades de Amazon Cognito.

Amazon Cognito Sync

Amazon Cognito registra todas las operaciones de la API de Amazon Cognito Sync como eventos de administración. Para obtener una lista de las operaciones de la API Amazon Cognito Sync en las que Amazon Cognito inicia sesión, consulte la referencia de CloudTrail la API Amazon Cognito Sync.

Explicación de los eventos de inicio de sesión de Amazon Cognito

Un rastro puede entregar eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la

fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

Temas

Ejemplos de CloudTrail eventos para una suscripción a una interfaz de usuario alojada

Los siguientes CloudTrail eventos de ejemplo muestran la información que Amazon Cognito registra cuando un usuario se registra a través de la interfaz de usuario alojada.

Amazon Cognito registra el siguiente evento cuando un usuario nuevo navega a la página de inicio de sesión de la aplicación.


```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-04-06T05:38:12Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Login_GET",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "errorCode": "",
    "errorMessage": "",
    "additionalEventData":
```

```
    {
        "responseParameters":
        {
            "status": 200.0
        },
        "requestParameters":
        {
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "response_type":
            [
                "token"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        }
    },
    "eventID": "382ae09a-151d-4116-8f2b-6ac0a804a38c",
    "readOnly": true,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra el siguiente evento cuando un usuario nuevo elige Sign up (Inscripción) en la página de inicio de sesión correspondiente a su aplicación.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-05T23:21:43Z",
```

```
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Signup_GET",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "response_type":
            [
                "code"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "7a63e7c2-b057-4f3d-a171-9d9113264fff",
    "eventID": "5e7b27a0-6870-4226-adb4-f86cd51ac5d8",
    "readOnly": true,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra el siguiente evento cuando un usuario nuevo elige un nombre de usuario, ingresa una dirección de correo electrónico y elige una contraseña en la página de inicio de sesión de la aplicación. Amazon Cognito no registra la información de identificación sobre la identidad del usuario. CloudTrail


```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-05T23:22:05Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Signup_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 302
        },
        "requestParameters":
        {
            "password":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "requiredAttributes[email]":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "response_type":
            [
                "code"
            ],
            "_csrf":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
```

```
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ],
            "username":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "9ad58dd8-3517-4aa8-96a5-d17a01df9eb4",
    "eventID": "c75eb7a5-eb8c-43d1-8331-f4412e756e69",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra el siguiente evento cuando un usuario nuevo accede a la página de confirmación de usuario en la interfaz de usuario alojada después de registrarse.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-05T23:22:06Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Confirm_GET",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
```

```
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "response_type":
            [
                "code"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "58a5b170-3127-45bb-88cc-3e652d779e0b",
    "eventID": "7f87291a-6d50-409a-822f-e3a5ec7e60da",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```


Amazon Cognito registra el siguiente evento cuando, en la página de confirmación de usuario de la interfaz de usuario alojada, un usuario introduce un código que Amazon Cognito le envió en un mensaje de correo electrónico.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-05T23:23:32Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Confirm_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 302
        },
        "requestParameters":
        {
            "confirm":
            [
                ""
            ],
            "deliveryMedium":
            [
                "EMAIL"
            ],
            "sub":
            [
                "704b1e47-34fe-40e9-8c41-504997494531"
            ],
            "code":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "destination":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "response_type":
```

```
            [
                "code"
            ],
            "_csrf":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "cognitoAsfData":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ],
            "username":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "9764300a-ed35-4f87-8a0f-b18b3fe2b11e",
    "eventID": "e24ac6e5-2f70-4c6e-ad4e-2f08a547bb36",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Ejemplo de CloudTrail evento para una solicitud de SAML

Amazon Cognito registra el siguiente evento cuando un usuario que se ha autenticado con su proveedor de identidad de SAML envía la afirmación de SAML a su punto de conexión `/saml2/idpresponse`.


```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-06T00:50:57Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "SAML2Response_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 302
        },
        "requestParameters":
        {
            "RelayState":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "SAMLResponse":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "4f6f15d1-c370-4a57-87f0-aac4817803f7",
    "eventID": "9824b50f-d9d1-4fb8-a2c1-6aa78ca5902a",
    "readOnly": false,
```

```
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "625647942648",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Ejemplos de CloudTrail eventos para solicitudes al punto final del token

En el ejemplo, se muestran eventos de las solicitudes a Punto de conexión de token.

Amazon Cognito registra el siguiente evento cuando un usuario que se ha autenticado y ha recibido un código de autorización envía el código a su punto de conexión `/oauth2/token`.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-12T22:12:30Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Token_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "code":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
```

```
            "grant_type":
            [
                "authorization_code"
            ],
            "redirect_uri":
            [
                "https://www.amazon.com"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "f257f752-cc14-4c52-ad5b-152a46915238",
    "eventID": "0bd1586d-cd3e-4d7a-abaf-fd8bfc3912fd",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra el siguiente evento cuando el sistema backend envía una solicitud `client_credentials` de un token de acceso al punto de conexión `/oauth2/token`.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-12T21:07:05Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Token_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
```

```
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "grant_type":
            [
                "client_credentials"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "4f871256-6825-488a-871b-c2d9f55caff2",
    "eventID": "473e5cbc-a5b3-4578-9ad6-3dfdcb8a6d34",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra el siguiente evento cuando su aplicación cambia un token de actualización por un ID y un token de acceso nuevos con su punto de conexión `/oauth2/token`.

```
{
    "eventVersion": "1.08",
    "userIdentity":
    {
```

```
        "accountId": "123456789012"
    },
    "eventTime": "2022-05-12T22:16:40Z",
    "eventSource": "cognito-idp.amazonaws.com",
    "eventName": "Token_POST",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData":
    {
        "responseParameters":
        {
            "status": 200
        },
        "requestParameters":
        {
            "refresh_token":
            [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ],
            "grant_type":
            [
                "refresh_token"
            ],
            "client_id":
            [
                "1example23456789"
            ]
        },
        "userPoolDomain": "mydomain.us-west-2.amazoncognito.com",
        "userPoolId": "us-west-2_aaaaaaaaa"
    },
    "requestID": "2829f0c6-a3a9-4584-b046-11756dfe8a81",
    "eventID": "12bd3464-59c7-44fa-b8ff-67e1cf092018",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
```

```
    "eventCategory": "Management"
}
```

Ejemplo de CloudTrail evento para CreateIdentityPool

El siguiente ejemplo muestra una entrada de registro de una solicitud de la acción `CreateIdentityPool`. La solicitud fue realizada por una usuaria de IAM llamada Alicia.

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "['EXAMPLE_KEY_ID']",
        "userName": "Alice"
    },
    "eventTime": "2016-01-07T02:04:30Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "CreateIdentityPool",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "USER_AGENT",
    "requestParameters": {
        "identityPoolName": "TestPool",
        "allowUnauthenticatedIdentities": true,
        "supportedLoginProviders": {
            "graph.facebook.com": "000000000000000"
        }
    },
    "responseElements": {
        "identityPoolName": "TestPool",
        "identityPoolId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
        "allowUnauthenticatedIdentities": true,
        "supportedLoginProviders": {
            "graph.facebook.com": "000000000000000"
        }
    },
    "requestID": "15cc73a1-0780-460c-91e8-e12ef034e116",
    "eventID": "f1d47f93-c708-495b-bff1-cb935a6064b2",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
```

```
}
```

Ejemplo de CloudTrail evento para GetCredentialsForIdentity

El siguiente ejemplo muestra una entrada de registro de una solicitud de la acción `GetCredentialsForIdentity`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetCredentialsForIdentity",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-credentials-for-identity",
    "requestParameters": {
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "aAaAaAaAaAaAab1111111111EXAMPLE",
            "expiration": "Jan 19, 2023 5:55:08 PM"
        },
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "659dfc23-7c4e-4e7c-858a-1abce884d645",
    "eventID": "6ad1c766-5a41-4b28-b5ca-e223ccb00f0d",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
```

```
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
}
```

Ejemplo de CloudTrail evento para GetId

El siguiente ejemplo muestra una entrada de registro de una solicitud de la acción `GetId`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:05Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetId",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-id",
    "requestParameters": {
        "identityPoolId": "us-east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE",
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
    },
    "responseElements": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "dc28def9-07c8-460a-a8f3-3816229e6664",
    "eventID": "c5c459d9-40ec-41fd-8f6b-57865d5a9975",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Data"
}
```

Ejemplo de CloudTrail evento para GetOpenIdToken

El siguiente ejemplo muestra una entrada de registro de una solicitud de la acción `GetOpenIdToken`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetOpenIdToken",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-open-id-token",
    "requestParameters": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
    },
    "responseElements": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "a506ba18-10d7-4fdb-9548-a8187b2e38bb",
    "eventID": "19ffc1a6-6ed8-4580-a4e1-3062c5ce6457",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
```

```
}
```

Ejemplo de CloudTrail evento para GetOpenIdTokenForDeveloperIdentity

El siguiente ejemplo muestra una entrada de registro de una solicitud de la acción `GetOpenIdTokenForDeveloperIdentity`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA1EXAMPLE:johns-AssumedRoleSession",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/johns-AssumedRoleSession",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA1EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2023-01-19T16:53:14Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetOpenIdTokenForDeveloperIdentity",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "27.0.3.154",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.get-open-id-token-for-developer-identity",
    "requestParameters": {
        "tokenDuration": 900,
        "identityPoolId": "us-east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE",
        "logins": {
            "JohnsDeveloperProvider": "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
    },
```

```
    "responseElements": {
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "b807df87-57e7-4dd6-b90c-b06f46a61c21",
    "eventID": "f26fed91-3340-4d70-91ae-cdf555547b76",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
}
```

Ejemplo de CloudTrail evento para UnlinkIdentity

El siguiente ejemplo muestra una entrada de registro de una solicitud de la acción `UnlinkIdentity`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:08Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "UnlinkIdentity",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.4",
    "userAgent": "aws-cli/2.7.25 Python/3.9.11 Darwin/21.6.0 exe/x86_64 prompt/off
 command/cognito-identity.unlink-identity",
    "requestParameters": {
        "logins": {
            "cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa":
 "HIDDEN_DUE_TO_SECURITY_REASONS"
        },
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
        "loginsToRemove": ["cognito-idp.us-east-1.amazonaws.com/us-east-1_aaaaaaaaa"]
    },
```

```
    "responseElements": null,
    "requestID": "99c2c8e2-9c29-416f-bb17-b650a5cbada9",
    "eventID": "d8e26126-202a-43c2-b458-3f225efaedc7",
    "readOnly": false,
    "resources": [{
        "accountId": "111122223333",
        "type": "AWS::Cognito::IdentityPool",
        "ARN": "arn:aws:cognito-identity:us-east-1:111122223333:identitypool/us-
east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "111122223333",
    "eventCategory": "Data"
}
```

Análisis de CloudTrail eventos de Amazon Cognito con Amazon Logs Insights CloudWatch

Puede buscar y analizar sus CloudTrail eventos de Amazon Cognito con Amazon CloudWatch Logs Insights. Cuando configura su ruta para enviar eventos a CloudWatch Logs, CloudTrail envía solo los eventos que coinciden con la configuración de su ruta.

Para consultar o investigar sus CloudTrail eventos de Amazon Cognito, en la CloudTrail consola, asegúrese de seleccionar la opción Gestión de eventos en la configuración de la ruta para poder supervisar las operaciones de administración que se realizan en sus AWS recursos. También puede seleccionar la opción Eventos de Insights en la configuración de seguimiento si desea identificar errores, actividades inusuales o comportamiento inusual del usuario en la cuenta.

Consultas de ejemplo de Amazon Cognito

Puedes usar las siguientes consultas en la CloudWatch consola de Amazon.

Consultas generales

Buscar los 25 eventos de registro agregados más recientes.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com"
```

Obtenga una lista de los 25 eventos de registro agregados recientemente que incluyen excepciones.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com" and @message like /Exception/
```

Consultas de excepción y error

Busque los 25 eventos de registro agregados más recientes con el código de error `NotAuthorizedException` junto con el grupo de usuarios de Amazon Cognito`sub`.

```
fields @timestamp, additionalEventData.sub as user | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
 "NotAuthorizedException"
```

Busque el número de registros con la `sourceIPAddress` y el `eventName` correspondiente.

```
filter eventSource = "cognito-idp.amazonaws.com"
| stats count(*) by sourceIPAddress, eventName
```

Busque las 25 direcciones IP principales que desencadenaron un error de `NotAuthorizedException`.

```
filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
 "NotAuthorizedException"
| stats count(*) as count by sourceIPAddress, eventName
| sort count desc | limit 25
```

Busque las 25 direcciones IP principales que llamaron a la API de `ForgotPassword`.

```
filter eventSource = "cognito-idp.amazonaws.com" and eventName = 'ForgotPassword'
| stats count(*) as count by sourceIPAddress
| sort count desc | limit 25
```

Validación de la conformidad para Amazon Cognito

Los auditores externos evalúan la seguridad y la conformidad de Amazon Cognito como parte de varios programas de AWS conformidad. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para obtener una lista de AWS los servicios incluidos en el ámbito de los programas de conformidad específicos, consulte los AWS servicios incluidos en el ámbito de aplicación por programa de

conformidad y AWS los servicios incluidos . Para obtener información general, consulte Programas de conformidad de AWS.

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descargar informes en AWS Artifact .

Su responsabilidad de conformidad al utilizar Amazon Cognito se determina en función de la sensibilidad de los datos, los objetivos de conformidad de su empresa y la legislación y los reglamentos vigentes. En AWS se proporcionan los siguientes recursos para ayudar con la conformidad:

- Guías de inicio rápido de seguridad y conformidad: estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- AWS recursos de cumplimiento: esta colección de trabajo y guías puede aplicarse a su sector y ubicación.
- Evaluar los recursos con las reglas de la Guía para AWS Config desarrolladores: AWS Config evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar el cumplimiento de los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia en Amazon Cognito

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja demora. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte la infraestructura AWS global.

Temas

Consideraciones de datos regionales

Los grupos de usuarios de Amazon Cognito se crean cada uno en una AWS región y almacenan los datos del perfil de usuario únicamente en esa región. Los grupos de usuarios pueden enviar los datos de los usuarios a una AWS región diferente, en función de cómo estén configuradas las funciones opcionales.

- Si la configuración predeterminada de la dirección de email `no-reply@verificationemail.com` se utiliza para la verificación de direcciones de email con grupos de usuarios de Amazon Cognito, los email se dirigen a través de la misma región que el grupo de usuarios asociado.
- Si se utiliza una dirección de correo electrónico diferente para configurar Amazon Simple Email Service (Amazon SES) con los grupos de usuarios de Amazon Cognito, esa dirección de correo electrónico se envía a AWS través de la región asociada a la dirección de correo electrónico en Amazon SES.
- Los mensajes SMS de los grupos de usuarios de Amazon Cognito se enrutan a través de la misma región de Amazon SNS, a menos que se indique lo contrario en Configuring Email or Phone Verification (Configuración de la verificación del correo electrónico o del teléfono).
- Si los análisis de Amazon Pinpoint se utilizan con grupos de usuarios de Amazon Cognito, los datos de eventos se dirigen a la región US East (Virginia del Norte).

> **Note**
>
> Amazon Pinpoint está disponible en varias AWS regiones de Norteamérica, Europa, Asia y Oceanía. Las regiones de Amazon Pinpoint incluyen la API de Amazon Pinpoint. Si Amazon Cognito admite una región de Amazon Pinpoint, enviará eventos a proyectos de Amazon Pinpoint dentro de la misma región de Amazon Pinpoint. Si una región no es compatible con Amazon Pinpoint, Amazon Cognito solo admitirá el envío de eventos en us-east-1. Para obtener información detallada sobre la región de Amazon Pinpoint, consulte Cuotas y puntos

de enlace de Amazon Pinpoint y Uso de Amazon Pinpoint Analytics con grupos de usuarios de Amazon Cognito.

Seguridad de la infraestructura en Amazon Cognito

Como servicio gestionado, Amazon Cognito está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Las llamadas a la API AWS publicadas se utilizan para acceder a Amazon Cognito a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM principal. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Configuración y análisis de vulnerabilidades en grupos de usuarios de Amazon Cognito

AWS se encarga de las tareas de seguridad básicas, como la aplicación de parches al sistema operativo (SO) huésped y a las bases de datos, la configuración del firewall y la recuperación ante desastres. Estos procedimientos han sido revisados y certificados por los terceros pertinentes. Para obtener más detalles, consulte los siguientes recursos de :

- Validación de la conformidad para Amazon Cognito
- Modelo de responsabilidad compartida

AWS políticas gestionadas para Amazon Cognito

Para añadir permisos a usuarios, grupos y roles, es más fácil usar políticas AWS administradas que escribirlas usted mismo. Se necesita tiempo y experiencia para crear políticas administradas por el cliente de IAM que proporcionen a su equipo solo los permisos necesarios. Para empezar rápidamente, puedes usar nuestras políticas AWS gestionadas. Estas políticas cubren casos de uso comunes y están disponibles en tu AWS cuenta. Para obtener más información sobre las políticas AWS administradas, consulte las políticas AWS administradas en la Guía del usuario de IAM.

AWS los servicios mantienen y AWS actualizan las políticas gestionadas. No puede cambiar los permisos en las políticas AWS gestionadas. En ocasiones, los servicios agregan permisos adicionales a una política administrada por AWS para admitir características nuevas. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Es más probable que los servicios actualicen una política administrada por AWS cuando se lanza una nueva característica o cuando se ponen a disposición nuevas operaciones. Los servicios no eliminan los permisos de una política AWS administrada, por lo que las actualizaciones de la política no afectarán a los permisos existentes.

Además, AWS admite políticas administradas para funciones laborales que abarcan varios servicios. Por ejemplo, la política ReadOnlyAccess AWS gestionada proporciona acceso de solo lectura a todos los AWS servicios y recursos. Cuando un servicio lanza una nueva función, AWS agrega permisos de solo lectura para nuevas operaciones y recursos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte Políticas administradas de AWS para funciones de trabajo en la Guía del usuario de IAM.

Se dispone de una serie de políticas a través de la consola de IAM que puede utilizar para conceder acceso a Amazon Cognito:

- `AmazonCognitoPowerUser`: permisos para tener acceso a todos los aspectos de los grupos de identidades y de usuarios y con el fin de administrarlos. Para ver los permisos de esta política, consulte. AmazonCognitoPowerUser
- `AmazonCognitoReadOnly`: permisos de acceso de solo lectura a los grupos de identidades y de usuarios. Para ver los permisos de esta política, consulte AmazonCognitoReadOnly.
- `AmazonCognitoDeveloperAuthenticatedIdentities`: permisos para que el sistema de autenticación se integre en Amazon Cognito. Para ver los permisos de esta política, consulte AmazonCognitoDeveloperAuthenticatedIdentities.

El equipo de Amazon Cognito mantiene estas políticas, por lo que, aunque se agreguen nuevas API, sus usuarios seguirán teniendo el mismo nivel de acceso.

> **Note**
>
> Al crear un nuevo grupo de identidades, puede crear automáticamente nuevos roles para el acceso de usuarios autenticados e invitados. El administrador que crea el grupo de identidades con nuevos roles de IAM también debe tener permisos de IAM para crear roles.

Los grupos de identidades con acceso de invitados no autenticados aplican una política AWS administrada adicional, como política de sesión `AmazonCognitoUnAuthedIdentitiesSessionPolicy`, a los usuarios no autenticados. Esta política AWS administrada no tiene ningún uso administrativo previsto. En cambio, limita el alcance de los permisos que puede aplicar a los usuarios invitados en el flujo de autenticación mejorado de los grupos de identidades. Para obtener más información, consulte Roles de IAM.

Amazon Cognito actualiza las políticas gestionadas AWS

Consulte los detalles sobre las actualizaciones de las políticas AWS gestionadas de Amazon Cognito desde que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página Document history (Historial de documentos) de Amazon Cognito.

Cambio	Descripción	Fecha
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` : política nueva	Se agregó una política AWS administrada para reducir el alcance de los privilegios de los usuarios invitados en los grupos de identidades.	14 de julio de 2023

Cambio	Descripción	Fecha
`AmazonCognitoPowerUser` y `AmazonCognitoReadOnly` : actualizaciones de las políticas existentes	Se han añadido nuevos permisos para permitir a los usuarios avanzados ver y gestionar las asociaciones de ACL AWS WAF web con los grupos de usuarios de Amazon Cognito. Se han añadido nuevos permisos para permitir a los usuarios de solo lectura ver las asociaciones de ACL AWS WAF web con los grupos de usuarios de Amazon Cognito.	19 de julio de 2022
`AmazonCognitoPowerUser` : actualización de una política actual	Se agregó un nuevo permiso para que Amazon Cognito pueda llamar al Amazon Simple Notification Service de`PutIdentityPolicy` y a las operaciones de `ListConfigurationSets` . Este cambio permite a los grupos de usuarios de Amazon Cognito actualizar las políticas de autorización de envío de Amazon SES y aplicar conjuntos de configuración de Amazon SES cuando se configura el envío de correo electrónico en su grupo de usuarios.	17 de noviembre de 2021

Cambio	Descripción	Fecha
`AmazonCognitoPowerUser` : actualización de una política actual	Se agregó un nuevo permiso para que Amazon Cognito pueda llamar a la operación `GetSMSSandboxAccountStatus` de Amazon Simple Notification Service. Este cambio permite a los grupos de usuarios de Amazon Cognito decidir si es necesario salir del entorno de pruebas de Amazon Simple Notification Service para enviar mensajes a todos los usuarios finales a través de los grupos de usuarios.	1 de junio de 2021
Amazon Cognito comenzó el seguimiento de los cambios	Amazon Cognito comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	1 de marzo de 2021

Etiquetado de recursos de Amazon Cognito

Una tag (etiqueta) es una etiqueta de metadatos que usted oAWS asignan a un recurso de AWS. Cada etiqueta consta de una key (clave) y un value (valor). En el caso de etiquetas que usted asigna, debe definir la clave y el valor. Por ejemplo, puede definir la clave como `stage` y el valor de un recurso como `test`.

Las etiquetas le ayudan a hacer lo siguiente:

- Identificar y organizar sus recursos de AWS. Muchos servicios de AWS admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios. Esto le ayuda a indicar qué recursos están relacionados. Por ejemplo, podría asignar la misma etiqueta a un grupo de usuarios de Amazon Cognito que asigne a una tabla de Amazon DynamoDB.
- Realizar un seguimiento de los costos de AWS. Las etiquetas se activan en el panel de AWS Billing and Cost Management. AWS usa las etiquetas de asignación de costes para categorizar sus costes y enviarle un informe mensual de asignación de costes. Para obtener más información, consulte Uso de etiquetas de asignación de costes en la Guía del usuario de AWS Billing.
- Controle el acceso a sus recursos basándose en las etiquetas que se les asignan. El acceso se controla especificando las claves y los valores de etiqueta en las condiciones de una política (IAM) de AWS Identity and Access Management. Por ejemplo, podría permitir que un usuario actualice un grupo de usuarios solo si este grupo tiene una etiqueta `owner` con un valor del nombre de ese usuario. Para obtener más información, consulte Control del acceso mediante etiquetas en la Guía del usuario de IAM.

Puede utilizar la AWS Command Line Interface o la API de Amazon Cognito para agregar, editar o eliminar etiquetas tanto de grupos de usuarios como de identidades. También puede administrar las etiquetas para los grupos de usuarios utilizando la consola de Amazon Cognito.

Para obtener sugerencias acerca del uso de etiquetas, consulte la publicación sobre estrategias de etiquetado de AWS en el blog de Respuestas de AWS.

En las siguientes secciones, se ofrece más información sobre las etiquetas de Amazon Cognito.

Recursos admitidos en Amazon Cognito

Los siguientes recursos de Amazon Cognito admiten el etiquetado:

- Grupos de usuarios
- Grupos de identidades

Restricciones de las etiquetas

Las siguientes restricciones se aplican a las etiquetas en los recursos de Amazon Cognito:

- Cantidad máxima de etiquetas que puede asignar a un recurso: 50
- Longitud máxima de la clave: 128 caracteres Unicode.
- Longitud máxima del valor: 256 caracteres Unicode.
- Caracteres válidos para claves y valores: a-z, A-Z, 0-9, espacio y los siguientes caracteres: _ . : / = + - @
- Las claves y los valores distinguen entre mayúsculas y minúsculas
- No utilice `aws:` como prefijo para claves, ya que está reservado para AWS.

Administración de etiquetas mediante la consola de Amazon Cognito

Puede utilizar la consola de Amazon Cognito para administrar las etiquetas que se asignan a los grupos de usuarios.

Para añadir etiquetas a un grupo de usuarios

1. Vaya a la consola de Amazon Cognito. Si se le solicita, escriba sus credenciales de AWS.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios.
4. Elija la pestaña User pool properties (Propiedades del grupo de usuarios) y localice las Tags (Etiquetas).
5. ElijaAdd tag (Agregar etiqueta) para agregar su primera etiqueta. Si ya ha asignado etiquetas a este grupo de usuarios, en Manage tags (Administrar etiquetas), elija Add another (Agregar otra).
6. Especifique los valores de Tag Key (Clave de etiqueta) y Tag Value (Valor de etiqueta).
7. Para cada etiqueta adicional que desee añadir, elija Add another (Agregar otra).
8. Cuando haya terminado de añadir etiquetas, elija Save changes (Guardar cambios).

En la página Manage tags (Administrar etiquetas), también puede editar las claves y los valores de las etiquetas existentes. Para quitar una etiqueta, elija Remove (Quitar).

Ejemplos del AWS CLI

La AWS CLI proporciona comandos que puede utilizar para administrar las etiquetas que asigna a los grupos de usuarios y de identidades de Amazon Cognito.

Asignación de etiquetas

Utilice los siguientes comandos para asignar etiquetas a sus grupos de usuarios y de identidades existentes.

Example **`tag-resource`** Comando para grupos de usuarios

Asigne etiquetas a un grupo de usuarios utilizando `tag-resource` en el conjunto de comandos de `cognito-idp`:

```
$ aws cognito-idp tag-resource \
> --resource-arn user-pool-arn \
> --tags Stage=Test
```

Este comando incluye los siguientes parámetros:

- `resource-arn`: el nombre de recurso de Amazon (ARN) del grupo de usuarios al que va a aplicar las etiquetas. Para buscar el ARN, elija el grupo de usuarios de la consola de Amazon Cognito y consulte el valor de Pool ARN (ARN de grupo) en la pestaña General settings (Configuración general).
- `tags` – Los pares de clave-valor de las etiquetas, en el formato *`key`*=*`value`*.

Para asignar varias etiquetas a la vez, especifíquelas en una lista separada por comas:

```
$ aws cognito-idp tag-resource \
> --resource-arn user-pool-arn \
> --tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Example **`tag-resource`** Comando para grupos de identidades

Asigne etiquetas a un grupo de identidades utilizando `tag-resource` en el conjunto de comandos de `cognito-identity`:

```
$ aws cognito-identity tag-resource \
> --resource-arn identity-pool-arn \
> --tags Stage=Test
```

Este comando incluye los siguientes parámetros:

- `resource-arn`: el nombre de recurso de Amazon (ARN) del grupo de identidades al que va a aplicar las etiquetas. Para buscar el ARN, elija el grupo de identidades en la consola de Amazon Cognito y elija Edit identity pool (Editar grupo de identidades). A continuación, en Identity pool ID (ID de grupo de identidades), elija Show ARN (Mostrar ARN).
- `tags` – Los pares de clave-valor de las etiquetas, en el formato *key*=*value*.

Para asignar varias etiquetas a la vez, especifíquelas en una lista separada por comas:

```
$ aws cognito-identity tag-resource \
> --resource-arn identity-pool-arn \
> --tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Visualización de etiquetas

Utilice los siguientes comandos para ver las etiquetas que ha asignado a sus grupos de usuarios y de identidades.

Example **`list-tags-for-resource`** Comando para grupos de usuarios

Consulte las etiquetas que están asignadas a un grupo de usuarios utilizando `list-tags-for-resource` en el conjunto de comandos de `cognito-idp`:

```
$ aws cognito-idp list-tags-for-resource --resource-arn user-pool-arn
```

Example **`list-tags-for-resource`** Comando para grupos de identidades

Consulte las etiquetas que están asignadas a un grupo de identidades utilizando `list-tags-for-resource` en el conjunto de comandos de `cognito-identity`:

```
$ aws cognito-identity list-tags-for-resource --resource-arn identity-pool-arn
```

Eliminación de etiquetas

Utilice los siguientes comandos para eliminar etiquetas de sus grupos de usuarios y de identidades.

Example **`untag-resource`** Comando para grupos de usuarios

Elimine etiquetas de un grupo de usuarios utilizando `untag-resource` en el conjunto de comandos de `cognito-idp`:

```
$ aws cognito-idp untag-resource \
> --resource-arn user-pool-arn \
> --tag-keys Stage CostCenter Owner
```

Para el parámetro de `--tag-keys`, especifique una o varias claves de etiquetas. No incluya los valores de etiqueta. Claves separadas con espacios.

Example **`untag-resource`** Comando para grupos de identidades

Elimine etiquetas de un grupo de identidades utilizando `untag-resource` en el conjunto de comandos de `cognito-identity`:

```
$ aws cognito-identity untag-resource \
> --resource-arn identity-pool-arn \
> --tag-keys Stage CostCenter Owner
```

Para el parámetro de `--tag-keys`, especifique una o varias claves de etiquetas. No incluya los valores de etiqueta.

> **Important**
>
> Después de eliminar un grupo de usuarios o de identidades, las etiquetas relacionadas con el grupo eliminado todavía pueden aparecer en la consola o en las llamadas a la API hasta treinta días después de su eliminación.

Aplicación de etiquetas al crear recursos

Utilice los siguientes comandos para asignar etiquetas en el momento de crear un grupo de usuarios o un grupo de identidades.

Example **`create-user-pool`** Comando con etiquetas

Cuando se crea un grupo de usuarios mediante el comando `create-user-pool`, es posible especificar etiquetas con el parámetro `--user-pool-tags`:

```
$ aws cognito-idp create-user-pool \
> --pool-name user-pool-name \
> --user-pool-tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Los pares clave-valor para las etiquetas deben tener el formato *`key`*=*`value`*. Si va a agregar varias etiquetas, especifíquelas en una lista separada por comas.

Example **`create-identity-pool`** Comando con etiquetas

Cuando se crea un grupo de identidades mediante el comando `create-identity-pool`, es posible especificar etiquetas con el parámetro `--identity-pool-tags`:

```
$ aws cognito-identity create-identity-pool \
> --identity-pool-name identity-pool-name \
> --allow-unauthenticated-identities \
> --identity-pool-tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Los pares clave-valor para las etiquetas deben tener el formato *`key`*=*`value`*. Si va a agregar varias etiquetas, especifíquelas en una lista separada por comas.

Administración de etiquetas mediante la API de Amazon Cognito

Puede utilizar las siguientes acciones de la API de Amazon Cognito para administrar las etiquetas de sus grupos de usuarios y de identidades.

Acciones de la API para las etiquetas de grupos de usuarios

Utilice las siguientes acciones de la API para asignar, ver y eliminar etiquetas de los grupos de usuarios.

- TagResource
- ListTagsForResource
- UntagResource
- CreateUserPool

Acciones de la API para las etiquetas de grupos de identidades

Utilice las siguientes acciones de la API para asignar, ver y eliminar etiquetas de los grupos de identidades.

- TagResource
- ListTagsForResource
- UntagResource
- CreateIdentityPool

Cuotas en Amazon Cognito

Amazon Cognito tiene cuotas predeterminadas, anteriormente conocidas como límites, para obtener la cantidad máxima de operaciones que puede realizar en su cuenta. Amazon Cognito también tiene cuotas para el número máximo y tamaño de los recursos de Amazon Cognito.

Cada cuota de Amazon Cognito representa un volumen máximo de solicitudes de una Región de AWS en una. Cuenta de AWS Por ejemplo, sus aplicaciones pueden realizar solicitudes de API hasta la tasa de cuota predeterminada (RPS) para operaciones `UserAuthentication` en todos sus grupos de usuarios en Este de EE. UU. (Norte de Virginia). Sus aplicaciones en Asia Pacífico (Tokio) pueden generar el mismo volumen de solicitudes para todos los grupos de usuarios de su propia región. AWS solo puede conceder una solicitud de aumento de cuota en una región a la vez. Un aumento correcto de la cuota en Este de EE. UU. (Norte de Virginia) no tiene ningún efecto sobre su tasa máxima de solicitudes en Asia Pacífico (Tokio).

Temas

Descripción de las cuotas de la tasa de solicitudes de la API

Categorización de cuotas

Amazon Cognito impone una tasa máxima de solicitudes para las operaciones de API. Para obtener más información sobre las operaciones de la API que Amazon Cognito pone a su disposición, consulte Referencias de la API y el punto de conexión de Amazon Cognito Para los grupos de usuarios, estas operaciones se agrupan en categorías de casos de uso común como `UserAuthentication` o `UserCreation`. Para obtener una lista de las operaciones de API de grupos de usuarios por categoría, consulte. Categorías y cuotas de las tasas de solicitudes de operaciones de API de grupos de usuarios de Amazon Cognito

En la consola de Service Quotas, puede realizar un seguimiento del uso de la cuota por grupos de usuarios de categoría y grupos de identidades. Si la tasa de solicitudes de sus grupos de usuarios de Amazon Cognito supera o supera una cuota, puede adquirir capacidad adicional. Puede realizar un seguimiento del uso de la cuota de su grupo de usuarios por categoría y comprar los aumentos de cuota en la consola Service Quotas.

Las cuotas de operación se definen como el número máximo de solicitudes por segundo (RPS) para todas las operaciones dentro de una categoría. El servicio de grupos de usuarios de Amazon Cognito aplica cuotas a todas las operaciones de cada categoría. Por ejemplo, la categoría `UserCreation` incluye cuatro operaciones: `SignUp`, `ConfirmSignUp`, `AdminCreateUser` y `AdminConfirmSignUp`. Se asigna con una cuota combinada de 50 RPS. Si se realizan varias operaciones al mismo tiempo, cada operación dentro de esta categoría puede llamar hasta 50 RPS por separado o en conjunto.

> **Note**
>
> Las cuotas de categoría solo se aplican a los grupos de usuarios. Amazon Cognito aplica cada cuota de grupo de identidades a una sola operación. Para las cuotas de solicitudes por categoría y por operación, AWS mide la tasa agregada de todas las solicitudes de todos los grupos de usuarios o grupos de identidades de su Cuenta de AWS región.

Operaciones de API de grupo de usuarios de Amazon Cognito con control de tasas de solicitud especiales

Las cuotas de operaciones se miden y se aplican en función de la cantidad total de solicitudes combinadas en el nivel de la categoría, excepto las operaciones `AdminRespondToAuthChallenge` y `RespondToAuthChallenge`, en la que se aplican reglas especiales de control.

La `UserAuthentication` categoría incluye cuatro operaciones en la API de grupos de usuarios de Amazon Cognito:`AdminInitiateAuth`, `InitiateAuthAdminRespondToAuthChallenge`, y. `RespondToAuthChallenge` Además, la autenticación de usuarios en la interfaz de usuario alojada contribuye a esta cuota. Las operaciones `InitiateAuth` y `AdminInitiateAuth` se miden y se aplican por cuota de categoría. Las operaciones coincidentes `RespondToAuthChallenge` y `AdminRespondToAuthChallenge` están sujetas a una cuota distinta que triplica el límite de la categoría `UserAuthentication`. Esta cuota elevada se adapta a varios desafíos de autenticación configurados en sus aplicaciones. La cuota es suficiente para cubrir la gran mayoría de los casos

de uso. Una vez que tu aplicación dé hasta tres respuestas a los desafíos de autenticación, las solicitudes adicionales se tendrán en cuenta para la cuota de `UserAuthentication` categorías. La autenticación multifactor (MFA), la autenticación de dispositivos y la autenticación personalizada son ejemplos de solicitudes de desafío que puede diseñar en su grupo de usuarios.

Por ejemplo, si tu cuota para la `UserAuthentication` categoría es de 80 RPS, puedes llamar `RespondToAuthChallenge` o `AdminRespondToAuthChallenge` a una tarifa de hasta 240 RPS (3 * 80 RPS). Si su grupo de usuarios solicita cuatro rondas de impugnación por autenticación y 70 usuarios inician sesión por segundo, el total `RespondToAuthChallenge` es de 280 RPS (70 x 4), es decir, 40 RPS por encima de la cuota. Los 40 RPS adicionales se agregan a 70 llamadas `InitiateAuth`, lo que hace un uso total de categoría `UserAuthentication` de 110 RPS (40 + 70). Como este valor supera la cuota de categoría establecida en 80 RPS por 30 RPS, Amazon Cognito limita las solicitudes de su aplicación.

Monthly active users (Usuarios activos mensuales)

Cuando Amazon Cognito calcula la facturación del grupo de usuarios, le cobra una tarifa por cada usuario activo mensual (MAU). Tenga en cuenta el recuento de MAU actual y previsto al planificar las solicitudes de aumento de cuota. Un usuario se cuenta como MAU si, dentro de un mes natural, hay una operación de identidad relacionada con ese usuario. Entre las actividades que activan a un usuario se incluyen las siguientes.

- Inscripción y creación administrativa de un usuario
- Sign-in (Inicio de sesión)
- Cerrar sesión
- Confirmación de la cuenta de usuario o verificación de atributos
- Restablecimiento de la contraseña
- Cambiar los atributos de usuario, la pertenencia a grupos o las preferencias de MFA
- Consultar los atributos detallados de un usuario
- Activación, desactivación o eliminación de usuarios

> **Note**
>
> La categoría Consulta los atributos detallados de un usuario incluye la operación de la API AdminGetUser, pero no ListUsers. Una user-by-user consulta detallada en un grupo de usuarios grande puede tener un impacto significativo en su AWS factura. Para evitar cargos

excesivos, recopile los datos de los usuarios `ListUsers` o almacene la información de los usuarios en una base de datos externa.

Administración de las cuotas de la tasa de solicitudes de la API

Identificación de los requisitos de cuota

Important

Si aumentas las cuotas de Amazon Cognito para categorías como`UserAuthentication`, o `UserCreationAccountRecovery`, puede que tengas que aumentar las cuotas para otras. Servicios de AWS Por ejemplo, los mensajes que Amazon Cognito envía con Amazon Simple Notification Service (Amazon SNS) o Amazon Simple Email Service (Amazon SES) pueden fallar si las cuotas de la tasa de solicitudes son insuficientes en esos servicios.

Para calcular los requisitos de cuota, determine cuántos usuarios activos interactuarán con su aplicación durante un período específico. Por ejemplo, si espera que en la aplicación inicien sesión un promedio de un millón de usuarios activos durante un periodo de ocho horas, tiene que poder autenticar un promedio de 35 usuarios por segundo.

Además, si supone que la sesión promedio de usuario es de dos horas y ha configurado los tokens para que caduquen después de una hora, cada usuario debe actualizar sus tokens una vez durante la sesión. La cuota promedio obligatoria de la categoría `UserAuthentication` para admitir esta carga es de 70 RPS.

Si asume una peak-to-average proporción de 3:1 al tener en cuenta la variación de la frecuencia de inicio de sesión de los usuarios durante el período de ocho horas, necesitará la cuota deseada `UserAuthentication` de 200 RPS.

Note

Si llama a varias operaciones para cada acción del usuario, debe resumir las tasas de llamada de operación individuales según el nivel de la categoría.

Optimice las tasas de solicitud para cumplir con los límites de cuota

Dado que aumentar los límites de las tasas de las API supone un coste adicional para tu AWS factura, considera la posibilidad de realizar ajustes en tu modelo de uso antes de solicitar un aumento de cuota. Los siguientes son algunos ejemplos de la arquitectura de la aplicación que optimiza las tasas de solicitudes.

Vuelva a intentarlo después de un período de retardo

Puede detectar errores en cada llamada a la API y, a continuación, volver a intentarlo después de un periodo de retardo. Puede ajustar el algoritmo de retardo de acuerdo con las necesidades del negocio y la carga. Los SDK de Amazon tienen una lógica de reintento incorporada. Para obtener más información, consulte Herramientas sobre AWS las que construir.

Uso de una base de datos externa para atributos que se actualizan con frecuencia

Si la aplicación exige varias llamadas a un grupo de usuarios para leer o escribir atributos personalizados, utilice almacenamiento externo. Puede utilizar su base de datos preferida para almacenar atributos personalizados o utilizar una capa de memoria caché para cargar un perfil de usuario durante el inicio de sesión. Puede referenciar este perfil desde la memoria caché cuando sea necesario, en lugar de volver a cargar el perfil de usuario desde un grupo de usuarios.

Valide los tokens web JSON (JWT) en el lado del cliente

Las aplicaciones deben validar tokens JWT antes de confiar en ellos. Puede verificar la firma y la validez de los tokens en el lado del cliente sin enviar solicitudes de API a un grupo de usuarios. Después de validar el token, puede confiar en las notificaciones del token y usarlas, en lugar de hacer más llamadas a la API de `getUser`. Para obtener más información, consulte Verificación de un JSON Web Token (JWT).

Limite el tráfico a la aplicación web con una sala de espera

Si espera tráfico de un gran número de usuarios que inicien sesión durante un evento de duración limitada, como rendir un examen o asistir a un evento en vivo, puede optimizar el tráfico de solicitudes con mecanismos de autolimitación. Por ejemplo, puede configurar una sala de espera en la que los usuarios puedan quedarse hasta que haya una sesión disponible, lo que le permite procesar solicitudes cuando tiene capacidad disponible. Consulte la Solución de sala de espera virtual de AWS para obtener una arquitectura de referencia de una sala de espera.

Almacene en caché los JWT

Reutilice los tokens de acceso hasta que caduquen. Para ver un ejemplo de marco con almacenamiento en caché de tokens en una API Gateway, consulteAlmacenamiento en caché de tokens. En lugar de generar solicitudes de API para consultar la información del usuario, almacene en caché los identificadores de identidad hasta que caduquen y leer los atributos de usuario de la caché.

Para obtener más información sobre cómo trabajar con las tasas de solicitudes de API AWS, consulte Administrar y monitorear la limitación de las API en sus cargas de trabajo. Para obtener información sobre cómo optimizar las operaciones de Amazon Cognito que añaden costes a su AWS factura, consulte. Administración de los costos de

Seguimiento del uso de cuotas

Amazon Cognito genera `CallCount` `ThrottleCount` métricas en Amazon CloudWatch para cada categoría de operación de la API a nivel de cuenta. Puede utilizar `CallCount` para hacer seguimiento de la cantidad total de llamadas que realizan los clientes en relación con una categoría. Puede utilizar `ThrottleCount` para hacer seguimiento de la cantidad total de llamadas con limitación controlada en relación con una categoría. Puede utilizar las métricas `CallCount` y `ThrottleCount` con la estadística `Sum` para contar la cantidad total de llamadas en una categoría. Para obtener más información, consulte las métricas CloudWatch de uso.

Al monitorear las cuotas de servicio, la utilización es el porcentaje de una cuota de servicio en uso. Por ejemplo, si el valor de la cuota es de 200 recursos y hay 150 en uso, la utilización es del 75 %. El uso es la cantidad de recursos u operaciones que se utilizan para una cuota de servicio.

Realizar un seguimiento del uso a través de CloudWatch métricas

Puede realizar un seguimiento de las métricas de uso de los grupos de usuarios de Amazon Cognito y recopilarlas con ellas. CloudWatch El CloudWatch panel muestra las métricas de todos los Servicio de AWS dispositivos que utiliza. Con CloudWatch él, puede crear alarmas métricas para notificarle o cambiar un recurso específico que esté monitoreando. Para obtener más información sobre CloudWatch las métricas, consulta Realizar un seguimiento de tus métricas CloudWatch de uso.

Seguimiento del uso con métricas de Service Quotas

Los grupos de usuarios de Amazon Cognito están integrados con Service Quotas, una interfaz de consola para mostrar y gestionar el uso de las cuotas de servicio. En la consola Service Quotas,

puede buscar el valor de una cuota específica, ver la información de supervisión, solicitar un aumento de cuota o configurar CloudWatch alarmas. Una vez que su cuenta haya estado activa durante un tiempo, podrá ver un gráfico de la utilización de los recursos.

La columna Valor de cuota a nivel de cuenta aplicado de la consola de Service Quotas para los grupos de usuarios de Amazon Cognito y los grupos de identidades de Amazon Cognito muestra su cuota actual. La columna Utilización muestra la tasa actual de uso de la cuota. Las cuotas ajustables de los grupos de usuarios requests-per-second (RPS) de Amazon Cognito muestran su uso actual. La consola Service Quotas también puede navegar hasta CloudWatch las métricas para ver más de cerca una métrica de cuota seleccionada. Para obtener más información sobre cómo ver las cuotas en la consola de Service Quotas, consulte Visualización de Service Quotas.

Realiza un seguimiento de los usuarios activos (MAU) mensuales

El número de usuarios activos mensuales (MAU) de su grupo de usuarios aporta datos importantes a la hora de planificar el aumento de las cuotas de solicitudes. Puede comparar las tasas de solicitudes de la API con la cantidad de usuarios que tuvo activos en un período de tiempo determinado. Con esta información, puede calcular cómo afectará el aumento de los usuarios activos de sus aplicaciones a las cuotas de su modelo de uso. Por ejemplo, imagine que sus aplicaciones combinadas en el oeste de EE. UU. (Oregón) dan como resultado 2 millones de usuarios activos en un mes y que su `UserAuthentication` categoría recibe errores de limitación ocasionales con la cuota predeterminada de 120 solicitudes por segundo (RPS). El mes anterior, antes del éxito de su campaña publicitaria, tenía 1 millón de MAU y sus aplicaciones nunca superaron los 80 RPS. Si prevé un aumento similar como consecuencia de un nuevo anuncio de televisión, podría adquirir 40 RPS adicionales para dar cabida al siguiente millón de usuarios con una cuota ajustada de 160 RPS.

Para revisar tus MAU

Accede a la AWS Billing consola y consulta una factura reciente. En Cargos por servicio, puedes filtrar Cognito para ver un desglose de tus MAU para ese período de facturación.

Solicitud de aumento de cuota

Amazon Cognito tiene una cuota para el número máximo de operaciones por segundo que puede realizar en sus grupos de usuarios y grupos de identidades de cada uno. Región de AWS Puede adquirir un aumento de las cuotas ajustables de las tasas de solicitud de la API de los grupos de usuarios de Amazon Cognito. Comprueba tu cuota actual y compra un aumento en la consola de Service Quotas o con las operaciones de la API Service Quotas `ListAWSDefaultServiceQuotas` y`RequestServiceQuotaIncrease`.

- Para comprar un aumento de cuota mediante la consola de Service Quotas, consulte Solicitar un aumento de cuota de API en la Guía del usuario de Service Quotas.
- AWS tiene como objetivo completar las solicitudes de aumento de cuota en un plazo de 10 días. Sin embargo, varias consideraciones pueden provocar que el tiempo de procesamiento de la solicitud supere los 10 días. Algunas solicitudes, por ejemplo, pueden requerir que Amazon Cognito aprovisione capacidad de hardware adicional, y los aumentos estacionales en los volúmenes de solicitudes pueden provocar retrasos.
- Si la cuota no está disponible en Service Quotas, utilice el formulario de aumento del límite de servicio.

> **Important**
>
> Solo se pueden aumentar las cuotas ajustables. Debe adquirir una mayor capacidad de cuota. Para obtener más información sobre los precios de las cuotas, consulte los precios de Amazon Cognito.

Categorías y cuotas de las tasas de solicitudes de operaciones de API de grupos de usuarios de Amazon Cognito

Como Amazon Cognito tiene clases superpuestas de operaciones de la API con diferentes modelos de autorización, cada operación pertenece a una categoría. Cada categoría tiene su propia cuota agrupada para todas las operaciones de API de los miembros, a través de todos los grupos de usuarios en una Región de AWS en su cuenta. Solo puede solicitar un aumento de las cuotas de categorías ajustables. Para obtener más información, consulte Solicitud de aumento de cuota. Los ajustes de cuotas se aplican a los grupos de usuarios de su cuenta en una única región. Amazon Cognito restringe las operaciones en algunas categorías[3] a cinco solicitudes por segundo (RPS), por grupo de usuarios. La cuota predeterminada (RPS) también se aplica a todos los grupos de usuarios de un. Cuenta de AWS

> **Note**
>
> La cuota de cada categoría se mide en usuarios activos mensuales (MAU). Las Cuentas de AWS con menos de dos millones de MAU pueden operar según la cuota predeterminada. Si tiene menos de un millón de MAU y Amazon Cognito está limitando las solicitudes, considere

la posibilidad de optimizar su aplicación. Para obtener más información, consulte Optimice las tasas de solicitud para cumplir con los límites de cuota.

Las cuotas de operaciones por categoría se aplican a todos los usuarios de todos los grupos de usuarios en una Región de AWS. Amazon Cognito también mantiene una cuota para el número de solicitudes que su aplicación puede generar para un usuario. Debe limitar las solicitudes de API por usuario como se muestra en la siguiente tabla.

Cuotas de tasa de solicitudes por usuario de los grupos de usuarios de Amazon Cognito

Operación	Operaciones por usuario por segundo
Leer perfil de usuario Ejemplos: `GetUser`, `GetDevice`	10
Escribir perfil de usuario Ejemplos: `UpdateUserAttributes` , `SetUserSettings`	10

Debe limitar las solicitudes de API por categoría como se muestra en la siguiente tabla.

Cuotas de tasa de solicitudes por categoría de los grupos de usuarios de Amazon Cognito

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserAuthentication` • InitiateAuth • Actualización del token con `InitiateAuth` o Punto de conexión de token	Operaciones con las que se autentifica a un usuario (inicia sesión). Estas operaciones están sujetas a Operaciones de API de grupo de usuarios	120	Sí

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
• RespondToAuthChallenge[1] • AdminInitiateAuth • AdminRespondToAuthChallenge[1] • Inicio de sesión en la IU alojada y MFA en concesiones implícitas o de código de autorización[2]	de Amazon Cognito con control de tasas de solicitud especiales .		
`UserCreation` • SignUp • ConfirmSignUp • AdminCreateUser • AdminConfirmSignUp	Operaciones que crean o confirman un usuario local de Amazon Cognito. Este es un usuario que los grupos de usuarios de Amazon Cognito crean y verifican directamente.	50	Sí

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserFederation` Operaciones con las que se federan (autentican) usuarios con un proveedor de identidad de terceros en sus grupos de usuarios de Amazon Cognito.	Operaciones que envían una respuesta del IdP a un punto de conexión de federación de grupo de usuarios. Las operaciones del OIDC o de los proveedores sociales que generan un token de IdP y todas las solicitudes de SAML contribuyen a esta cuota.	25	Sí

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserAccountRecovery` • ChangePassword • ConfirmForgotPassword • ForgotPassword • AdminResetUserPassword • AdminSetUserPassword • RespondToAuthChallenge[1] • AdminRespondToAuthChallenge[1] • Restablecimiento de contraseña de interfaz de usuario alojada	Operaciones con las que se recupera la cuenta de un usuario, o se cambia o actualiza la contraseña de un usuario.	30	No
`UserRead` • AdminGetUser • GetUser	Operaciones con las que se recupera un usuario de los grupos de usuarios	120	Sí

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserUpdate` • AdminAddUserToGroup • AdminDeleteUserAttributes • AdminUpdateUserAttributes • AdminDeleteUser • AdminDisableUser • AdminEnableUser • AdminLinkProviderForUser • AdminDisableProviderForUser • VerifyUserAttribute • DeleteUser • DeleteUserAttributes • UpdateUserAttributes • AdminUserGlobalSignOut • GlobalSignOut • AdminRemoveUserFromGroup	Operaciones que se utilizan para administrar usuarios y atributos de usuario.	25	No
`UserToken` • RevokeToken	Operaciones para la administración de tokens	120	Sí

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserResourceRead` • AdminGetDevice • AdminListGroupsForUser • AdminListDevices • GetDevice • ListDevices • GetUserAttributeVerificationCode • ResendConfirmationCode • AdminListUserAuthEvents	Operaciones con las que se recupera la información de los recursos de usuario de Amazon Cognito, como un dispositivo recordado o una pertenencia a un grupo.	50	Sí

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserResourceUpdate` • AdminForgetDevice • AdminUpdateAuthEventFeedback • AdminSetUserPreferencia de MFA • AdminSetUserSettings • AdminUpdateDeviceStatus • UpdateDeviceStatus • UpdateAuthEventFeedback • ConfirmDevice • SetUserPreferencia de MFAP • SetUserSettings • VerifySoftwareToken • AssociateSoftwareToken • ForgetDevice	Operaciones con las que se actualiza la información de los recursos de usuario, como un dispositivo recordado o una pertenencia a un grupo.	25	No

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserList` • ListUsers • ListUsersInGroup	Operaciones con las que se devuelve una lista de usuarios	30	No
`UserPoolRead` • DescribeUserPool • ListUserPools	Operaciones con las que se leen los grupos de usuarios	15	No
`UserPoolUpdate` • CreateUserPool • UpdateUserPool • DeleteUserPool	Operaciones con las que se crean, actualizan o eliminan grupos de usuarios.	15	No

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserPoolResourceRead` • DescribeIdentityProvider • DescribeResourceServer • DescribeUserImportJob • DescribeUserPoolDomain • GetCSVHeader • GetGroup • GetSigningCertificate • GetIdentityProviderByIdentifier • GetUserPoolMfaConfig • ListGroups • ListIdentityProviders • ListResourceServers • ListTagsForResource • ListUserImportJobs • DescribeRiskConfiguration • GetUICustomization	Operaciones que recuperan información sobre recursos, como grupos o servidores de recursos, de un grupo de usuarios.[3]	20	No

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
`UserPoolResourceUpdate` • AddCustomAttributes • CreateGroup • CreateIdentityProvider • CreateResourceServer • CreateUserImportJob • CreateUserPoolDomain • DeleteGroup • DeleteIdentityProvider • DeleteResourceServer • DeleteUserPoolDomain • SetUserPoolMfaConfig • StartUserImportJob • StopUserImportJob • UpdateGroup • UpdateIdentityProvider • UpdateResourceServer	Operaciones con las que se modifican recursos, como grupos o servidores de recursos, de un grupo de usuarios.[3]	15	No

Categoría	Descripción	Cuota predeterminada (RPS)	Ajustable
• UpdateUserPoolDomain • SetRiskConfiguration • SetUICustomization • TagResource • UntagResource			
`UserPoolClientRead` • DescribeUserPoolClient • ListUserPoolClients	Operaciones con las que se recupera información sobre los clientes del grupo de usuarios.[3]	15	No
`UserPoolClientUpdate` • CreateUserPoolClient • DeleteUserPoolClient • UpdateUserPoolClient	Operaciones con las que se crean, actualizan y eliminan los clientes de un grupo de usuarios.[3]	15	No
`ClientAuthentication` Solicitudes de tipo de concesión `client_credentials` al punto de conexión del token.	Operaciones que generan credenciales para utilizarlas en la autorización de solicitudes machine-to-machine	150	No

[1] A `RespondToAuthChallenge` o una `AdminRespondToAuthChallenge` respuesta con una o más se `ChallengeName` tienen en `NEW_PASSWORD_REQUIRED` cuenta para la `UserAccountRecovery` categoría. Todas las demás respuestas al desafío cuentan para la `UserAuthentication` categoría.

[2] Cada operación de la interfaz de usuario alojada durante el inicio de sesión contribuye con una solicitud a la cuota. Por ejemplo, un usuario que inicia sesión y proporciona un código MFA aporta dos solicitudes. El canje de fichas en las concesiones con códigos de autorización está sujeto a una asignación de cuota adicional igual a la cuota de la categoría. `UserAuthentication`

[3] Cualquier operación individual de esta categoría tiene una restricción que impide que la operación se cancele a una velocidad superior a 5 RPS para un solo grupo de usuarios.

Cuotas de las tasas de solicitudes de operaciones de API de grupos de identidades de Amazon Cognito (identidades federadas)

Operación	Descripción	Cuota predeterminada (RPS)[1]	Ajustable	Idoneidad para el aumento de cuota
`GetId`	Recuperación de un ID de identidad de un grupo de identidades.	25	Sí	Contáctese con el equipo de cuentas.
`GetOpenIdToken`	Recuperación de un token OpenID de un grupo de identidades del flujo de trabajo clásico.	200	Sí	Contáctese con el equipo de cuentas.
`GetCredentialsForIdentity`	Recupere AWS las credenciales de un grupo de identidades en el	200	Sí	Contáctese con el equipo de cuentas.

Operación	Descripción	Cuota predeterminada (RPS)[1]	Ajustable	Idoneidad para el aumento de cuota
	flujo de trabajo mejorado.			
`GetOpenIdTokenForDeveloperIdentity`	Recuperación de un token OpenID de un grupo de identidades del flujo de trabajo de desarrolladores.	50	Sí	Contáctese con el equipo de cuentas.
`ListIdentities`	Recupere una lista de identificadores de identidad en un grupo de identidades.	5	Sí	Contáctese con el equipo de cuentas.
`DeleteIdentities`	Elimine una o más identidades registradas de un grupo de identidades.	10	Sí	Contáctese con el equipo de cuentas.
`TagResource`	Aplique una etiqueta a un grupo de identidades.	5	Sí	Contáctese con el equipo de cuentas.
`UntagResource`	Elimine una etiqueta de un grupo de identidades.	5	Sí	Contáctese con el equipo de cuentas.

Operación	Descripción	Cuota predeterminada (RPS)[1]	Ajustable	Idoneidad para el aumento de cuota
`ListTagsForResource`	Muestre una lista de las etiquetas aplicadas a un grupo de identidades.	10	Sí	Contáctese con el equipo de cuentas.

[1] La cuota predeterminada es la cuota mínima de solicitudes para los grupos de identidades de cualquiera Región de AWS de sus países Cuenta de AWS. Es posible que su cuota de RPS sea mayor en algunas regiones.

Cuotas sobre el número y el tamaño de los recursos

Las cuotas de recursos son el número o tamaño máximo de recursos, campos de entrada, duración de tiempo y otras características diversas en Amazon Cognito.

Puede solicitar un ajuste de algunas cuotas de recursos en la consola de Service Quotas o desde un formulario de aumento del límite de servicio. Para solicitar una cuota mediante la consola Service Quotas, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas. Si la cuota no está disponible en Service Quotas, utilice el formulario de aumento del límite de servicio.

> **Note**
>
> Las cuotas de recursos a Cuenta de AWS nivel, como los grupos de usuarios por región, se aplican a los recursos de Amazon Cognito en cada región. Región de AWS Por ejemplo, puede tener 1000 grupos de usuarios en Este de EE. UU. (Norte de Virginia) y otros 1000 en Europa (Estocolmo).

En las tablas siguientes se indican las cuotas de recursos predeterminadas y si son ajustables.

Cuotas de recursos de grupos de usuarios de Amazon Cognito

Recurso	Cuota	Ajustable	Cuota máxima
Clientes de aplicaciones por grupo de usuarios	1 000	Sí	10 000
Grupos de usuarios por región	1 000	Sí	10 000
Proveedores de identidad por grupo de usuarios	300	Sí	1 000
Servidores de recursos por grupo de usuarios	25	Sí	300
Usuarios por grupo de usuarios	40 000 000	Sí	Contáctese con el equipo de cuentas.
Cambios totales combinados en el desencadenador de Lambda previo a la generación del token [1]	5 000	Sí	Contáctese con el equipo de cuentas.
Atributos personalizados por grupo de usuarios	50	No	N/A
Caracteres por atributo	2048 bytes	No	N/A
Caracteres en el nombre del atributo personalizado	20	No	N/A

Recurso	Cuota	Ajustable	Cuota máxima
Caracteres de contraseña mínimos requeridos en la política de contraseñas	6–99	No	N/A
Mensajes de correo electrónico enviados diariamente por Cuenta de AWS[2]	50	No	N/A
Caracteres en el asunto del correo electrónico	140	No	N/A
Caracteres en el mensaje de correo electrónico	20 000	No	N/A
Caracteres en el mensaje de verificación de SMS	140	No	N/A
Caracteres en la contraseña	256	No	N/A
Caracteres en el nombre del proveedor de identidad	32	No	N/A
Identificadores por proveedor de identidad	50	No	N/A
Identidades vinculadas a un usuario	5	No	N/A

Recurso	Cuota	Ajustable	Cuota máxima
Devolución de llamada por cliente de aplicación	100	No	N/A
URL de cierre de sesión por cliente de aplicación	100	No	N/A
Alcances por servidor de recursos	100	No	N/A
Alcances por cliente de aplicación	50	No	N/A
Dominios personalizados por cuenta	4	No	N/A
Grupos a los que puede pertenecer cada usuario	100	No	N/A
Grupos por grupo de usuarios	10 000	No	N/A

[1] Es posible que esta cuota se encuentre en los tokens de Desencadenador de Lambda anterior a la generación del token. El número de reclamaciones existentes y agregadas más los alcances de los tokens de acceso e identidad deben sumar un número inferior o igual a esta cuota. Las reclamaciones y los alcances suprimidos no contribuyen a esta cuota.

[2] Esta cuota solo se aplica si utiliza la característica de correo electrónico predeterminada para un grupo de usuarios de Amazon Cognito. Para un mayor volumen de entrega de correo electrónico, configure el grupo de usuarios para usar la configuración de correo electrónico de Amazon SES. Para obtener más información, consulte Configuración de correo electrónico para grupos de usuarios de Amazon Cognito.

Parámetros de validación de sesión de grupos de usuarios de Amazon Cognito

Token	Cuota
Token de ID	5 minutos – 1 día
Token de actualización	1 hora – 3650 días
Token de acceso	5 minutos – 1 día
Cookie de sesión de interfaz de usuario alojada	1 hora
Token de sesión de autenticación	De 3 minutos a 15 minutos

Cuotas de recursos de seguridad de código de grupos de usuarios de Amazon Cognito (no ajustables)

Recurso	Cuota
Periodo de validez de código de confirmación de registro	24 horas
Periodo de validez del código de verificación de atributo de usuario	24 horas
Periodo de validez del código de autenticación multifactor (MFA)	De 3 minutos a 15 minutos
Período de validez de código de contraseña olvidada	1 hora
Número máximo de solicitudes `ForgotPassword` y `ConfirmForgotPassword` solicitudes por usuario por hora [1]	5–20
Número máximo de solicitudes de `ResendConfirmationCode` por usuario por hora	5
Número máximo de solicitudes de `ConfirmSignUp` por usuario por hora	15

Recurso	Cuota
Número máximo de solicitudes de `ChangePassword` por usuario por hora	5
Número máximo de solicitudes de `GetUserAttributeVerificationCode` por usuario por hora	5
Número máximo de solicitudes de `VerifyUserAttribute` por usuario por hora	15

[1] Amazon Cognito evalúa los factores de riesgo de la solicitud para actualizar contraseñas y asigna una cuota vinculada al nivel de riesgo evaluado. Para obtener más información, consulte Comportamiento de contraseña olvidada.

Cuotas de recursos de trabajos de importación de grupos de usuarios de Amazon Cognito

Recurso	Cuota	Ajustable	Cuota máxima
Trabajos de importación de usuarios por grupo de usuarios	1 000	Sí	Contáctese con el equipo de cuentas.
Número máximo de caracteres por fila CSV de importación de usuarios	16,000	No	N/A
Tamaño máximo de archivo CSV	100 MB	No	N/A
Número máximo de usuarios por archivo CSV	500.000	No	N/A

Cuotas de recursos de grupos de identidades de Amazon Cognito (identidades federadas)

Recurso	Cuota	Ajustable	Cuota máxima
Grupos de identidades por cuenta	1 000	Sí	N/A
Proveedores de grupos de usuarios de Amazon Cognito por grupo de identidades	50	Sí	1 000
Longitud de caracteres de un nombre de grupo de identidades	128 bytes	No	N/A
Longitud de caracteres de un nombre de proveedor de inicio de sesión	2048 bytes	No	N/A
Identidades por grupo de identidades	Sin límite	No	N/A
Proveedores de identidad para los que se pueden especificar asignación de roles	10	No	N/A
Resultados de una sola llamada de lista o de búsqueda	60	No	N/A
Reglas de control de acceso basado en roles (RBAC)	25	No	N/A

Cuotas de recursos de Amazon Cognito Sync

Recurso	Cuota	Ajustable	Cuota máxima
Conjuntos de datos por identidad	20	Sí	Contáctese con el equipo de cuentas.
Registros por conjunto de datos	1 024	Sí	Contáctese con el equipo de cuentas.
Tamaño de un solo conjunto de datos	1 MB	Sí	Contáctese con el equipo de cuentas.
Caracteres en el nombre del conjunto de datos	128 bytes	No	N/A
Tiempo de espera para una publicación en masa tras una solicitud efectuada correctamente	24 horas	No	N/A

Referencias de la API y el punto de conexión de Amazon Cognito

Las siguientes referencias describen los puntos de conexión del servicio para cada característica de Amazon Cognito. Los grupos de usuarios de Amazon Cognito tienen las siguientes opciones: puntos de conexión de grupos de usuarios con un dominio de grupo de usuarios y la API de grupos de usuarios. Para obtener un desglose de las clases de operaciones de API con la API de los grupos de usuarios de Amazon Cognito, consulte Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios.

Para obtener una lista de los puntos de conexión de servicio para la API de los grupos de usuarios de Región de AWS, consulte Puntos de conexión de servicio en la Referencia general de AWS.

Temas

- Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada
- Referencia de la API de grupos de usuarios de Amazon Cognito
- Referencia de la API de grupos de identidades de Amazon Cognito (identidades federadas)
- Referencia de la API de sincronización de Amazon Cognito

Referencia de puntos de conexión de federación de grupo de usuarios e interfaz de usuario alojada

Amazon Cognito activa las páginas web públicas enumeradas aquí cuando asigna un dominio a su grupo de usuarios. Su dominio sirve de punto de acceso central para todos sus clientes de aplicación. Incluyen la IU alojada, donde los usuarios pueden registrarse e iniciar sesión (el Punto de conexión Login) y cerrar la sesión (el Punto de conexión Logout). Para obtener más información acerca de estos recursos, consulte Configuración y uso de la interfaz de usuario alojada y los puntos de conexión de federación de Amazon Cognito.

Estas páginas también incluyen los recursos web públicos que permiten a su grupo de usuarios comunicarse con proveedores de identidad SAML, OpenID Connect (OIDC) y OAuth 2.0 de terceros (). IdPs Para iniciar sesión con un usuario mediante un proveedor de identidades federado, los usuarios deben iniciar una solicitud a la interfaz de usuario alojada interactiva Punto de conexión

Login o Autorizar punto de conexión de OIDC. El punto de conexión de autorización redirige a los usuarios a la interfaz de usuario alojada o a la página de inicio de sesión de IdP.

La aplicación también puede permitir el inicio de sesión de usuarios locales con la API de grupos de usuarios de Amazon Cognito. Un usuario local existe exclusivamente en el directorio del grupo de usuarios sin federación a través de un IdP externo.

Además de la interfaz de usuario alojada y los puntos de enlace de federación, Amazon Cognito se integra con los SDK para Android JavaScript, iOS y más. Los SDK proporcionan herramientas para realizar operaciones de la API del grupo de usuarios con los puntos de conexión de servicio de la API de Amazon Cognito. Para obtener más información acerca de los puntos de conexión de servicio, consulte Puntos de conexión y cuotas de Amazon Cognito Identity.

> **Warning**
>
> No fije los certificados de seguridad de la capa de transporte (TLS) de la entidad final o intermedio para los dominios de Amazon Cognito. AWS administra todos los certificados de todos los puntos de enlace y dominios de prefijo de su grupo de usuarios. Las autoridades de certificación (CA) de la cadena de confianza que admite los certificados de Amazon Cognito se rotan y renuevan de forma dinámica. Al anclar la aplicación a un certificado intermedio o hoja, la aplicación puede fallar sin previo aviso al AWS rotar los certificados.
> En su lugar, fije su aplicación a todos los certificados raíz de Amazon disponibles. Para obtener más información, consulte las prácticas recomendadas y las recomendaciones en Asignación de certificados en la Guía del usuario de AWS Certificate Manager .

Temas

- Referencia de puntos de conexión de interfaz de usuario alojada
- Referencia de puntos de conexión de federación OAuth 2.0, OpenID Connect y SAML 2.0
- Concesiones de OAuth 2.0
- El uso de PKCE en las concesiones de códigos de autorización con grupos de usuarios de Amazon Cognito
- Respuestas de error de IU alojada y federación

Referencia de puntos de conexión de interfaz de usuario alojada

Amazon Cognito activa los puntos de conexión de interfaz de usuario alojados en esta sección cuando agrega un dominio al grupo de usuarios. Son páginas web en las que los usuarios pueden completar las operaciones de autenticación principales de un grupo de usuarios. Incluyen páginas para la administración de contraseñas, la autenticación multifactor (MFA) y la verificación de atributos. Para obtener más información sobre la experiencia de usuario en la interfaz de usuario alojada, consulte Registro e inicio de sesión con la interfaz de usuario alojada.

Las páginas web que componen la interfaz de usuario alojada son una aplicación web frontend para sesiones de usuario interactivas con sus clientes. Su aplicación debe invocar la interfaz de usuario alojada en los navegadores de sus usuarios. Amazon Cognito no admite el acceso mediante programación a las páginas web de este capítulo. Los puntos de conexión de federación en Referencia de puntos de conexión de federación OAuth 2.0, OpenID Connect y SAML 2.0 que devuelven una respuesta JSON pueden consultarse directamente en su código de aplicación. El Autorizar punto de conexión redirige a la interfaz de usuario alojada o a una página de inicio de sesión del IdP y también debe abrirse en los navegadores de los usuarios.

En los temas de esta guía se describen detalladamente los puntos de conexión de la interfaz de usuario alojada que se utilizan con frecuencia. Amazon Cognito pone a su disposición las páginas web que aparecen a continuación cuando asigna un dominio a su grupo de usuarios.

Puntos de conexión de interfaz de usuario alojados

URL del punto de conexión	Descripción	Cómo se accede
https://*El dominio del grupo de usuarios*/login	Inicia sesión en el grupo de usuarios locales y federados.	Redireccione desde puntos de conexión como Autorizar punto de conexión, `/logout` y `/confirmforgotPassword`. Consulte Punto de conexión Login.
https://*El dominio del grupo de usuarios*/logout	Cierra la sesión de los usuarios del grupo de usuarios.	Enlace directo. Consulte Punto de conexión Logout.

URL del punto de conexión	Descripción	Cómo se accede
https://*Dominio del grupo de usuarios*/confirmUser	Confirma a los usuarios que han seleccionado un enlace de correo electrónico para verificar su cuenta de usuario.	Enlace seleccionado por el usuario en un mensaje de correo electrónico.
https://*Dominio del grupo de usuarios*/signup	Inscribe a un usuario nuevo. La página `/login` dirige a su usuario a `/signup` cuando selecciona Sign up (Registrarse).	Enlace directo con los mismos parámetros que `/oauth2/authorize` .
https://*Dominio del grupo de usuarios*/confirm	Cuando el grupo de usuarios envía un código de confirmación a un usuario que se haya registrado, se lo pedirá al usuario.	Redirija solo desde `/signup`.
https://*dominio de su grupo de usuarios*/forgotPassword	Solicita al usuario su nombre de usuario y le envía un código de restablecimiento de contraseña. La página `/login` redirige al usuario a `/forgotPassword` cuando selecciona Forgot your password? (¿Ha olvidado su contraseña?).	1. Desde el enlace Olvidé mi contraseña en `/login`. 2. Enlace directo con los mismos parámetros que `/oauth2/authorize` .

URL del punto de conexión	Descripción	Cómo se accede
https://*dominio de su grupo de usuarios*/confirmUser	Solicita al usuario su código de restablecimiento de contraseña y una nueva contraseña. La página `/forgotPassword` redirige al usuario a `/confirmforgotPassword` cuando selecciona Reset your password (Restablecer su contraseña).	Redirija solo desde `/forgotPassword` .
https://*Dominio del grupo de usuarios*/resendcode	Envía un nuevo código de confirmación a un usuario que se ha registrado en el grupo de usuarios.	Redirija solo desde el enlace Enviar un nuevo código a `/confirm`.

Temas

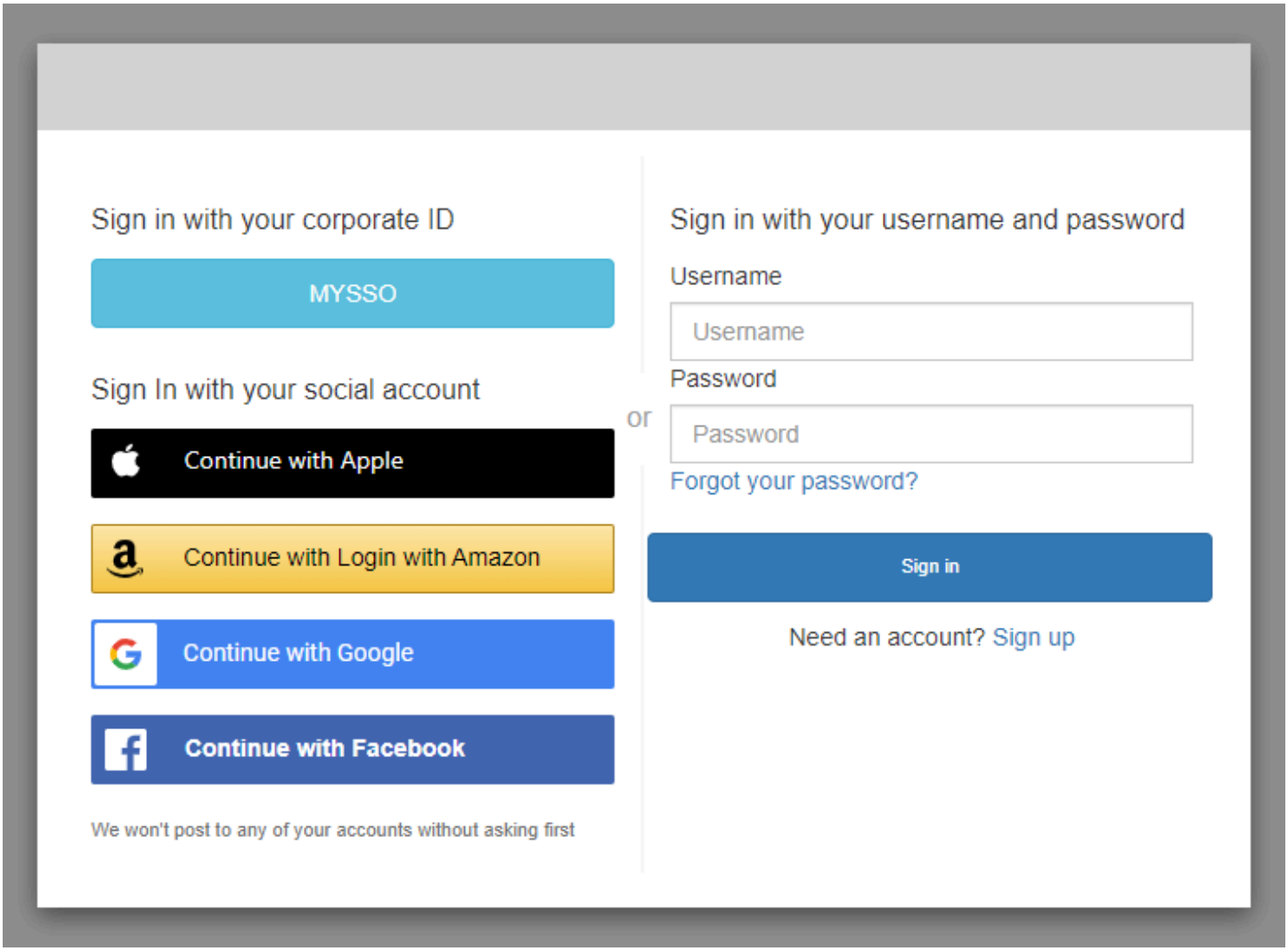
- Punto de conexión Login
- Punto de conexión Logout

Punto de conexión Login

El punto de conexión de inicio de sesión es un servidor de autenticación y un destino de redireccionamiento desde Autorizar punto de conexión. Es el punto de entrada a la interfaz de usuario alojada cuando no especifica un proveedor de identidades. Al generar un redireccionamiento al punto de conexión de inicio de sesión, se carga la página de inicio de sesión, que muestra al usuario las opciones de autenticación configuradas para el cliente.

> **Note**
>
> El punto de conexión de inicio de sesión es un componente de la interfaz de usuario alojada. En la aplicación, invoque las páginas de interfaz de usuario federadas y alojadas que

redirigen al punto de conexión de inicio de sesión. El acceso directo de los usuarios al punto de conexión de inicio de sesión no es una práctica recomendada.



GET /login

El punto de conexión `/login` solo admite `HTTPS GET` para la solicitud inicial del usuario. La aplicación invoca la página en un navegador como Chrome o Firefox. Cuando redireccionas a `/login` desdeAutorizar punto de conexión, se transmiten todos los parámetros que proporcionaste en tu solicitud inicial. El punto de conexión de inicio de sesión admite todos los parámetros de solicitud del punto de conexión autorizado. También puede acceder directamente al punto de conexión de inicio de sesión. Como práctica recomendada, origine todas las sesiones de los usuarios en `/oauth2/authorize`.

Ejemplo: pida al usuario que inicie sesión

En este ejemplo se muestra la pantalla de inicio de sesión.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/login?
                response_type=code&
                client_id=ad398u21ijw3s9w3939&
                redirect_uri=https://YOUR_APP/redirect_uri&
                state=STATE&
                scope=openid+profile+aws.cognito.signin.user.admin
```

Ejemplo: respuesta

El servidor de autenticación redirige a la aplicación con el código y el estado de autorización. El servidor debe devolver el código y el estado en los parámetros de la cadena de consulta y no en el fragmento.

```
HTTP/1.1 302 Found
                    Location: https://YOUR_APP/redirect_uri?
code=AUTHORIZATION_CODE&state=STATE
```

Solicitud de inicio de sesión iniciada por el usuario

Una vez que el usuario cargue el punto de conexión `/login`, podrá ingresar un nombre de usuario y una contraseña y elegir Iniciar sesión. Cuando lo hace, genera una solicitud `HTTPS POST` con los mismos parámetros de solicitud de encabezado que la solicitud `GET` y un cuerpo de solicitud con el nombre de usuario, contraseña y la huella digital del dispositivo.

Punto de conexión Logout

El punto de conexión `/logout` es un punto de conexión de redirección. Cierra la sesión del usuario y lo redirige a una URL de cierre de sesión autorizada para el cliente de la aplicación o al `/login` punto final. Los parámetros disponibles en una solicitud GET al punto de conexión `/logout` se adaptan a los casos de uso de la interfaz de usuario alojada en Amazon Cognito.

Para redirigir al usuario a la interfaz de usuario alojada para volver a iniciar sesión, agregue un parámetro `redirect_uri` a la solicitud. Una solicitud `logout` con un parámetro `redirect_uri` también debe incluir parámetros para la solicitud posterior a Punto de conexión Login, como `client_id`, `response_type` y `scope`.

El punto de conexión es una aplicación web frontend para sesiones de usuario interactivas con sus clientes. Su aplicación debe invocar este y otros puntos de conexión de la interfaz de usuario alojada en los navegadores de sus usuarios.

Para redirigir al usuario a la página que elija, agregue las URL de cierre de sesión permitidas al cliente de la aplicación. En las solicitudes de los usuarios al punto de conexión `logout`, agregue `logout_uri` y los parámetros `client_id`. Si el valor de `logout_uri` es una de las URL de cierre de sesión permitidas para el cliente de la aplicación, Amazon Cognito redirige a los usuarios a esa URL.

Con el cierre de sesión único (SLO) para SAML 2.0, Amazon IdPs Cognito redirige primero al usuario al punto de enlace de SLO que definió en la configuración de su IdP. Una vez que su IdP redirija al usuario de nuevo a, Amazon `saml2/logout` Cognito responde con una redirección más hacia o desde su solicitud. `redirect_uri` `logout_uri` Para obtener más información, consulte Flujo de cierre de sesión de SAML.

El punto de cierre de sesión no cierra la sesión de los usuarios en OIDC ni en los proveedores de identidad social (). IdPs Para cerrar la sesión de los usuarios con un IdP externo, diríjalos a la página de cierre de sesión de ese proveedor.

GET /logout

El punto de enlace `/logout` solo admite `HTTPS GET`. Normalmente, el cliente de grupo de usuarios realiza esta solicitud a través del navegador del sistema. El navegador suele ser la pestaña Chrome personalizada en Android o el controlador de vista de Safari en iOS.

Parámetros de solicitud

client_id

El ID de cliente de aplicación de su aplicación. Para obtener un ID de cliente de aplicación, debe registrar la aplicación en el grupo de usuarios. Para obtener más información, consulte Clientes de aplicación de grupo de usuarios.

Obligatorio.

logout_uri

Redirija al usuario a una página de cierre de sesión personalizada con un parámetro logout_uri. Establecer su valor en la URL de cierre de sesión del cliente de aplicación donde quiere redirigir

al usuario después de que se cierre la sesión. Use logout_uri solo con un parámetro client_id. Para obtener más información, consulte Clientes de aplicación de grupo de usuarios.

También puede utilizar el parámetro logout_uri para redirigir al usuario a la página de inicio de sesión de otro cliente de la aplicación. Establezca la página de inicio de sesión para el otro cliente de aplicación como Allowed callback URL (URL de devolución de llamada permitida) en el cliente de aplicación. En su solicitud al punto de conexión `/logout`, establezca el valor del parámetro logout_uri en la página de inicio de sesión codificada en URL.

Amazon Cognito exige un parámetro logout_uri o redirect_uri en la solicitud al punto de conexión `/logout`. Un parámetro logout_uri redirige al usuario a otro sitio web. Si los parámetros tanto logout_uri como redirect_uri se incluyen en su solicitud para el punto de conexión `/logout`, Amazon Cognito utilizará exclusivamente el parámetro logout_uri, anulando el parámetro redirect_uri.

redirect_uri

Redirija al usuario a la página de inicio de sesión para autenticarse con un parámetro redirect_uri. Establecer su valor en la URL de devolución de llamada permitida del cliente de aplicación donde quiere redirigir al usuario después de que se inicie sesión de nuevo. Añada los parámetros client_id, scope, state y response_type que quiera pasar a su punto de conexión `/login`.

Amazon Cognito exige un parámetro logout_uri o redirect_uri en la solicitud al punto de conexión `/logout`. Para redirigir al usuario al *`/login`* punto final para volver a autenticarse y pasar los tokens a la aplicación, añade un parámetro redirect_uri. Si se incluyen los parámetros logout_uri y redirect_uri en la solicitud al punto de conexión, `/logout` Amazon Cognito anula el parámetro redirect_uri y procesa el parámetro logout_uri exclusivamente.

response_type

La respuesta de OAuth 2.0 que desea recibir de Amazon Cognito después de que el usuario inicie sesión. `code` y `token` son los valores válidos para el parámetro response_type.

Necesario si utiliza un parámetro redirect_uri.

estado

Cuando la aplicación añade un parámetro de estado a una solicitud, Amazon Cognito devuelve su valor a la aplicación cuando el `/oauth2/logout` punto final redirige al usuario.

Agregue este valor a sus solicitudes de protección contra ataques CSRF.

No se puede establecer el valor de un parámetro `state` a una cadena JSON codificada en URL. Para pasar una cadena que coincida con este formato a un `state` parámetro, codifique la cadena en base64 y, a continuación, decodifíquela en su aplicación.

Se recomienda encarecidamente usarlo si se utiliza un parámetro redirect_uri.

scope

Los ámbitos de OAuth 2.0 que desea solicitar a Amazon Cognito después de cerrar sesión con un parámetro redirect_uri. Amazon Cognito redirige a su usuario al punto de conexión `/login` con el parámetro scope en la solicitud al punto de conexión `/logout`.

Necesario si se utiliza un parámetro redirect_uri. Si no se incluye un parámetro scope, Amazon Cognito redirige al usuario al punto de conexión `/login` con un parámetro scope. Cuando Amazon Cognito redirige al usuario y se rellena automáticamente `scope`, el parámetro incluye todos los ámbitos autorizados para su cliente de aplicación.

Ejemplos de solicitudes

Ejemplo: cerrar sesión y redirigir al usuario al cliente

A excepción de `logout_uri` y`client_id`, todos los parámetros de consulta posibles para este punto final se transfieren aAutorizar punto de conexión. Amazon Cognito redirige las sesiones de usuario a la URL con el valor de`logout_uri`, ignorando todos los demás parámetros de solicitud, cuando las solicitudes incluyen `logout_uri` y `client_id`. Esta URL debe ser una URL de cierre de sesión autorizada para el cliente de aplicaciones.

El siguiente es un ejemplo de solicitud de cierre de sesión y redireccionamiento a `https://www.example.com/welcome`.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/logout?
  client_id=1example23456789&
  logout_uri=https%3A%2F%2Fwww.example.com%2Fwelcome
```

Ejemplo: cerrar sesión y solicitar al usuario que inicie sesión como otro usuario

Cuando las solicitudes omiten `logout_uri`, pero proporcionan los parámetros que componen una solicitud con el formato correcto al punto de conexión autorizado, Amazon Cognito redirige a los usuarios al inicio de sesión de la UI alojada. El punto de conexión de cierre de sesión anexa los parámetros de la solicitud original al destino de redireccionamiento. El parámetro `redirect_uri` de

una solicitud al punto de conexión de cierre de sesión no es una URL de cierre de sesión, sino una URL de inicio de sesión por la que se quiere pasar al punto de conexión de autorización.

El siguiente es un ejemplo de solicitud que cierra la sesión de un usuario, lo redirige a la página de inicio de sesión y le proporciona un código de autorización `https://www.example.com` después de iniciar sesión.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/logout?
  response_type=code&
  client_id=1example23456789&
  redirect_uri=https%3A%2F%2Fwww.example.com&
  state=example-state-value&
  nonce=example-nonce-value&
  scope=openid+profile+aws.cognito.signin.user.admin
```

Referencia de puntos de conexión de federación OAuth 2.0, OpenID Connect y SAML 2.0

Amazon Cognito activa los puntos de conexión en esta sección cuando agrega un dominio al grupo de usuarios. Los puntos de conexión de federación no son interactivos para el usuario. Desempeñan una función de servicio para que tu aplicación se comunique con proveedores de identidad de OAuth 2.0, OIDC y SAML 2.0 de terceros (). IdPs

En los temas de esta guía se describen varios puntos de conexión de OAuth 2.0 y OIDC que se utilizan con frecuencia. Amazon Cognito crea los siguientes puntos de conexión cuando asigna un dominio al grupo de usuarios.

Puntos de conexión de federación de grupo de usuarios

URL del punto de conexión	Descripción	Cómo se accede
https://*El dominio del grupo de usuarios*/oauth2/authorize	Redirige un usuario a la interfaz de usuario alojada o a iniciar sesión con el IdP.	Se invocan en el navegador del cliente para iniciar la autenticación del usuario. Consulte Autorizar punto de conexión.
https://*El dominio del grupo de usuarios*/oauth2/token	Devuelve los tokens en función de un código de	La aplicación lo solicitó para recuperar los tokens. Consulte Punto de conexión de token.

URL del punto de conexión	Descripción	Cómo se accede
	autorización o de una solicitud de credenciales del cliente.	
https://*El dominio del grupo de usuarios*/oauth2/userInfo	Devuelve los atributos de usuario en función de los alcances y la identidad del usuario de OAuth 2.0 en un token de acceso.	La aplicación lo solicitó para recuperar el perfil de usuario. Consulte Punto de conexión de UserInfo.
https://*El dominio del grupo de usuarios*/oauth2/revoke	Revoca un token de actualización y los tokens de acceso asociados.	La aplicación solicita la revocación de un token. Consulte Revocación de puntos de conexión.
https://cognito-idp.*Región*.amazonaws.com/*ID de grupo de usuarios*/.well-known/openid-configuration	Directorio de la arquitectura OIDC del grupo de usuarios.	La aplicación lo solicitó para localizar los metadatos del emisor del grupo de usuarios.
https://cognito-idp.*Región*.amazonaws.com/*ID de grupo de usuarios*/.well-known/jwks.json	Claves públicas que puede usar para validar tokens de Amazon Cognito.	Solicitado por la aplicación para verificar los JWT.
https://*Dominkio de grupo de usuarios*/oauth2/idpresponse	Los proveedores de identidad es social deben redirigir a los usuarios a este punto de conexión con un código de autorización. Amazon Cognito canjea el código por un token cuando autentica al usuario federado.	Se redirigió desde el inicio de sesión del IdP de OIDC como URL de devolución de llamada del cliente de IdP.

URL del punto de conexión	Descripción	Cómo se accede
https://*Dominio del grupo de usuarios*/oauth2/idpresponse	La URL de Assertion Consumer Response (ACS) para la integración con los proveedores de identidad de SAML 2.0.	Redirigido desde el IdP de SAML 2.0 como URL de ACS o punto de origen del inicio de sesión iniciado por el IdP. [1]
https://El dominio de su grupo de usuarios / saml2/logout	La URL de cierre de sesión único (SLO) para la integración con los proveedores de identidad de SAML 2.0.	Redirigido desde el IdP de SAML 2.0 como URL de cierre de sesión único (SLO). Solo acepta enlaces POST.

[1] Para obtener más información sobre el inicio de sesión SAML iniciado por el IdP, consulte. Uso del inicio de sesión SAML iniciado por el IdP

Para obtener más información acerca de los estándares OpenID Connect y OAuth, consulte OpenID Connect 1.0 y OAuth 2.0.

Temas

- Autorizar punto de conexión
- Punto de conexión de token
- Punto de conexión de UserInfo
- Revocación de puntos de conexión
- punto final saml2/idpreresponse

Autorizar punto de conexión

El punto de conexión `/oauth2/authorize` es un punto de conexión de redirección que admite dos destinos de redireccionamiento. Si incluye un parámetro `identity_provider` o `idp_identifier` en la URL, redirige al usuario de forma silenciosa a la página de inicio de sesión de ese proveedor de identidades (IdP). De lo contrario, redirige al Punto de conexión Login con los mismos parámetros de URL incluidos en la solicitud.

El punto de conexión de autorización redirige a la interfaz de usuario alojada o a la página de inicio de sesión de IdP. El destino de una sesión de usuario en este punto de conexión es una página web con la que su usuario debe interactuar directamente en su navegador.

Para usar el punto de conexión de autorización, invoque el navegador de su usuario en `/oauth2/authorize` con parámetros que proporcionan a su grupo de usuarios información sobre los siguientes detalles del grupo de usuarios.

- El cliente de aplicación en el que desea iniciar sesión.
- La URL de devolución de llamada en la que quiere terminar.
- Los ámbitos de OAuth 2.0 que desea solicitar en el token de acceso de su usuario.
- De manera opcional, el IdP de terceros que desea usar para iniciar sesión.

También puede suministrar los parámetros `state` y `nonce` que Amazon Cognito utiliza para validar las notificaciones entrantes.

GET **`/oauth2/authorize`**

El punto de enlace `/oauth2/authorize` solo admite `HTTPS GET`. Por lo general, la aplicación inicia esta solicitud en el navegador del usuario. Solo puede hacer solicitudes a los puntos de conexión de `/oauth2/authorize` sobre HTTPS.

Puede obtener más información sobre la definición del punto de conexión de autorización en el estándar OpenID Connect (OIDC) en Punto de conexión de autorización.

Parámetros de solicitud

`response_type`

(Obligatorio) El tipo de respuesta. Debe ser `code` o `token`.

Una solicitud exitosa con un `response_type` de `code` devuelve una concesión de código de autorización. Una concesión de código de autorización es un parámetro `code` que Amazon Cognito añade a la URL de redireccionamiento. Su aplicación puede intercambiar el código con el Punto de conexión de token para tokens de acceso, ID y actualización. Como práctica recomendada de seguridad, y para recibir tokens de actualización para sus usuarios, use un código de autorización de concesión en su aplicación.

Una solicitud exitosa con un `response_type` de `token` devuelve una concesión de código de autorización. Una concesión implícita es un identificador y un token de acceso que Amazon

Cognito añade a la URL de redireccionamiento. Una concesión implícita es menos segura porque expone los tokens y la posible información de identificación a los usuarios. Puede desactivar la compatibilidad con las concesiones implícitas en la configuración del cliente de su aplicación.

`client_id`

(Obligatorio) El ID del cliente de la aplicación.

El valor de `client_id` debe ser el ID de un cliente de aplicación del grupo de usuarios en el que se realiza la solicitud. El cliente de la aplicación debe admitir el inicio de sesión de los usuarios locales de Amazon Cognito o de al menos un IdP de terceros.

`redirect_uri`

(Obligatorio) La URL a la que el servidor de autenticación redirige el navegador después de que Amazon Cognito autorice al usuario.

Un identificador uniforme de recursos (URI) de redirección debe tener los siguientes atributos:

- Ser un URI absoluta
- Debe haber registrado el URI previamente en un cliente.
- No puede incluir un componente fragmento.

Consulte OAuth 2.0 - Redirection Endpoint.

Amazon Cognito requiere que el URI de redireccionamiento use HTTPS, excepto para `http://localhost`, que puede configurar como URL de devolución de llamada para pruebas.

Amazon Cognito también admite las URL de devolución de llamada de aplicación como `myapp://example`.

`state`

(Opcional, recomendado) Cuando la aplicación añade un parámetro de estado a una solicitud, Amazon Cognito devuelve su valor a la aplicación cuando el `/oauth2/authorize` punto final redirige al usuario.

Agregue este valor a sus solicitudes de protección contra ataques CSRF.

No se puede establecer el valor de un parámetro `state` a una cadena JSON codificada en URL. Para pasar una cadena que coincida con este formato a un `state` parámetro, codifique la cadena en base64 y, a continuación, decodifíquela en su aplicación.

identity_provider

(Opcional) Agrega este parámetro para omitir la interfaz de usuario alojada y redirigir al usuario a la página de inicio de sesión de un proveedor. El valor delidentity_provideres el nombre del proveedor de identidad (IdP) tal como aparece en el grupo de usuarios.

- Para los proveedores sociales, puedes usar los valores de identity_provider`Facebook`,, y`Google`. `LoginWithAmazon` `SignInWithApple`
- Para los grupos de usuarios de Amazon Cognito, utilice el valor. `COGNITO`
- Para los proveedores de identidad SAML 2.0 y OpenID Connect (OIDC) (IdPs), usa el nombre que asignaste al IdP en tu grupo de usuarios.

idp_identifier

(Opcional) Agregue este parámetro para redirigirlo a un proveedor con un nombre alternativo para el nombre identity_provider. Puede introducir los identificadores de SAML 2.0 y OIDC IdPs desde la pestaña Experiencia de inicio de sesión de la consola de Amazon Cognito.

scope

(Opcional) Puede ser una combinación de cualquier ámbito reservado por el sistema o un ámbito personalizado que esté asociado a un cliente. Los ámbitos deben estar separados por espacios. Los ámbitos reservados por el sistema son `openid`, `email`, `phone`, `profile` y `aws.cognito.signin.user.admin`. Todo ámbito utilizado debe estar asociado al cliente o se ignorará en el tiempo de ejecución.

Si el cliente no solicita ningún ámbito, en el servidor de autenticación se utilizarán todos los ámbitos asociados al cliente.

Solo se devuelve un token de ID si se solicita el ámbito `openid`. El token de acceso solo se puede utilizar en grupos de usuarios de Amazon Cognito si se solicita el ámbito `aws.cognito.signin.user.admin`. Los ámbitos `phone`, `email` y `profile` solo se pueden solicitar si se solicita también el ámbito `openid`. Estos ámbitos dictan las notificaciones que se incluyen en el token de ID.

code_challenge_method

(Opcional) El protocolo de hash que utilizó para generar el desafío. En el PKCE RFC, se definen dos métodos, S256 y sin formato; sin embargo, el servidor de autenticación de Amazon Cognito solo admite S256.

`code_challenge`

(Opcional) El desafío que generaste a partir del`code_verifier`.

Obligatorio solo cuando se especifica un parámetro `code_challenge_method`.

`nonce`

(Opcional) Un valor aleatorio que puedes añadir a la solicitud. El valor nonce que proporciona se incluye en el token de ID que emite Amazon Cognito. Para protegerse de los ataques de reproducción, su aplicación puede inspeccionar la reclamación de `nonce` en el token de identificación y compararlo con el generado. Para obtener más información sobre la reclamación de `nonce`, consulte ID token validation (Validación de token de ID) en el estándar de OpenID Connect.

Ejemplo de solicitudes con respuestas positivas

Los siguientes ejemplos ilustran el formato de las solicitudes HTTP al `/oauth2/authorize` punto final.

Concesión de código de autorización

Este es un ejemplo de solicitud de concesión de código de autorización.

Ejemplo: solicitud GET

La siguiente solicitud inicia una sesión para recuperar un código de autorización que el usuario pasa a la aplicación en el `redirect_uri` destino. En esta sesión, se solicitan los ámbitos de los atributos de usuario y el acceso a las operaciones de la API de autoservicio de Amazon Cognito.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=openid+profile+aws.cognito.signin.user.admin
```

Ejemplo: respuesta

El servidor de autenticación de Amazon Cognito redirige a la aplicación con el código y el estado de autorización. El código de autorización es válido durante cinco minutos.

```
HTTP/1.1 302 Found
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Concesión de código de autorización con PKCE

Este es un ejemplo de solicitud de concesión de un código de autorización con el PKCE.

Ejemplo: solicitud GET

La siguiente solicitud agrega un `code_challenge` parámetro a la solicitud anterior. Para completar el intercambio de un código por un token, debe incluir el `code_verifier` parámetro en la solicitud al `/oauth2/token` punto final.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin&
code_challenge_method=S256&
code_challenge=a1b2c3d4...
```

Ejemplo: respuesta

El servidor de autenticación redirige de nuevo a su aplicación con el código y el estado de autorización. El código y el estado deben devolverse en los parámetros de la cadena de consulta y no en el fragmento:

```
HTTP/1.1 302 Found
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Concesión de token sin ámbito **`openid`**

Este es un ejemplo de solicitud que genera una concesión implícita y devuelve los JWT directamente a la sesión del usuario.

Ejemplo: solicitud GET

La siguiente solicitud es para una concesión implícita de su servidor de autorización. El token de acceso de Amazon Cognito autoriza las operaciones de la API de autoservicio.

```
GET https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=token&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin
```

Ejemplo: respuesta

El servidor de autorización de Amazon Cognito redirige a la aplicación con el token de acceso. Dado que no se ha solicitado el ámbito `openid`, Amazon Cognito no devuelve un token de ID. Además, Amazon Cognito no devuelve un token de actualización en este flujo. Amazon Cognito devuelve el token de acceso y el estado en el fragmento y no en la cadena de consulta:

```
HTTP/1.1 302 Found
Location: https://YOUR_APP/
redirect_uri#access_token=ACCESS_TOKEN&token_type=bearer&expires_in=3600&state=STATE
```

Concesión de token con ámbito **`openid`**

Este es un ejemplo de solicitud que genera una concesión implícita y devuelve los JWT directamente a la sesión del usuario.

Ejemplo: solicitud GET

La siguiente solicitud es para una concesión implícita de su servidor de autorización. El token de acceso de Amazon Cognito autoriza el acceso a los atributos del usuario y a las operaciones de la API de autoservicio.

```
GET
https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=token&
client_id=1example23456789&
redirect_uri=https://www.example.com&
state=abcdefg&
scope=aws.cognito.signin.user.admin+openid+profile
```

Ejemplo: respuesta

El servidor de autorización redirige de nuevo a tu aplicación con el token de acceso y el token de identificación (porque el `openid` alcance estaba incluido):

```
HTTP/1.1 302 Found
Location: https://
www.example.com#id_token=eyJra67890EXAMPLE&access_token=eyJra12345EXAMPLE&token_type=bearer&exp
```

Ejemplos de respuestas negativas

Amazon Cognito podría denegar su solicitud. Las solicitudes negativas vienen con un código de error HTTP y una descripción que puede utilizar para corregir los parámetros de la solicitud. Los siguientes son ejemplos de respuestas negativas.

- Si `client_id` y `redirect_uri` son válidos, pero los parámetros de la solicitud no tienen el formato correcto, el servidor de autenticación redirige el error al del cliente `redirect_uri` y agrega un mensaje de error a un parámetro de URL. A continuación se muestran ejemplos de un formato incorrecto.
  - La solicitud no incluye ningún `response_type` parámetro.
  - La solicitud de autorización proporcionó un `code_challenge` parámetro, pero no un `code_challenge_method` parámetro.
  - El valor del `code_challenge_method` parámetro no lo es`S256`.

  A continuación se muestra la respuesta a un ejemplo de solicitud con un formato incorrecto.

  ```
  HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_request
  ```
- Si el cliente solicita `code` o `token` entra`response_type`, pero no tiene permiso para estas solicitudes, el servidor de autorización de Amazon Cognito vuelve `unauthorized_client` al del cliente de la siguiente `redirect_uri` manera:

  ```
  HTTP 1.1 302 Found Location: https://client_redirect_uri?error=unauthorized_client
  ```
- Si el cliente solicita un ámbito no válido, desconocido o con un formato incorrecto, el servidor de autorización de Amazon Cognito devuelve `invalid_scope` al `redirect_uri` del cliente, tal y como se indica a continuación:

  ```
  HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_scope
  ```

- Si se produce un error inesperado en el servidor, el servidor de autenticación vuelve `server_error` al servidor del `redirect_uri` cliente. Como el error HTTP 500 no se envía al cliente, no se muestra en el navegador del usuario. El servidor de autorización devuelve el siguiente error.

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=server_error
```

- Cuando Amazon Cognito se autentica mediante la federación con un tercero, Amazon IdPs Cognito puede experimentar problemas de conexión, como los siguientes:
  - Si se produce un tiempo de espera de conexión al solicitar un token desde el IdP, el servidor de autenticación redirecciona el error al `redirect_uri` del cliente como se muestra a continuación:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=Timeout+occurred+in+calling+IdP+token
+endpoint
```

  - Si se agota el tiempo de espera de la conexión al `jwks_uri` punto final para validar el token de identificación, el servidor de autenticación redirige con un error al servidor del cliente de la siguiente manera: `redirect_uri`

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=error_description=Timeout+in+calling+jwks
+uri
```

- Al autenticarse mediante la federación a un tercero IdPs, es posible que los proveedores devuelvan respuestas de error. Esto puede deberse a errores de configuración o a otros motivos, como los siguientes:
  - Si se recibe una respuesta de error de otros proveedores, el servidor de autenticación redirige el error al `redirect_uri` del cliente como se muestra a continuación:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=[IdP name]+Error+-+[status code]+error
 getting token
```

  - Si se recibe una respuesta de error de Google, el servidor de autenticación redirige el error al `redirect_uri` del cliente como se muestra a continuación:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=Google+Error+-+[status code]+[Google-
provided error code]
```

- Cuando Amazon Cognito detecta una excepción de comunicación al conectarse a un IdP externo, el servidor de autenticación redirige con un error al del cliente con uno de los siguientes `redirect_uri` mensajes:
  - ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=Connection+reset
    ```
  - ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=Read+timed+out
    ```

Punto de conexión de token

El punto de conexión de tokens de OAuth 2.0 en `/oauth2/token` emite tokens web JSON (JWT).

El servidor de autorización de OAuth 2.0 de su grupo de usuarios emite tokens web JSON (JWT) desde el punto de enlace del token para los siguientes tipos de sesiones:

1. Usuarios que han completado una solicitud de concesión de un código de autorización. Al canjear correctamente un código, se obtienen tokens de ID, acceso y actualización.
2. M achine-to-machine (M2M) sesiones que han completado una concesión de credenciales de cliente. Una autorización correcta con el secreto del cliente devuelve un token de acceso.
3. Usuarios que han iniciado sesión anteriormente y han recibido tokens de actualización. La autenticación con un token de actualización devuelve un nuevo identificador y un nuevo token de acceso.

> **Note**
>
> Los usuarios que inicien sesión con un código de autorización otorgado en la interfaz de usuario alojada o mediante la federación siempre pueden actualizar sus tokens desde el punto de enlace del token. Los usuarios que inician sesión con las operaciones de la API `InitiateAuth` y `AdminInitiateAuth` pueden actualizar sus tokens con el punto final del token cuando los dispositivos recordados no están activos en su grupo de usuarios. Si los dispositivos recordados están activos, actualiza los tokens con las solicitudes

`AuthFlow` `REFRESH_TOKEN_AUTH` de `InitiateAuth` entrada o `AdminInitiateAuth` API.

El punto de conexión del token pasa a estar disponible públicamente cuando agrega un dominio a su grupo de usuarios. Acepta solicitudes HTTP POST. Para garantizar la seguridad de las aplicaciones, usa PKCE con tus eventos de inicio de sesión con el código de autorización. PKCE verifica que el usuario que pasa un código de autorización es el mismo que se autenticó. Para obtener más información sobre el PKCE, consulte la RFC 7636 del IETF.

Puede obtener más información sobre los clientes de aplicación del grupo de usuarios y sus tipos de concesión, secretos de cliente, ámbitos autorizados e ID de cliente en Clientes de aplicación de grupo de usuarios. Puede obtener más información sobre la autorización M2M, la concesión de credenciales de cliente y la autorización con alcances de token de acceso en. Autorización de alcances, M2M y API con servidores de recursos

Para recuperar información sobre un usuario desde su token de acceso, pásala a tu solicitud Punto de conexión de UserInfo o a una solicitud de GetUserAPI.

POST /oauth2/token

El punto de enlace `/oauth2/token` solo admite `HTTPS POST`. El cliente de grupo de usuarios realiza solicitudes a este punto de enlace directamente y no mediante un navegador.

El punto de conexión de token admite la autenticación `client_secret_basic` y `client_secret_post`. Para obtener más información sobre la especificación OpenID Connect, consulte Autenticación de cliente. Para obtener más información sobre el punto de conexión de token de la especificación OpenID Connect, consulte Punto de conexión de token.

Parámetros de la solicitud en el encabezado

`Authorization`

Si se le emitió un secreto al cliente, debe pasar su `client_id` y `client_secret` en el encabezado de la autorización a través de la autorización HTTP `client_secret_basic`. También puede incluir el `client_id` y `client_secret` en el cuerpo de la solicitud como autorización `client_secret_post`.

La cadena de encabezado de autorización es Basic `Base64Encode(client_id:client_secret)`. El siguiente ejemplo es un encabezado

de autorización para un cliente de aplicación `djc98u3jiedmi283eu928` con un secreto de cliente`abcdef01234567890`, que utiliza la versión de la cadena codificada en Base64: `djc98u3jiedmi283eu928:abcdef01234567890`

```
Authorization: Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw
```

`Content-Type`

Establezca el valor del parámetro en `'application/x-www-form-urlencoded'`.

Parámetros de la solicitud en el cuerpo

`grant_type`

(Obligatorio) El tipo de concesión de la OIDC que desea solicitar.

Debe ser `authorization_code`, `refresh_token` o `client_credentials`. Puedes solicitar un token de acceso para un ámbito personalizado desde el punto final del token en las siguientes condiciones:

- Has activado el ámbito solicitado en la configuración del cliente de tu aplicación.
- Has configurado el cliente de la aplicación con un secreto de cliente.
- Usted habilita la concesión de credenciales de cliente en el cliente de su aplicación.

`client_id`

(Opcional) El ID de un cliente de aplicaciones de tu grupo de usuarios. Especifique el mismo cliente de aplicación que autenticó al usuario.

Debe proporcionar este parámetro si el cliente es público y no tiene un secreto o no tiene `client_secret_post` autorización. `client_secret`

`client_secret`

(Opcional) El secreto del cliente de la aplicación que autenticó al usuario. Obligatorio si el cliente de aplicación tiene un secreto de cliente y no ha enviado un encabezado `Authorization`.

`scope`

(Opcional) Puede ser una combinación de cualquier ámbito personalizado asociado a un cliente de aplicación. Cualquier ámbito que solicite debe estar activado para el cliente de la aplicación. Si no, Amazon Cognito lo ignorará. Si el cliente no solicita ningún ámbito, el servidor

de autenticación asigna todos los ámbitos personalizados que usted autorizó en la configuración del cliente de la aplicación.

Solo se usa si `grant_type` es `client_credentials`.

`redirect_uri`

(Opcional) Debe ser el mismo `redirect_uri` que se usó para entrar. `authorization_code` / `oauth2/authorize`

Debe proporcionar este parámetro, si lo `grant_type` está`authorization_code`.

`refresh_token`

(Opcional) Para generar nuevos tokens de acceso e ID para la sesión de un usuario, establece el valor de un `refresh_token` parámetro de tu `/oauth2/token` solicitud en un token de actualización emitido anteriormente desde el mismo cliente de la aplicación.

`code`

(Opcional) El código de autorización de una concesión de código de autorización. Debe proporcionar este parámetro si su solicitud de autorización incluía uno `grant_type` de`authorization_code`.

`code_verifier`

(Opcional) El valor arbitrario que utilizó para calcular una solicitud de `code_challenge` concesión de código de autorización con el PKCE.

Ejemplo de solicitudes con respuestas positivas

Intercambio de un código de autorización para los tokens

Ejemplo: solicitud POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token&
                            Content-Type='application/x-www-form-urlencoded'&

 Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

                            grant_type=authorization_code&
                            client_id=1example23456789&
                            code=AUTHORIZATION_CODE&
```

```
                            redirect_uri=com.myclientapp://myclient/redirect
```

Ejemplo: respuesta

```
HTTP/1.1 200 OK

                           Content-Type: application/json

                           {
                            "access_token":"eyJra1example",
                            "id_token":"eyJra2example",
                            "refresh_token":"eyJj3example",
                            "token_type":"Bearer",
                            "expires_in":3600
                           }
```

> **Note**
>
> El punto de enlace del token devuelve `refresh_token` solo cuando `grant_type` es `authorization_code`.

Intercambio de credenciales de cliente para un token de acceso: secreto del cliente en un encabezado de autorización

Ejemplo: solicitud POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
                            Content-Type='application/x-www-form-urlencoded'&

 Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

                            grant_type=client_credentials&
                            client_id=1example23456789&

 scope=resourceServerIdentifier1/scope1 resourceServerIdentifier2/scope2
```

Ejemplo: respuesta

```
HTTP/1.1 200 OK
```

```
                            Content-Type: application/json

                            {
                            "access_token":"eyJra1example",
                            "token_type":"Bearer",
                            "expires_in":3600
                            }
```

Intercambio de credenciales de cliente para un token de acceso: secreto del cliente en un cuerpo de solicitud

Ejemplo: solicitud POST

```
POST /oauth2/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
X-Amz-Target: AWSCognitoIdentityProviderService.Client credentials request
User-Agent: USER_AGENT
Accept: /
Accept-Encoding: gzip, deflate, br
Content-Length: 177
Referer: http://auth.example.com/oauth2/token
Host: auth.example.com
Connection: keep-alive

grant_type=client_credentials&client_id=1example23456789&scope=my_resource_server_identifier%2F
```

Ejemplo: respuesta

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Tue, 05 Dec 2023 16:11:11 GMT
x-amz-cognito-request-id: 829f4fe2-a1ee-476e-b834-5cd85c03373b

{
"access_token": "eyJra12345EXAMPLE",
"expires_in": 3600,
"token_type": "Bearer"
}
```

Intercambio de una concesión de código de autorización con PKCE para los tokens

Ejemplo: solicitud POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token
                                Content-Type='application/x-www-form-urlencoded'&

 Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

                                grant_type=authorization_code&
                                client_id=1example23456789&
                                code=AUTHORIZATION_CODE&
                                code_verifier=CODE_VERIFIER&
                                redirect_uri=com.myclientapp://myclient/redirect
```

Ejemplo: respuesta

```
HTTP/1.1 200 OK
                            Content-Type: application/json

                            {
                            "access_token":"eyJra1example",
                            "id_token":"eyJra2example",
                            "refresh_token":"eyJj3example",
                            "token_type":"Bearer",
                            "expires_in":3600
                            }
```

> **Note**
>
> El punto de enlace del token devuelve `refresh_token` solo cuando `grant_type` es `authorization_code`.

Intercambio de un token de actualización para los tokens

Ejemplo: solicitud POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
                           Content-Type='application/x-www-form-urlencoded'&

 Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw
```

```
                        grant_type=refresh_token&
                        client_id=1example23456789&
                        refresh_token=eyJj3example
```

Ejemplo: respuesta

```
HTTP/1.1 200 OK
                        Content-Type: application/json

                        {
                        "access_token":"eyJra1example",
                        "id_token":"eyJra2example",
                        "token_type":"Bearer",
                        "expires_in":3600
                              }
```

> **Note**
>
> El punto de enlace del token devuelve `refresh_token` solo cuando `grant_type` es `authorization_code`.

Ejemplos de respuestas negativas

Ejemplo: respuesta de error

```
HTTP/1.1 400 Bad Request
                        Content-Type: application/json;charset=UTF-8

                        {
                        "error":"invalid_request|invalid_client|invalid_grant|
unauthorized_client|unsupported_grant_type"
                        }
```

`invalid_request`

Falta un parámetro necesario en la solicitud, la solicitud incluye un valor de parámetro no admitido (distinto de `unsupported_grant_type`) o la solicitud tiene un formato incorrecto. Por ejemplo, `grant_type` es `refresh_token` pero `refresh_token` no está incluido.

`invalid_client`

Error de autenticación del cliente. Por ejemplo, cuando el cliente incluye `client_id` y `client_secret` en el encabezado de la autorización, pero no existe un cliente con esos `client_id` y `client_secret`.

`invalid_grant`

El token de actualización se ha revocado.

El código de autorización ya se ha utilizado o no existe.

El cliente de la aplicación no tiene acceso de lectura a todos los atributos en el ámbito solicitado. Por ejemplo, su aplicación solicita el ámbito `email` y su cliente de aplicación puede leer el atributo `email`, pero no `email_verified`.

`unauthorized_client`

El cliente no tiene permiso para el flujo de concesión de códigos o para la actualización de tokens.

`unsupported_grant_type`

Se devuelve si `grant_type` es distinto de `authorization_code`, `refresh_token` o `client_credentials`.

Punto de conexión de UserInfo

El punto de conexión `userInfo` es un punto de conexión userInfo de OpenID Connect (OIDC). Responde con atributos de usuario cuando los proveedores de servicios presentan tokens de acceso que su Punto de conexión de token ha emitido. Los ámbitos del token de acceso de su usuario definen los atributos de usuario que el punto de conexión userInfo devuelve en su respuesta. El ámbito `openid` debe ser una de las notificaciones del token de acceso.

Amazon Cognito emite tokens de acceso en respuesta a solicitudes de la API de grupos de usuarios como InitiateAuth. Como no contienen ningún ámbito, el punto de conexión userInfo no acepta estos tokens de acceso. En su lugar, debe presentar los tokens de acceso desde el punto de conexión del token.

Su proveedor de identidades (IdP) externo de OAuth 2.0 también aloja un punto de conexión userInfo. Cuando el usuario se autentica con ese IdP, Amazon Cognito intercambia silenciosamente un código de autorización con el punto de enlace del IdP. `token` Su grupo de usuarios pasa el token

de acceso del IdP para autorizar la recuperación de la información del usuario desde el punto final del IdP. `userInfo`

GET /oauth2/userInfo

Su aplicación hace peticiones a este punto de conexión directamente y no a través de un navegador.

Para obtener más información, consulte el tema sobre el punto de conexión UserInfo en la especificación OpenID Connect (OIDC).

Temas

- Parámetros de la solicitud en el encabezado
- Ejemplo: solicitud
- Ejemplo: respuesta positiva
- Ejemplo de respuestas negativas

Parámetros de la solicitud en el encabezado

`Authorization: Bearer` ***`<access_token>`***

Pase el token de acceso al campo del encabezado de autorización.

Obligatorio.

Ejemplo: solicitud

```
GET /oauth2/userInfo HTTP/1.1
Content-Type: application/x-amz-json-1.1
Authorization: Bearer eyJra12345EXAMPLE
User-Agent: [User agent]
Accept: */*
Host: auth.example.com
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Ejemplo: respuesta positiva

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: [Integer]
```

```
Date: [Timestamp]
x-amz-cognito-request-id: [UUID]
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Server: Server
Connection: keep-alive
{
    "sub": "[UUID]",
    "email_verified": "true",
    "custom:mycustom1": "CustomValue",
    "phone_number_verified": "true",
    "phone_number": "+12065551212",
    "email": "bob@example.com",
    "username": "bob"
}
```

Para obtener una lista de las notificaciones OIDC, consulte el tema sobre notificaciones estándar. Actualmente, Amazon Cognito devuelve los valores para `email_verified` y `phone_number_verified` como cadenas.

Ejemplo de respuestas negativas

Ejemplo: solicitud incorrecta

```
HTTP/1.1 400 Bad Request
WWW-Authenticate: error="invalid_request",
error_description="Bad OAuth2 request at UserInfo Endpoint"
```

`invalid_request`

A la solicitud le falta un parámetro obligatorio, incluye un valor de parámetro no admitido o está mal formada por algún motivo.

Ejemplo: token incorrecto

```
HTTP/1.1 401 Unauthorized
```

```
WWW-Authenticate: error="invalid_token",
error_description="Access token is expired, disabled, or deleted, or the user has
 globally signed out."
```

`invalid_token`

El token de acceso ha caducado, está revocado, tiene un formato incorrecto o no es válido.

Revocación de puntos de conexión

El `oauth2/revoke` punto de enlace/revoca el token de acceso de un usuario que Amazon Cognito emitió inicialmente con el token de actualización que usted proporciona. Este punto final también revoca todos los tokens de acceso e identidad posteriores del mismo token de actualización. Después de que el punto de conexión revoque los tokens, no podrá usar los tokens de acceso revocados para acceder a las API que autentican los tokens de Amazon Cognito.

POST/oauth2/revoke

El punto de enlace `/oauth2/revoke` solo admite `HTTPS POST`. El cliente del grupo de usuarios realiza solicitudes a este punto de enlace directamente y no a través del navegador del sistema.

Parámetros de la solicitud en el encabezado

`Authorization`

Si el cliente de la aplicación tiene un secreto de cliente, la aplicación debe pasar su nombre `client_id` y `client_secret` el encabezado de autorización mediante la autorización HTTP básica. El secreto es Basic `Base64Encode(client_id:client_secret)`.

`Content-Type`

Debe ser siempre `'application/x-www-form-urlencoded'`.

Parámetros de la solicitud en el cuerpo

`token`

(Obligatorio) El token de actualización que el cliente quiere revocar. La solicitud también revoca todos los tokens de acceso que Amazon Cognito emitió desde este token de actualización.

Obligatorio.

`client_id`

(Opcional) El ID de cliente de la aplicación para el token que quieres revocar.

Obligatorio si el cliente es público y no tiene ningún secreto.

Ejemplo de solicitud de revocación

Ejemplo 1: revocar un token para un cliente de aplicación sin secreto de cliente

```
POST /oauth2/revoke HTTP/1.1
    Host: https://mydomain.auth.us-east-1.amazoncognito.com
    Accept: application/json
    Content-Type: application/x-www-form-urlencoded
    token=2YotnFZFEjr1zCsicMWpAA&
    client_id=djc98u3jiedmi283eu928
```

Ejemplo 2: revocar un token para un cliente de aplicación con un secreto de cliente

```
POST /oauth2/revoke HTTP/1.1
    Host: https://mydomain.auth.us-east-1.amazoncognito.com
    Accept: application/json
    Content-Type: application/x-www-form-urlencoded
    Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
    token=2YotnFZFEjr1zCsicMWpAA
```

Respuesta de error de revocación

En una respuesta satisfactoria, se incluye un cuerpo vacío. La respuesta de error es un objeto JSON con un campo `error` y, en algunos casos, un campo `error_description`.

Errores de punto de conexión

- Se devuelve HTTP 400 y el error `invalid_request` si el token no está presente en la solicitud o si la característica se desactiva para el cliente de aplicación.
- Si el token que Amazon Cognito envió en la solicitud de revocación no es un token de actualización, recibirá un HTTP 400 y un error `unsupported_token_type`.
- Si las credenciales de cliente no son válidas, recibirá un HTTP 401 y un error `invalid_client`.

- Si el token se ha revocado o si el cliente ha enviado un token que no es válido, recibirá un HTTP 200 OK.

punto final saml2/idpreresponse

`/saml2/idpresponse`Recibe las afirmaciones de SAML. Al iniciar sesión service-provider-initiated (iniciado por SP), tu proveedor de identidad (IdP) de SAML 2.0 redirige al usuario a este punto final con su respuesta de SAML. En el inicio de sesión iniciado por el SP, la aplicación no interactúa con este punto final. Configure su IdP con la ruta a su URL `saml2/idpresponse` como URL del servicio de consumo de aserciones (ACS). Para obtener más información sobre el inicio de la sesión, consulte. Inicio de sesión SAML en grupos de usuarios de Amazon Cognito

En el inicio de sesión iniciado por el IdP, tus usuarios pueden iniciar sesión con tu IdP mediante tu propio proceso y enviar una afirmación de SAML en el cuerpo de una solicitud a través de HTTPS. `HTTP POST` El cuerpo de la `POST` solicitud debe ser un parámetro y un parámetro. `SAMLResponse` `Relaystate` Para obtener más información, consulte Uso del inicio de sesión SAML iniciado por el IdP.

PUBLICAR **`/saml2/idpresponse`**

Para usar el `/saml2/idpresponse` punto final en un inicio de sesión iniciado por el IdP, genere una solicitud POST con parámetros que proporcionen a su grupo de usuarios información sobre la sesión de su usuario.

- El cliente de la aplicación en el que quieren iniciar sesión.
- La URL de devolución de llamada en la que quieren terminar.
- Los alcances de OAuth 2.0 quieren solicitar en el token de acceso de tu usuario.
- El IdP que inició la solicitud de inicio de sesión.

Parámetros del cuerpo de la solicitud iniciada por el IDP

Respuesta SAML

Una afirmación SAML codificada en Base64 de un IdP asociado a un cliente de aplicaciones válido y a una configuración de IdP de su grupo de usuarios.

RelayState

Un `RelayState` parámetro contiene los parámetros de solicitud que, de otro modo, pasaría al punto final. `oauth2/authorize` Para obtener información detallada sobre estos parámetros, consulteAutorizar punto de conexión.

response_type

El tipo de concesión de OAuth 2.0.

client_id

El ID de cliente de aplicación.

redirect_uri

La dirección URL a la que el servidor de autenticación redirige el navegador después de que Amazon Cognito autorice al usuario.

identity_provider

El nombre del proveedor de identidad al que quieres redirigir a tu usuario.

idp_identifier

El identificador del proveedor de identidad al que quieres redirigir a tu usuario.

scope

Los ámbitos de OAuth 2.0 que quieres que tu usuario solicite al servidor de autorización.

Ejemplos de solicitudes con respuestas positivas

Ejemplo: solicitud POST

La siguiente solicitud es para la concesión de un código de autorización para un usuario desde el IdP `MySAMLIdP` en el cliente de la aplicación. `1example23456789` El usuario lo redirige a `https://www.example.com` con su código de autorización, que se puede cambiar por tokens que incluyan un token de acceso con los alcances `openid` de OAuth 2.0, y. `email` `phone`

```
POST /saml2/idpresponse HTTP/1.1
User-Agent: USER_AGENT
Accept: */*
Host: example.auth.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded
```

```
SAMLResponse=[Base64-encoded SAML assertion]&RelayState=identity_provider
%3DMySAMLIdP%26client_id%3D1example23456789%26redirect_uri%3Dhttps%3A%2F
%2Fwww.example.com%26response_type%3Dcode%26scope%3Demail%2Bopenid%2Bphone
```

Ejemplo: respuesta

La siguiente es la respuesta a la solicitud anterior.

```
HTTP/1.1 302 Found
Date: Wed, 06 Dec 2023 00:15:29 GMT
Content-Length: 0
x-amz-cognito-request-id: 8aba6eb5-fb54-4bc6-9368-c3878434f0fb
Location: https://www.example.com?code=[Authorization code]
```

Concesiones de OAuth 2.0

El servidor de autorización de OAuth 2.0 del grupo de usuarios de Amazon Cognito emite tokens en respuesta a tres tipos de concesiones de autorización de OAuth 2.0. Puede configurar los tipos de concesión admitidos para cada cliente de aplicaciones del grupo de usuarios. No puede habilitar concesiones de credenciales de cliente en el mismo cliente de la aplicación que cualquiera de las concesiones de códigos implícitos o de autorización. Las solicitudes de concesiones de códigos implícitos y de autorización comienzan en Autorizar punto de conexión y las solicitudes de concesiones de credenciales de clientes comienzan en Punto de conexión de token.

Concesión de código de autorización

> En respuesta a la solicitud de autenticación correcta, el servidor de autorización agrega un código de autorización en un parámetro `code` a la URL de devolución de llamada. A continuación, debe intercambiar el código para los tokens de ID, acceso y actualización con Punto de conexión de token. Para solicitar la concesión de un código de autorización, establezca `response_type` en `code` en la solicitud. Para obtener una solicitud de ejemplo, consulte Concesión de código de autorización.
>
> La concesión del código de autorización es la forma más segura de concesión de autorización. No muestra el contenido del token directamente a los usuarios. En cambio, la aplicación es responsable de recuperar y almacenar de forma segura los tokens de los usuarios. En Amazon Cognito, la concesión de un código de autorización es la única forma de obtener los tres tipos de token (ID, acceso y actualización) del servidor de autorización. También puede obtener los tres tipos de token mediante la autenticación a través de la API de grupos

de usuarios de Amazon Cognito, pero la API no emite tokens de acceso con otros ámbitos que `aws.cognito.signin.user.admin`.

Implicit grant (Concesión implícita)

En respuesta a la solicitud de autenticación correcta, el servidor de autorización agrega un token de acceso a un parámetro `access_token` y un token de identificación en un parámetro `id_token`, a la URL de devolución de llamada. Una concesión implícita no requiere ninguna interacción adicional con Punto de conexión de token. Para solicitar una concesión implícita, establezca `response_type` en `token` en la solicitud. La concesión implícita solo genera un identificador y un token de acceso. Para obtener una solicitud de ejemplo, consulte Concesión de token sin ámbito `openid`.

La concesión implícita es una concesión de autorización antigua. A diferencia de lo que ocurre con la concesión del código de autorización, los usuarios pueden interceptar e inspeccionar los tokens. Para evitar la entrega de tokens mediante una concesión implícita, configure el cliente de la aplicación para que solo admita la concesión de códigos de autorización.

Client credentials (Credenciales del cliente)

Las credenciales del cliente son una concesión de acceso únicamente con autorización. machine-to-machine Para recibir una concesión de credenciales de cliente, omita Autorizar punto de conexión y genere una solicitud directamente al Punto de conexión de token. El cliente de la aplicación debe tener un secreto de cliente y admitir solo la concesión de credenciales de cliente. En respuesta a la solicitud correcta, el servidor de autorización devuelve un token de acceso.

El token de acceso de la concesión de credenciales de un cliente es un mecanismo de autorización que contiene los ámbitos de OAuth 2.0. Por lo general, el token contiene notificaciones de alcance personalizadas que autorizan las operaciones HTTP a las API protegidas por el acceso. Para obtener más información, consulte Autorización de alcances, M2M y API con servidores de recursos.

La concesión de credenciales de cliente añade costes a tu factura. AWS Para obtener más información, consulte Precios de Amazon Cognito.

El uso de PKCE en las concesiones de códigos de autorización con grupos de usuarios de Amazon Cognito

Amazon Cognito admite la autenticación con clave de prueba para intercambio de códigos (PKCE) en las concesiones de códigos de autorización. PKCE es una extensión de la concesión de

códigos de autorización OAuth 2.0 para clientes públicos. La PKCE evita el canje de los códigos de autorización interceptados.

Cómo utiliza Amazon Cognito PKCE

Para iniciar la autenticación con PKCE, la aplicación debe generar un valor de cadena único. Esta cadena es el verificador de código, un valor secreto que Amazon Cognito utiliza para comparar el cliente que solicita la concesión de autorización inicial con el cliente que intercambia el código de autorización por tokens.

La aplicación debe aplicar un hash SHA256 a la cadena del verificador de código y codificar el resultado en base64. Pasa la cadena cifrada a Autorizar punto de conexión como `code_challenge` parámetro en el cuerpo de la solicitud. Cuando tu aplicación intercambie el código de autorización por tokens, debe incluir la cadena verificadora de código en texto simple como `code_verifier` parámetro en el cuerpo de la solicitud. Punto de conexión de token Amazon Cognito realiza la misma hash-and-encode operación en el verificador de código. Amazon Cognito solo devuelve los tokens de ID, acceso y actualización si determina que el verificador de código genera el mismo desafío de código que recibió en la solicitud de autorización.

Para implementar el flujo de concesión de autorizaciones con el PKCE

1. Abra la consola de Amazon Cognito. Si se le solicita, introduzca sus AWS credenciales.
2. Elija User Pools (Grupos de usuarios).
3. Elija un grupo de usuarios existente en la lista o cree un grupo de usuarios. Si crea un grupo de usuarios, se le pedirá que configure un cliente de aplicaciones y que configure la interfaz de usuario alojada durante el asistente.

    a. Si crea un nuevo grupo de usuarios, configure un cliente de aplicaciones y configure la interfaz de usuario alojada durante la configuración guiada.

    b. Si configura un grupo de usuarios existente, añada un dominio y un cliente de aplicación pública, si aún no lo ha hecho.
4. Genera una cadena alfanumérica aleatoria, normalmente un identificador único universal (UUID), para crear un desafío de código para el PKCE. Esta cadena es el valor del `code_verifier` parámetro que enviará en su solicitud al. Punto de conexión de token
5. Aplica un hash a la `code_verifier` cadena con el algoritmo SHA256. Codifique el resultado de la operación de hash en base64. Esta cadena es el valor del `code_challenge` parámetro que enviará en su solicitud a. Autorizar punto de conexión

En el siguiente Python ejemplo se genera un `code_verifier` y se calcula el`code_challenge`:

```
#!/usr/bin/env python3

import random
from base64 import urlsafe_b64encode
from hashlib import sha256
from string import ascii_letters
from string import digits

# use a cryptographically strong random number generator source
rand = random.SystemRandom()

code_verifier = ''.join(rand.choices(ascii_letters + digits, k=128))
code_verifier_hash = sha256(code_verifier.encode()).digest()
code_challenge = urlsafe_b64encode(code_verifier_hash).decode().rstrip('=')

print(f"code challenge: {code_challenge}")
print(f"code verifier: {code_verifier}")
```

El siguiente es un ejemplo de resultado del Python script:

```
code challenge: Eh0mg-OZv7BAyo-tdv_vYamx1boOYDulDklyXoMDtLg
code verifier: 9D-aW_iygXrgQcWJd0y0tNVMPSXSChIc2xceDhvYVdGLCBk-
JWFTmBNjvKSdOrjTTYazOFbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFGyE8yA-05-_v7Dxf3EIYJH
```

6. Complete el inicio de sesión en la interfaz de usuario alojada con una solicitud de concesión de código de autorización a PKCE. A continuación se muestra un ejemplo de URL:

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://
www.example.com&code_challenge=Eh0mg-OZv7BAyo-
tdv_vYamx1boOYDulDklyXoMDtLg&code_challenge_method=S256
```

7. Recopile la autorización `code` y canjéela por fichas con el punto final del token. A continuación, se muestra un ejemplo de solicitud:

```
POST /oauth2/token HTTP/1.1
Host: mydomain.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 296

redirect_uri=https%3A%2F%2Fwww.example.com&
client_id=1example23456789&
code=7378f445-c87f-400c-855e-0297d072ff03&
grant_type=authorization_code&
code_verifier=9D-aW_iygXrgQcWJd0y0tNVMPSXSChIc2xceDhvYVdGLCBk-
JWFTmBNjvKSdOrjTTYazOFbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFGyE8yA-05-_v7Dxf3EIYJH
```

8. Revisa la respuesta. Contendrá los identificadores de identificación, acceso y actualización. Para obtener más información sobre el uso de los tokens de grupos de usuarios de Amazon Cognito, consulte. Uso de tokens con grupos de usuarios

Respuestas de error de IU alojada y federación

Es posible que un proceso de inicio de sesión en la interfaz de usuario alojada o el inicio de sesión federado devuelva un error. A continuación, se muestran algunas condiciones que pueden provocar que la autenticación finalice con un error.

- Un usuario realiza una operación que el grupo de usuarios no puede realizar.
- Un desencadenador de Lambda no responde con la sintaxis esperada.
- El proveedor de identidades (IdP) devuelve un error.
- Amazon Cognito no pudo validar la información de atributos proporcionada por el usuario.
- El IdP no envió reclamaciones que se asignan a los atributos obligatorios.

Cuando Amazon Cognito encuentra un error, lo comunica de una de las siguientes maneras.

1. Amazon Cognito envía una URL de redireccionamiento con el error en los parámetros de la solicitud.
2. Amazon Cognito muestra un error en la IU alojada.

Los errores que Amazon Cognito agrega a los parámetros de la solicitud tienen el siguiente formato.

```
https://<Callback URL>/?error_description=error+description&error=error+name
```

Cuando ayude a los usuarios a enviar información de error cuando no puedan realizar una operación, pídales que capturen la URL y el texto o una captura de pantalla de la página.

> **Note**
>
> Las descripciones de errores de Amazon Cognito no son cadenas fijas y no debe utilizar una lógica que se base en un patrón o formato fijo.

Mensajes de error de OIDC y del proveedor de identidad social

Es posible que el proveedor de identidades devuelva un error. Cuando un IdP de OIDC u OAuth 2.0 devuelve un error que cumple con los estándares, Amazon Cognito redirige al usuario a la URL de devolución de llamada y agrega la respuesta de error del proveedor a los parámetros de la solicitud de error. Amazon Cognito agrega el nombre del proveedor y el código de error HTTP a las cadenas de error existentes.

La siguiente URL es un ejemplo de redireccionamiento desde un IdP que devolvió un error a Amazon Cognito.

```
https://www.amazon.com/?error_description=LoginWithAmazon+Error+-+400+invalid_request
+The+request+is+missing+a+required+parameter+%3A+client_secret&error=invalid_request
```

Dado que Amazon Cognito solo devuelve lo que recibe de un proveedor, es posible que el usuario vea un subconjunto de esta información.

Cuando el usuario encuentra un problema con el inicio de sesión inicial a través del IdP, el IdP envía los mensajes de error directamente al usuario. Amazon Cognito transmite un mensaje de error al usuario cuando genera una solicitud al IdP para validar la sesión del usuario. Amazon Cognito transmite los mensajes de error del IdP de OAuth y OIDC desde los siguientes puntos de conexión.

`/token`

Amazon Cognito intercambia un código de autorización de IdP por un token de acceso.

`/.well-known/openid-configuration`

Amazon Cognito descubre la ruta hacia los puntos de conexión del emisor.

`/.well-known/jwks.json`

Para verificar los JSON Web Tokens (JWT) de los usuarios, Amazon Cognito descubre las claves web JSON (JWK) que el IdP utiliza para firmar los tokens.

Dado que Amazon Cognito no inicia sesiones salientes con proveedores de SAML 2.0 que es posible que devuelvan errores HTTP, los errores de los usuarios durante una sesión con un IdP SAML 2.0 no incluyen este tipo de mensaje de error del proveedor.

Referencia de la API de grupos de usuarios de Amazon Cognito

Con los grupos de usuarios de Amazon Cognito, puede registrar usuarios e iniciar su sesión con su aplicación web y móvil. Es posible cambiar las contraseñas de los usuarios autenticados e iniciar flujos de contraseña olvidada para los usuarios sin autenticar. Para obtener más información, consulte Flujo de autenticación de los grupos de usuarios y Uso de tokens con grupos de usuarios.

La API de grupos de usuarios de Amazon Cognito incluye operaciones para ver y modificar sus grupos de usuarios y usuarios, así como para realizar la autenticación y autorización de usuarios. Para obtener una descripción de las clases de operaciones de la API que se combinan en la API de grupos de usuarios de Amazon Cognito, consulte Uso de la API de grupos de usuarios de Amazon Cognito y los puntos de conexión de grupos de usuarios.

Para obtener una lista detallada de las operaciones y la sintaxis de la API de grupos de usuarios de Amazon Cognito, consulte Referencia de la API de grupos de usuarios de Amazon Cognito. Cada página de la referencia de la API de los grupos de usuarios de Amazon Cognito enlaza con material de referencia con sintaxis y ejemplos para diferentes SDK de AWS.

Referencia de la API de grupos de identidades de Amazon Cognito (identidades federadas)

Con un grupo de identidades de Amazon Cognito, sus usuarios de aplicaciones web y móviles pueden obtener credenciales temporales de AWS con privilegios limitados con las que puedan acceder a otros servicios de AWS.

Para ver una referencia completa de la API de grupos de identidades (identidades federadas), consulte Referencia de la API de Amazon Cognito.

Referencia de la API de sincronización de Amazon Cognito

Amazon Cognito Sync es un servicio de AWS y una biblioteca cliente con la que se permite la sincronización entre dispositivos de datos de usuarios relacionados con la aplicación.

Para obtener más información sobre la referencia de la API de Amazon Cognito Sync, consulte Referencia de la API de Amazon Cognito Sync.

Historial de documentos de Amazon Cognito

En la siguiente tabla se describen los cambios importantes de la documentación de Amazon Cognito. Realizamos también actualizaciones menores frecuentes de la documentación en respuesta al feedback que envíe. Para enviar comentarios, busque el enlace de Feedback (Comentarios) en la parte inferior de cualquier página de la documentación de Amazon Cognito.

Cambio	Descripción	Fecha
Se agregó soporte para objetos complejos en el disparador Lambda previo al token	Ahora puede añadir matrices y objetos JSON a las notificaciones de identificación y de acceso a los tokens.	30 de mayo de 2024
Información actualizada sobre los permisos verificados y Amazon Cognito.	Amazon Verified Permissions ahora tiene una integración más directa con Amazon Cognito.	15 de mayo de 2024
Identidades verificadas de Amazon SES multirregional.	En algunos casos Regiones de AWS sin Amazon SES, los grupos de usuarios de Amazon Cognito equilibran la carga del correo electrónico entre dos regiones remotas.	10 de mayo de 2024
Se agregó información sobre la autorización M2M y la administración de los costos.	Aprenda a utilizar las concesiones de credenciales de cliente para casos de uso machine-to-machine (M2M) con los grupos de usuarios de Amazon Cognito.	9 de mayo de 2024
Amazon Cognito ya está disponible en Europa (España) y Asia Pacífico (Hyderabad). Regiones de AWS	Ahora puede crear recursos de Amazon Cognito en las regiones de Europa (España) y Asia Pacífico (Hyderabad).	15 de abril de 2024

Amazon Cognito ya está disponible en Asia Pacífico (Melbourne). Región de AWS	Ahora puede crear recursos de Amazon Cognito en la región Asia Pacífico (Melbourne).	4 de abril de 2024
Se agregó una aplicación Android de ejemplo en Flutter para grupos de usuarios de Amazon Cognito.	Puede crear una aplicación móvil de inicio para Amazon Cognito a partir de una aplicación Flutter de ejemplo. GitHub	4 de abril de 2024
Nuevo contenido de introducción	Contenido ampliado para empezar, escenarios comunes, prácticas recomendadas para varios usuarios y acceso a los recursos después de iniciar sesión.	1 de abril de 2024
Amazon Cognito ya está disponible en Europa (Zúrich). Región de AWS	Ahora puede crear recursos de Amazon Cognito en la región de Europa (Zúrich).	14 de marzo de 2024
Amazon Cognito ya está disponible en Oriente Medio (Emiratos Árabes Unidos). Región de AWS	Ahora puede crear recursos de Amazon Cognito en la región de Oriente Medio (EAU).	8 de marzo de 2024
Nuevas funciones de SAML y contenido mejorado.	Ahora puedes firmar las solicitudes de SAML, cifrar las respuestas de SAML y configurar el SSO de SAML iniciado por el IdP.	1 de febrero de 2024
Aumentos de cuota disponibles.	Ahora puede adquirir capacidad adicional para las cuotas de tasa de solicitud de Amazon Cognito.	25 de enero de 2024

Los grupos de identidades de Amazon Cognito admiten las tasas de solicitud en Service Quotas.	Ahora puede supervisar las cuotas requests-per-second (RPS) de los grupos de identidades de Amazon Cognito y solicitar un aumento en la consola de Service Quotas.	19 de diciembre de 2023
Se agregó una nueva función para personalizar el contenido de los tokens de acceso.	Ahora puede agregar, modificar y eliminar las reclamaciones y los ámbitos de los tokens de acceso a los grupos de usuarios.	12 de diciembre de 2023
Se ha mejorado el contenido sobre los clientes de aplicaciones y los ámbitos de OAuth.	Ediciones de claridad y correcciones a Clientes de aplicación de grupo de usuarios y Autorización de alcances, M2M y API con servidores de recursos. Eliminadas las instrucciones de consola heredadas.	14 de noviembre de 2023
Contenido mejorado sobre dispositivos y autenticación de dispositivos.	Nuevo contenido sobre el uso de las claves de los dispositivos y la autenticación SRP de los dispositivos.	18 de octubre de 2023
AWS Management Console Guía actualizada.	Se ha eliminado la referencia de la consola de grupos de usuarios y se redistribuyeron los temas dentro de temas relacionados, y se ha agregado una guía para la organización por pestañas en la consola de Amazon Cognito.	30 de agosto de 2023

Se ha restado importancia al acceso directo al punto final LOGIN.	Se ha agregado información general visual del grupo de usuarios Punto de conexión Login y enfatizado el inicio de la autenticación con Autorizar punto de conexión.	30 de agosto de 2023
Amazon Cognito ya está disponible en Asia Pacífico (Osaka) e Israel (Tel Aviv). Regiones de AWS	Ahora puede crear recursos de Amazon Cognito en las regiones de Asia Pacífico (Osaka) e Israel (Tel Aviv).	30 de agosto de 2023
Se introdujo información sobre la autorización de Amazon Cognito con permisos verificados de Amazon.	En la aplicación, puede invocar la API de permisos verificados para que una autoridad central tome las decisiones de acceso.	1 de agosto de 2023
Se agregó una nueva función para registrar la actividad detallada de los usuarios del grupo de usuarios en Amazon CloudWatch Logs.	Ahora puede registrar los errores de entrega de correos electrónicos y mensajes SMS en los grupos de CloudWatch registro.	1 de agosto de 2023
Información actualizada sobre la política AWS administrada para los usuarios invitados del grupo de identidades.	La reducción del alcance de los permisos para los usuarios invitados del grupo de identidades ahora incluye una política de sesión integrada y una AWS política de sesión administrada.	16 de mayo de 2023

Mejora del contenido y nuevas instrucciones de consola para los grupos de identidades de Amazon Cognito.	Se han agregado nuevos tutoriales de consola para reflejar la nueva experienc ia de la consola y se han mejorado los detalles de integración de códigos para los grupos de identidades.	16 de mayo de 2023
Adiciones y mejoras en la página principal del servicio y en la página principal de los grupos de usuarios.	Páginas de información general actualizadas para Amazon Cognito y grupos de usuarios.	16 de mayo de 2023
Mejoras generales en la documentación de los tokens del grupo de usuarios.	Se han actualizado los tokens de ejemplo y se ha añadido nueva información sobre la verificación de los tokens.	16 de febrero de 2023
Ahora puede registrar los eventos de datos de los grupos de identidades de Amazon Cognito. AWS CloudTrail	CloudTrail admite la selección de agrupaciones de identidad es de Amazon Cognito, operaciones de API de gran volumen en registros que registran eventos de datos.	15 de febrero de 2023
Ejemplos y descripciones actualizados de los activador es Lambda.	Los ejemplos de activador es Lambda se actualizan a la JavaScript versión 3. Ahora puede correlacionar directame nte disparadores de Lambda con acciones de la API.	31 de enero de 2023

Los grupos de identidades de Amazon Cognito aplican una política AWS administrada a las sesiones no autenticadas.	Los usuarios del grupo de identidades que se autentican mediante el flujo mejorado ahora tienen una política AWS administrada adicional que se aplica a su sesión.	31 de enero de 2023
Se han añadido ejemplos de código.	Esta guía ahora incluye un código de ejemplo para su aplicación de Amazon Cognito en diversos lenguajes de programación.	23 de enero de 2023
Se agregó información sobre los modelos de API y la autenticación con los grupos de usuarios de Amazon Cognito.	Los grupos de usuarios de Amazon Cognito disponen de varias interfaces de API y formatos para la autorización de solicitudes.	15 de diciembre de 2022
Amazon Cognito ya está disponible en Europa (Milán). Región de AWS	Ahora puede crear grupos de usuarios de Amazon Cognito en la región de Europa (Milán).	6 de diciembre de 2022
Se agregó información sobre la protección contra la eliminación de grupos de usuarios.	Al crear un nuevo grupo de usuarios con el AWS Management Console, ahora está protegido contra la eliminación de forma predeterminada.	20 de octubre de 2022

Se agregó una guía del usuario para la interfaz de usuario alojada e información sobre el MFA TOTP en la interfaz de usuario alojada.	Sus usuarios ahora pueden registrar un dispositivo MFA con TOTP en la interfaz de usuario alojada en Amazon Cognito. Ahora puede obtener una vista previa de la interfaz de usuario alojada predeterminada.	8 de septiembre de 2022
Se agregó información sobre AWS WAF Amazon Cognito.	Ahora puede asociar una ACL AWS WAF web a un grupo de usuarios de Amazon Cognito.	3 de agosto de 2022
Se agregaron más AWS CloudTrail eventos de ejemplo.	Amazon Cognito ahora registra las solicitudes de interfaz de usuario alojadas y federadas en su seguimiento.	15 de junio de 2022
Se agregó información sobre la verificación de atributos en dos pasos.	Ahora puede elegir si el usuario debe verificar una nueva dirección de correo electrónico o número de teléfono antes de poder iniciar sesión con ellos.	9 de junio de 2022
Documentación de federación actualizada. Nueva función de propagación de direcciones IP.	Tutoriales actualizados para configurar un grupo social de usuarios. IdPs Se ha agregado información sobre los perfiles de usuario federados y la asignación de atributos. Se agregó nueva información sobre las huellas digitales de los dispositivos para una seguridad avanzada.	31 de mayo de 2022

Inicie sesión con los usuarios federados sin interactuar con la interfaz de usuario alojada	Se agregó una nueva página sobre cómo marcar aplicaciones como favoritas para que Amazon Cognito dirija silenciosamente a los usuarios al inicio de sesión federado.	29 de mayo de 2022
Mensajes de correo electrónico y SMS dentro de la región para grupos de usuarios de Amazon Cognito	Ahora puede utilizar Amazon Simple Notification Service para los mensajes SMS y Amazon Simple Email Service para los mensajes de correo electrónico al Región de AWS mismo tiempo que su grupo de usuarios.	14 de marzo de 2022
Actualizaciones de la página de cuotas	Se agregaron y aclararon las cuotas de recursos y tasas de solicitud.	10 de enero de 2022
Nueva experiencia de consola de grupos de usuarios de Amazon Cognito	Se han actualizado las instrucciones para crear y administrar grupos de usuarios en la nueva versión de la consola de Amazon Cognito.	18 de noviembre de 2021
RevokeToken API y punto final de revocación	Puede utilizar la RevokeTok en operación para revocar un token de actualización para un usuario.	10 de junio de 2021
Mejores prácticas para varios inquilinos	Se agregaron las mejores prácticas para las aplicaciones de varios inquilinos.	4 de marzo de 2021

Atributos para controlar el acceso	Los grupos de identidades de Amazon Cognito proporcionan atributos para el control de acceso (AFAC) como una forma de que los clientes concedan a los usuarios acceso a los recursos. AWS La autorización puede realizarse en función de los atributos de los usuarios del proveedor de identidad que utilizaron para federarse con Amazon Cognito.	15 de enero de 2021
Activador Lambda de remitente de SMS personalizado y disparador Lambda de remitente de correo electrónico personalizado	Con el desencadenador de Lambda para remitentes personalizados de SMS y el desencadenador para remitentes personalizados de correos electrónicos, se puede habilitar a un proveedor de terceros para enviar notificaciones de correo electrónico y SMS a sus usuarios desde el código de función de Lambda.	30 de noviembre de 2020
Actualizaciones del token de Amazon Cognito	Se agregó la información actualizada sobre el vencimiento a los tokens de acceso, ID y actualización.	29 de octubre de 2020

Cuotas de Amazon Cognito Service	Service Quotas está disponible para las cuotas de categorías de Amazon Cognito. Puede usar la consola Service Quotas para ver el uso de la cuota, solicitar un aumento de la cuota y crear CloudWatch alarmas para supervisar el uso de la cuota. Como parte de este cambio, se actualizó la sección CloudWatch Métricas disponibles para los grupos de usuarios de Amazon Cognito para reflejar la nueva información. El nombre de la nueva sección es: Seguimiento de cuotas y uso en CloudWatch y Service Quotas	29 de octubre de 2020
Categorización de cuotas de Amazon Cognito	Con las categorías de cuotas, es más fácil monitorear el uso de cuotas y solicitar un aumento. Las cuotas se agrupan en categorías en función de los casos de uso común.	17 de agosto de 2020
Amazon Cognito es compatible con GovCloud de EE. UU. AWS	Amazon Cognito ahora es compatible con la región AWS GovCloud (EE. UU.).	13 de mayo de 2020

Actualizaciones de documentos de Amazon Cognito Pinpoint	Se agregó un nuevo rol vinculado al servicio. Se actualizaron las instrucciones de “Uso del análisis de Amazon Pinpoint con grupos de usuarios de Amazon Cognito”.	13 de mayo de 2020
Nuevo capítulo de seguridad dedicado a Amazon Cognito	El capítulo de seguridad puede ayudar a su organización a obtener información detallada sobre la seguridad integrada y configurable de AWS los servicios. Nuestros nuevos capítulos proporcionan información sobre la seguridad de la nube y en la nube.	30 de abril de 2020
Amazon Cognito Identity Pools ahora admite el inicio de sesión con Apple	Sign in with Apple está disponible en todas las regiones donde opera Amazon Cognito, excepto en la región cn-north-1.	7 de abril de 2020
Nuevo control de versiones de la API de Facebook	Se ha agregado la selección de versiones a la API de Facebook.	3 de abril de 2020
Actualización de la insensibilidad entre mayúsculas y minúsculas	Se ha añadido una recomendación sobre cómo deshabilitar la sensibilidad a mayúsculas y minúsculas del nombre de usuario antes de crear un grupo de usuarios.	11 de febrero de 2020

Nueva información sobre AWS Amplify	Se agregó información sobre la integración de Amazon Cognito con su aplicación web o móvil mediante bibliotecas y AWS Amplify SDK. Se eliminó información sobre el uso de los SDK de Amazon Cognito anteriores a AWS Amplify.	22 de noviembre de 2019
Nuevo atributo para los activadores del grupo de usuarios	Amazon Cognito ahora incluye un `clientMetadata` parámetro en la información de eventos que transfiere a las AWS Lambda funciones de la mayoría de los activadores de grupos de usuarios. Puede utilizar este parámetro para mejorar el flujo de trabajo de autenticación personalizado con datos adicionales.	4 de octubre de 2019
Límite actualizado	Se ha actualizado el límite de limitación de la acción ListUsers de la API.	25 de junio de 2019
Nuevo límite	Los límites flexibles de los grupos de usuarios ahora incluyen un límite para el número de usuarios.	17 de junio de 2019

Configuración de correo electrónico de Amazon SES para grupos de usuarios de Amazon Cognito	Puede configurar un grupo de usuarios para que Amazon Cognito envíe correos electrónicos a sus usuarios con la configuración de Amazon SES. Con esta configuración, Amazon Cognito puede enviar mensajes de correo electróni co con un mayor volumen de entrega al que sería posible de otro modo.	8 de abril de 2019
Soporte de etiquetado	Se agregó información sobre el etiquetado de recursos de Amazon Cognito.	26 de marzo de 2019
Cambie el certificado de un dominio personalizado	Si utiliza un dominio personali zado para alojar la IU alojada de Amazon Cognito, puede cambiar el certificado SSL de este dominio según sea necesario.	19 de diciembre de 2018
Nuevo límite	Se ha añadido un nuevo límite para el número máximo de grupos al que cada usuario puede pertenecer.	14 de diciembre de 2018
Límites actualizados	Se han actualizado los límites flexibles de los grupos de usuarios.	11 de diciembre de 2018

Actualización de la documentación para verificar las direcciones de correo electrónico y los números de teléfono	Se ha añadido información acerca de cómo configurar el grupo de usuarios para requerir la verificación del correo electrónico o del número de teléfono cuando el usuario se registra en la aplicación.	20 de noviembre de 2018
Actualización de la documentación para probar los correos electrónicos	Se agregaron instrucciones acerca de cómo crear correos electrónicos de Amazon Cognito mientras se prueba la aplicación.	13 de noviembre de 2018
Seguridad avanzada de Amazon Cognito	Se han agregado características de seguridad nuevas que permiten a los desarrolladores proteger sus aplicaciones y usuarios de bots malintencionados, proteger las cuentas de los usuarios frente a las credenciales atacadas y ajustar automáticamente los desafíos necesarios para iniciar sesión en función del riesgo calculado del intento de inicio de sesión.	14 de junio de 2018
Dominios personalizados para la interfaz de usuario alojada de Amazon Cognito	Con ellos, los desarrolladores pueden utilizar sus propios dominios totalmente personalizados para la IU alojada de los grupos de usuarios de Amazon Cognito.	4 de junio de 2018

Grupo de usuarios de Amazon Cognito: proveedor de identidad OIDC	Añadido el inicio de sesión de grupos de usuarios a través de un proveedor de identidad OpenID Connect (OIDC), como Salesforce o Ping Identity.	17 de mayo de 2018
Activador de migración a Amazon Cognito Lambda	Se han agregado páginas que tratan la característica de disparador de migración de Lambda	8 de abril de 2018
Actualización de la guía para desarrolladores de Amazon Cognito	Se agregó el nivel superior “Qué es Amazon Cognito” e “Introducción a Amazon Cognito”. También se han agregado situaciones comunes y se ha reorganizado el índice de grupos de usuarios. Se agregó una nueva sección; “Introducción a los grupos de usuarios de Amazon Cognito”.	6 de abril de 2018

Beta de seguridad avanzada de Amazon Cognito	Se han añadido características de seguridad nuevas que permiten a los desarrolladores proteger sus aplicaciones y usuarios de bots malintenc ionados, proteger las cuentas de los usuarios con credencia les publicadas que se han visto comprometidas en otros sitios de Internet, y ajustar automáticamente los desafíos necesarios para iniciar sesión en función del riesgo calculado del intento de inicio de sesión.	28 de noviembre de 2017
Integración con Amazon Pinpoint	Se agregó la posibilidad de utilizar Amazon Pinpoint a fin de proporcionar análisis para las aplicaciones de grupos de usuarios de Amazon Cognito y enriquecer los datos de usuario destinados a las campañas de Amazon Pinpoint.	26 de septiembre de 2017

Funciones de federación y de interfaz de usuario de aplicaciones integradas de los grupos de usuarios de Amazon Cognito	Se ha añadido la posibilidad de permitir a los usuarios iniciar sesión en el grupo de usuarios a través de Facebook, Google, Login with Amazon o un proveedor de identidad SAML. Se ha añadido una interfaz de aplicaciones integradas personalizable y compatibilidad de OAuth 2.0 con notificaciones personalizadas.	10 de agosto de 2017
Cambios en las funciones relacionados con el cumplimiento de la HIPAA y la PCI	Se ha añadido la posibilidad de permitir a los usuarios utilizar un número de teléfono o una dirección de correo electrónico como su nombre de usuario.	6 de julio de 2017
Funciones de control de acceso basadas en roles y grupos de usuarios	Se ha añadido una capacidad administrativa de creación y administración de grupos de usuarios. Los administradores pueden asignar roles de IAM a usuarios basados en la pertenencia a grupos y en reglas creadas por el administrador.	15 de diciembre de 2016
Actualización de la documentación	Ejemplos actualizados que muestran cómo usar los AWS Lambda activadores con grupos de usuarios.	27 de noviembre de 2016
Actualización de la documentación	Ejemplos de código de iOS actualizados.	18 de noviembre de 2016

Actualización de la documentación	Se ha añadido información acerca del flujo de confirmación de las cuentas de usuario.	9 de noviembre de 2016
Función de creación de cuentas de usuario	Se agregó la funcionalidad administrativa para crear cuentas de usuario a través de la consola de Amazon Cognito y la API.	6 de octubre de 2016
Función de importación de usuarios	Se ha añadido la capacidad de importación masiva de grupos de usuarios de Cognito. Utilice esta característica para migrar usuarios de su proveedor de identidad actual a un grupo de usuarios de Amazon Cognito.	1 de septiembre de 2016
Disponibilidad general de los grupos de usuarios de Cognito	Se ha añadido la característica de grupos de usuarios de Cognito. Utilice esta característica para crear y mantener un directorio de usuarios y añadir la inscripción y el inicio de sesión a la aplicación móvil o la aplicación web mediante grupos de usuarios.	28 de julio de 2016
Soporte para SAML	Se ha añadido compatibilidad con la autenticación con proveedores de identidad mediante el lenguaje SAML 2.0 (Security Assertion Markup Language 2.0).	23 de junio de 2016
CloudTrail integración	Integración añadida con AWS CloudTrail.	18 de febrero de 2016

Integración de eventos con Lambda	Le permite ejecutar una AWS Lambda función en respuesta a eventos importantes en Amazon Cognito.	9 de abril de 2015
Transmisión de datos a Amazon Kinesis	Proporciona control e información de los flujos de datos.	4 de marzo de 2015
Soporte para OpenID Connect	Activa la compatibilidad con los proveedores de OpenID Connect.	23 de noviembre de 2014
Sincronización push	Activa la compatibilidad con la sincronización mediante inserción silenciosa.	6 de noviembre de 2014
Se agregó soporte para identidades autenticadas por el desarrollador	Esto permite tratar a los desarrolladores propietarios de sus propios sistemas de administración de identidad es y autenticación como proveedores de identidad en Amazon Cognito.	29 de septiembre de 2014
Disponibilidad general de Amazon Cognito		10 de julio de 2014

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.